Actes des 17es Journées d'Intelligence Artificielle Fondamentale et des 18es Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes

Zied Bouraoui, François Schwarzentruber, Anaelle Wilczynski

AfIA

Association française
pour l'Intelligence Artificielle

JIAF-JFPDA

Journées d'Intelligence Artificielle Fondamentale et Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes

PFIA 2023

Table des matières

Éditorial

Journées d'Intelligence Artificielle Fondamentale et Journées Francophones sur la Planification, la Décision et l'Apprentissagepour la conduite de systèmes

Les Journées d'Intelligence Artificielle Fondamentale (JIAF) et les Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes (JFPDA) constituent un rendez-vous annuel de la communauté francophone travaillant sur l'Intelligence Artificielle Fondamentale et la Planification. En 2023, pour la première fois, JIAF et JFPDA sont réunies au sein d'une même conférence, qui a été hébergée par la Plate-Forme Intelligence Artificielle (PFIA) 2023, organisée à Strasbourg les 6 et 7 juillet 2023.

Les thématiques de recherche abordées lors des JIAF-JFPDA portent généralement sur :

- La définition de modèles de *représentation des informations* (croyances, connaissances, préférences, obligations et permissions, actions, incertitude, confiance, réputation) : langages des logiques classiques ou non classiques, modèles possibilistes, ontologies, langages à base de contraintes, représentations graphiques, etc ;
- La définition et l'automatisation de *raisonnements* sur ces informations : raisonnement spatio-temporel, dynamique des informations, révision de croyances, fusion d'informations symboliques, raisonnement par argumentation, raisonnement causal, raisonnement abductif, raisonnement à partir de cas, etc ;
- La mise au point de méthodes de *codage* des informations et d'*algorithmes* de traitement efficaces : compilation de connaissances, SAT, contraintes, ASP, etc ;
- La modélisation formelle de l'*interaction* : entre utilisateurs et systèmes informatiques, entre entités informatiques autonomes (agents), intégration de ces deux aspects dans les divers agents conversationnels, agents de recherche, assistants personnels ;
- Le choix social, la théorie des jeux, les algorithmes pour les *jeux* ;
- Des objectifs de décision, planification, ordonnancement, diagnostic, apprentissage et dans différents contextes d'*application*, comme par exemple le Web sémantique ;
- La prise de décision séquentielle sous incertitude et la planification : problèmes d'apprentissage par renforcement, processus décisionnels de Markov, théorie du contrôle, programmation dynamique, etc.

Les JIAF-JFPDA entretiennent des liens privilégiés avec le collège « Représentation et Raisonnement » de l'AFIA, et avec le GDR RADIA, groupe de recherche sur les « Aspects Formels et Algorithmiques de l'Intelligence Artificielle » du CNRS. Le comité de programme des journées est composé d'une trentaine de membres, et a été animé sur l'édition 2023 par Zied Bouraoui (CRIL, Université d'Artois & CNRS), François Schwarzentruber (IRISA, ENS Rennes) et Anaëlle Wilczynski (CentraleSupélec, Université Paris-Saclay).

Zied Bouraoui, François Schwarzentruber, Anaëlle Wilczynski

JIAF-JFPDA@PFIA 2023

Comité de programme

Présidence

— Zied Bouraoui (CRIL, Univ Artois & CNRS) ;
— François Schwarzentruber (IRISA, ENS Rennes) ;
— Anaëlle Wilczynski (MICS, CentraleSupélec, Université Paris-Saclay).

Membres

— Francesco Belardinelli (IBISC, Université d'Évry) ;
— Nawal Benabbou (LIP6, Sorbonne Université) ;
— Elise Bonzon (LIPADE, Université Paris Descartes) ;
— Nadia Creignou (LIS, Aix-Marseille Université) ;
— Aurélie Beynier (LIP6, Sorbonne Université) ;
— Olivier Buffet (INRIA / LORIA) ;
— Martin Cooper (IRIT, Université Paul Sabatier) ;
— Tiago de Lima (CRIL, Univ Artois & CNRS) ;
— Sylvie Doutre (IRIT, Université de Toulouse) ;
— Alain Dutech (Loria - Inria) ;
— Jérôme Euzenat (LIG, INRIA) ;
— Hugo Gilbert (LAMSADE, Université Paris-Dauphine) ;
— Sébastien Konieczny (CRIL, CNRS) ;
— Jean Lieber (LORIA, INRIA) ;
— Jérôme Lang (CNRS, LAMSADE, Université Paris-Dauphine) ;
— Frédéric Maris (IRIT, Université de Toulouse) ;
— Pierre Marquis (CRIL, IUF, Univ Artois & CNRS) ;
— Amedeo Napoli (LORIA Nancy, CNRS - Inria - Université de Lorraine) ;
— Célia da Costa Pereira (I3S, Université Nice Sophia Antipolis) ;
— Damien Pellier (Laboratoire d'Informatique de Grenoble) ;
— Laurent Perrussel (IRIT, Université de Toulouse) ;
— Sophie Pinchinat (IRISA, INRIA) ;
— Philippe Preux (INRIA, LIFL, Université de Lille) ;
— Emmanuel Rachelson (ISAE-SUPAERO) ;
— Stéphanie Roussel (ONERA) ;
— Julien Rossit (LIPADE, Université Paris Decartes) ;
— Régis Sabbadin (INRAE) ;
— Vincent Thomas (LORIA) ;
— Paul Weng (UM-SJTU Joint Institute) ;
— Bruno Zanuttini (GREYC, UNICAEN).

Session 1 : Argumentation

Actes JIAF-JFPDA 2023

Similarity Measures between Order-Sorted Logical Arguments

Victor David[1] **Jérôme Delobelle**[2] **Jean-Guy Mailly**[2]

[1] Department of Mathematics and Computer Science University of Perugia, Italy
[2] Université Paris Cité, LIPADE, F-75006 Paris, France
victor.david@unipg.it
jerome.delobelle@u-paris.fr
jean-guy.mailly@u-paris.fr

Abstract

Similarity in formal argumentation has received some attention recently, since one can argue that, in some context, using similar arguments to reach a conclusion is not the same as using dissimilar ones. While existing work consider arguments built from propositional logic, in this work we adapt the notion of similarity measures to arguments built from Order-Sorted First Order Logic, an extension of First Order Logic which allows to represent complex information, considering the type of the data. We study and evaluate our approach with respect to an adaptation of axioms from the literature. This paves the way to new reasoning modes taking into account similarity between arguments in complex settings like ontologies.

1 Introduction

Formal argumentation has become a major topic in Knowledge Representation and Reasoning (KRR), with various applications like decision making [29], defeasible reasoning [16], dealing with inconsistent knowledge bases [12], as well as in multi-agent systems [23]. So, when agents use logic-based information for reasoning, it is possible to build arguments from this information, where typically an argument is a pair made of a set of formulae (called support) and a single formula (called conclusion). The conclusion should be a logical consequence of the support. Examples of arguments are $A = \langle\{p \wedge q \wedge r\}, p \wedge q\rangle$, $B = \langle\{p \wedge q\}, p \wedge q\rangle$ and $C = \langle\{p, q\}, p \wedge q\rangle$. From the definition of arguments, one can identify attacks between them, and then use a semantics to evaluate the arguments. Finally, conclusions of the "strong" arguments are inferred from the base. In the literature, there exist several families of semantics (e.g. extension-based, ranking-based or gradual semantics) to determine which arguments are "strong". We refer the reader to [1] for a recent overview of the existing families of semantics in abstract argumentation and the differences between these approaches (e.g., definition, outcome, application). Among the existing gradual semantics, like h-Categorizer [12], some of them satisfy the Counting (or Strict Monotony) principle defined in [2]. This principle states that each attacker of an argument contributes to weakening the argument. For instance, if the argument $D = \langle\{\neg p \vee \neg q\}, \neg p \vee \neg q\rangle$ is attacked by A, B, C, then each of the three arguments will decrease the strength of D. However, the three attackers are somehow similar, thus D will lose more than necessary. Consequently, the authors in [4] have motivated the need for investigating the notion of similarity between pairs of such logical arguments. They introduced a set of principles that a reasonable similarity measure should satisfy, and provided several measures that satisfy them. In [3, 5, 6] several extensions of h-Categorizer that take into account similarities between arguments have been proposed. All these works consider propositional logic. In this paper, we suggest to adapt the principles behind similarity measures for logical arguments to a much more expressive framework, namely Order-Sorted First Order Logic ($\mathtt{OS-FOL}$) [24], a formalism which generalizes (standard) First Order Logic ($\mathtt{FOL}$) . Fragments of $\mathtt{OS-FOL}$ have been used for reasoning in various domains (*e.g.* [17] uses FOL for reasoning about policies, and [22] proposes an architecture for building cognitive agents capable of deduction on facts and rules inferred directly from natural language). More generally, many KRR formalisms can be captured through $\mathtt{OS-FOL}$, like Description Logics [11]. While $\mathtt{FOL}$ has already interesting modelling capabilities, $\mathtt{OS-FOL}$ allows to naturally model situations where variables belong to a given domain, and there can be relations between the domains of the variables (*e.g.*, the domains made of all the penguins is a subset of the domain contai-

ning all the birds). So, by studying logical arguments built from OS − FOL, we are able to apply our work to existing argumentation frameworks based on FOL [13, 10], but also other rich frameworks like Description Logics [11]. This paves the way to applications of argumentation (and similarity measures) to inconsistent knowledge expressed in these rich structured frameworks.

2 Background

2.1 Logic and Arguments

We assume that the reader is familiar with propositional logic and First Order Logic (FOL). First Order Logic is a rich framework that develops information about the objects and can also express the relationship between them (using predicates). An example is "Tweety is a penguin, all penguins are birds and all birds have wings, so Tweety has wings" which can be expressed as $penguin(Tweety) \wedge (\forall x, penguin(x) \rightarrow bird(x)) \wedge (\forall x, bird(x) \rightarrow haveWings(x))$ for the premises, and $haveWings(Tweety)$ as the consequence. However, this framework does not allow to distinguish between various types of objects. This means that it would be possible to write a FOL formula like $hasRoots(Tweety)$, which does not make sense since Tweety is a bird, not a plant. Since we want to apply our method to contexts where data can have a specific type, we use Order-Sorted FOL [24], a generalization of (standard) FOL where all the variables are associated with a *sort* (as well as the parameters of the predicates). [1] Then, when interpreting a formula, the domain of variables is constrained by its sort. An additional constraint can be added to these sorts, as a partial order over them, corresponding to inclusion relations between the domains associated to the sorts.

Definition 1 (Order-Sorted FOL) *Let* $\mathbf{So} = \{s_1, \ldots, s_n\}$ *be a set of sorts, and* $\prec \subseteq \mathbf{So} \times \mathbf{So}$ *a partial order over* $\mathbf{So}$. *An* Order-Sorted First Order Language OS − FOL, *is a set of formulae built up by induction from :*
– *a set* $\mathbf{C}$ *of constants (*$\mathbf{C} = \{a_1, \ldots, a_l\}$*),*
– *a set* $\mathbf{V}$ *of variables (*$\mathbf{V} = \{x^s, y^s, z^s, \ldots \mid s \in \mathbf{So}\}$*),*
– *a set* $\mathbf{P}$ *of predicates (*$\mathbf{P} = \{P_1, \ldots, P_m\}$*),*
– *a function* $\mathtt{ar} : \mathbf{P} \rightarrow \mathbb{N}$ *which gives the arity of predicates,*
– *a function* sort *s.t. for* $P \in \mathbf{P}$, $\mathtt{sort}(P) \in \mathbf{So}^{\mathtt{ar}(P)}$, *and for* $c \in \mathbf{C}$, $\mathtt{sort}(c) \in \mathbf{So}$,
– *the usual connectives (*$\neg, \vee, \wedge, \rightarrow, \leftrightarrow$*), Boolean constants* $\top$ *(true) and* $\bot$ *(false) and quantifier symbols (*$\forall, \exists$*).*
A grounded formula *is a formula without any variable.*

We use lowercase greek letters (*e.g.* ϕ, ψ) to denote formulae, and uppercase ones (*e.g.* Φ, Ψ) to denote sets of formulae. The set of all Order-Sorted FOL formulae is denoted by OS − FOL. We assume formulae to be *prenex*, *i.e.* written as $Q_1 x_1, \ldots, Q_k x_k \phi$ where Q_i is a quantifier (for each $i \in \{1, \ldots, k\}$) and ϕ is a non-quantified formula. A formula ϕ is in negation normal form (NNF) if and only if it does not contain implication or equivalence symbols, and every negation symbol occurs directly in front of an atom. Following [21], we slightly abuse words and denote by $\mathtt{NNF}(\phi)$ the formula in NNF obtained from ϕ by "pushing down" every occurrence of $\neg$ (using De Morgan's law) and eliminating double negations. For instance, $\mathtt{NNF}(\neg((P(a) \rightarrow Q(a)) \vee \neg Q(b))) = P(a) \wedge \neg Q(a) \wedge Q(b)$. In that case, we call *literal* either an atom (*i.e.* a predicate with its parameters) or the negation of an atom. The set of grounded atoms is denoted by $\mathbf{A}$. We denote by $\mathtt{Lit}(\phi)$ the set of literals occurring in $\mathtt{NNF}(\phi)$, hence $\mathtt{Lit}(\neg((P(a) \rightarrow Q(a)) \vee \neg Q(b))) = \{P(a), \neg Q(a), Q(b)\}$. For a given set of predicates $\mathbf{P}$, we define $\mathbf{L} = \{P(x_1^{s_1}, \ldots, x_k^{s_k}), \neg P(x_1^{s_1}, \ldots, x_k^{s_k}) \mid P \in \mathbf{P}, \mathtt{sort}(P) = (s_1, \ldots, s_k)\}$ the set of literals. We say that a literal L is *negative* when it starts with a negation, denoted by $\mathtt{Pol}(L) = -$. Otherwise we say that it is *positive*, denoted by $\mathtt{Pol}(L) = +$. And we say that two literals have the same *polarity* if they are either both positive or both negative. Finally, given a grounded literal $L = \pm P(a_1, \ldots, a_k)$ where $\pm$ indicates the polarity of L, $\mathtt{Pred}(L) = P$ corresponds to the name of the predicate underlying L, and $\mathtt{Para}(L) = \langle a_1, \ldots, a_k \rangle$.

Consider $\phi \in$ OS − FOL, ϕ is in a conjunctive normal form (CNF) if it is a conjunction of clauses $\bigwedge_i cl_i$ where each clause cl_i is a disjunction of literals $\bigvee_j l_j$. For instance $P(a) \wedge (Q(a) \vee Q(b))$ is in CNF while $(P(a) \wedge Q(a)) \vee Q(b)$ is not. CNF formulae are particular NNF formulae. Clauses are also usually represented as sets of literals, and CNF formulae as sets of clauses.

In OS − FOL, the partial order $\prec$ represents "sub-type" relations between groups of entities. For instance, the fact that dogs are a special type of mammals can be represented by such a sub-type relation. In the case where $s_1 \prec s_2$, a predicate which expects a parameter of type s_2 can be applied to a constant or variable of type s_1 (for instance, a predicate about mammals can be applied to dogs).

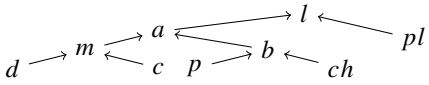


Figure 1 – Hierarchy of sorts from Example 1. An arrow from s_1 to s_2 means $s_1 \prec s_2$.

Example 1 OS − FOL *formulae can be used to reason about ontological information. Assume that we have the following information : mammals and birds are animals, dogs and cats are mammals, penguins and chickens are birds. Moreover, Zazu is a bird, Tweety is a penguin, and Dogmatix is a dog. Finally, animals are living beings, as well as*

1. In this paper, we restrict ourselves to formulae without functions.

plants. This can be represented by the following sorts and constants :
– $\mathbf{So} = \{m, b, a, d, c, p, ch, l, pl\}$ *with* $m \prec a$, $b \prec a$, $d \prec m$, $c \prec m$, $p \prec b$, $ch \prec b$, $a \prec l$, $pl \prec l$ *(see Figure 1),*
– $Z \in \mathbf{C}$ *with* $\texttt{sort}(Z) = b$ *is a constant for Zazu,*
– $T \in \mathbf{C}$ *with* $\texttt{sort}(T) = p$ *is a constant for Tweety,*
– $D \in \mathbf{C}$ *with* $\texttt{sort}(D) = d$ *is a constant for Dogmatix.*

We know that all birds have wings, and both mammals and birds are warm-blooded. Also, some birds and some mammals fly, but not all of them. If a bird is wounded, then it cannot fly. If a bird is penguin, then it cannot fly. Some birds are wounded. Finally, Tweety is a penguin. This information can be represented by the predicates $\mathbf{P} = \{hW, wB, f, w, p\}$*, standing respectively for "haveWings", "warmBlooded", "fly", "wounded" and "penguin" s.t.* $\texttt{ar}(P_i) = 1$ *and* $\texttt{sort}(P_i) = a$ *for each* $P_i \in \mathbf{P}$.

We can build, e.g. *the formula* $\forall x^b, hW(x^b)$ *meaning that all birds have wings (because the variable* x^b *has the sort b). The other pieces of information are represented by*

$\forall x^b wB(x^b)$	$\forall x^m wB(x^m)$
$\exists x_1^b, x_2^b f(x_1^b) \wedge \neg f(x_2^b)$	$\exists x_1^m, x_2^m f(x_1^m) \wedge \neg f(x_2^m)$
$\forall x^b w(x^b) \rightarrow \neg f(x^b)$	$\forall x^b p(x^b) \rightarrow \neg f(x^b)$
$\exists x^b w(x^b)$	$p(T)$

However formulae like $\exists x^l, f(x^l)$ *or* $\forall x^{pl}, wB(x^{pl})$ *are not well-formed, since the predicates* fly *and* wB *cannot be applied to living beings or plants.*

$\texttt{OS}-\texttt{FOL}$ formulae are evaluated via a notion of structure :

Definition 2 (Structure) *Given* $n \in \mathbb{N}$*, a* n-sorted structure *is* $\mathbf{St} = (Dom, Rel, Cons)$ *where :*
– $Dom = \{D_1, \ldots, D_n\}$ *are the (non-empty) domains,*
– $Rel = \{R_1, \ldots, R_m\}$ *are relations over the domains,*
– $Cons = \{c_1, \ldots, c_l\}$ *are constants in the domains.*

Example 2 *A structure associated with the* $\texttt{OS}-\texttt{FOL}$ *from Example 1 is* $\mathbf{St} = (Dom, Rel, Cons)$ *where*
– $Dom = \{D_1 \ldots D_9\}$ *are the sets of all individuals of the various types (*e.g. *D_1 is the set of mammals, corresponding to the sort symbol m ; D_2 is the set of birds, corresponding to the sort symbol b ; etc),*
– $Rel = \{R_1, \ldots, R_5\}$ *are the relations corresponding to the predicate symbols (*e.g. *R_1 identifies winged animals,. . .)*
– $Cons = \{Zazu, Tweety, Dogmatix\}$ *are respectively a particular bird (an element of the domain D_2 associated with the sort b), a particular penguin (an element of D_6 associated with the sort p) and a particular dog (an element of D_4 associated with the sort d).*

Classical first order logic formulae can be evaluated via 1-sorted structures. For this reason, any fragment of first order logic is captured by $\texttt{OS}-\texttt{FOL}$. Now, we show how $\texttt{OS}-\texttt{FOL}$ formulae are interpreted.

Definition 3 (Interpretation) *An* interpretation $\mathbf{I_{St}}$ over a structure $\mathbf{St}$ *assigns to elements of the* $\texttt{OS}-\texttt{FOL}$ *vocabulary some values in the structure* $\mathbf{St}$*. Formally,*
– $\mathbf{I_{St}}(s_i) = D_i$*, for* $i \in \{1, \ldots, n\}$ *s.t. for each* $s_i, s_j \in \mathbf{So}$*, if* $s_i \preceq s_j$ *then* $\mathbf{I_{St}}(s_i) \subseteq \mathbf{I_{St}}(s_j)$ *(each sort symbol is assigned to a domain s.t. the sub-type relations are respected),*
– $\mathbf{I_{St}}(P_i) = R_i$*, for* $i \in \{1, \ldots, m\}$ *(each predicate symbol is assigned to a relation),*
– $\mathbf{I_{St}}(a_i) = c_i$*, for* $i \in \{1, \ldots, l\}$ *(each constant symbol is assigned to a constant value). As a shorthand, we write* $\mathbf{I_{St}}((s_1, \ldots, s_k)) = \mathbf{I_{St}}(s_1) \times \cdots \times \mathbf{I_{St}}(s_k)$*. Then satisfaction of formulae is recursively defined by :*
– $\mathbf{I_{St}} \models P_i(x_1, \ldots, x_k)$*, where* $(x_1, \ldots, x_k) \in \mathbf{I_{St}}((s_1, \ldots, s_k))$ *with* $\texttt{sort}(x_i) = s_i$ *for each* $i \in \{1, \ldots, k\}$*, iff* $(x_1, \ldots, x_k) \in R_i$,
– $\mathbf{I_{St}} \models \exists x^{s_i} \phi$ *iff* $\mathbf{I_{St}, x^{s_i} \leftarrow v} \models \phi$ *for some* $v \in D_i$,
– $\mathbf{I_{St}} \models \forall x^{s_i} \phi$ *iff* $\mathbf{I_{St}, x^{s_i} \leftarrow v} \models \phi$ *for each* $v \in D_i$,
– $\mathbf{I_{St}} \models \phi \wedge \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *and* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \phi \vee \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *or* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \neg\phi$ *iff* $\mathbf{I_{St}} \not\models \phi$,
where $\mathbf{I_{St}, x^{s_i} \leftarrow v}$ *is a modified version of* $\mathbf{I_{St}}$ *s.t. the variable* x^{s_i} *is replaced by a value v in the domain* D_i *corresponding to the sort symbol* s_i*. Finally, if* Φ *is a set of formulae, then* $\mathbf{I_{St}} \models \Phi$ *iff* $\mathbf{I_{St}} \models \phi$ *for each* $\phi \in \Phi$.

Observe that Definition 3 does not specify the satisfaction of implications and equivalences, but they can be defined as usual by $(\phi \rightarrow \psi) \equiv (\neg\phi \vee \psi)$, and $(\phi \leftrightarrow \psi) \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. We use $\texttt{Mod}(\Phi)$ to denote the set of interpretations satisfying a set of formulae Φ, and we call Φ *consistent* if $\texttt{Mod}(\Phi) \neq \emptyset$.

Example 3 *Continuing Example 1, we define* $\mathbf{I_{St}}$ *by :*
– $\mathbf{I_{St}}(m) = D_1$, $\mathbf{I_{St}}(b) = D_2$, . . . , $\mathbf{I_{St}}(pl) = D_9$,
– $\mathbf{I_{St}}(hW) = R_1, \ldots, \mathbf{I_{St}}(p) = R_5$,
– $\mathbf{I_{St}}(Z) = Zazu$, $\mathbf{I_{St}}(T) = Tweety$, $\mathbf{I_{St}}(D) = Dogmatix$.
The formula $\phi = \forall x^b hW(x^b)$ *is satisfied by* $\mathbf{I_{St}}$*, since all elements of the domain* D_2 *associated with the sort symbol b actually have wings. On the contrary, consider the set of formulae* $\Phi = \{\forall x^b f(x^b), \forall x^p \neg f(x^p)\}$*. This set of formulae is not satisfied, because* $p \prec b$*, and so the domains satisfy* $D_6 \subset D_2$*, meaning that all penguins are birds. Then, from* Φ *we can deduce that any penguin can fly (because of the first formula) and cannot fly (because of the second formula) at the same time. So, this formula is not satisfied by* $\mathbf{I_{St}}$*. Notice that we could not define an interpretation* $\mathbf{I'_{St}}$ *s.t.* $\mathbf{I'_{St}}(Z) = Tweety$ *and* $\mathbf{I'_{St}}(T) = Zazu$*, since Zazu is a bird, and T has the sort p (*i.e. *it can only be a penguin, not any kind of bird).*

Now we introduce the concept of instantiation, *i.e.* grounded formulae which are compatible with a given $\texttt{OS}-\texttt{FOL}$ formula.

Definition 4 (Instantiation) *Given* Φ *a set of* $\texttt{OS}-\texttt{FOL}$ *formulae and* $\mathbf{I_{St}}$ *an interpretation over a structure* $\mathbf{St}$*, the set of* instantiations *of* Φ *is defined recursively by :*

– $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\Phi\}$ *if* $\Phi = \{\phi\}$*, where* ϕ *is a grounded formula s.t.* $\mathbf{I_{St}} \models \phi$*,*
– $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_{x^s \leftarrow v} \mid \mathbf{I_{St}} \models \phi_{x^s \leftarrow v}, v \in \mathbf{I_{St}}(s)\})\}$ *if* $\Phi = \{\forall x^s \phi\}$*,*
– $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_{x^s \leftarrow v} \mid \mathbf{I_{St}} \models \phi_{x^s \leftarrow v}, v \in V\}) \mid \emptyset \subset V \subseteq \mathbf{I_{St}}(s)\}$ *if* $\Phi = \{\exists x^s \phi\}$*,*
– $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{I_1 \cup I_2 \mid I_1 \in \texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_1\}), I_2 \in \texttt{Inst}_{\mathbf{I_{St}}}(\Phi_2), \mathbf{I_{St}} \models I_1 \cup I_2\}$ *if* $\Phi = \{\phi_1\} \cup \Phi_2$ *with* $\phi_1 \notin \Phi_2$*, where* $\phi_{x^s \leftarrow v}$ *is the formula* ϕ *s.t. all the occurrences of the variable* x^s *are replaced by the value* v *(from the domain associated with the sort* s*).*

The idea is that formulae with quantified variables may be instanciated in various ways. Assuming that the domain of a variable x is $\{A, B\}$, then the formula $\exists x P(x)$ means that either $P(A)$ is true, or $P(B)$, or both at the same time. And $\forall x P(x)$ means that $P(A)$ and $P(B)$ are both true. This is what is captured by the notion of instantiation. Moreover, an instantiation is consistent because of the constraint $\mathbf{I_{St}} \models I_1 \cup I_2$ in the last part of the definition. This constraint means that, if *e.g.* we consider the set of formulae $\{\exists x P(x), \exists x \neg P(x)\}$, then we keep the instantiations where $P(A)$ is true and $P(B)$ is false, or the opposite. But we exclude situations where $P(A)$ is both true (because of the first formula) and false (because of the second formula) at the same time.

Example 4 *Consider the set of formulae* $\Phi = \{\phi_1 = \exists x^b w(x^b), \phi_2 = \forall x^b w(x^b) \rightarrow \neg f(x^b)\}$*. We assume here that the domain associated with the sort* b *is the set* $\{Tweety, Zazu\}$*. Applying Definition 4,* $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{I_1 \cup I_2 \mid I_1 \in \texttt{Inst}_{\mathbf{I_{St}}}(\{\exists x^b w(x^b)\}), I_2 \in \texttt{Inst}_{\mathbf{I_{St}}}(\{\forall x^b w(x^b) \rightarrow \neg f(x^b)\}), \mathbf{I_{St}} \models I_1 \cup I_2\}$.

We start with the first formula, i.e. $\phi_1 = \exists x^b w(x^b)$. $\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_1\}) = \{\{w(Tweety)\}, \{w(Zazu)\}, \{w(Tweety), w(Zazu)\}\}$. *For* $\phi_2 = \forall x^b w(x^b) \rightarrow \neg f(x^b)$*, there is only one possible instantiation :* $\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_2\}) = \{\{w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}\}$.

We conclude that $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) =$
$\{\{w(Tweety), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}, \{w(Zazu), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}, \{w(Tweety), w(Zazu), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}\}$

From the notions of structure and interpretation, we can define the consequence relation over $\texttt{OS} - \texttt{FOL}$ formulae.

Definition 5 (Consequence Relation) *Let* ϕ *and* ψ *be two* $\texttt{OS} - \texttt{FOL}$ *formulae. We say that* ψ is a consequence of ϕ*, denoted by* $\phi \vdash \psi$*, if for any structure* $\mathbf{St}$*, and any interpretation* $\mathbf{I_{St}}$ *over* $\mathbf{St}$*,* $\mathbf{I_{St}} \models \phi$ *implies* $\mathbf{I_{St}} \models \psi$*. Two formulae* ϕ*,* ψ *are* equivalent *(denoted* $\phi \equiv \psi$*) iff* $\phi \vdash \psi$ *and* $\psi \vdash \phi$*.*

A *logic* is a pair $(L, |\sim)$ where L is a set of formulae (*i.e.* a language) and $\vdash \subseteq L \times L$ is a consequence relation. We can lift the consequence relation to sets of formulae by $\{\psi_1, \ldots, \psi_n\} |\sim \phi$ if $\psi_1 \wedge \cdots \wedge \psi_n |\sim \phi$. An example of logic consists of $(\mathcal{L}, \vdash)$ where $\mathcal{L}$ is an $\texttt{OS} - \texttt{FOL}$ language following Definition 1 and $\vdash$ is the consequence relation from Definition 5. Classical logic can be used to define arguments, *i.e.* logic-based representation of reasons supporting a specific conclusion. Logical arguments usually need to satisfy some constraints [12] :

Definition 6 (Logical Argument) *An* argument *built under a logic* $(L, |\sim)$ *is a pair* $\langle \Phi, \phi \rangle$*, where* [2] $\Phi \subseteq_f L$ *and* $\phi \in L$*, s.t.* Φ *is consistent,* $\Phi |\sim \phi$*, and* $\nexists \Phi' \subset \Phi$ *s.t.* $\Phi' |\sim \phi$*. An argument* $A = \langle \Phi, \phi \rangle$ *is* trivial *iff* $\Phi = \emptyset$ *and* $\phi \equiv \top$*.* Φ *is called the support of the argument (*$\texttt{Supp}(A) = \Phi$*) and* ϕ *its conclusion (*$\texttt{Conc}(A) = \phi$*). The set of all arguments built under* $(L, \vdash)$ *is denoted* $\texttt{Arg}(L)$*.*

In the rest of this paper, we assume a $\texttt{OS} - \texttt{FOL}$ language $\mathcal{L}$, and we focus on the set of arguments $\texttt{Arg}(\mathcal{L})$ built under the logic $(\mathcal{L}, \vdash)$ as defined previously.

Example 5 *Let* A_1 *and* A_2 *be two examples of arguments :*
$A_1 = \langle \{\exists x^b w(x^b), \forall x^b w(x^b) \rightarrow \neg f(x^b)\}, \exists x^b \neg f(x^b) \rangle$
$A_2 = \langle \{p(Tweety), \forall x^b p(x^b) \rightarrow \neg f(x^b)\}, \neg f(Tweety) \rangle$

Note that two sets of formulae $\Phi, \Psi \subseteq_f \mathcal{L}$ are *equivalent*, denoted by $\Phi \cong \Psi$, iff there is a bijection $f : \Phi \rightarrow \Psi$ s.t. $\forall \phi \in \Phi$, $\phi \equiv f(\phi)$. We use this restricted equivalence notion to avoid equivalences that could be false due to incorrect information. For example the sets $\{Square(a), Square(a) \rightarrow Rectangle(a)\}$ and $\{Rectangle(a), Rectangle(a) \rightarrow Square(a)\}$ should not be equivalent. However, we may want to consider that a set of formulae is equivalent with the conjunction of its elements (*e.g.* $\{P(a), Q(a)\}$ and $\{P(a) \wedge Q(a)\}$ are equivalent). To make them equivalent, we borrow the method used in [7]. We transform every formula into a CNF, then we split it into a set containing its clauses. In our approach, we consider one CNF per formula. For that purpose, we will use a finite sub-language $\mathcal{F}$ that contains one formula per equivalent class and the formula should be in CNF.

Definition 7 (Finite CNF over Language $\mathcal{F}$)
Let $\mathcal{F} \subset_f \mathcal{L}$ *s.t.* $\forall \phi \in \mathcal{L}$*, there is a unique* $\psi \in \mathcal{F}$ *s.t.* $\phi \equiv \psi$*,* $\texttt{Lit}(\phi) = \texttt{Lit}(\psi)$ *and* ψ *is a CNF formula. We define* $\texttt{CNF}(\phi) = \psi$*.*

While we do not specify the elements of $\mathcal{F}$, we use concrete formulae in the examples, and they are assumed to belong to $\mathcal{F}$.

Now we introduce $\texttt{UC}(\Phi)$ as the representation of the formulae in Φ as one set of clauses. Intuitively, recall that any formula can be seen as a set of clauses, associated with a sequence of quantifiers. A set of formulae can then be seen

2. $X \subseteq_f Y$ means X is a finite subset of Y

as set of clauses and a sequence of quantifiers, such that variables are renamed to avoid ambiguities. As an example, assume $\phi_1 = \exists x P(x) \wedge Q(x)$ and $\phi_2 = \exists x Q(x) \vee R(x)$. We have $\mathtt{UC}(\{\phi_1, \phi_2\}) = \exists x, x' \{P(x), Q(x), Q(x') \vee R(x')\}$. Formally, for $\Phi = \{Q_{\phi_i}\phi_i \mid i \in \mathbb{N}\} \subseteq_f \mathcal{F}$, where ϕ_i is a non-quantified CNF formula (i.e. a set of clauses $\mathtt{CNF}(\psi)$ for some $\psi \in \mathcal{F}$), and Q_{ϕ_i} is the sequence of quantifiers associated with ϕ_i, we define $\mathtt{UC}(\Phi) = (Q^*_{\phi_1} \ldots Q^*_{\phi_n}, \bigcup_{\phi\in\Phi} \bigcup_{\delta\in\phi} \delta^*)$, where a renaming is applied to each clause (δ^*) and each sequence of quantifiers ($Q^*_{\phi_i}$) in order to guarantee that no variable is shared between quantifiers $Q^*_{\phi_i}$ and $Q^*_{\phi_j}$ (with $i \neq j$) or between clauses coming from different formulae ϕ_i and ϕ_j (with $i \neq j$). We simply write $\mathtt{UC}(\phi)$ instead of $\mathtt{UC}(\{\phi\})$, for $\phi \in \mathcal{F}$.

Implicitly, in the rest of the paper, we consider $\mathtt{UC}(\Phi)$ as the set made of a single formula such that the sequence of quantifiers is the concatenation of $Q^*_{\phi_1} \ldots Q^*_{\phi_n}$ and the non-quantified part is the CNF formula corresponding to the set of clauses $\bigcup_{\phi\in\Phi} \bigcup_{\delta\in\phi} \delta^*$.

Note that $\mathtt{UC}(\{P(a), Q(a)\}) = \mathtt{UC}(\{P(a) \wedge Q(a)\}) = \{P(a), Q(a)\}$ or with some quantifiers $\mathtt{UC}(\{\forall x \exists y\ P(x,y), \forall x\ Q_1(x) \vee Q_2(x)\}) = \mathtt{UC}(\{\forall x_1 \exists x_2\ P(x_1,x_2) \wedge \forall x_3\ Q_1(x_3) \vee Q_2(x_3)\}) = \{\forall x_1 \exists x_2\ P(x_1,x_2), \forall x_3\ Q_1(x_3) \vee Q_2(x_3)\}$.

Let us now introduce the notion of compiled argument.

Definition 8 (Compiled Argument) *The* compilation *of* $A \in \mathtt{Arg}(\mathcal{L})$ *is* $A^* = \langle \mathtt{UC}(\mathtt{Supp}(A)), \mathtt{Conc}(A)\rangle$.

Example 6 *Consider* $A, B, C \in \mathtt{Arg}(\mathcal{L})$ *such that*
$A = \langle\{P(a) \wedge Q(a) \wedge Q(b)\}, P(a) \wedge Q(a)\rangle$,
$B = \langle\{P(a) \wedge Q(a)\}, P(a) \wedge Q(a)\rangle$, *and*
$C = \langle\{P(a), Q(a)\}, P(a) \wedge Q(a)\rangle$.
The compilations of the three arguments A, B, C *are :*
$A^* = \langle\{P(a), Q(a), Q(b)\}, P(a) \wedge Q(a)\rangle$,
$B^* = \langle\{P(a), Q(a)\}, P(a) \wedge Q(a)\rangle$, *and*
$C^* = \langle\{P(a), Q(a)\}, P(a) \wedge Q(a)\rangle$.

We can see in Example 6 that argument A is not concise, meaning that it has irrelevant information ($Q(b)$) for implying its conclusion. As it was shown in [7], using clausal arguments ensure that the arguments are concise.

Definition 9 (Equivalent Arguments) *Two arguments* $A, B \in \mathtt{Arg}(\mathcal{L})$ *are equivalent, denoted by* $A \approx B$, *iff* $\mathtt{UC}(\mathtt{Supp}(A)) = \mathtt{UC}(\mathtt{Supp}(B))$ *and* $\mathtt{UC}(\mathtt{Conc}(A)) = \mathtt{UC}(\mathtt{Conc}(B))$. *We denote by* $A \not\approx B$ *when* A *and* B *are not equivalent.*

Definition 10 (Sub-argument) *Given two arguments* $A = \langle\Phi, \phi\rangle$ *and* $B = \langle\Psi, \psi\rangle$, *we say that* A *is a sub-argument of* B *if* $\mathtt{UC}(\Phi) \subseteq \mathtt{UC}(\Psi)$.

2.2 Binary Similarity Measure between OS − FOL Argument

A similarity measure is used to indicate whether two arguments are similar or not, *i.e.* whether they share some parts of the reasoning mechanism used to build the arguments.

Definition 11 (Similarity Measure) *Let* $\mathbb{X}$ *be a set of objects. A similarity measure on* $\mathbb{X}$, *denoted by* $\mathtt{sim}^{\mathbb{X}}$, *is a function from* $\mathbb{X} \times \mathbb{X}$ *to* $[0, 1]$.

In this section, we focus on similarity measures over arguments, *i.e.* $\mathbb{X} = \mathtt{Arg}(\mathcal{L})$. Intuitively, $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}(A, B)$ is close to 0 if the difference between A and B is important, while it is close to 1 if the arguments are similar. Several principles that similarity measures should satisfy have been discussed in the literature [4, 8, 7]. Some of the principles (Maximality, Symmetry, Substitution, and Syntax Independence) can be stated exactly as in the literature [7], since they do not concern the internal structure of the arguments. Notice that some authors have argued against the fact that a similarity measures should absolutely satisfy symmetry [28, 19]. Some of the principles can be stated exactly as in the literature [7], since they do not concern the internal structure of the arguments. It is the case of these principles : Maximality states that the similarity between an argument and itself should be maximal ; Symmetry states that the similarity measure should be symmetric [3] ; Substitution states that two fully similar arguments should be equally similar to any third argument ; and Syntax Independence states that similarity between arguments should be independent from the syntax. For the other ones, we may need to adapt them to our OS − FOL-based arguments.

First, we adapt the Minimality principle. It states that, if two arguments do not have anything in common in their content, then their degree of similarity should be minimal. While, in propositional logic, determining the set of common propositional variables is enough, here we need to consider (domains of) predicates and constants. We do not consider variables here since they are used in the context of quantifiers : there is no reason to assume that there is something common between $\forall x, P(x)$ and $\forall x, Q(x)$.

Before presenting the Minimality principle, let us introduce some useful notations. Given a formula ϕ, $\mathtt{Dom}(\phi) = \bigcup_{P\in\mathtt{Pred}(\phi)} \mathtt{sort}(P)$ represents the domains of the predicates in ϕ (or, more precisely, the sort symbols associated with these domains). We extend the notation to $\mathtt{Dom}(\Phi) = \bigcup_{\phi\in\Phi} \mathtt{Dom}(\phi)$ for Φ a set of formulae.

Principle 1 (Minimality)
A similarity measure $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}$ *satisfies* Minimality *iff for all* $A, B \in \mathtt{Arg}(\mathcal{L})$, *if*

3. Notice that some authors have argued against the fact that a similarity measures should absolutely satisfy symmetry [28, 19].

1. one of A,B is not trivial,
2. $\forall s_i \in \texttt{Dom}(\texttt{Supp}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Supp}(B))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
3. $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Conc}(B))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, B) = 0$.

The first condition excludes the case where the arguments have no formula in the support and therefore no sort to compare and the second and third conditions ensure that each argument has completely different information.

The second (resp. third) principle states that the more an argument shares formulae in its support (resp. conclusion) with an another one, the higher is their similarity. For these principles, we need to introduce the notation $\mathbb{C}$ which represents the set of all grounded clauses in $\texttt{OS} - \texttt{FOL}$.

Notice that we consider in the two next principles only arguments having no irrelevant information (i.e., $A^*, B^*, C^* \in \texttt{Arg}(\mathcal{L})$) allowing safe handling of their similarity. The first conditions allow us to isolate the specific behaviours on second and third conditions. For Principle 2 focusing on supports we ensure that we have identical or completely different conclusions such that it does not contradict that (A, B) is more similar than (A, C). We cannot, as in Principle 3, use the fact that the conclusions of B and C are equivalent as this would prevent conditions 2 and 3 from being satisfied (due to the minimality of the supports of an argument, e.g. the case of one support included in another is not possible). Please note that the constraints $C_A \setminus B_A \subseteq \mathbb{C}$ ensure that the distinct elements in C cannot have similarity with A.

Principle 2 (Monotony – Strict Monotony)
A similarity measure $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}$ *satisfies* Monotony *iff for all* $A, B, C, A^*, B^*, C^* \in \texttt{Arg}(\mathcal{L})$, *if*
1. $\texttt{UC}(\texttt{Conc}(A)) = \texttt{UC}(\texttt{Conc}(B))$ *or* $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Conc}(C))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
2. $\texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(C)) \subseteq \texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(B))$,
3. for $B_A = \texttt{UC}(\texttt{Supp}(B)) \setminus \texttt{UC}(\texttt{Supp}(A))$ *and* $C_A = \texttt{UC}(\texttt{Supp}(C)) \setminus \texttt{UC}(\texttt{Supp}(A))$, $B_A \subseteq C_A$, $C_A \setminus B_A \subseteq \mathbb{C}$ *and* $\forall s_i \in \texttt{Dom}(\texttt{Supp}(A))$, $\nexists s_j \in \texttt{Dom}(C_A \setminus B_A)$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, B) \geq \texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, C)$.
(Monotony)
– If the inclusion in condition 2. is strict or, $\texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(C)) \neq \emptyset$ *and* $B_A \subset C_A$,
then $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, B) > \texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, C)$.
(Strict Monotony)

Principle 3 (Dominance – Strict Dominance)
A similarity measure $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}$ *satisfies* Dominance *iff for all* $A, B, C, A^*, B^*, C^* \in \texttt{Arg}(\mathcal{L})$, *if*
1. $\texttt{UC}(\texttt{Supp}(B)) = \texttt{UC}(\texttt{Supp}(C))$,
2. $\texttt{UC}(\texttt{Conc}(A)) \cap \texttt{UC}(\texttt{Conc}(C)) \subseteq \texttt{UC}(\texttt{Conc}(A)) \cap \texttt{UC}(\texttt{Conc}(B))$,
3. for $B_A = \texttt{UC}(\texttt{Conc}(B)) \setminus \texttt{UC}(\texttt{Conc}(A))$ *and* $C_A = \texttt{UC}(\texttt{Conc}(C)) \setminus \texttt{UC}(\texttt{Conc}(A))$, $B_A \subseteq C_A$, $C_A \setminus B_A \subseteq \mathbb{C}$ *and* $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(C_A \setminus B_A)$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, B) \geq \texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, C)$.
(Dominance)
– If the inclusion in cond. 2. is strict or, $\texttt{UC}(\texttt{Conc}(A)) \cap \texttt{UC}(\texttt{Conc}(C)) \neq \emptyset$ *and* $B_A \subset C_A$, *then* $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, B) > \texttt{sim}^{\texttt{Arg}(\mathcal{L})}(A, C)$.
(Strict Dominance)

3 Similarity Models

To define the similarity between two arguments, we will split the reasoning in several steps, corresponding to the different levels used in the construction of the arguments. At each level, different similarity measures can be used to compare the objects, and various aggregation functions can then be used to go from the comparison of objects to the comparison of sets of objects (leading to the next level). This level structure is based on the fact that our arguments are built from CNF formulae. More precisely,
Level 1 : compute the similarity between two literals, by combining the similarity between their polarity, the predicate involved, and the predicates parameters (Section 3.1);
Level 2 : then we use the previous level and aggregate the result of comparing literals in order to compare grounded clauses (Section 3.2);
Level 3 : next, we aggregate the similarity between grounded clauses to obtain the similarity between sets of grounded clauses (Section 3.3);
Level 4 : finally, we can define the similarity between sets of instantiations, since each instantiation is a set of grounded clauses (Section 3.4).
The similarity between two arguments is obtained by computing the similarity between the instantiations of their supports and the similarity between their conclusions, so Level 4 is the last level of abstraction that we need.

3.1 Similarity between literals

Recall that a literal is a predicate with or without a negation operator "¬". To know how similar are two literals, we compute the similarity between two atoms (*i.e.* without the literals' polarity) and combine these scores according to the polarity. At the level of atoms, we identify two parameters influencing the similarity : the value of the predicates and those of their vectors of parameters. Thus the similarity between two atoms can be seen as a combination of three functions : c to compute the similarity between two vectors of constants, p between two predicates and g to aggregate these scores.

Definition 12 (Similarity between Atoms)
Let $\mathbf{c} : \bigcup_{j,k=1}^{+\infty} \mathbf{C}^j \times \mathbf{C}^k \to [0,1]$ *be a similarity measure between a pair of vectors of constants,* $\mathbf{p} : \mathbf{P} \times \mathbf{P} \to [0,1]$ *be a similarity measure between a pair of predicates and* $\mathbf{g} : [0,1] \times [0,1] \to [0,1]$ *be an aggregation function. Given two atoms* $A_1 = P_1(a_1, \ldots, a_j)$ *and* $A_2 = P_2(b_1, \ldots, b_k)$*, to compute the similarity score between* A_1 *and* A_2 *we define* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle} : \mathbf{A} \times \mathbf{A} \to [0,1]$ *s.t.* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}(A_1, A_2) = \mathbf{g}\Big(\mathbf{p}(\mathtt{Pred}(A_1), \mathtt{Pred}(A_2)), \mathbf{c}\big(\mathtt{Para}(A_1), \mathtt{Para}(A_2)\big)\Big)$.

A possible $\mathbf{p}$ is the function returning 1 if the predicates are the same, 0 otherwise.

Definition 13 (Function Equal) *Let* x, y *be two arbitrary objects. The function* $\mathtt{eq} : \mathbb{X} \times \mathbb{X} \to \{0,1\}$ *is defined by* $\mathtt{eq}(x,y) = 1$ *if* $x = y$*; or* $\mathtt{eq}(x,y) = 0$ *otherwise.*

We propose an instance of $\mathbf{c}$ suited to vectors of objects. Other methods could be used and are kept for future work.

Definition 14 (Pointwise Similarity)
Let $X = \langle x_1, \ldots, x_j \rangle, Y = \langle y_1, \ldots, y_k \rangle$ *be arbitrary vectors of objects. The* pointwise similarity *between X and Y is :*

$$\mathtt{pws}(X,Y) = \begin{cases} 1 & X = Y = \emptyset \\ \frac{\sum_{i=1}^{\min(j,k)} \mathtt{eq}(x_i, y_i)}{\max(j,k)} & \text{otherwise} \end{cases}$$

Having a similarity score between two atoms, we propose to use the polarities as binary factors of acceptance or not of the similarity between atoms.

Definition 15 (Similarity between Literals) *Consider two literals* $l_1, l_2 \in \mathbf{L}$*, such that the respective atoms are* A_1 *and* A_2*. We define* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle} : \mathbf{L} \times \mathbf{L} \to [0,1]$*, the similarity measure between two literals according to a similarity measure between atoms* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$ *s.t. :* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}(l_1, l_2) =$

$$\begin{cases} \mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}(A_1, A_2) & \text{if } \mathtt{Pol}(l_1) = \mathtt{Pol}(l_2) \\ 0 & \text{otherwise} \end{cases}$$

Example 7 $\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P(A,B), \neg P(A,C)) = 0$ *because the polarity is not the same. Conversely, we have* $\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P(A,B), P(A,C)) = \frac{1}{2}$ *because :*
$\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P(A,B), P(A,C))$
$= \mathtt{simA}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P(A,B), P(A,C))$
$= \mathtt{min}(\mathtt{eq}(P,P), \mathtt{pws}(\langle A,B\rangle, \langle A,C\rangle))$
$= \mathtt{min}(1, \frac{\mathtt{eq}(A,A)+\mathtt{eq}(B,C)}{2}) = \mathtt{min}(1, \frac{1}{2}) = \frac{1}{2}$.

3.2 Similarity between grounded clauses

From the level two of the definition of our similarity measures on arguments, we will need several mathematical tools that can be defined in an abstract way. In this part, we apply these tools only for level 2 (the comparison of two CNF formulae), but they will be applicable also at the next levels. Let us start with the notion of aggregation function.

Definition 16 (Aggregation Function) *Let* $\mathbb{X}$ *be a set of objects and* $\{x_1, x_2, \ldots\} \subseteq \mathbb{X}$*.We say that* $\oplus$ *is an aggregation function if* $\forall k \in \mathbb{N}$*,* $\oplus$ *is a mapping* $[0,1]^k \to [0,1]$ *such that :*
– *if* $x_i \geq x_i'$*, then* $\oplus(x_1, \ldots, x_i, \ldots, x_k) \geq \oplus(x_1, \ldots, x_i', \ldots, x_k)$ **(non-decreasingness)**
– $\oplus(0, \ldots, 0) = 0$ **(weak minimality)**
– $\forall i \in \{1, \ldots, k\}, \oplus(x_i) = x_i$ **(identity)**

These properties are satisfied by *e.g.* `min`, `max` and `avg`.

Now we introduce the notion of *membership* function which expresses how much an object is similar to the elements of a set.

Definition 17 (Membership Function) *Given* $\mathbb{X}$ *a set of objects,* $x \in \mathbb{X}$ *an object,* $X \subseteq \mathbb{X}$*,* $\oplus$ *an aggregation function and* `sim` *a similarity measure the* membership function *of x in X,* $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}} : \mathbb{X} \times 2^{\mathbb{X}} \to [0,1]$ *is defined by :* $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x, X) = \oplus_{x' \in X}(\mathtt{sim}^{\mathbb{X}}(x, x'))$.

Let us note that classical set-membership can be captured by $\varepsilon_{\mathtt{max},\mathtt{eq}}$ where `eq` is the equality function from Definition 13. Now we can evaluate how much a literal is similar to a clause, *i.e.* a set of literals : given $l \in \mathbf{L}$ a literal, $L \subseteq \mathbf{L}$ a set of literals and $\oplus^1$ an aggregation function, we define the function $\mathtt{s}^{\mathbf{L}} = \varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}}$. Then, the similarity between two grounded clauses is computed by $\mathtt{simC}^{\mathtt{s}^{\mathbf{L}}}$.

Definition 18 (Membership of a literal in a set of literals)
Let $l \in \mathbf{L}$ *be a literal,* $L \subseteq \mathbf{L}$ *be a set of literals and* $\oplus^1$ *be an aggregation function. We define the membership of a literal in a set of literals by the function* $\varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{s}^{\mathbf{L}}} : \mathbf{L} \times 2^{\mathbf{L}} \to [0,1]$ *s.t. :*

$$\varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{s}^{\mathbf{L}}}(l, L) = \oplus^1_{l' \in L}(\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}(l, l'))$$

Definition 19 (Similarity measure between two clauses)
Let $\delta_1 = l_1 \vee \ldots \vee l_j, \delta_2 = l_1' \vee \ldots l_k' \in \mathtt{OS-FOL}$ *be two grounded clauses. The similarity measure between two grounded clauses,* $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{s}^{\mathbf{L}}}} : \mathtt{OS-FOL} \times \mathtt{OS-FOL} \to [0,1]$.

Roughly speaking, what we mean in Definition 19 (and subsequent similar definitions) is that the similarity between two grounded clauses must be computed using a similarity measure (in the sense of Definition 11), and ideally this measure should use the membership function $\varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{s}^{\mathbf{L}}}$ to compare a given literal with a set of literals (*i.e.* with a grounded clause). But at this level of abstraction, we do not explicitly defined one function realizing this computation, Def. 19 characterizes the general meaning of what a similarity measure between clauses should be. In the rest of this paper, we will use one concrete approach to define similarity measures, namely Tversky's ratio model [28], but other approaches could be used instead as soon as they satisfy the requirements of Def. 19 (and Def. 11).

Tversky's ratio model [28] is a general similarity measure which encompasses different well known similarity measure as the Jaccard measure [18], Dice measure [15], Sorensen one [27], Symmetric Anderberg [9] and Sokal and Sneath 2 [26]. We propose to extend it in two different ways. Firstly, instead of using the usual operators of membership of an element to a set, we propose to use our parameterisable membership function ε (see Definition 17). Then a new parameter γ allows us to have a lower evaluation between a set of literals than a set of clauses (or instantiations), i.e. when sets of objects are interpreted disjunctively or conjunctively.

Definition 20 (Extended Tversky Measure) *Let $X, Y \subseteq \mathbb{X}$ be arbitrary sets of objects. Let $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}$ be a membership function with $\oplus$ an aggregation function and* sim *a similarity measure. We denote by* avg *the average function. Let us consider*

— $a = \mathtt{avg}\Big(\sum\limits_{x\in X} \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x,Y), \sum\limits_{y\in Y} \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(y,X)\Big)$,

— $b = \sum\limits_{x\in X} (1 - \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x,Y))$,

— $c = \sum\limits_{y\in Y} (1 - \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(y,X))$,

— $\alpha, \beta \in [0, +\infty[$ *and* $\gamma \in]0, +\infty[$.

The extended Tversky measure *between X and Y is :*

$$\mathtt{Tve}^{\alpha,\beta,\gamma,\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}}(X,Y) = \begin{cases} 1 & \text{if } X = Y = \emptyset \\ \left(\frac{a}{a+\alpha\cdot b+\beta\cdot c}\right)^{\gamma} & \text{otherwise} \end{cases}$$

Classical similarity measures (see Table 1 in [4] for the definitions) can be obtained with $\alpha = \beta = 2^{-n}$ and $\gamma = 1$ and the classical set-membership. In particular, the Jaccard measure (i.e. jac) is obtained with $n = 0$, Dice (i.e. dic) with $n = 1$, Sorensen (i.e. sor) with $n = 2$, Anderberg (i.e. adb) with $n = 3$, and Sokal and Sneah 2 (i.e. $\mathtt{ss}_2$) with $n = -1$. Under some reasonable assumptions, Tversky measure s.t. $\alpha = \beta$ are symmetric.

Proposition 1 *For any $X, Y \subseteq \mathbb{X}$, any $\gamma \in]0, +\infty[$, any membership function $\varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$ s.t.* sim *is symmetric, we have $\mathtt{Tve}^{\alpha,\alpha,\otimes}(X,Y) = \mathtt{Tve}^{\alpha,\alpha,\otimes}(Y,X)$, where $\otimes = \gamma, \varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$.*

In the rest of the paper we will focus our study on the membership function using the aggregator function max. Table 1 denotes the set of parametric (non-)symmetric extended versions of the well known similarity measures, where fixing α and β corresponds to choosing among Jaccard, Dice, Sorensen, Anderberg, or Sokal and Sneah.

The other parameters of the different similarity measures are only the coefficient γ and the similarity function $\mathtt{sim}^{\mathbb{X}}$. Let us prove that any such measure satisfies some intuitive properties : two sets are maximally similar if they are identical (in the symmetric case), or at least included in one another (non-symmetric case).

Symmetric Measures	Non-Symmetric Measures
$\mathtt{Tve}^{1,1,\otimes}(X,Y) = \mathtt{jac}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,1,\otimes}(X,Y) = \mathtt{ns\text{-}jac}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.5,0.5,\otimes}(X,Y) = \mathtt{dic}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.5,\otimes}(X,Y) = \mathtt{ns\text{-}dic}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.25,0.25,\otimes}(X,Y) = \mathtt{sor}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.25,\otimes}(X,Y) = \mathtt{ns\text{-}sor}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.125,0.125,\otimes}(X,Y) = \mathtt{adb}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.125,\otimes}(X,Y) = \mathtt{ns\text{-}adb}^{\ominus}(X,Y)$
$\mathtt{Tve}^{2,2,\otimes}(X,Y) = \mathtt{ss}_2{}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,2,\otimes}(X,Y) = \mathtt{ns\text{-}ss}_2{}^{\ominus}(X,Y)$

TABLE 1 – Set of parametric (non-)symmetric measures, where $\otimes$ is $\gamma, \varepsilon^{\mathbb{X}}_{\mathtt{max},\mathtt{sim}}$ and $\ominus$ is $\gamma, \mathtt{sim}^{\mathbb{X}}$

Proposition 2 *If $\mathtt{sim}^{\mathbb{X}}$ satisfies Maximality [4] and $\otimes = \gamma, \varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$, then, for any $\gamma \in]0, +\infty[$, $\alpha \neq 0$, if*
– $Y = X$ then $\mathtt{Tve}^{\alpha,\alpha,\otimes}(X,Y) = 1$ (symmetric case),
– $Y \subseteq X$ then $\mathtt{Tve}^{0,\alpha,\otimes}(X,Y) = 1$ (non-symmetric case).

Example 8 *Consider $P_1 = P(A,B)$, $P_2 = P(A,C)$ and $P_3 = P(C,B)$. Consider $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}$.* $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}}(P_1, P_2 \vee P_3) = \mathtt{Tve}^{1,1,1,\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}}(P_1, P_2 \vee P_3) = \frac{a}{a+b+c} = \frac{1}{3}$ *with :*

- $a = \mathtt{avg}(\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3), \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_2, P_1) + \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_3, P_1)) = \mathtt{avg}(\frac{1}{2}, 1) = \frac{3}{4}$
- $b = 1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3) = \frac{1}{2}$
- $c = (1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_2, P_1)) + (1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_3, P_1)) = \frac{1}{2} + \frac{1}{2} = 1$, *with* $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3) = \frac{1}{2} = \mathtt{max}(\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_2), \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_3))$, $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2) = \mathtt{max}(\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_2)) = \frac{1}{2}$ *(idem for $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_3)$).*

3.3 Similarity between sets of grounded clauses

Recall that $\mathbb{C}$ is the set of all grounded clauses in OS−FOL.

Definition 21 (Grounded clause membership) *Let $\delta \in \mathbb{C}$ be a grounded clause and $\Delta \subseteq \mathbb{C}$ be a set of grounded clauses. Let $\oplus^{\mathtt{c}}$ and $\oplus^{\mathtt{l}}$ be two aggregation functions and $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$ be a similarity measure between a pair of clauses with $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$. The membership function of a grounded clause in a set of grounded clauses, denoted $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}} : \mathbb{C} \times 2^{\mathbb{C}} \to [0,1]$, is $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}(\delta, \Delta) = \oplus^{\mathtt{c}}_{\delta' \in \Delta}(\mathtt{s}^{\mathbb{C}}(\delta, \delta'))$.*

Definition 22 (Similarity between sets of grounded clauses) *Let $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}$ be a membership function with $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$. A similarity measure between two sets of grounded clauses is defined as $\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}} : 2^{\mathbb{C}} \times 2^{\mathbb{C}} \to [0,1]$.*

Example 9 *Let Δ_1 and Δ_4 be two sets of grounded clauses.*
$\Delta_1 = \{w(T), \neg w(T) \vee \neg f(T), \neg w(Z) \vee \neg f(Z)\}$
$\Delta_4 = \{p(T), \neg p(T) \vee \neg f(T), \neg p(Z) \vee \neg f(Z)\}$

$\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}}(\Delta_4,\Delta_1) = \mathtt{Tve}^{1,1,1,\varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}}(\Delta_4,\Delta_1) = \frac{a}{a+b+c} = \frac{1}{8}$ *with :*

- $a = \mathtt{avg}(\varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(p(T),\Delta_1) + \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg p(T) \vee \neg f(T),\Delta_1) + \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg p(Z) \vee \neg f(Z),\Delta_1),$
$\varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(w(T),\Delta_4) + \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg w(T) \vee \neg f(T),\Delta_4) + \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg w(Z) \vee \neg f(Z),\Delta_4)) = \mathtt{avg}(0+\frac{1}{3}+\frac{1}{3}, 0+\frac{1}{3}+\frac{1}{3}) = \frac{2}{3}$
- $b = (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(p(T),\Delta_1)) + (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg p(T) \vee \neg f(T),\Delta_1)) + (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg p(Z) \vee \neg f(Z),\Delta_1)) = 1 + \frac{2}{3} + \frac{2}{3} = \frac{7}{3}$
- $c = (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(w(T),\Delta_4)) + (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg w(T) \vee \neg f(T),\Delta_4)) + (1 - \varepsilon^{\mathbb{C}}_{\mathtt{max},\mathtt{s}^{\mathbb{C}}}(\neg w(Z) \vee \neg f(Z),\Delta_4)) = 1 + \frac{2}{3} + \frac{2}{3} = \frac{7}{3}$

3.4 Similarity between instantiations

Now, define $\mathbb{I}$ the set of all instantiations in $\mathtt{OS}-\mathtt{FOL}$.

Definition 23 (Instantiation membership) *Consider an instantiation $\Delta \in \mathbb{I}$ and a set of instantiations $I \subseteq \mathbb{I}$. Let $\oplus^{\mathtt{i}}$, $\oplus^{\mathtt{c}}$ and $\oplus^{\mathtt{l}}$ be three aggregation functions and $\mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$ be a similarity measure between a pair of set of clauses with $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$. The membership function of an instantiation in a set of instantiations, $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}} : \mathbb{I} \times 2^{\mathbb{I}} \to [0,1]$, is $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}(\Delta, I) = \oplus^{\mathtt{i}}_{\Delta' \in I}(\mathtt{s}^{\mathbb{I}}(\Delta,\Delta'))$.*

Definition 24 (Similarity between sets of instantiations) *Let $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}$ be a membership function with $\mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$, $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$. The similarity measure between two set of instantiations is defined as $\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}} : 2^{\mathbb{I}} \times 2^{\mathbb{I}} \to [0,1]$.*

Example 10 *Let I_1 and I_2 be two sets of instantiations s.t. :*
$I_1 = \{\Delta_1, \Delta_2, \Delta_3\}$ *with*
- $\Delta_1 = \{w(T), \neg w(T) \vee \neg f(T), \neg w(Z) \vee \neg f(Z)\}$
- $\Delta_2 = \{w(Z), \neg w(T) \vee \neg f(T), \neg w(Z) \vee \neg f(Z)\}$
- $\Delta_3 = \{w(T), w(Z), \neg w(T) \vee \neg f(T), \neg w(Z) \vee \neg f(Z)\}$

$I_2 = \{\Delta_4\}$ *with*
- $\Delta_4 = \{p(T), \neg p(T) \vee \neg f(T), \neg p(Z) \vee \neg f(Z)\}$

$\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}}(I_1,I_2) = \mathtt{Tve}^{1,1,1,\varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}}(I_1,I_2) = \frac{a}{a+b+c} = \frac{73}{1143} \simeq 0.064$ *with :*
- $a = \mathtt{avg}\Big(\sum_{x\in I_1} \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(x,I_2), \sum_{y\in I_2} \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(y,I_1)\Big)$
$= \mathtt{avg}\Big(\varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_1,I_2) + \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_2,I_2) + \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_3,I_2), \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_4,I_1)\Big)$
$= \mathtt{avg}(\frac{1}{8}+\frac{1}{8}+\frac{2}{19}, \frac{1}{8}) = \frac{73}{304}$
- $b = \sum_{x\in I_1} 1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(x,I_2) = (1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_1,I_2)) + (1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_2,I_2)) + (1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_3,I_2)) = \frac{7}{8} + \frac{7}{8} + \frac{17}{19} = \frac{201}{76}$
- $c = \sum_{y\in I_2} 1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(y,I_1) = 1 - \varepsilon^{\mathbb{I}}_{\mathtt{max},\mathtt{s}^{\mathbb{I}}}(\Delta_4,I_1) = \frac{7}{8}$

Let us now define a similarity measure between sets of formulae.

Definition 25 (Similarity Models) *A* Similarity Model *(*$\mathtt{SM}$*) is a tuple $\mathbf{M} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}, \mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}\rangle$. Let $\Phi, \Psi \subseteq \mathtt{OS}-\mathtt{FOL}$ be two sets of formulae and $\mathbf{I_{St}}$ be an interpretation over a structure $\mathbf{St}$. The similarity between Φ and Ψ is $\mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\Phi,\Psi) = \mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi), \mathtt{Inst}_{\mathbf{I_{St}}}(\Psi))$.*

Finally, using the measure of similarity between sets of formulae, we can extend the definition from [4] to asses the similarity between two $\mathtt{OS}-\mathtt{FOL}$ arguments.

Definition 26 (Similarity between OS-FOL Arguments) *Consider a coefficient $0 < \eta < 1$, a* $\mathtt{SM}$ $\mathbf{M}$ *and $\mathbf{I_{St}}$ an interpretation over a structure $\mathbf{St}$. We define $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I_{St}},\eta} : \mathtt{Arg}(\mathcal{L}) \times \mathtt{Arg}(\mathcal{L}) \to [0,1]$ by $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I_{St}},\eta}(A,B) = \eta \cdot \mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\mathtt{UC}(\mathtt{Supp}(A)), \mathtt{UC}(\mathtt{Supp}(B)))$
$+(1-\eta) \cdot \mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\mathtt{UC}(\mathtt{Conc}(A)), \mathtt{UC}(\mathtt{Conc}(B)))$.*

Example 11 *Let $\mathbf{M}_{\mathtt{jac}} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{jac}^{2,\mathtt{s}^{\mathbf{L}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{jac}^{1,\mathtt{s}^{\mathbb{C}}}, \mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}}\rangle$ be a similarity instantiation model and let A_1 and A_2 be the two OS-FOL arguments from Example 5. Their respective instantiations are given in Example 4 for the premises and the conclusions. Let us compute the similarity between A_1 and A_2 with $\eta = 0.5$.*
$\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}},0.5}(A_1,A_2) =$
$0.5 \cdot \mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Supp}(A_1), \mathtt{Supp}(A_2)) +$
$0.5 \cdot \mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Conc}(A_1), \mathtt{Conc}(A_2))$
$= 0.5 \cdot \frac{73}{1143} + 0.5 \cdot \frac{5}{11} \simeq 0.2592$ *where* $\mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Supp}(A_1), \mathtt{Supp}(A_2)) =$
$\mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Supp}(A_1)), \mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Supp}(A_2))) = \frac{73}{1143} \simeq 0.064$ *and* $\mathtt{sim}^{\mathtt{OS}-\mathtt{FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Conc}(A_1), \mathtt{Conc}(A_2)) =$
$\mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Conc}(A_1)), \mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Conc}(A_2))) = \frac{5}{11} \simeq 0.4545$.

4 Axiomatic Evaluation

Before determining the principles satisfied by our similarity measures, we introduce the notion of well-behaved $\mathtt{SM}$. It is a bridge between the (lower level) properties of the measures that we use (*e.g.* the Tversky measures) and the (higher level) properties of the similarity measure between arguments defined from such a $\mathtt{SM}$.

TABLE 2 – Principles satisfaction by similarity measures. • (resp. ○) means the measure satisfies (resp. violates) the principle. $\mathtt{sim}_x$ is a shorthand for $\mathtt{sim}_x^{\mathtt{Arg}(\mathcal{L})}$.

	$\mathtt{sim}_{\mathtt{jac}}$	$\mathtt{sim}_{\mathtt{dic}}$	$\mathtt{sim}_{\mathtt{sor}}$	$\mathtt{sim}_{\mathtt{adb}}$	$\mathtt{sim}_{\mathtt{ss}_2}$	$\mathtt{sim}_{\mathtt{ns}}$-jac	$\mathtt{sim}_{\mathtt{ns}}$-dic	$\mathtt{sim}_{\mathtt{ns}}$-sor	$\mathtt{sim}_{\mathtt{ns}}$-adb	$\mathtt{sim}_{\mathtt{ns}}$-$\mathtt{ss}_2$
Maximality	•	•	•	•	•	•	•	•	•	•
Symmetry	•	•	•	•	•	○	○	○	○	○
Substitution	•	•	•	•	•	○	○	○	○	○
Syntax Independence	•	•	•	•	•	•	•	•	•	•
Minimality	•	•	•	•	•	•	•	•	•	•
Monotony	•	•	•	•	•	•	•	•	•	•
Strict Monotony	•	•	•	•	•	○	○	○	○	○
Dominance	•	•	•	•	•	•	•	•	•	•
Strict Dominance	•	•	•	•	•	○	○	○	○	○

Definition 27 (Well-Behaved SM)
A $\mathtt{SM}$ $\mathbf{M} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\mathcal{E}^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\mathcal{E}^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}, \mathtt{simSI}^{\mathcal{E}^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}\rangle$ *is well-behaved iff :*

1. *(a) i.* $\mathbf{g}(1,1) = 1$,
   ii. $\mathbf{g}(0,0) = 0$,
   (b) i. $\mathbf{p}(P,P) = 1$,
   ii. $\mathbf{p}(P,Q) = 0$ *iff* $P \neq Q$,
   (c) i. $\mathbf{c}(\langle a_1,\ldots,a_k\rangle, \langle a_1,\ldots,a_k\rangle) = 1$,
   ii. if $\forall i \in \{1,\ldots,k\}, \nexists j \in \{1,\ldots,n\}$ *s.t.* $a_i = b_j$ *then* $\mathbf{c}(\langle a_1,\ldots,a_k\rangle, \langle b_1,\ldots,b_n\rangle) = 0$,
2. *Given* $\mathbb{X}$ *a set of objects,*
   (a) $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X,X) = 1$ *for any set of objects* $X \subseteq \mathbb{X}$,
   (b) if $\forall x \in X, \forall x' \in X', \mathtt{s}(x,x') = 0$ *then* $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X,X') = 0$,
   (c) consider $X_0, X_1, X_2 \subseteq \mathbb{X}$ *s.t.* $X_1 \subset X_2$ *and* $X_2 \setminus X_1 = \{x_2\}$. *If* $\exists x_0 \in X_0$ *s.t.* $\mathtt{s}(x_0,x_2) = \mathtt{s}(x_2,x_0) = 1$ *then* $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_2) \geq \mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_1)$,
   (d) consider $X_0, X_1, X_2 \subseteq \mathbb{X}$ *s.t.* $X_1 \subset X_2$ *and* $X_2 \setminus X_1 = \{x_2\}$. *If* $\forall x_0 \in X_0$, $\mathtt{s}(x_0,x_2) = \mathtt{s}(x_2,x_0) = 0$ *then* $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_1) \geq \mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_2)$.

In the last item, $\mathbb{X}$ can be the set of all literals (for characterizing $\mathtt{simC}^{\mathcal{E}^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$), the set of all grounded clauses (for characterizing $\mathtt{simI}^{\mathcal{E}^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$) or the set of instantiations (for characterizing $\mathtt{simSI}^{\mathcal{E}^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}$). Now we can show that a well-behaved SM guarantees that the corresponding similarity measure satisfies some principles. Let us recall that the set of principles can be found in Section 2.2.

Theorem 1 *For any* $\mathbf{M} \in \mathtt{SM}$, *if* $\mathbf{M}$ *is well-behaved then* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathtt{St}},\eta}$ *satisfies the following principles : Maximality, Minimality, Monotony and Dominance.*

To satisfy other principles we propose additional constraints.

Theorem 2 *Let* $\mathbf{M} \in \mathtt{SM}$ *be a well-behaved and* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathtt{St}},\eta}$ *be a similarity based on* $\mathbf{M}$.
– $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathtt{St}},\eta}$ *satisfies Symmetry (resp. Syntax Independence) if all the functions in* $\mathbf{M}$ *are symmetric (resp. syntax independent).*
– $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathtt{St}},\eta}$ *satisfies Strict Monotony and Strict Dominance if it satisfies condition 2.(c') : consider* $X_0, X_1, X_2 \subseteq \mathbb{X}$ *s.t.* $X_1 \subset X_2$ *and* $X_2 \setminus X_1 = \{x_2\}$. *If* $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_1) < 1$ *and* $\exists x_0 \in X_0$ *s.t.* $\mathtt{s}(x_0,x_2) = \mathtt{s}(x_2,x_0) = 1$ *then* $\mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_2) > \mathtt{sim}^{\mathcal{E},\mathtt{s}}(X_0,X_1)$.

We extend some results from [4].

Proposition 3 *Let* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}$ *be a similarity measure.*
– *Consider* $A, B \in \mathtt{Arg}(\mathcal{L})$. *If* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}$ *satisfies Maximality, Monotony, Strict Monotony and Strict Dominance then* $A \approx B$ *iff* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}(A,B) = 1$.
– *If* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}$ *satisfies Symmetry, Maximality, Strict Monotony, Dominance, and Strict Dominance then* $\mathtt{sim}^{\mathtt{Arg}(\mathcal{L})}$ *satisfies Substitution.*

Let us prove that the functions $\mathbf{g}$, $\mathbf{p}$ and $\mathbf{c}$ used in the paper satisfy the expected properties of a well-behaved SM.

Lemma 1 *For* $\mathbf{g} \in \{\mathtt{min}, \mathtt{avg}\}$, $\mathbf{p} = \mathtt{eq}$ *and* $\mathbf{c} = \mathtt{pws}$, $\langle \mathbf{g}, \mathbf{p}, \mathbf{c}\rangle$ *satisfies item 1. of Def.* 27.

We can show similar results for the Tversky measures that we use to define $\mathtt{simC}^{\mathcal{E}^{\mathbf{L}}_{\oplus^{\mathtt{l}},\mathtt{s}^{\mathbf{L}}}}$, $\mathtt{simI}^{\mathcal{E}^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$ and $\mathtt{simSI}^{\mathcal{E}^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}$. We consider the measures described in Table 1.

Lemma 2 *If* $\mathtt{Tve}^{\alpha,\beta,\gamma,\mathcal{E}^{\mathbb{X}}_{\oplus,\mathtt{sim}}}$ *is a Tversky measure, with* $\oplus = \mathtt{max}$, *and* $\mathtt{sim}$ *is*
– *either* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c}\rangle}$ *(from Definition 15) s.t.* $\langle \mathbf{g}, \mathbf{p}, \mathbf{c}\rangle$ *satisfies item 1. of Def. 27,*
– *or a similarity measure satisfying the item 2. of Def. 27, then* $\mathtt{Tve}^{\alpha,\beta,\gamma,\mathcal{E}^{\mathbb{X}}_{\oplus,\mathtt{sim}}}$ *satisfies the item 2. of Def. 27.*

Proposition 4 *For* $x \in \{\mathtt{jac}, \mathtt{dic}, \mathtt{sor}, \mathtt{adb}, \mathtt{ss}_2, \mathtt{ns}\text{-}\mathtt{jac}, \mathtt{ns}\text{-}\mathtt{dic}, \mathtt{ns}\text{-}\mathtt{sor}, \mathtt{ns}\text{-}\mathtt{adb}, \mathtt{ns}\text{-}\mathtt{ss}_2\}$, *define* $\mathtt{sim}_x^{\mathtt{Arg}(\mathcal{L})}$. *Then define the similarity model* $\mathtt{SM}$ $\mathbf{M}_x = \langle \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}, x^{2,\mathtt{sim}^{\mathbf{L}}}, x^{1,\mathtt{sim}^{\mathbb{C}}}, x^{1,\mathtt{sim}^{\mathbb{I}}}\rangle$. *The satisfaction of principles by the measures is given in Table* 2.

Notice that Proposition 4 implies that all the principles are compatible. Moreover with the result of item 1 of Proposition 3, we can deduce that our 5 symmetric extended Tversky measures satisfying a stronger form of maximality, since equivalent arguments are maximally similar. For non-symmetric measures, we show that they can obtain full similarity in a particular case of sub-argument.

Proposition 5 *Let* $A, B \in \texttt{Arg}(\mathcal{L})$ *be two arguments. Assume that* $\mathbf{M}$ *is a* SM *s.t.* $\texttt{simC}^{\mathcal{E}^{\mathbf{L}}_{\oplus^{\mathbf{l}},\mathrm{s}^{\mathbf{L}}}}$, $\texttt{simI}^{\mathcal{E}^{\mathbb{C}}_{\oplus^{\mathbb{C}},\mathrm{s}^{\mathbb{C}}}}$ *and* $\texttt{simSI}^{\mathcal{E}^{\mathbb{I}}_{\oplus^{\mathbb{i}},\mathrm{s}^{\mathbb{I}}}}$ *are Tversky measures s.t.* $\alpha \neq \beta$ *for at least one of them (*i.e. *it is non-symmetric). If* B *is a sub-argument of* A*, then* $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathbf{St}},\eta}(A, B) \geq \eta$*. Moreover, if* $\texttt{UC}(\texttt{Conc}(B)) \subseteq \texttt{UC}(\texttt{Conc}(A))$*, then* $\texttt{sim}^{\texttt{Arg}(\mathcal{L})}_{\mathbf{M},\mathbf{I}_{\mathbf{St}},\eta}(A, B) = 1$.

5 Conclusion

In this paper, we have proposed the rich methodology of similarity models which are able to express large families of similarity measures between Order-Sorted First Order Logic ($\texttt{OS} - \texttt{FOL}$) arguments, thanks to various parameters which allow to define generalized versions of similarity measures from the literature. For the first time in the logical argumentation literature, we define non-symmetric similarity measures. A set of nine principles for these $\texttt{OS} - \texttt{FOL}$ arguments has been proposed with a set of well-behaved properties ensuring some principles. We have shown that our symmetric measures satisfy all the principles, while their non-symmetric counterparts only satisfy a subset.

This work paves the way to several interesting research questions. First of all, we can consider additional measures (*e.g.* Ochiai [25], Kulczynski [20]) and principles (*e.g.* triangular inequality, non-zero, independent distribution [14]) to allow a more accurate comparison of similarity measures. Another research line could be to consider situations where different predicates are partially similar. For instance, one can consider that $greaterOrEqual(A, B)$ is somehow similar to $strictlyGreater(A, B)$. Following the same idea as in [6], we also plan to use our similarity measures as a parameter of acceptability semantics. Finally, we want to apply our work on real data expressed in fragments of $\texttt{OS} - \texttt{FOL}$.

6 Acknowledgement

This work benefited from the support of the project AGGREEY ANR-22-CE23-0005 of the French National Research Agency (ANR) and the Project GIUSTIZIA AGILE, CUP J89J22000900005.

Références

[1] Amgoud, L.: *A Replication Study of Semantics in Argumentation*. Dans *IJCAI'19*, pages 6260–6266, 2019.

[2] Amgoud, L. et J. Ben-Naim: *Axiomatic Foundations of Acceptability Semantics*. Dans *KR'16*, pages 2–11, 2016.

[3] Amgoud, L., E. Bonzon, J. Delobelle, D. Doder, S. Konieczny et N. Maudet: *Gradual Semantics Accounting for Similarity between Arguments*. Dans *KR'18*, pages 88–97, 2018.

[4] Amgoud, L. et V. David: *Measuring Similarity between Logical Arguments*. Dans *KR'18*, pages 98–107, 2018.

[5] Amgoud, L. et V. David: *An Adjustment Function for Dealing with Similarities*. Dans *COMMA'20*, pages 79–90, 2020.

[6] Amgoud, L. et V. David: *A General Setting for Gradual Semantics Dealing with Similarity*. Dans *AAAI'21*, 2021.

[7] Amgoud, L. et V. David: *Similarity Measures Based on Compiled Arguments*. Dans *ECSQARU'21*, pages 32–44, 2021.

[8] Amgoud, L., V. David et D. Doder: *Similarity Measures Between Arguments Revisited*. Dans *ECSQARU'19*, pages 3–13, 2019.

[9] Anderberg, M.: *Cluster analysis for applications. Monographs and textbooks on probability and mathematical statistics*, 1973.

[10] Arioua, A., M. Croitoru et S. Vesic: *Logic-based argumentation with existential rules*. Int. J. Approx. Reason., 90 :76–106, 2017.

[11] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi et P. F. Patel-Schneider (rédacteurs): *The Description Logic Handbook : Theory, Implementation, Applications*. 2003.

[12] Besnard, P. et A. Hunter: *A logic-based theory of deductive arguments*. Artificial Intelligence, 128(1-2) :203–235, 2001.

[13] Besnard, P. et A. Hunter: *Practical first-order argumentation*. Dans *AAAI'05*, pages 590–595, 2005.

[14] David, V.: *Dealing with Similarity in Argumentation*. Thèse de doctorat, Univ. Toulouse III, 2021.

[15] Dice, L.: *Measures of the amount of ecologic association between species*. Ecology, 26(3) :297–302, 1945.

[16] Governatori, G., M. Maher, G. Antoniou et D. Billington: *Argumentation Semantics for Defeasible Logic*. J. Log. Comput., 14(5) :675–702, 2004.

[17] Halpern, J. et V. Weissman: *Using first-order logic to reason about policies*. TISSEC, 11(4) :1–41, 2008.

[18] Jaccard, P.: *Nouvelles recherches sur la distributions florale*. Bulletin de la societe Vaudoise des sciences naturelles, 37 :223–270, 1901.

[19] Jantke, K. P.: *Nonstandard Concepts of Similarity in Case-Based Reasoning*. Dans *Information Systems in Data Analysis : Prospects – Foundations – Applications*, pages 28–43, 1994.

[20] Kulczynski, S.: *Classe des sciences mathématiques et naturelles*. Bull. Internat. de l'Academie Polonaise des Sciences et des Lettres, pages 57–203, 1927.

[21] Lang, J., P. Liberatore et P. Marquis: *Propositional independence-formula-variable independence and forgetting*. J. Artif. Intell. Res., 18 :391–443, 2003.

[22] Longo, C., F. Longo et C. Santoro: *CASPAR : Towards decision making helpers agents for IoT, based on natural language and first order logic reasoning*. Eng. Appl. Artif. Intell., 104 :104269, 2021.

[23] McBurney, Peter, Simon Parsons et Iyad Rahwan (rédacteurs): *Proc. ArgMAS'11*, 2012.

[24] Oberschelp, Arnold: *Order sorted predicate logic*. Dans Bläsius, Karl Hans, Ulrich Hedtstück et Claus Rainer Rollinger (rédacteurs) : *Sorts and Types in Artificial Intelligence*, pages 7–17, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg, ISBN 978-3-540-46965-0.

[25] Ochiai, A.: *Zoogeographical studies on the Soleoid fishes found in Japan and its neighbouring regions*. Bull. Jpn. Soc. scient. Fish., 22 :526–530, 1957.

[26] Sneath, P. et R. Sokal: *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

[27] Sørensen, T.: *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons*. Biol. Skr., 5 :1–34, 1948.

[28] Tversky, A.: *Features of Similarity*. Psychological Review, 84(4) :327–352, 1977.

[29] Zhong, Q., X. Fan, X. Luo et F. Toni: *An explainable multi-attribute decision model based on argumentation*. Expert Sys. and Appl., 117 :42–61, 2019.

Actes JIAF-JFPDA 2023

A Gradual Semantic to Model Opinion using Bipolar Argumentation Graphs

Louise Dupuis de Tarlé

LAMSADE, Paris-Dauphine University, PSL

louise.dupuis@dauphine.eu

Résumé

L'argumentation abstraite est une méthode de formalisation des discussions argumentatives largement utilisée dans la représentation des connaissances et la construction de protocoles multi-agents. Les sémantiques graduelles ont récemment été proposées comme une extension des sémantiques classiques, permettant une évaluation plus fine des arguments. Dans cet article, nous proposons une structure générale pour représenter l'opinion des agents à partir de graphes d'argumentation bipolaires valués et caractériser cette opinion grâce à l'application d'une sémantique graduelle. Nous identifions certaines propriétés désirables d'une telle sémantique, en particulier les propriétés d'ouverture d'esprit et de dualité, et proposons une nouvelle sémantique graduelle qui les vérifie, que nous comparons à l'*Euler based semantic* de Amgoud et al. [1]. Nous discutons également des conséquences de l'application de cette nouvelle sémantique à notre structure d'opinion. Ce travail ouvre la voie à une analyse plus fine des dynamiques argumentatives dans les protocoles multi-agents utilisant les outils de l'argumentation abstraite.

Abstract

Abstract argumentation is a method for formalizing argumentative discussions which is widely used in the representation of knowledge and the construction of multi-agent protocols. Gradual semantics have recently been proposed as an extension of classical semantics, allowing a finer evaluation of arguments. In this article, we propose a general structure to represent the opinion of agents extracted from valued bipolar argumentation graphs thanks to the application of a gradual semantic. We identify some desirable properties of such a semantic, in particular open-mindedness and duality properties, and propose a new gradual semantic that verifies them, which we compare to the Euler based semantic of Amgoud et al. [1]. We also discuss the consequences of applying this semantic to our opinion framework. This work opens the way to a finer analysis of argumentative dynamics in multi-agent protocols which uses the tools of abstract argumentation.

Introduction

Abstract argumentation is a method for formalizing argumentative discussions. By representing debates in the form of graphs, it becomes possible to formally define the acceptability of arguments from the perspective of a rational agent. The simplicity and flexibility of this representation makes it an ideal tool for the representation of knowledge, and the construction of multi-agent protocols.

Many works use abstract argumentation to study dynamics that are explicitly argumentative : [12] model a strategic game of persuasion of an audience, [4] develops a protocol inspired by debates conducted on online platforms. Other works use abstract argumentation because it allows for a finer modeling of exchanges between agents, with the emergence of more varied opinion dynamics [21]. The use of argumentation to model the reasoning process of agents is justified by the recent advancements in cognitive psychology by Mercier and Sperber [14] and their theory of argumentative reasoning, which states that our reasoning abilities are derived from our capacity to produce arguments. The growing interest in these models justifies the enhancement of abstract argumentation with new tools, such as gradual semantics.

The semantics introduced by Dung [8] are functions that determine the set of acceptable arguments within an argumentation graph. Recently, a new type of semantic called *gradual* has been proposed [13, 2, 1] : these semantics assign an acceptability score to each argument, allowing for a more nuanced evaluation of the acceptability of the arguments. The inherent expressiveness of gradual semantics makes them highly valuable for the development of multi-agent protocols, as they enable precise analysis of argumentative dynamics. For instance, [9] creates a multi-agent model where the opinion of the agents is a number obtained through the application of a gradual semantic, which

makes it possible to study the dynamics of the agents' opinions. In this work, we seek to build a tool which would enable similar analyses in the case of bipolar argumentation graphs.

Indeed, the first works on abstract argumentation consider only one relation between the arguments : the attack. Bipolar graphs are an extension of classical abstract argumentation graphs which consider an additional relation, that of *support* [5]. These graphs are more expressive and have been validated by empirical experiments as more representative of the way in which humans actually reason [16]. In most cases, these graphs are *weighted*, which means that arguments are equipped with a weight ; which can represent their intrinsic strength, trust in their source, or support in the form of a vote. Several gradual semantics for weighted bipolar graphs have been proposed [1, 19].

Building upon the work of [9], we present a general framework that enables the representation of agents' knowledge in the form of a bipolar graph, and the characterization of their opinion with the application of a gradual semantic. This leads us to identify some desirable properties of such semantics : in particular, the properties of open-mindedness and duality identified by Potyka [17, 18]. We propose a new semantic which verifies these principles.

The first section of this article presents an overview of the fundamental concepts of abstract argumentation frameworks and gradual semantics. Subsequently, in the second section, we introduce a general framework that effectively represents agents' opinions using argumentation graphs. This framework serves as a foundation for the exploration of various desirable properties of semantics in the subsequent third section. To address these properties, the fourth section introduces a novel gradual semantics that ensures, among others, the presence of open-mindedness and duality. Moreover, we delve into the implications of applying this semantics to our opinion framework within this section.

1 Abstract Argumentation and Gradual Semantics

1.1 Bipolar Graphs

Abstract argumentation, introduced by Dung [8], is a method for formalizing argumentative discussions that considers arguments as abstract objects and focuses on the relations that link these arguments together. Since the introduction of abstract argumentation frameworks, or graphs, the attack relation which was considered originally has been supplemented by a *support* relation, giving rise to **bipolar argumentation graphs** [5].

A very common extension of bipolar argumentation graphs consists in equipping each argument with a "weight", which corresponds to an intrinsic quality of the argument. Thus, we can define weighted bipolar argumentation graphs.

Definition 1 (Weighted Bipolar Argumentation Graphs) *A weighted bipolar argumentation graphs B is a quadruple $B = \langle A, R, S, W\rangle$ where A is a finite set of arguments, $R \subseteq A \times A$, $S \subseteq A \times A$ are two binary relations on arguments, respectively attack and support, and W is a function from A to $[0, 1]$.*

In this article, we will also focus on a subclass of these graphs, namely non-weighted bipolar graphs.

Non Weighted Bipolar Graphs correspond to the case $B = \langle A, R, S, W\rangle$ where W is a constant function. This amounts to choosing a base weight w_{base} such that $W(a) = w_{base}$ for all $a \in A$.

In order to simplify notations, we will use $B \setminus A'$ with $A' \subset A$ to denote graph B without the arguments of A' and the attack and support relations featuring these arguments. We now also define the set of attacker and supporters of an argument.

Definition 2 (Set of attackers and supporters) *Let $B = \langle A, R, S, W\rangle \in G$ a weighted bipolar graph and $a \in A$ an argument from this graph. The set of attackers and supporters of a in B are respectively defined as $Att(B, a) = \{b \in A | (b, a) \in R\}$ and $Supp(B, a) = \{b \in A | (b, a) \in S\}$.*

1.2 Gradual Semantics

A major challenge of abstract argumentation is the characterisation of the *acceptability* of arguments based on the information contained in an argumentation graph. **Gradual semantics** are functions which evaluate the acceptability of arguments through an *acceptability score*. In this work, we only consider gradual semantics whose image set is an interval of $\mathbb{R}$: it is a necessary condition to define the relevant properties presented in Section 3. This restriction enables us to consider all the gradual semantics for bipolar graphs proposed by [1, 15].

Definition 3 (Gradual Semantic) *Let G the set of all weighted bipolar graphs, $B = \langle A, R, S, W\rangle \in G$, and D an interval of $\mathbb{R}$. A gradual semantic on B is a function $\sigma : G \times A \to D$ and for all $a \in A$, $\sigma(B, a)$ denotes the acceptability score of a in B.*

We have a specific focus on a particular type of gradual semantic known as **modular** semantics, as identified by Mossakowski et al. [15]. These semantics consist of two functions : one aggregates the scores of the attackers and supports of a given argument, and the other determines their influence on the base weight of this argument. Consequently, the various semantics proposed for bipolar argument graphs in existing literature can be analyzed as combinations of an aggregation function and an influence function.

Definition 4 (Modular Gradual Semantic) *Let $B = \langle A, R, S, W\rangle \in G$, and D an interval of $\mathbb{R}$. A modular gradual semantic on B is a function $\sigma : G \times A \rightarrow D$ where $\sigma(B, a) = \iota(\alpha(B, a))$ with α and ι respectively the aggregation function and influence function.*

In the rest of this article, we will only consider modular gradual semantics, which will simplify the definition and verification of their properties. As noted by [15], all the gradual semantics for weighted bipolar graphs presented in the literature are modular.

2 Opinion Model

In this section, we place ourselves within the framework of an argumentative discussion between agents, and we define a way to characterize the opinion of the agents through a bipolar graph, their **opinion graph**. We don't specify a multi-agent protocol governing what actions are performed each turn, so we use the terminology "framework" or "structure". Our goal is to create a flexible model that can serve as a basis for the creation of various multi-agent protocols, whose specific characteristics would make it possible to study various phenomena.

The opinion graph can be interpreted either as the agent's knowledge base, or the arguments that she takes into account in her evaluation of a debate. Here, we favor the first interpretation, and say that an agent *knows* an argument when it belongs to her opinion graph.

Following the methodology of [4] and [9], all of the opinion graphs contain an argument with a special status, the *issue*, which constitutes the focus of the debate. These graphs are **issue-oriented**, which means that all the arguments of a graph belong to a path of supports and attacks directed towards the issue. Furthermore, we consider the opinion of the agents to be a real number, which belongs to the image set of the gradual semantic that we use (in most case, this interval is $[0, 1]$), and represent their opinion *about the issue*. This focus on a single issue is warranted by the context of argumentative discussions, although this framework could easily be extended to include multi-dimensional opinions about several issues. Many seminal opinion dynamics models represent the opinion of agents as a real number in the interval $[0, 1]$: such is the case of the *bounded-confidence* type models [10, 7]. These models make the assumption that the opinion of agents can be represented as a real number for the sake of simplicity, citing the example of "an expert who has to assess a certain magnitude" [10]. In our case, as we are studying argumentative discussions, one natural interpretation of the opinion is a **degree of belief** of the agent in the acceptability of the issue.

We distinguish two cases, the one where the graph is not weighted and the one where it is. It is important that our structure takes into account the cases where the arguments are equipped with weights because it enables greater expressiveness : for example a protocol could aggregate votes coming from agents and transform them into weights as described in [13]. The non-weighted case is also necessary, because it allows for a simplification of the multi-agent protocol. Indeed, according to the KISS approach (*Keep it Simple, Stupid!*) [3], multi-agent protocols must be as simple as possible, and use a minimum number of parameters. As we will see later, the need to accommodate weighted and non-weighted cases is a non-trivial constraint on the semantic used.

We can now formally define our framework, starting with opinion graphs.

Definition 5 (Issue Oriented Bipolar Graphs) *Let $B = \langle A, R, S, W\rangle$ a weighted bipolar graph. B is **issue-oriented** if there exists $i \in A$ such that for all $a \in A$, there is a path from a to i.*

Suppose we have a semantic for weighted bipolar graphs $\sigma : G \times A \rightarrow D \in \mathbb{R}$. We can then define the opinion of agents as the evaluation of the acceptability of the issue by the gradual semantic σ applied to their opinion graph.

Definition 6 (Opinion of an agent) *Let an agent k equipped with an issue-oriented bipolar graph of issue i, $B_k = \langle A, R, S, W\rangle \in G$ and $\sigma : G \times A \longrightarrow D \in \mathbb{R}$ a modular gradual semantic well defined on B_k. We define the **opinion** of agent k as $O_k = \sigma(B_k, i)$.*

The main contribution of our framework is that the agents' opinions are derived from graphs forming their knowledge base, using gradual semantics which can express certain ideals of rationality. With an accurate choice of image set D for the semantic as $[0, 1]$, the results of multi-agent protocols built using our framework could be directly compared with that of the bounded-confidence type models.

We can see that the semantic used plays a major role in the evaluation of the agents' opinions. The following section describes necessary and desirable properties of a semantic for this framework.

3 Desirable properties

3.1 First principles

Amgoud et al. [1] carry out an extensive study of gradual semantics for weighted bipolar graphs. The authors identify twelve desirable properties that can be verified by such semantics. Table 1 offers an intuitive explanation of each of these principles. We refer the reader to the original article for a complete formalization.

The authors compare existing semantics for weighted bipolar graphs based on these principles. They propose a

novel semantic called *Euler Based Semantic* (EBS) and show that it is the only semantic that verifies their twelve principles.

3.2 EBS and non-weighted graphs

The model for representing agents' knowledge and opinions presented in Section 2 gives rise to constraints on the semantic used. In particular, a semantic must be defined for the type of graph considered, depending on whether it is weighted or not. These properties are not trivial : despite the fact that it satisfies many desirable principles, we show here that the EBS semantic is not appropriate for a protocol using non-weighted graphs.

Most of the gradual semantics for bipolar graphs proposed in the literature are defined for weighted graphs. The underlying logic is that of "Whoever can do the most can do the least.", i.e. a semantic capable of accommodating an additional level of complexity can *a fortiori* deal with simpler cases, here non-weighted graphs. It would suffice to choose the base weight well to obtain a semantics that behaves correctly. This is a method successfully applied for attack (and support) gradual semantics : thus, the h-categorizer [13] semantic can be adapted to non-weighted attack graphs and retains desirable properties [18]. We will see that this is not the case with EBS.

Let us define formally the Euler base semantic introduced by [1]. This semantic is defined exclusively in the case of acyclic bipolar graphs and is based on a quantity, the energy, which aggregates the scores of the supports and direct attackers of an argument.

Definition 7 (Energy of an argument) *Let $B = \langle A, R, S, W\rangle \in G$ a weighted bipolar graph. For an argument $a \in A$, $Supp(B,a)$ and $Att(B,a)$ are respectively the set of argument attacking and supporting a in B. The energy E is defined as the function $E : G \times A \to \mathbb{R}$ such that for all $a \in A$:*

$$E(B,a) = \sum_{x \in Supp(B,a)} \sigma(x) - \sum_{x \in Att(B,a)} \sigma(x)$$

Definition 8 (*Euler Based Semantic* (EBS)) *Let $B = \langle A, R, S, W\rangle \in G$ an acyclic weighted bipolar graph. $Ebs(B)$ is the score function $\sigma : G \times A \to [0,1]$ recursively defined as : For all $a \in A$ of weight w_a*

$$\sigma(B,a) = 1 - \frac{1 - w_a^2}{1 + w_a e^{E(B,a)}}, \qquad (1)$$

If one wishes to apply EBS to non-weighted graphs, it is necessary to choose a base weight w_{base} for all the arguments, which will correspond to their evaluation when they are neither attacked nor supported (a natural choice for such a weight would be 0.5 for example). However, as noted in

Table 1 – Principles defined by [1] which can be verified by a gradual semantic for weighted bipolar graphs.

Property	**Intuition**
Anonymity	The score of an argument is independant from its identity.
Bivariate Independance	The score of an argument is independant from every argument that is not linked to it.
Bivariate Directionnality	Only the relations directed towards the argument influence its score, and not relations directed away from it.
Bivariate Equivalence	The score of an argument only depends on its base weight and on the score of its direct attackers and supporters.
Stability	If an argument is neither attacked nor supported, its score must be equal to its weight.
Neutrality	Attackers and supporters of score equal to zero have no effect on their targets.
Bivariate Monotony	If an argument a is as much or less attacked than an argument b, and as much or less supported than b, then the score of a must be at least as great as that of b.
Bivariate Reinforcement	An argument's score increases if the quality of its attackers is reduced and the quality of its supports is increased.
Resilience	If an argument's weight is positive, its score cannot be reduced to zero by attacks. If the weight is lower than 1, it cannot reach 1 with supports.
Strict Franklin	Attacks are as important as supports.
Weakening / Strengthening	If attacks are greater than supports, the score of the argument is lower than its weight, and conversely.

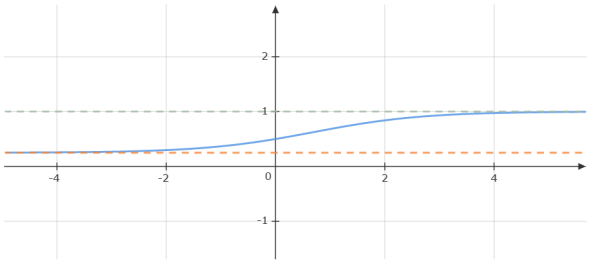


FIGURE 1 – Variation of the score of an argument according to EBS (y-axis) against its energy $E(B, a)$ (x-axis), in the case $w_{base} = 0.5$. We see that the value of the score is between 1 (grey line) and $w^2_{base} = 0.25$ (orange line).

[18], the acceptability score of an argument evaluated by EBS cannot be less than the square of the weight of the argument. Thus, this semantic adapted to a non-weighted graph would no longer affect a score between 0 and 1 but in the interval $[w^2_{base}, 1]$. Figure 1 illustrates this situation in the case of $w_{base} = 0.5$. We see that using EBS for a protocol using non-weighted graphs would amount to limiting the opinions of the agents to the interval $[w^2_{base}, 1]$, which among other things limits the possibilities of comparison with *bounded confidence* type models [1].

Another consequence of this behavior is that the impacts of supports are proportionally greater than those of attacks, regardless of the weight of the argument.

3.3 Open Mindedness and Duality

Potyka's works [18, 17] define two other properties of gradual semantic for bipolar graphs : open-mindedness and duality.

Intuitively, open mindedness corresponds to the fact that the score of an argument can vary freely between the limits of interval D, whatever its base weight : for instance, it may approach as closely as we want 0 or 1 as long as we add enough attacks and supports.

Definition 9 (Open Mindedness) *Let $\sigma : G \times A \rightarrow D$ a semantic for weighted bipolar graphs on an interval D. The semantic is open-minded for all graph $B = \langle A, R, S, W\rangle$ if for all argument $a \in A$ and all $\epsilon > 0$, the following condition is satisfied : there exists a number $N \in \mathbb{N}$ such that if we add N new arguments whose base score is maximal : $A_N = a_1, .., a_N$, $A \cap A_N = \emptyset$, then :*

1. *For graph $B_{att} = \langle A \cup A_N, R \cup \{(a_i, a) | 1 \leq i \leq N\}, S, W'\rangle$, we obtain $|\sigma(B_{att}, a) - min(D)| < \epsilon$*
2. *For graph $B_{supp} = \langle A \cup A_N, S \cup \{(a_i, a) | 1 \leq i \leq N\}, S, W'\rangle$, we obtain $|\sigma(B_{supp}, a) - max(D)| < \epsilon$*

where $W'(b) = W(b)$ for all $b \in A$ and $W'(a_i) = max(D)$ for $i \in [\![1, N]\!]$.

Potyka [17] also defines a property which illustrates the intuitive notion of symmetry between the actions of attacks and supports : duality. To illustrate, let's take the example of EBS : the asymmetric nature of this semantic causes an imbalance between the action of attacks and supports. One would expect symmetrical actions of attacks and supports when the initial weight is 0.5. On the other hand, when the initial weight is greater or less than 0.5, we cannot expect perfect symmetry because the weight of the argument is now closer to one of the limits of the interval $[0, 1]$. [17] generalizes this symmetry intuition in the following way : suppose that the initial weights of a and b are shifted relative to 0.5 by in different directions, and that the attackers of a have the same strength as the supports of b and vice versa. Then if the application of a dual semantics to a transforms its weight into a score shifted by δ, the score of b should be shifted by $-\delta$ with respect to its base weight. We define this property, duality, in the context of modular gradual semantics.

Definition 10 (Duality) *Let $\sigma : G \times A \rightarrow D$ be a modular semantic for weighted bipolar graphs, with α its aggregation function and ι its influence function. The semantic σ verifies **duality** for all $B = \langle A, R, S, W\rangle$ if and only if it verifies the following property :*

Let $a, b \in A$ such that $w_a = 0.5 + \epsilon, w_b = 0.5 - \epsilon$ for an $\epsilon \in [0, 0.5]$, and the supporters and attackers of a and b are such that :

$$\alpha(B \setminus Att(B, a), a) = \alpha(B \setminus Supp(B, b), b)$$
$$\alpha(B \setminus Att(B, b), b) = \alpha(B \setminus Supp(B, a), a)$$

then $\sigma(B, a) - w_a = w_b - \sigma(B, b)$.

These two properties are not necessary ; however, depending on the context, they may be very important. Concerning open-mindedness, apart from not being very elegant, real problems can emerge from a situation where the entirety of the opinion space is not accessible to agents. For instance, if one wanted to create an argumentative model studying epistemic communities where, like in Hegselmann's work [11], the success of the agents if measured by a distance between their opinion and a truth value, it would be very important that the opinion of the agents could approach any value of the interval $[0, 1]$. Similar problems would arise if we wanted to study extremism, another phenomenon investigated in bounded-confidence type models [6]. Duality, on the other hand, imposes a form of symmetry between the actions of attacks and supports. If it is not verified, the dynamics may differ from one side of the opinion space to another, which could be problematic for certain protocols.

EBS verifies neither open-mindedness, nor duality. In the following section, we propose a novel semantic which verifies both of these properties.

1. Note that by choosing a base weight of 0, we would ensure an opinion interval of $[0, 1]$ but that in this case, all the scores of the arguments would be equal to 0.

4 Novel Semantic

We define a modular gradual semantic by combining the *energy* aggregation function with a modified logistic influence function. Like EBS, our semantic is defined exclusively for acyclic graphs.

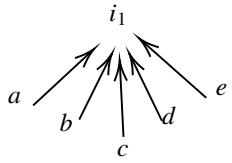
Definition 11 (*Logistic Sum Semantic* (LSS)) *Let* $B = \langle A, R, S, W \rangle$ *be an acyclic bipolar graph. LSS is the score function* $\sigma : G \times A \rightarrow [0, 1]$ *recursively defined by : For all* $a \in A$ *of weight* w_a*,*

$$\sigma(B, a) = 1 - \frac{1}{1 + e^{E(B,a)+b(w_a)}}, \quad b(w_a) = ln(\frac{1}{1 - w_a} - 1) \tag{2}$$

Property 1 *The LSS semantic verifies the twelve principles defined by [1] (see Table 1).*

Property 2 *The LSS semantic verifies open mindedness and duality.*

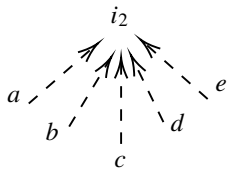
The following example compares the behavior of LSS and EBS on two simple argumentation graphs, and illustrates open-mindedness and duality.



$\sigma_{EBS}(B_1, i_1) = 0.28$

$\sigma_{LSS}(B_1, i_1) = 0.08$

Graph B_1



$\sigma_{EBS}(B_2, i_2) = 0.89$

$\sigma_{LSS}(B_2, i_2) = 0.92$

Graph B_2

Consider the above graphs B_1 and B_2, where we use full arrows to indicate attack relations and dotted arrows to indicate supports. The issue of B_1 is attacked by 5 arguments, and the issue of B_2 is supported by 5 arguments. We fix all of the weights of the arguments at 0.5, thus the aggregation of the attackers of i_1 is equal to the inverse of the aggregation of the supporters of i_2 (using the energy function).

The example illustrates the problem mentioned above, which is that EBS is limited to the interval $[0.25, 1]$. Indeed, the value of i_1 is 0.28 according to EBS, while LSS is able to assign a lower value of 0.08. We can also note that if we were to add attacks to i_1, EBS would not show much modification because the score of i_1 is already close to the limit of 0.25, while LSS would be more expressive, but on the other hand their evaluations of i_2 are much more similar.

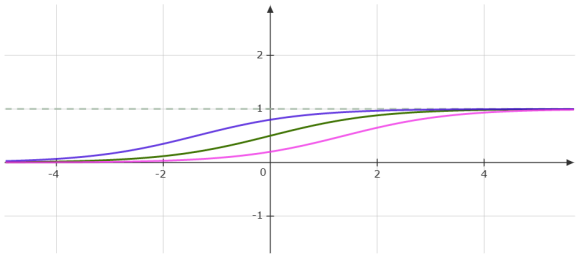


FIGURE 2 – Variation of the score of an argument according to LSS (y-axis) against its energy $E(B, a)$ (x-axis), for three basic weights $w_a = 0.2, 0.5$ and 0.8 respectively in pink, green and blue. We see that the value of the score is between 1 (grey line) and 0 and that when $E(B, a) = 0$, the score is equal to the base weight.

In this context, duality is verified if the sum of the evaluation of i_1 and i_2 is equal to 1, and we can see that it is verified by LSS and not by EBS.

Another illustration of these two properties in the case of LSS can be found on Figure 2, where we represent the score of an argument against its energy for three different weights. We see that in all three cases, open mindedness is verified as the score covers the whole interval. We can see that the purple and the pink curve, which corresponds to the acceptability score of an argument with a weight of 0.8 and 0.2 respectively, exhibit a central symmetry around the point $(0, 0.5)$. This symmetry corresponds to duality : if we take two arguments a and b with $w_a = 0.8$ and $w_b = 0.2$, and if their energies are inverse from each other, then the sum of their acceptability scores will be equal to the sum of their weights, which is 1.

4.1 Convergence of Opinions

Let us now place ourselves in the general framework defined in Section 2 : consider agents equipped with an opinion graph and let us use the LSS semantic. Suppose that two agents communicate by exchange of arguments, what can be said about their respective opinions ?

This question is important because it will allow us to compare directly any protocol built with our framework to the bounded confidence type models, where any communication between agents automatically results in a convergence of their opinions.

[9] show that in the case of attack graphs, with the h-categorizer semantic, the communication between agents does not automatically result in a convergence of their opinions. However, their simulations show an empiric convergence when many interactions take place.

In order to study this problem, we need to formally define what we mean by communication through exchange of arguments. For this, we make a number of simplification

assumptions.

— The opinion graphs of the agents are acyclic.
— When agents are aware of the same arguments, they are also aware of the same attack and support relations between them.
— Agents all agree on the base weights of their shared arguments.

The initial assumption enables the use of our LSS semantic. The other two assumptions, which are aligned with [9], are rather restrictive and allow us to define communication as a strict exchange of arguments without requiring a merging mechanism for attacks, supports, and weights. It is worth noting that these constraints, within which a wide variety of protocols can still be constructed, could be relaxed within our framework given that we ensure the existence of a compatible semantic and establish a process for merging argumentation graphs.

In accordance with the idea that opinion graphs are agent's knowledge bases, we say that an agent *learns* an argument when she adds it to her opinion graph.

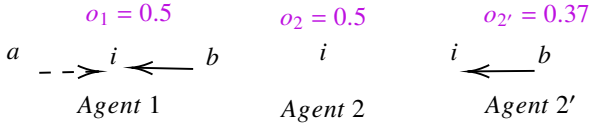
Definition 12 (Learning an argument) *Let an agent k equipped with opinion graph $B_k = \langle A, R, S, W\rangle$, and (a, R_a, S_a, w_a) a tuple composed of an argument a, relations R_a and S_a such that $R_a = \{(a,x)|x \in A_0 \subset A\} \cup \{(x,a)|x \in A_1 \subset A\}$ and $S_a = \{(a,x)|x \in A_2 \subset A\} \cup \{(x,a)|x \in A_3 \subset A\}$, and a base weight $w_a \in [0,1]$. Agent k **learns** argument a by transforming her opinion graph to $B'_k = \langle A \cup \{a\}, R \cup R_a, S \cup S_a, W'\rangle$ with $\forall x \in A, W' : x \rightarrow W(x)$ and $W'(a) = w_a$.*

We suppose that the attack and support relations R_a and S_a that link argument a to the arguments of the opinion graph of agent k are known. Depending on the specifics of the protocol, they could be obtained from another agent's opinion graph, or generated dynamically. Thus, we can define communication between agents as the learning of arguments from other's opinion graphs.

Under these constraints, we show the following property.

Property 3 *In our opinion model with LSS semantic, the opinion of two agents does not necessarily converge when they exchange arguments from their opinion graphs.*

It is easy to generate an example that illustrates (and proves) Property 3. Consider two agents 1 and 2 whose opinion graphs are shown below.



When applying the LSS semantic on their initial opinion graphs, their opinions are the same : $o_1 = o_2 = 0.5$. Consider what happens if Agent 2 adds one of the arguments of Agent 1 to her opinion graph : this is the situation denoted Agent 2′ above. Her opinion $o_{2'} = 0.37$ is now further from that of agent 1, even though their opinion graphs are now more similar [2].

Property 3 is also verified when using the EBS semantic. Therefore, it seems that gradual semantics exhibit non-trivial properties that justify the interest of their study in the context of multi-agent models.

5 Conclusion

We proposed a general structure which represents the knowledge and the opinion of agents with weighted bipolar argumentation graphs and a gradual semantic. We have discussed various desirable properties for such a semantic, in particular the principles of open-mindedness and duality, and proposed a new gradual semantics that verifies them. We would like to continue to study this semantic, in particular its behavior on bipolar graphs which may include cycles. Finally, we have identified that the use of this semantic within the framework of our opinion model gives rise to non-trivial dynamics : the opinions of agents do not necessarily converge when they communicate. Knowing the similar result obtained by [9], we are convinced that this behavior is not limited to our semantic. A natural extension to this work would be to characterize a minimal set of properties that must be checked by a gradual semantic to guarantee this behavior. We also plan an empirical study of the impact of various semantic on the dynamics of agents' opinions.

Références

[1] Amgoud, Leila et Jonathan Ben-Naim: *Weighted bipolar argumentation graphs : Axioms and semantics*. Dans *Twenty-Seventh International Joint Conference on Artificial Intelligence-IJCAI 2018*, pages 5194–5198, 2018.

[2] Amgoud, Leila, Jonathan Ben-Naim, Dragan Doder et Srdjan Vesic: *Acceptability semantics for weighted argumentation frameworks*. Dans *Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*. International Joint Conferences on Artifical Intelligence (IJCAI), 2017.

[3] Axelrod, Robert: *The complexity of cooperation*. Dans *The Complexity of Cooperation*. Princeton university press, 1997.

[4] Bonzon, Elise et Nicolas Maudet: *On the outcomes of multiparty persuasion*. Dans *Argumentation in Multi-Agent Systems : 8th International Workshop, ArgMAS 2011, Taipei, Taiwan, May 3, 2011, Revised Selected Papers 8*, pages 86–101. Springer, 2012.

2. The notion of similarity between graphs is not developed here, but consider for example a graph edit distance [20].

[5] Cayrol, Claudette et Marie Christine Lagasquie-Schiex: *On the acceptability of arguments in bipolar argumentation frameworks*. Dans *Symbolic and Quantitative Approaches to Reasoning with Uncertainty : 8th European Conference, ECSQARU 2005, Barcelona, Spain, July 6-8, 2005. Proceedings 8*, pages 378–389. Springer, 2005.

[6] Deffuant, Guillaume, Frédéric Amblard, Gérard Weisbuch et Thierry Faure: *How can extremism prevail ? A study based on the relative agreement interaction model*. Journal of artificial societies and social simulation, 5(4), 2002.

[7] Deffuant, Guillaume, David Neau, Frederic Amblard et Gérard Weisbuch: *Mixing beliefs among interacting agents*. Advances in Complex Systems, 3(01n04) :87–98, 2000.

[8] Dung, Phan Minh: *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*. Artificial intelligence, 77(2) :321–357, 1995.

[9] Dupuis de Tarlé, Louise, Elise Bonzon et Nicolas Maudet: *Multiagent Dynamics of Gradual Argumentation Semantics.* Dans *AAMAS*, pages 363–371, 2022.

[10] Hegselmann, Rainer et Ulrich Krause: *Opinion dynamics and bounded confidence : models, analysis and simulation*. 2002.

[11] Hegselmann, Rainer, Ulrich Krause *et al.*: *Truth and cognitive division of labor : First steps towards a computer aided social epistemology*. Journal of Artificial Societies and Social Simulation, 9(3) :10, 2006.

[12] Kohan Marzagão, David, Josh Murphy, Anthony P Young, Marcelo Matheus Gauy, Michael Luck, Peter McBurney et Elizabeth Black: *Team Persuasion*. Dans *Theory and Applications of Formal Argumentation : 4th International Workshop, TAFA 2017, Melbourne, VIC, Australia, August 19-20, 2017, Revised Selected Papers 4*, pages 159–174. Springer, 2018.

[13] Leite, Joao et Joao Martins: *Social abstract argumentation*. Dans *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[14] Mercier, Hugo et Dan Sperber: *The enigma of reason*. Harvard University Press, 2017.

[15] Mossakowski, Till et Fabian Neuhaus: *Modular semantics and characteristics for bipolar weighted argumentation graphs*. arXiv preprint arXiv :1807.06685, 2018.

[16] Polberg, Sylwia et Anthony Hunter: *Empirical evaluation of abstract argumentation : Supporting the need for bipolar and probabilistic approaches*. International Journal of Approximate Reasoning, 93 :487–543, 2018.

[17] Potyka, Nico: *Continuous Dynamical Systems for Weighted Bipolar Argumentation.* Dans *KR*, pages 148–157, 2018.

[18] Potyka, Nico: *Open-mindedness of gradual argumentation semantics*. Dans *Scalable Uncertainty Management : 13th International Conference, SUM 2019, Compiègne, France, December 16–18, 2019, Proceedings 13*, pages 236–249. Springer, 2019.

[19] Rago, Antonio, Francesca Toni, Marco Aurisicchio, Pietro Baroni *et al.*: *Discontinuity-Free Decision Support with Quantitative Argumentation Debates.* KR, 16 :63–73, 2016.

[20] Sanfeliu, Alberto et King Sun Fu: *A distance measure between attributed relational graphs for pattern recognition*. IEEE transactions on systems, man, and cybernetics, (3) :353–362, 1983.

[21] Taillandier, Patrick, Nicolas Salliou et Rallou Thomopoulos: *Introducing the Argumentation Framework Within Agent-Based Models to Better Simulate Agents' Cognition in Opinion Dynamics : Application to Vegetarian Diet Diffusion*. Journal of Artificial Societies and Social Simulation, 24(2), 2021.

Actes JIAF-JFPDA 2023

A New Evolutive Generator for Graphs with Communities and its Application to Abstract Argumentation

Jean-Marie Lagniez[1] **Emmanuel Lonca**[1] **Jean-Guy Mailly**[2] **Julien Rossit**[2]

[1] CRIL, Univ. Artois, CNRS, France
[2] Université Paris Cité, LIPADE, F-75006 Paris, France
{lagniez, lonca}@cril.fr
{jean-guy.mailly, julien.rossit}@u-paris.fr

Abstract

Graph generators are a powerful tool to provide benchmarks for various subfields of KR (e.g. abstract argumentation, description logics, etc.) as well as other domains of AI (e.g. resources allocation, gossip problem, etc.). In this paper, we describe a new approach for generating graphs based on the idea of communities, i.e. parts of the graph which are densely connected, but with fewer connections between different communities. We discuss the design of an application named crusti_g2io implementing this idea, and then focus on a use case related to abstract argumentation. We show how crusti_g2io can be used to generate structured hard argumentation instances which are challenging for the fourth International Competition on Computational Models of Argumentation (ICCMA'21) solvers.

1 Introduction

Graph-based models are widespread in many fields of Knowledge Representation and Reasoning [10] (e.g. abstract argumentation [12], description logics [28], etc.) as well as other domains of Artificial Intelligence like multi-agent systems (e.g. resources allocation [4], gossip problem [11], etc.).

The popularity of this kind of representation appeals automated graphs generation approaches to provide challenging benchmarks that can put to the test practical tools developed within these various frameworks. The literature offers different methods to generate graphs, which exhibit different properties and various applicabilities to concrete problems and scenarios. In particular, one challenge consists in generating *structured* instances, i.e. random graphs which present interesting patterns that are relevant for some specific application. A well-known example of such a structured generation model is the Watts-Strogatz model [31], where the generated graphs have a *small world* property. Among the variety of graphs that have been studied, some recent works are interested in the generation of graphs with communities of nodes, i.e. parts of the graphs which are densely connected, but with fewer connections between different communities [20]. Such models include BTER [23] and Darwini [17], that propose to link nodes inside so-called affinity blocks, and then to add links between the nodes from different blocks. This kind of graphs is for example able to model interactions between people, including in social networks [26]. The importance of generators for this kind of graphs is amplified by the privacy issues that come when using real social networks data [32].

Being a model of choice to represent people communities, graphs with communities are a *de facto* candidate to encode large debates, which could be the source of argumentative reasoning. Computational argumentation has become an important sub-field of Knowledge Representation and Reasoning, being a prominent formalism for non-monotonic reasoning [12] in general, and reasoning with inconsistent knowledge in particular [3].

However, until recently, there was an important lack of practical approach for computing the solutions of argumentation problems. Although there were some algorithmic approaches proposed in the literature, few pieces of software were actually available for the community. This has changed (mainly) thanks to the organization of the First International Competition on Computational Models of Argumentation (ICCMA), in 2015. Since then, some solvers have been proposed, based either on original techniques dedicated to argumentation frameworks [19, 21, 22], or on translation into other frameworks which have already proven efficient computational benefits (namely Boolean satisfaction problem (SAT) [16, 24, 29], Answer Set Programming (ASP) [15]

or Constraint Satisfaction Problems (CSP) [5]). The efforts of the community at the occasion of the various editions of ICCMA have seen a general increase of the quality of the computational approaches for argumentation, both with respect to the correctness of the approaches and their runtime efficiency. However, the lack of challenging and realistic benchmarks for argumentation is still an issue for the community. Using (community-based) graph generators was naturally quickly considered to fill this hole.

BTER and Darwini approches are customizable in the sense that some metrics can be given to produce graphs with communities of expected shapes, but the manner the communities are linked is tied with the community generation algorithm which follows the Erdös-Rényi model [18]. In this paper, we propose a new generation approach and we apply it to abstract argumentation.

Our approach is based on three components : we first generate an *outer graph* which gives a global skeleton for the structure of the generated instance ; then in each node of the outer graph, we generate an *inner graph* i.e. a community of nodes ; and finally when two nodes of the outer graph are connected, we use a *linker* to add some relations between the corresponding inner graphs. We then show how our method can be applied to generate structured, challenging graphs for argumentation purpose. The added value of our approach compared to the previous ones lies in its ability to be generic and modular, since any of the three components can be easily replaced by other versions. In particular, the outer and inner graphs can be generated through classical generation models like Erdös-Rényi [18], Watts-Strogatz [31] or Barabási-Albert [1], but any other model could be plugged instead (including BTER and Darwini graphs themselves). Our contribution includes a documented, open-source graph generator following this inner/outer template. This application has been made to be easily used by any user, but also to be convenient for developers who want to add new features like graph generators, linkers or output formats.

The paper is organized as follows. After providing some necessary background in Section 2, we first introduce the inner/outer model in Section 3. This model is then instantiated to generate abstract argumentation benchmarks in Section 4. Section 5 presents some related works. Necessary and relevant features of our framework are presented in Section 6, followed by some experiments in Section 7. Finally, Section 8 draws some conclusions and highlights avenues for future work.

2 Background on Graph Generators

Let us first describe various classical graph generation models, which are later used in the conception of our new approach. In the following, we use $G = \langle N, E \rangle$ to denote any graph, where N are the nodes and E are the edges. In the case of a directed graph, $E \subseteq N \times N$, while in the case of a non-directed graph, $E \subseteq \{\{a, a'\} \mid a, a' \in N\}$. We also consider simple models like *paths* and *trees*.

Erdös-Rényi The Erdöz-Rényi (or binomial graph) generation model [18] takes into consideration two parameters $n_e \in \mathbb{N}$ and $p_e \in]0, 1]$ to construct graphs $\langle N, E \rangle$ with $|N| = n_e$ nodes, where for each couple $(a_i, a_j) \in N \times N$ there is a probability p_e to add an edge (a_i, a_j) in E.

Watts-Strogatz The model proposed in [31] considers a number of arguments $n_w \in \mathbb{N}$ and an even number $k_w \in \mathbb{N}$ (s.t. $k_w < n_w$) to construct a ring lattice made of n_w nodes, where each node is linked to k_w other nodes. Then, for each node a and each edge (a, b) of this node, there is a probability p_w of re-wiring the edge (avoiding to duplicate an existing edge or to link the node a with itself). Such graphs are called *small worlds*, i.e. for any two nodes in the graph, the shortest path between them has a logarithmic length in the number of nodes.

Barabási-Albert The preferential attachment model by [1] is based on two parameters $n_b, m_b \in \mathbb{N}$. It allows to generate graphs $\langle N, E \rangle$ where $|N| = n_b$, which are built by incrementally enlarging an initial graph (possibly made of a single node), such that each new node is attached to m_b nodes with a preference for existing nodes with the higher degree (formally, the probability to attach a new node a to an existing node b is $p_b = \frac{deg(b)}{\sum_c deg(c)}$ where $deg(b)$ (resp. $deg(c)$) is the degree of b. (resp. of c), and c iterates over the set of nodes already present in the graph).

Community-based Graphs Some models have already been proposed in the literature to incorporate the notion of community within the structure of the graphs, such as BTER and Darwini. BTER [23] splits a set of k nodes into so-called affinity blocks (i.e. the communities of nodes), which are then locally linked, and finally nodes from different blocks are linked together. Affinity blocks are linked following the Erdös-Rényi model, while the links between different blocks use the Chung-Lu model [9] (which is an extension of the Erdös-Rényi model). Darwini [17] performs a similar process, with an additional starting point which consists in mapping each node with its degree and clustering coefficient.

Directed/Undirected Graphs In the definition that we provide for the Erdös-Rényi model, we assume that the graph is directed. It is easy to obtain a non-directed graph by choosing to add an (undirected) edge $\{a_i, a_j\}$ with a probability p_e (instead of considering both the directed edges (a_i, a_j) and (a_j, a_i)). Similarly, obtaining a directed path is easy (once the non-directed graph made of a single

path $(a_1, a_2, \ldots, a_n)$ is built, each edge is directed from a_i to a_{i+1}, for each $i \in \{1, \ldots, n-1\}$). In the case of trees, we can also easily build a directed graph, for instance with the edges going "down" from the root to the leaves.

Unfortunately, the graphs generated by the other models are generally non-directed. When a directed graph is required, it could be possible to randomly select the orientation of each edges. However, depending the targeted application, this solution is still not satisfactory. For example, when considering the problem of generating argumentation frameworks, it is important to consider symmetrical attacks between argument in order to cover a wide range of cases. To do so, an option consists in considering a parameter $p_s \in [0, 1]$ representing the probability that a given edge should be symmetrical. Then, for an edge $\{a_i, a_j\}$ in the non-directed graph, there will be a probability p_s to have both (a_i, a_j) and (a_j, a_i) in the directed version of the graph, and a probability $\frac{1-p_s}{2}$ for either (a_i, a_j) or (a_j, a_i).

3 The Inner/outer Model

As mentioned earlier, existing community-based graphs generators suffer from being tied to the model used to build their communities. In order to overcome this issue, we propose a new approach for generating graphs that considers underlying graph structures. Roughly speaking, we implement the reverse approach of the BTER process : we first generate the relations between the communities, then we generate communities and finally we link them by connecting some of their inner elements. More precisely, an *outer graph* $G_{\mathcal{G}^O}$ that will be used as a skeleton for the instance is first constructed from a graph generator $\mathcal{G}^O$. Then, each node of this graph is associated with a fresh *inner graph* (fresh in the sense where nodes of each inner graph are disjoint) built by another generator $\mathcal{G}^I$. In order to link inner graphs together, we successively consider each inner graph G_n rooted to a node n of $G_{\mathcal{G}^O}$ and add edges between it and the inner graphs $G_{n'}$ rooted to a node n' when an edge exists in the outer graph between n and n'. The final graph is then the set of inner graphs together with the added edges. Interestingly, such generation process can handle both directed and undirected graphs (with the constraint that both generators and the added edges involve edges of the same kind). Formally, the function in charge of linking inner graphs together in the directed case is defined as follows :

Definition 1 (Directed linker) *A* linker over directed graphs *is a mapping* $\mathcal{L}_d$ *such that, for any* $G_1 = \langle N_1, E_1 \rangle$ *and* $G_2 = \langle N_2, E_2 \rangle$ *:* $\mathcal{L}_d(G_1, G_2) \subseteq (N_1 \times N_2) \cup (N_2 \times N_1)$.

For the undirected case the linker is defined as follows :

Definition 2 (Undirected linker) *A* linker over undirected graphs *is a mapping* $\mathcal{L}_u$ *such that, for any* $G_1 = \langle N_1, E_1 \rangle$ *and* $G_2 = \langle N_2, E_2 \rangle$ *:* $\mathcal{L}_u(G_1, G_2) \subseteq \{\{n_1, n_2\} \mid n_1 \in N_1, n_2 \in N_2\}$.

Without loss of generality, in the following we only consider the directed case. Algorithm 1 formalizes our approach.

Algorithm 1 Inner/outer graph generation

Input: an outer graph generator $\mathcal{G}^O$, an inner graph generator $\mathcal{G}^I$ and a linker $\mathcal{L}$
Output: an inner/outer graph
1: $G_{\mathcal{G}^O} \leftarrow \langle N, E \rangle$ a $\mathcal{G}^O$-generated graph
2: **for** $n \in N$ **do**
3: $\quad G_n \leftarrow \langle N_n, E_n \rangle$ a $\mathcal{G}^I$-generated graph
4: **end for**
5: $L = \emptyset$
6: **for** $(n, n') \in E$ **do**
7: $\quad L \leftarrow L \cup \mathcal{L}(G_n, G_{n'})$
8: **end for**
9: **return** $\langle (\bigcup_{n \in N} N_n), (\bigcup_{n \in N} E_n) \cup L \rangle$

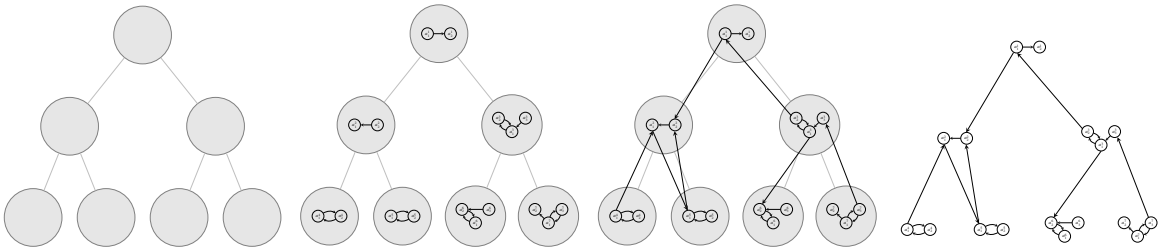
The generation process starts with the generation of the outer graph, i.e. the graph which is used as the skeleton of the instance (line 1). Then, each node of this outer graph is associated with an inner graph which is built by the dedicated graph generator $\mathcal{G}^I$ (line 3). The rest of the algorithm consists in building some links between the different inner graphs, with respect to the structure of the outer graph. To do so, for each edge in the outer graph, the inner graphs associated with the two outer graph nodes under consideration are passed to the linker (line 7) ; the resulting set of edges is stored. At the end, the algorithm returns the union of the inner graphs plus the edges returned by the linker, producing the final inner/outer graph.

Our approach offers the advantage of being flexible and allows, for instance, to generate a community graph such that the outer graph is a tree ($\mathcal{T}$) and inner graphs are Erdös-Rényi graphs ($\mathcal{ER}$). It is also possible to generate paths of Barabási-Albert ($\mathcal{BA}$) graphs, or Watts-Strogatz ($\mathcal{WS}$) graphs made of $\mathcal{WS}$ communities, etc.

Example 1 *Let us illustrate the generation algorithm with* $\mathcal{G}^O = \mathcal{T}$, $\mathcal{G}^I = \mathcal{ER}$, *and* $\mathcal{L}$ *a function which returns a random set of edges between two graphs. An example of generation process is given at Figure 1. Figure 1a shows the outer graph, which is thus a balanced binary tree. Then, in each node of the tree, an inner graph is generated thanks to the Erdös-Rényi model (Figure 1b). Figure 1c shows the addition of edges between the inner graphs thanks to the linker. And finally, the resulting graph is shown at Figure 1d.*

4 Application to Abstract Argumentation

From a practical point of view, it seems reasonable to assume that large debates may be structured in smaller



(a) Outer Graph (b) Inner Graphs (c) Edges between Inner Graphs (d) Final Graph

FIGURE 1 – Generation process.

sub-debates, which are only connected by few links; this would follow how people are themselves structured in social networks [26]. More precisely, this can be the case, for instance, in argumentation frameworks related to multi-issue negotiation, where each sub-debate corresponds to the arguments focusing on one issue, and the links between sub-debates correspond e.g. to the concessions ("If I accept to pay more for this car, then I want the company to deliver it faster" makes the link between the sub-debate about the price of the car and the sub-debate about the delivery date). So, in some sense, these sub-debates represent communities of arguments which are strongly related (i.e. there is a high density of attacks in such a community), and there are fewer relations between different communities. In this section, we briefly recall basic notions of abstract argumentation.

Definition 3 *An* abstract argumentation framework *(AF) [12] is a directed graph $\mathcal{F} = \langle A, R\rangle$ where A is a set of* arguments *and $R \subseteq A \times A$ is the* attack relation *between arguments.*

We say that an argument a *attacks* an argument b if $(a, b) \in R$. This is generalized to sets of arguments : S attacks b (resp. S') if there is some $a \in S$ which attacks b (resp. some $b \in S'$). A set S *defends* an argument a if for any b attacking a, there is a $c \in S$ attacking b. Acceptability of arguments is usually evaluated thanks to the notion of extensions, i.e. sets of collectively acceptable arguments. Various semantics exist for defining extension [12]. Formally, a semantics is a function $\sigma : \mathcal{F} = \langle A, R\rangle \mapsto \mathcal{E} \subseteq 2^A$.

Definition 4 *Given an AF $\mathcal{F} = \langle A, R\rangle$, and a set of argument $S \subseteq A$,*

- *$S \in \mathtt{cf}(\mathcal{F})$ iff $\forall a, b \in S$, $(a, b) \notin R$,*
- *$S \in \mathtt{ad}(\mathcal{F})$ iff $S \in \mathtt{cf}(\mathcal{F})$ and S defends all its elements,*
- *$S \in \mathtt{co}(\mathcal{F})$ iff $S \in \mathtt{ad}(\mathcal{F})$ and S does not defend any argument in $A \setminus S$,*
- *$S \in \mathtt{pr}(\mathcal{F})$ if S is a $\subseteq$-maximal element of $\mathtt{ad}(\mathcal{F})$,*
- *$S \in \mathtt{stb}(\mathcal{F})$ iff $S \in \mathtt{cf}(\mathcal{F})$ and S attacks all the arguments in $A \setminus S$,*
- *$S \in \mathtt{gr}(\mathcal{F})$ iff S is the $\subseteq$-minimal element of $\mathtt{co}(\mathcal{F})$.*

where cf, ad, co, pr, stb *and* gr *stand respectively for* conflict-free, admissible, complete, preferred, stable *and* grounded.

See e.g. [12, 2] for more details about these semantics as well as other semantics defined in the literature. Let us illustrate the complete, preferred, stable and grounded semantics with the following example :

Example 2 *The extensions for* co, pr, stb *and* gr *of the AF $\mathcal{F} = \langle A, R\rangle$ depicted in Figure 2 are given in Table 1.*

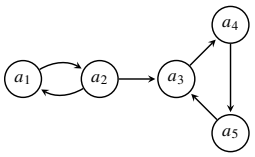


FIGURE 2 – The AF $\mathcal{F}$

Semantics σ	Extensions $\sigma(\mathcal{F})$
co	$\emptyset, \{a_1\}, \{a_2, a_4\}$
pr	$\{a_1\}, \{a_2, a_4\}$
stb	$\{a_2, a_4\}$
gr	$\emptyset$

TABLE 1 – Extensions of the AF $\mathcal{F}$

Recall that reasoning with AFs is generally hard, with many classical problems at the first or second level of the polynomial hierarchy [14].

5 Related Works

The next sections presents the application we developed to generate inner/outer graphs and its application to generate AF benchmarks. There already exists tools for generating AFs from random graph generators. But, from the best of our knowledge, these tools do not modify the underlying graph generated by these models. In [7], the authors

propose the C++ framework `AFBenchGen`. It is an AF generator based on the Erdös-Rényi model ($\mathcal{ER}$). In [8], the same authors proposed an extension of `AFBenchGen`, called `AFBenchGen2` which is written in Java, that also consider two additional random graph generator models, which are the Watts-Strogatz ($\mathcal{WS}$) and Barabási-Albert ($\mathcal{BA}$) models. For these two generators the random graphs are used as such. Our tool is much more general than the `AFBenchGen` family of AFs generators. Indeed, by considering the simple graph consisting in one node as outer graph, it is possible to have the exactly same behaviour.

In [25], we introduced a new method for generating challenging benchmarks for the ICCMA'21 competition. This generator is the fundamental basis of our tool. More precisely, we have proposed three variants of our generator $\langle \mathcal{G}^O, \mathcal{G}_i^I, \mathcal{L} \rangle$, with $i \in \{1, 2, 3\}$, defined as follows. In our case $\mathcal{G}^O = \mathcal{T}$, meaning that the underlying graph is actually a perfectly balanced d-tree of height h, where d and h are fixed and provided as parameters. The only difference between the three variants is the inner graphs generator : $\mathcal{G}_1^I = \mathcal{ER}$, $\mathcal{G}_2^I = \mathcal{BA}$, while $\mathcal{G}_3^I$ is a random pick of either $\mathcal{ER}$ or $\mathcal{BA}$, which means that in the first case all the local graphs are Erdös-Rényi graphs, in the second case they are all Barabási-Albert graphs, and in the last case they can be either of them with a probability 0.5.

Once the outer graph has been generated, the inner graphs are linked as follows. For this generation model, the iteration over the set of edges (line 6 in Algorithm 1) is a breadth-first graph traversal from the root to the leaves of the tree. For each inner graph associated with an outer node o, `k` nodes are randomly selected (`k` varies from 5 up to 12 for the benchmarks generated for the ICCMA'21 competition). The descendants $\{o_1, \ldots, o_m\}$ of o are iteratively considered. For each o_i, between 20% and 70% of the inner nodes contained in o_i are randomly selected. Then, for each node n_1 picked in o and with each node n_2 picked in o_i one of the attacks (n_1,n_2) or (n_2,n_1) is added randomly.

In this paper a slightly modified version of the tool proposed for generating the ICCMA'21 benchmarks has been considered. Inner graphs are only linked with their children (and not with any of their descendants). Moreover, a ratio of 20% has been considered for selecting the edges that are added between communities (instead of a ratio between 20% and 70% of the nodes).

6 The crusti_g2io graph generator

We built a command line application called crusti_g2io, dedicated to the generation of inner/outer graphs. It is made available under the terms of the GNU GPLv3 on Github account of the *Centre de Recherche en Informatique de Lens*. [1]

We took advantage of the Rust programming language to provide an efficient, memory-safe application, even in parallel context. In addition, Rust allows crusti_g2io to be both an application and a library (the project in mainly a Rust library with additional code to create the application). Interestingly, Rust libraries can be turned into C libraries (static or dynamic) or be linked with them. This makes crusti_g2io able to use any library that can be turned into a C library or to be used itself with any program that can load C libraries, allowing for example Go and Python bindings.

The application can be used to generate both directed and undirected graphs. In the following, we describe how to use the application for directed graphs only ; however, going from directed to undirected is as simple as replacing `directed` by `undirected` in the commands.

The first goal of crusti_g2io is to be easy to install and to use. The only requirement to use it is to have a Rust compiler installed (except of course if you were given an already compiled version) ; then, executing a standard release build command (`cargo build -release`) produces the executable (in the `target/release` directory on UNIX systems). The user can also use the `cargo install` command to compile and install the program on its computer.

From a user perspective, crusti_g2io is made to be used without looking at its documentation. Calling crusti_g2io with `-h`, `-help` displays the list of the commands and what they do. Calling crusti_g2io with a command and one of the two help flags displays the help message associated with the command. For example, calling `crusti_g2io generate-directed -h` explains what `generate-directed` does, gives its mandatory and optional options (along with their descriptions).

The goal of crusti_g2io is to generate a graph from an outer graph generator, an inner graph generator and a linker, and to output it using a graph output format. Thus, these exact four options form the exact set of mandatory options for the `generate-directed` command. Again, they can be recalled by typing `crusti_g2io generate-directed -h` in a terminal. Concerning the lists of the available graph generators, linkers and graph output formats, they can all be retrieved by a crusti_g2io command (respectively `generators-directed`, `linkers-directed` and `display-engines-directed`) ; calling these commands also indicates how to parameterize the generators, linkers or formats which need it. Figure 3 shows how to build a tree-like outer graph (`-o`) of 10 inner (`-i`) Erdös-Rényi graphs of 100 nodes with a probability of 0.5 where links (`-l`) are created between lowest degree nodes, and export (`-x`) it in the file `t_10_er_100_50.dot` using the dot format (`-f`). The required parameters for generators and linkers (when needed) are given after a slash

1. At the time of submission, it is here : `https://www.cril.univ-artois.fr/~lonca/crusti_g2io-94dfb5e8b6e14a3c13bf9f861b0ad221533815de.zip`.

```
me@PC:~/crusti_g2io generate-directed -o tree/10 -i er/100,0.5 -l min_incoming -x t_10_er_100_50.dot -f dot
![INFO ] [2023-03-03 10:54:39] crusti_g2io 0.1.0
[...]
![INFO ] [2023-03-03 10:54:39] random seed is 6203895736620038422
![INFO ] [2023-03-03 10:54:39] beginning the outer graph generation
![INFO ] [2023-03-03 10:54:39] beginning the inner graphs generation
![INFO ] [2023-03-03 10:54:39] beginning the linking
![INFO ] [2023-03-03 10:54:39] generated a graph with 1000 nodes and 24882 edges
![INFO ] [2023-03-03 10:54:39] exiting successfully after 45.6625ms
```

FIGURE 3 – Example on invocation of crusti_g2io.

and split by commas (see `tree/10` and `er/100,0.5` in the figure). Embedded graph generators include the famous Erdös-Rényi, Watts-Strogatz and Barabási-Albert models, trees and chains. Concerning the linkers, one is a random one, one links nodes with the least incoming edges, and the last one links the nodes with index 0 — which can have some meaning, in particular if a graph is initialized with a special value like in the Barabási-Albert model. Finally, The Graphviz DOT and GraphML formats are available, just like the abstract argumentation related format APX we use in next section.

```rust
#[derive(Default)]
pub struct ErdosRenyiGeneratorFactory;

impl<Ty, R> NamedParam<BoxedGenerator<Ty, R>> for ErdosRenyiGeneratorFactory
where
    R: Rng,
    Ty: EdgeType,
{
    fn name(&self) -> &'static str {
        "er"
    }

    fn description(&self) -> Vec<&'static str> {
        vec![
            "A generator following the Erdős-Rényi model.",
            "First parameter gives the number of nodes of the graph, while t
        ]
    }

    fn expected_parameter_types(&self) -> Vec<ParameterType> {
        vec![ParameterType::PositiveInteger, ParameterType::Probability]
    }

    fn try_with_params(
        &self,
        parameter_values: Vec<ParameterValue>,
    ) -> Result<BoxedGenerator<Ty, R>> {
        let n = parameter_values[0].unwrap_usize();
        let p = parameter_values[1].unwrap_f64();
        Ok(Box::new(move |r| {
            petgraph_gen::random_gnp_graph(r, n, p).into()
        }))
    }
}
```

FIGURE 4 – Implementation of a new graph generator for Erdös-Rényi graphs using the `petgraph` library.

These generators, linkers and formats are a very small subset of what is offered by the literature. This is the reason why we tried to make the addition of new content as easy as possible for developers. For example, to add a new generator, it is only required to create a structure that implements the four functions of the dedicated trait and to register it in the set of generators. Concerning the trait, the implementation of three functions out of four is straightforward (see Figure 4 for an example of implementation for $\mathcal{ER}$ graphs using the `petgraph` library – `https://crates.io/crates/petgraph`), as they respectively return the name of the generator to be used on the command line interface, the description of the generator, and the types of the expected parameters. The last function is the one dedicated to the generation of graphs : it takes as input the (checked) parameter values as given on the command line interface (i.e. the content following the slash) and returns a closure which takes a pseudo-random number generator (PRNG) and produces a graph. The registration of the new generator consist of adding an import statement and a single line of code. Adding a new linker requires a similar process, except that the closure takes a PRNG and two graphs, and returns a vector of edges. When invoking crusti_g2io, the graph can be printed out on the standard output (this is the default behaviour) or exported to a file. The default behavior mixes log messages and the graph; this can be prevented by hiding the log messages (e.g. by setting the corresponding option) or by exporting the graph to a file. Adding a new output format is similar to adding a new generator or linker.

Finally, crusti_g2io is made to produce reproductible results. By default, it uses an unpredictable random seed; in order to get reproductible results, the user can set the random seed with the `-s` option on the command line. Regardless of the fact the seed was specified or randomly specified, it is logged so the results can be reproduced. An effort was made in order to mix reproducibility and the use of the full power of the computers, as the application computes the inner graphs and the links between these graphs in a parallel fashion. In order to get reproductible results, the program first computes the outer graph using the global PRNG initialized with the provided seed. Then, each outer node is sequentially associated a random seed using the global PRNG. This way, each inner graph generation process can receive a PRNG which directly depends on the CLI-provided seed, enforcing the reproducibility of the generation for a given seed. The same approach is used for the linking process.

7 Using crusti_g2io to generate challenging abstract argumentation problems

In this section, we use crusti_g2io to generate structured instances for abstract argumentation solvers. The goal is to generate overall challenging instances composed of multiple communities. In addition, we want to generate instances with a large amount of small communities, but also instances with less communities of a greater size. To achieve this, we aim at drawing the frontier between hard and too-hard instances for a set of community sizes, densities and counts.

In order to evaluate the difficulty induced by the generated argumentation graphs, we chose to compute extensions (putting acceptance queries aside) to consider the whole graphs instead of problems that could be related to a reduced area of the graph. We arbitrary selected a problem of the first level of the polynomial hierarchy (SE-ST : compute an extension for the stable semantics) and one of the second level (SE-PR : compute an extension for the preferred semantics). For both tracks, we used the solvers that got the best results at the ICCMA'21 competition, namely A-Folio-DPDB [2] for the SE-ST track and μ-Toksia [29] for the SE-PR track. As A-Folio-DPDB delegates the SE-ST problems to the μ-Toksia solver submitted at ICCMA'19, we finally used μ-Toksia (2019) for SE-ST problems. We chose to build communities of Erdös-Rényi graphs, since those graphs were already used to generate AFs and can be naturally generated as directed graphs. Communities were linked following a tree template (like ICCMA'21 instances). The linker processes in a way inspired by the $\mathcal{ER}$ generator : each possible edge from the source graph to the target graph is added with probability 0.2.

In the first part of our experiments, we sought which sizes of communities are small enough to be part of our graphs. We used crusti_g2io to generate single Erdös-Rényi graphs (by asking for an outer graph composed of a single node) with different number of nodes (from 100 to 1000) and probability for each edge to appear (0.1, 0.2 and 0.5). For each setting, we generated 10 different graphs by feeding the app with random seeds from 0 to 9 ; the computation times are averages of these 10 values, and a timeout of at least one makes the average be also timeout. We run experiments on machines equipped with Intel Xeon E5-2637 v4 processors and 128GB of RAM, and the timeout was fixed to 600s, as in ICCMA'21. Table 2 shows some experimental results.

First, we can note that for a given number of nodes, instances are more difficult for lower Erdös-Rényi probability values. This may be explained by the lower number of constraints, making preferred extensions admit more arguments, and stable extensions less common. This hypothesis would require further investigation, but is off-topic here since we are only interested in the difficulty of the instances.

ER proba.	ER nodes	SE-ST (s)	SE-PR (s)
0,1	100	0,01	0,03
	200	3,13	9,14
	300	—	—
	400	—	—
0,2	100	0,02	0,02
	200	1,85	4,13
	300	13,87	22,91
	400	—	—
0,5	100	0,01	0,02
	200	0,10	0,07
	300	0,14	0,37
	400	0,23	4,11
	500	1,81	13,97
	600	4,28	16,56
	700	3,34	41,23
	800	6,72	74,41
	900	11,27	141,24
	1000	14,32	67,37

TABLE 2 – CPU time required by μ-Toksia 2019 (resp. 2021) to compute a single stable (resp. preferred) extension for different sizes of Erdös-Rényi graphs. CPU times are average of 10 values. If a timeout was reached for at least one graph, — is reported.

Communities of 100 arguments seem easy for both SE-ST and SE-PR, whatever the probability setting. With a setting of 0.1, the problems begin to require multiple seconds to be solved for 200 nodes ; this value should not be exceeded for instances involving several communities. A single community of 300 nodes cannot be solved in this context. With a setting of 0.2, the limit in terms of number of nodes to consider for multiple communities seems to be between 200 and 300 ; for this value, a single community requires more than 10 seconds for SE-ST, and more than 20s for SE-PR. A setting of 0.5 allows to generate instances with a single community of at least 1000 nodes. Interestingly we remarked that in this case, all instances admit stable extensions, which is not the case for the other probability settings. This indicates that these instances have a special structure that might make solvers work differently on them. Finally, as expected, the SE-PR problem takes more time to be solved than SE-ST.

Now that we have bounds on the size of the communities to consider, we can experiment the difficulty induced by the number of communities. We generated complete binary trees of Erdös-Rényi communities, where each community is linked to the ones associated with its children.

For this second experiment session, we considered Erdös-Rényi with nodes between 100 and 500 with the same three

2. https://github.com/gorczyca/dp_on_dbs/tree/competition

probability settings. We assumed the multiplicity of the communities would make the instances very hard for the 0.5 probability for more than 500 nodes per community. We considered outer tree heights from 3 to 9, making the outer graphs contain from 7 to 511 nodes. For each setting, 10 instances were generated with random seeds going from 0 to 9. We used to same machines and timeout than before. Figures 5 and 6 report the interesting parts of these new results. The plots on Figure 5 correspond to the results for the SE-ST track, while Figure 6 reports the results for SE-PR. For each figure, the three subfigures are each associated with a density setting (0.1, 0.2 and 0.5). For each subfigure, the average computation time is given on the y-axis, while the x-axis gives the number of communities ; the lines gives the different community sizes.

We first focus on the SE-ST results, given by the plots at Figures 5a, 5b and 5c. Concerning the results of µ-Toksia 2021 for the 0.1 probability setting (Figure 5a), we can observe that the problems are too easy when the number of nodes per community is lower than 200 (all solved in few seconds even for 511 communities) and too hard when it is above this value (such problems cannot be solved when there are more than 31 communities). Thus, this setting does not allow us to draw a clear frontier between the hard and the too-hard instances. This is also the case for the 0.5 probability setting (Figure 5c) for which the instances are surprisingly very difficult even for low values of community sizes and community counts. This is not an unexpected result since as noted below, these instances have a special structure that might prevent µ-Toksia to solve them. By the way, we discovered that µ-Toksia was not able to prove the absence of stable extension in any community-based instance with this density. If such instances are included in our benchmarks, then µ-Toksia may suffer from this special kind of instances. Fortunately, the 0.2 case (Figure 5b) perfectly fits our needs of frontier as it shows multiple settings of community sizes and counts are solvable but difficult (hundreds of seconds required to solve) namely the sets of 511 communities of size 225, the sets of 255 communities of size 250 and the sets of 63 communities of size 275.

Now, we discuss the SE-PR results, given by the plots at Figures 6a, 6b and 6c. Just like for SE-ST, the 0.1 probability setting (Figure 6a) does not seem to be an interesting value for us since little changes in community sizes makes the difficulty a lot higher : see e.g. the difference between communities of 175 nodes — almost difficult instances when there are 511 of them — and 200 nodes — where instances are too difficult for 255 communities. Things are a little better for the 0.2 probability (Figure 6b) when considering communities of size between 225 and 300, but the real interesting setting in this case if the 0.5 probability (Figure 6c). In this case, we can find at least three cases of different community sizes for which hard instances exist : the sets of 511 communities of 175 nodes, the sets of 255 communities of 300 nodes and the sets of 127 communities of 500 nodes.

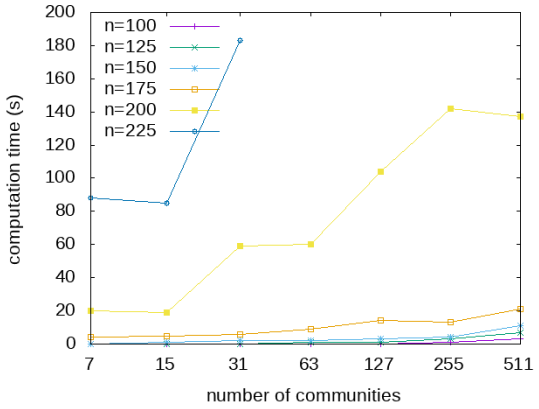
To conclude this section, it is worth noting that crusti_g2io generated the instances very fast. For the graph generation, we took advantage of machines with a higher number of processor cores. We dedicated to each process an Intel Xeon Gold 6248 (a 20-cores processor) and 192GB of RAM. The biggest instances we considered are the ones with 511 communities of 500 nodes with a probability setting of 0.5, for which the graph admits 255500 nodes and more than 89 millions edges. For these instances, the graph generation itself took less than 4s each. A little longer was necessary to translate the graphs into argumentation frameworks and store them using the (verbose) APX format on the hard disk. With these additional translation and writing times, the average wall-clock time was 19.62s.

8 Conclusion

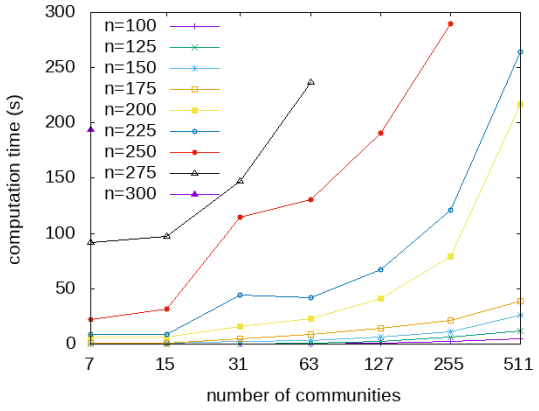
In this paper, we have defined a new approach for generating (directed or non-directed) graphs based on the concept of communities, which are graphs where some subparts of the graph are highly connected, but are loosely related to other subparts. Our approach uses a so-called inner/outer template, i.e. we first generate an outer graph representing the global structure of the graph, then in each node of the outer graph we generate an inner graph, and finally we use a linker to add edges between nodes of inner graphs which are connected in the outer graph structure. The proposed model is particularly generic and modular, since all the components (outer graph generator, inner graph generator and linker) can be replaced by other generators or linkers. Our model is particularly well suited for abstract argumentation, since large debates (i.e. large argumentation frameworks) can naturally be split into sub-debates which are only connected by a few arguments and attacks. We have described our open-source tool for the generation of graphs, and especially we have shown that this tool allows to generate meaningful argumentation framework instances with a level of difficulty for standard computational problems which can be adapted thanks to the choice of some parameters.

Several avenues for future work can be highlighted. Regarding the tool, a natural development direction is to design an even more generic framework, allowing several levels of nested graphs (i.e. the inner graph generator could generate graphs which also follow the inner/outer template). We also plan to improve the usability of the tool by describing the generation task in files (using e.g. the YAML or JSON format) instead of the command-line interface.

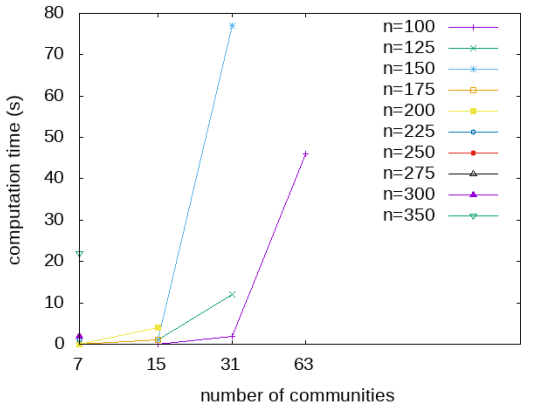
Regarding the issue of AF generation, we can improve the relevance of the tool by incorporating linkers which make sense in the context of abstract argumentation frameworks (for instance, we could add edges concerning in priority arguments which are skeptically accepted w.r.t. some given



(a) SE-ST, $\mathcal{ER}$ probability of 0.1

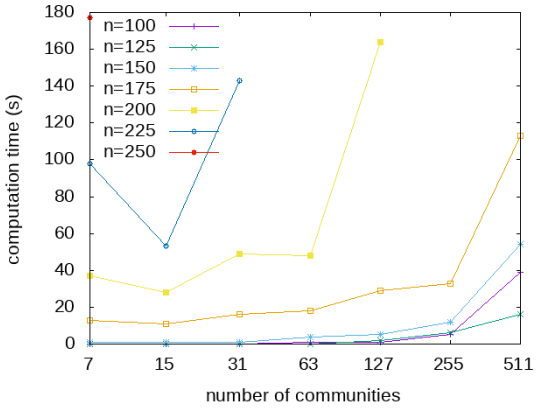


(b) SE-ST, $\mathcal{ER}$ probability of 0.2

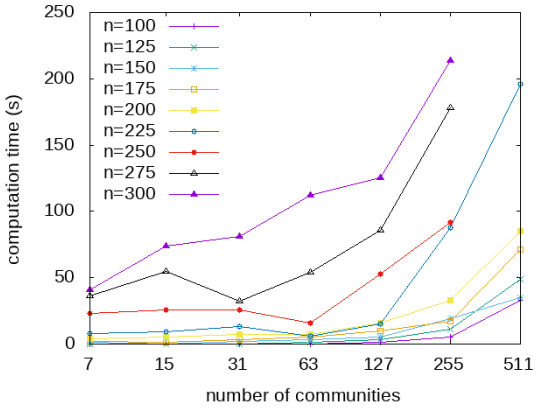


(c) SE-ST, $\mathcal{ER}$ probability of 0.5

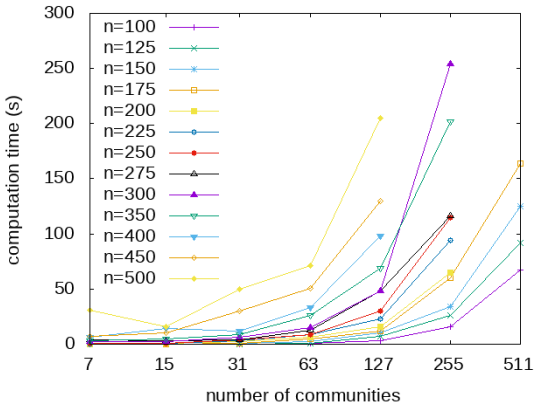
FIGURE 5 – CPU time (in seconds) required by µ-Toksia 2019 to compute a single stable extension for community graphs of different community sizes and different community count. CPU times are an average of 10 values.



(a) SE-PR, $\mathcal{ER}$ probability of 0.1



(b) SE-PR, $\mathcal{ER}$ probability of 0.2



(c) SE-PR, $\mathcal{ER}$ probability of 0.5

FIGURE 6 – CPU time (in seconds) required by µ-Toksia 2021 to compute a single preferred extension for community graphs of different community sizes and different community count. CPU times are an average of 10 values.

semantics). Another interesting future work consists in proposing generation models for more complex argumentation frameworks, which would require e.g. graphs with different kinds of edges or arguments (to incorporate supports [6] or incompleteness [27]) or graphs with weights associated with edges [13] or arguments [30].

Acknowledgements

This work has been partly supported by the CPER DATA Commode project from the "Hauts-de-France" Region, the ANR projects PING/ACK (ANR-18-CE40-0011) and AGGREEY (ANR-22-CE23-0005).

Références

[1] Barabási, A. et R. Albert: *Emergence of scaling in random networks*. Science, 286 :509–512, 1999.

[2] Baroni, P., M. Caminada et M. Giacomin: *Abstract Argumentation Frameworks and Their Semantics*. Dans *Handbook of Formal Argumentation*, pages 159–236. College Publications, 2018.

[3] Besnard, P. et A. Hunter: *Elements of Argumentation*. MIT Press, 2008, ISBN 978-0-262-02643-7.

[4] Beynier, A., Y. Chevaleyre, L. Gourvès, A. Harutyunyan, J. Lesca, N. Maudet et A. Wilczynski: *Local envy-freeness in house allocation problems*. Auton. Agents Multi Agent Syst., 33(5) :591–627, 2019.

[5] Bistarelli, S., F. Rossi et F. Santini: *ConArg : A Tool for Classical and Weighted Argumentation*. Dans *Proc. of COMMA 2016*. IOS Press, 2016.

[6] Cayrol, C. et M. C. Lagasquie-Schiex: *Bipolarity in argumentation graphs : Towards a better understanding*. Int. J. Approx. Reason., 54(7) :876–899, 2013.

[7] Cerutti, F., M. Giacomin et M. Vallati: *Generating Challenging Benchmark AFs*. Dans *Proc. of COMMA 2014*, 2014.

[8] Cerutti, F., M. Giacomin et M. Vallati: *Generating Structured Argumentation Frameworks : AFBenchGen2*. Dans *Proc. of COMMA 2016*, 2016.

[9] Chung, F. et Li. Lu: *The Average Distance in a Random Graph with Given Expected Degrees*. Internet Math., 1(1) :91–113, 2003.

[10] Cochez, M., M. Croitoru, P. Marquis et S. Rudolph (rédacteurs): *Proceedings of GKR 2020*, tome 12640 de *Lecture Notes in Computer Science*, 2021.

[11] Cooper, M. C., A. Herzig, F. Maffre, F. Maris et P. Régnier: *The epistemic gossip problem*. Discret. Math., 342(3) :654–663, 2019.

[12] Dung, P. M.: *On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games*. Artif. Intell., 77(2) :321–358, 1995.

[13] Dunne, P. E., A. Hunter, P. McBurney, S. Parsons et M. Wooldridge: *Weighted argument systems : Basic definitions, algorithms, and complexity results*. Artif. Intell., 175(2) :457–486, 2011.

[14] Dvorák, W. et P. E. Dunne: *Computational Problems in Formal Argumentation and their Complexity*. Dans *Handbook of Formal Argumentation*, pages 631–688. College Publications, 2018.

[15] Dvorák, W., S. A. Gaggl, A. Rapberger, J. P. Wallner et S. Woltran: *The ASPARTIX System Suite*. Dans *Proc. of COMMA 2020*, 2020.

[16] Dvorák, W., M. Järvisalo, J. P. Wallner et S. Woltran: *Complexity-sensitive decision procedures for abstract argumentation*. Artif. Intell., 206 :53–78, 2014.

[17] Edunov, S., D. Logothetis, C. Wang, A. Ching et M. Kabiljo: *Generating Synthetic Social Graphs with Darwini*. Dans *Proc. of ICDCS*, pages 567–577, 2018.

[18] Erdös, P. et A. Rényi: *On random graphs. I.* Publicationes Mathematicae, 6 :290–297, 1959.

[19] Geilen, N. et M. Thimm: *Heureka : A General Heuristic Backtracking Solver for Abstract Argumentation*. Dans *Proc. of TAFA 2017*, pages 143–149, 2017.

[20] Girvan, M. et M. Newman: *Community structure in social and biological networks*. Proc. of the NAS of the USA, 99(12) :7821–7826, 2002.

[21] Heinrich, M.: *The MatrixX Solver For Argumentation Frameworks*. CoRR, abs/2109.14732, 2021.

[22] Kinder, L., M. Thimm et B. Verheij: *A Labeling Based Backtracking Solver for Abstract Argumentation*. Dans *Proc. of SAFA 2022*, pages 111–123, 2022.

[23] Kolda, T., A. Pinar, T. Plantenga et C. Seshadhri: *A Scalable Generative Graph Model with Community Structure*. SIAM J. Sci. Comput., 36(5), 2014.

[24] Lagniez, J. M., E. Lonca et J. G. Mailly: *CoQuiAAS : A Constraint-Based Quick Abstract Argumentation Solver*. Dans *Proc. of ICTAI 2015*, pages 928–935, 2015.

[25] Lagniez, J. M., E. Lonca, J. G. Mailly et J. Rossit: *Design and Results of ICCMA 2021*. CoRR, abs/2109.08884, 2021.

[26] Leskovec, J., K. Lang, A. Dasgupta et M. Mahoney: *Statistical properties of community structure in large social and information networks*. Dans *Proc. of WWW 2008*, pages 695–704, 2008.

[27] Mailly, J. G.: *Yes, no, maybe, I don't know : Complexity and application of abstract argumentation with incomplete knowledge*. Argument Comput., 13(3) :291–324, 2022.

[28] Motik, B., B. Cuenca Grau, I. Horrocks et U. Sattler: *Representing ontologies using description logics, description graphs, and rules*. Artif. Intell., 173(14) :1275–1309, 2009.

[29] Niskanen, A. et M. Järvisalo: *μ-toksia : An Efficient Abstract Argumentation Reasoner*. Dans *Proc. of KR 2020*, pages 800–804, 2020.

[30] Rossit, J., J. G. Mailly, Y. Dimopoulos et P. Moraitis: *United we stand : Accruals in strength-based argumentation*. Argument Comput., 12(1) :87–113, 2021.

[31] Watts, D. et S. Strogatz: *Collective dynamics of "small-world" networks*. Nature, 393 :440–442, 1998.

[32] Wu, X., X. Ying, K. Liu et L. Chen: *A Survey of Privacy-Preservation of Graphs and Social Networks*. Dans *Managing and Mining Graph Data*, pages 421–453. 2010.

Actes JIAF-JFPDA 2023

Exploration des Sémantiques d'Argumentation Bipolaire: Une Analyse Basée Sur Les Principes

Liuwen Yu [1,2,3] **Caren Al Anaissy**[4] **Srdjan Vesic**[5] **Xu Li**[1] **Leendert van der Torre** [1,6]

[1] University of Luxembourg, Luxembourg
[2] University of Bologna, Italy
[3] University of Turin, Italy
[4] CRIL Univ. Artois & CNRS, France
[5] CRIL CNRS Univ. Artois, France
[6] Zhejiang University, China

Résumé

Dans cet article, nous introduisons et étudions sept types de sémantique pour les cadres d'argumentation bipolaires, chacun étendant l'interprétation de Dung de l'attaque avec une interprétation distincte du support. Premièrement, nous introduisons trois types de sémantique basés sur la défense en adaptant les notions de défense. Deuxièmement, nous introduisons deux types de sémantique basés sur la sélection, qui sélectionnent les extensions en comptant le nombre de supports. Troisièmement, nous analysons deux types de sémantiques traditionnels basés sur la réduction, sous des interprétations déductives et nécessaires du support. Nous fournissons une analyse complète de vingt-huit sémantiques d'argumentation bipolaire et dix principes au total.

1 Cadres d'Argumentation Bipolaires

Un cadre d'argumentation bipolaire est un triple $\langle Ar, att, sup\rangle$ où Ar est un ensemble fini d'arguments, et $att, sup \subseteq Ar \times Ar$ sont des relations binaires sur Ar appelées respectivement attaque et support.

La figure 1 illustre trois BAF, où les relations d'attaque sont représentées par des flèches pleines et les relations de support par des flèches en pointillés. Étant donné a, b dans Ar, $(a, b) \in att$ représente a attaque b, et $(a, b) \in sup$ représente a soutient b, les définitions de l'absence de conflit et de la défense fournies par Dung [6] sont appelées sans conflit$_0$ et défendu$_0$ dans le présent document.

Caren Al Anaissy et Srdjan Vesic ont bénéficié du soutien du projet AGGREEY ANR-22-CE23-0005 de l'Agence nationale de la recherche (ANR)

1.1 Sémantiques basées sur la défense

Nous introduisons trois nouvelles sémantiques de défense basées sur les notions de : sans-conflit$_0$, défendu$_1$, défendu$_2$, et défendu$_3$. En outre, nous définissons sans-conflit$_1$, sans-conflit$_2$ et sans-conflit$_3$ pour établir une définition générique des sémantiques basées sur la défense (définition 2). Défendu$_1$ nécessite que l'argument qui défend un autre argument, le soutient aussi, tandis que défendu$_2$ nécessite que l'argument qui défend un autre argument soit également soutenu. Défendu$_3$ va plus loin en exigeant d'attaquer à la fois les attaquants et ceux qui les soutiennent.

Definition 1 (Sans conflit$_{1-3}$ et Défendu$_{1-3}$)**.** *Soit $\mathcal{F} = \langle Ar, att, sup\rangle$ un BAF. Nous utilisons la même définition que Dung pour l'absence de conflit, c'est-à-dire $cf_1 \equiv cf_2 \equiv cf_3 \equiv cf_0$. En outre :*

- *l'ensemble des arguments défendus$_1$ par E, noté $d_1(\mathcal{F}, E)$, est l'ensemble des arguments a dans Ar tel que pour chaque argument b dans Ar attaquant a, il existe un argument c dans E attaquant b et soutenant a ;*
- *l'ensemble des arguments défendus$_2$ par E, noté $d_2(\mathcal{F}, E)$, est l'ensemble des arguments a dans Ar tel que pour tous les arguments b dans Ar attaquant a, il existe un argument c dans E attaquant b, et il existe un argument d dans E soutenant c ;*
- *l'ensemble des arguments défendus$_3$ par E, noté $d_3(\mathcal{F}, E)$, est l'ensemble des arguments a dans Ar tel que pour tous les arguments b dans Ar attaquant a, il existe un argument c dans E attaquant b, et pour tous les arguments d dans Ar soutenant b, il existe*

un argument e dans E attaquant d .

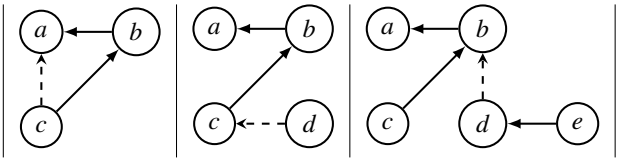


FIGURE 1 – Trois BAF illustrant les trois nouvelles notions de défense, pour la figure de gauche, $d_1(\mathcal{F}, \{c\}) = \{a, c\}$; pour la figure du milieu, $d_2(\mathcal{F}, \{c, d\}) = \{a, c, d\}$; pour la figure de droite, $d_3(\mathcal{F}, \{c, e\}) = \{a, c, e\}$

Différentes fonctions caractéristiques peuvent être définies en utilisant différentes notions de defence ainsi que la notion d'admissibilité.
Pour définir la sémantique complète (abrégée en c), préférée (p), ancrée (a) et stable (s) des cadres d'argumentation bipolaires, la définition suivante est générique et peut être utilisée avec n'importe quel type sans-conflit et de défense.

Definition 2 (Sémantiques$_{0-3}$)**.** *Une sémantique basée sur l'extension σ est une fonction qui fait correspondre un BAF ; $\mathcal{F} = \langle Ar, att, sup\rangle$ à un ensemble de sous-ensembles de Ar, écrit comme $\sigma_i^x(\mathcal{F})$, où $i \in \{0, 1, 2, 3\}$, $x \in \{c, p, s\}$ de la manière suivante :*

- $\sigma_i^c(\mathcal{F}) = \{E \subseteq Ar \mid cf_i(\mathcal{F}, E),\ d_i(\mathcal{F}, E) = E\}$;
- $\sigma_i^p(\mathcal{F}) = \{E \subseteq Ar \mid$ *pour tout ensemble admissible$_i$* $E', E \not\subseteq E'\}$;
- $\sigma_i^s(\mathcal{F}) = \{E \subseteq Ar \mid E$ *est admissible$_i$, et pour tous les arguments a qui ne sont pas dans E, il existe un argument b dans E attaquant a}.*
- $\sigma_i^a(\mathcal{F}) = \{E \subseteq Ar \mid E$ *est le plus petit point fixe de la fonction caractéristique* $d_i(\mathcal{F}, E)\}$

1.2 Sémantiques basées sur la sélection

Le support est pertinent dans l'étape de post-traitement de la théorie de Dung [7]. Les sémantique$_4$ et sémantique$_5$ sont des approches basées sur la sélection qui choisissent des extensions à partir de la sémantique$_0$. La sémantique$_4$ donne la priorité aux extensions ayant le plus grand nombre de supports internes, ce qui indique une plus grande cohésion au sein d'une coalition. La sémantique$_5$ sélectionne les extensions qui reçoivent le plus de supports externes, soulignant la force d'une coalition basée sur le support.

1.3 Sémantiques basées sur la réduction

Les approches basées sur la réduction ont été largement étudiées dans la littérature [3, 4, 5]. Sémantique$_6$ et sémantique$_7$ sont deux approches basées sur la réduction où le support est utilisé comme prétraitement pour la sémantique de Dung. Les cadres d'argumentation abstraits correspondants sont réduits en ajoutant des attaques indirectes issues de l'interaction entre l'attaque et le support avec différentes interprétations, c'est-à-dire le support déductif et le support nécessaire. L'attaque soutenue et l'attaque médiatisée proviennent de l'interaction entre l'attaque et le support déductif, tandis que l'attaque secondaire et l'attaque étendue proviennent de l'interaction entre l'attaque et le support nécessaire.

Nous utilisons les notions d'attaque soutenue, médiatisée, super-médiatisée, secondaire et étendue [5] pour définir les sémantiques 6 et 7.

2 Principes

Nous étudions nos sémantiques par rapport aux principes existants ainsi qu'aux nouveaux principes que nous introduisons dans ce travail. Pour des raisons d'espace, nous ne présentons pas les définitions des principes, mais seulement leur description.

Le principe de Transitivité exprime la transitivité du support. Le principe de Sélection des Extensions stipule que les supports peuvent être utilisés pour sélectionner les extensions. Les principes de Robustesse de Suppression du Support Interne et de Robustesse de Suppression du Support Externe font la distinction entre la sémantique$_4$ et la sémantique$_5$. L'ensemble des principes de Robustesse a été proposé par Rienstra et al. [10]. Le principe de Closure dit que si un argument est dans une extension, les arguments qu'il soutient sont également dans l'extension, tandis que le principe de Inverse Closure dit le contraire, c'est-à-dire que si un argument est dans une extension, les arguments qui le soutiennent devraient également être dans l'extension [2, 5, 9]. Le principe d'Équivalence de l'Extension reflète l'idée que s'il n'y a pas de relation de support, les extensions sous la sémantique σ_i^x sont équivalentes à celles de la sémantique de Dung. Les principes de Monotonie du Statut et de Croissance de l'Extension énoncent tous deux l'effet positif des supports sur les arguments soutenus. Le principe de Monotonie du Statut stipule que l'ajout de supports aux arguments ne modifie pas leur statut dans un ordre inférieur. Gargouri et al. [7] appellent cela la Monotonie, mais nous préférons utiliser un nom plus spécifique (c.-à-d. la Monotonie du Statut) pour le rendre plus précis et éviter toute ambiguïté. Le principe de Croissance de l'Extension montre qu'un argument accepté avec scepticisme reste accepté avec scepticisme lorsque des supports sont ajoutés [8]. Le principe de Directionnalité a été introduit par Baroni, Giacomin et Guida [1]. Il reflète l'idée que nous pouvons décomposer un cadre d'argumentation en sous-cadres de sorte que la sémantique puisse être définie localement.

Nous étudions toutes les sémantiques par rapport à tous les principes.

Références

[1] Baroni, Pietro, Massimiliano Giacomin et Giovanni Guida: *SCC-recursiveness : a general schema for argumentation semantics*. Artificial Intelligence, 168(1-2) :162–210, 2005.

[2] Boella, Guido, Dov M Gabbay, Leon van der Torre et Serena Villata: *Support in abstract argumentation*. Dans *Proceedings of the Third International Conference on Computational Models of Argument (COMMA'10)*, pages 40–51. Frontiers in Artificial Intelligence and Applications, IOS Press, 2010.

[3] Cayrol, Claudette et Marie Christine Lagasquie-Schiex: *On the acceptability of arguments in bipolar argumentation frameworks*. Dans *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.

[4] Cayrol, Claudette et Marie Christine Lagasquie-Schiex: *Bipolar abstract argumentation systems*. Dans *Argumentation in Artificial Intelligence*, pages 65–84. Springer, 2009.

[5] Cayrol, Claudette et Marie Christine Lagasquie-Schiex: *Bipolarity in argumentation graphs : Towards a better understanding*. International Journal of Approximate Reasoning, 54(7) :876–899, 2013.

[6] Dung, Phan M.: *On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games*. Artificial Intelligence, 77(2) :321–357, 1995.

[7] Gargouri, Anis, Sébastien Konieczny, Pierre Marquis et Srdjan Vesic: *On a Notion of Monotonic Support for Bipolar Argumentation Frameworks*. Dans *20th International Conference on Autonomous Agents and MultiAgent Systems*, 2020.

[8] Kaci, Souhila, Leendert van der Torre, Srdjan Vesic et Serena Villata: *Preference in Abstract Argumentation*. Dans Gabbay, Dov, Massimiliano Giacomin, Guillermo R. Simari et Matthias Thimm (rédacteurs) : *Handbook of Formal Argumentation, Volume 2*, pages 211–248. College Publications, 2021.

[9] Polberg, Sylwia: *Intertranslatability of abstract argumentation frameworks*. rapport technique, Technical Report DBAI-TR-2017-104, Institute for Information Systems . . ., 2017.

[10] Rienstra, Tjitze, Chiaki Sakama et Leendert van der Torre: *Persistence and monotony properties of argumentation semantics*. Dans *International Workshop on Theory and Applications of Formal Argumentation*, pages 211–225. Springer, 2015.

Session 2 : Information et croyances

Actes JIAF-JFPDA 2023

Opérateurs totalement informatifs et ordres linéaires partitionnés en révision de croyances

Khaled Belahcene[1] **Jérome Gaigne**[2] **Sylvain Lagrue**[2]

[1]MICS, CentraleSupélec, Université Paris-Saclay, Gif-Sur-Yvette, France
[2]Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 – 60 203 Compiègne Cedex
khaled.belahcene@centralesupelec.fr {jerome.gaigne,sylvain.lagrue}@hds.utc.fr

Résumé

Cet article traite d'opérateurs de révision des croyances qui conduisent à des situations totalement informées. Ce type de situation peut être représenté dans le cadre de révision de Katsuno et Mendelzon par une formule propositionnelle complète à partir de laquelle n'importe quelle formule ou sa négation peut être déduite. Nous caractérisons de tels opérateurs par un ensemble de postulats et fournissons un théorème de représentation qui conduit à totalement ordonner les interprétations. Nous présentons une famille d'opérateurs concrets basée sur des fonctions de départage. Néanmoins, les ordres linéaires sont extrêmement difficiles à obtenir dans la pratique. C'est pourquoi nous proposons dans une deuxième partie une structure ordonnée originale, appelée ordre linéaire partitionné (OLP). Ces structures peuvent être considérées comme des restrictions d'ordres linéaires indépendants, chacun d'entre eux représentant une situation locale totalement informée. Nous étudions les propriétés de la famille des OLP en termes d'expressivité, de représentations graphiques et de modèles interdits. Enfin, nous proposons des postulats rationnels et un théorème de représentation pour les opérateurs de révision aboutissant à un les OLP sur les interprétations, ainsi qu'un opérateur concret satisfaisant ces postulats.

Abstract

This paper focuses on belief revision operators leading to totally informed situations. This kind of situation can be represented in Katsuno Mendelzon revision framework by a complete propositional formula from which either any formula or its negation can be entailed. We characterize such operators by a set of postulates and provide a representation theorem that leads to linear orders on interpretations. We exhibit a concrete operator family satisfying this new set of postulates by combining KM revision operators with tie-breaking functions. Nevertheless, linear orders are extremely difficult to obtain in practice. For that reason, we propose in a second part a new ordered structure named partitioned linear orders (PLO). These structures can be viewed as split independent linear orders, each of them representing a local totally informed situation. We study the properties of PLO in terms of expressiveness, graphical representations, and forbidden patterns. Finally, we propose rational postulates and a representation theorem for revision operators handling PLOs.

1 Introduction

La révision de croyance correspond au problème pouvant survenir lorsqu'un agent reçoit une nouvelle information potentiellement en contradiction avec ses croyances actuelles. Cet article se place dans le cadre de Katsuno Mendelzon (KM), dans lequel les croyances d'un agent, c'est-à-dire sa représentation du monde (de la situation), sont représentées par une formule propositionnelle φ et la nouvelle information par une autre formule α. Il peut arriver que le processus de révision de croyance permette de raffiner les croyances courantes de l'agent et ainsi lui donner une meilleure appréciation de la situation. Par exemple, en se référant au postulat ($\boldsymbol{R_2}$), si la nouvelle information α est cohérente avec les croyances actuelles de l'agent φ, alors le résultat se doit d'être la conjonction $\varphi \wedge \alpha$.

Cet article approfondit cette idée et se focalise sur les opérateurs de révision menant à des *situations totalement informées*. Ce type de situation est représenté par une formule propositionnelle complète depuis laquelle toute formule ou sa négation peut être déduite. Autrement dit, il s'agit d'une formule satisfaite par un seul modèle. Ce type de d'information, sans équivoque, est particulièrement important dans une situation de décision. Par exemple, les procédures de vote mises effectivement en pratique sont *décisives*, en ce sens qu'elles désignent systématiquement un vainqueur

univoque, généralement en complétant une règle idéale respectant le principe d'anonymat (telle que le vote à majorité simple ou le décompte de Borda) à l'aide d'une procédure de départage, par exemple en privilégiant un électeur spécifique ou en se référant à une variable exogène (telle que l'âge du candidat). Dans le domaine de la décision dans l'incertain, dans une situation de complète ignorance, on obtient une procédure de décision quasi décisive en appliquant le principe d'indifférence de Laplace, qui consiste à considérer que les mondes possibles sont équirépartis et à choisir l'option majoritaire, et pour laquelle [34] propose le qualificatif de "pignistique". En décision multicritère, il est fréquent que les procédures paramétriques servant à agréger les points de vue soient élicitées de manière incomplète. On peut alors envisager de raisonner de manière prudente vis-à-vis de l'espace des versions du modèle, mais l'approche de loin la plus fréquente (c.f. par exemple [4] au sujet des modèles de tri multicritère) consiste à sélectionner un agrégateur jugé représentatif parmi l'ensemble des agrégateurs possibles en cherchant à optimiser une fonction objectif adéquate. Ainsi, la procédure d'agrégation n'aboutira jamais à une situation d'incomparabilité.

Un *opérateur de révision totalement informatif* est un opérateur menant à une situation totalement informée. Nous proposons dans cet article un jeu de postulats capturant ce comportement. Nous définissons aussi un opérateur basé sur celui proposé par Dalal qui, équipé avec une fonction de *tie-break*, satisfait tous ces postulats. Nous proposons aussi un théorème de représentation en termes de structures ordonnées et montrons que nos postulats mènent à des ordres linéaires, c'est-à-dire des ordres où la plausibilité relative de deux situations quelconques peut être comparée strictement.

Néanmoins, les ordres linéaires sont difficiles à obtenir en pratique. C'est pourquoi nous proposons, dans un second temps, une nouvelle structure ordonnée nommée *ordres linéaires partitionnés* (OLP). Cette structure peut être vue comme un ensemble d'ordres linéaires indépendants et séparés, chacun d'entre eux représentant localement une situation totalement informée. Nous étudions en détail ces structures et proposons un jeu de postulats les capturant. Nous proposons aussi un opérateur basé sur les diagrammes de Voronoï et les fonctions de tie-break vérifiant nos postulats de rationalité.

La section 2 rappelle les concepts nécessaires sur les structures ordonnées et la révision de croyance. Ensuite, la section 3 se concentre sur notre première contribution, la caractérisation des opérateurs de révisions totalement informatifs. Nous proposons aussi dans cette section un exemple d'opérateur. Comme les opérateurs de révisions totalement informatifs peuvent être considérés comme excessivement spécifiques pour la représentation de croyances, nous proposons et étudions une structure ordonnée originale, les ordres linéaires partitionnés (OLP) dans la section 4. Par la suite, la section 5 se focalise sur la révision de croyances utilisant les OLP. Finalement, avant de conclure, la section 6 propose une discussion sur les travaux associés.

2 Préliminaires

Nous proposons dans cette section les éléments nécessaires à la compréhension du papier. Dans une première partie, nous rappelons différentes notions sur les structures ordonnées. Puis, dans une seconde partie, nous traitons la logique propositionnelle, la révision de croyance ainsi que son lien avec les structures ordonnées.

2.1 Structures ordonnées

Relation binaire Soit $\mathcal{R}$ une relation binaire sur l'ensemble X. Alors $\mathcal{R}$ est :

- *réflexive* lorsque $\forall x \in X, x \mathcal{R} x$;
- *transitive* lorsque $\forall x, y, z \in X,$ si $x \mathcal{R} y$ et $y \mathcal{R} z$, alors $x \mathcal{R} z$;
- *antisymétrique* lorsque $\forall x, y \in X$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$;
- *asymétrique* lorsque $\forall x, y \in X$, si $x \mathcal{R} y$ alors $\neg(y \mathcal{R} x)$;
- *totale* lorsque $\forall x, y \in X,$ soit $x \mathcal{R} y$ soit $y \mathcal{R} x$.

Un *préordre*, noté $\precsim$, est une relation réflexive et transitive. Un *ordre*, noté $\preceq$, est un préordre antisymétrique. Un *ordre linéaire* (alias un ordre total) est un ordre total. Étant donné un préordre $\precsim$, nous notons par $\prec$ sa partie asymétrique, et par $\approx$ sa partie symétrique, et par $\sim$ la relation d'incomparabilité, c'est-à-dire :

- $x \prec y$ lorsque $x \precsim y$ et non $y \precsim x$;
- $x \approx y$ lorsque à la fois $x \precsim y$ et $y \precsim x$; et
- $x \sim y$ lorsque ni $x \precsim y$ ni $y \precsim x$.

Ces notations sont les mêmes pour les ordres $\preceq$.

Posets Un ensemble X associé à un (pré)ordre $\mathcal{R}$ est un ensemble (pré)ordonné noté $P = (X, \mathcal{R})$. Un ensemble partiellement ordonné, également appelé *poset*, est un ensemble ordonné par un ordre partiel.

L'*union de posets disjoints* est définie comme suit. Étant donné deux posets $P_a = (X_a, \precsim_a)$ et $P_b = (X_b, \precsim_b)$ tels que $X_a \cap X_b = \emptyset$, nous avons $P = (X, \precsim) = P_a \cup P_b$ tel que $X = X_a \cup X_b$ et $\forall x, y \in X, x \precsim y$ ssi

$$\begin{cases} x, y \in X_a \text{ et } x \precsim_a y, & \text{ou} \\ x, y \in X_b \text{ et } x \precsim_b y \end{cases}$$

Finalement, étant donnés un ensemble préordonné $P = (X, \precsim)$ et $Y \subseteq X$, nous en définissons les *éléments minimaux* de Y de la manière suivante :

$\text{Min}\,(Y, \precsim) = \{x \in Y : \forall y \in Y, \text{ non } y \prec x\}$

2.2 Rappel sur la révision de croyance

Nous commençons par établir les notations utilisées tout au long de l'article.

2.2.1 Logique propositionnelle

Nous notons par $\mathcal{P}$ un ensemble **fini** de variables propositionnelles et par $\mathcal{L}$ l'ensemble des formules propositionnelles qui peuvent être construites à partir de $\mathcal{P}$. Nous utilisons les connecteurs usuels $\vee$, $\wedge$, $\neg$, $\rightarrow$, $\leftrightarrow$ et les symboles $\top$ et $\bot$ pour représenter respectivement la formule toujours vraie et la formule toujours fausse. Les éléments de $\mathcal{L}$ sont représentés par des lettres grecques. L'ensemble de toutes les interprétations de $\mathcal{L}$ est noté Ω et $[\![\alpha]\!]$ représente l'ensemble des modèles de α, c'est-à-dire l'ensemble $\{\omega \in \Omega : \omega \models \alpha\}$. La formule α_M avec $M \subseteq \Omega$ dénote la formule ayant pour modèles M et seulement M. Une formule φ est dite complète si $\forall \mu \in \mathcal{L}$, nous avons soit $\varphi \vdash \mu$ ou $\varphi \vdash \neg\mu$. Autrement dit, $[\![\varphi]\!]$ est un singleton.

2.2.2 Révision de croyances

La révision de croyances correspond au problème apparaissant lorsqu'une nouvelle information à propos du monde est donnée à un agent, tel que cette nouvelle information et les croyances actuelles de l'agent soient potentiellement incohérentes. Le cadre AGM [1] propose un ensemble de postulats de rationalité qui sont la base du cadre AGR pour la révision de croyance qui est communément accepté. Une reformulation des postulats AGM en logique propositionnelle a eté développée dans [21] et a conduit aux postulats suivants :

($\boldsymbol{R_1}$) $\varphi \circ \alpha \vdash \alpha$
($\boldsymbol{R_2}$) si $\varphi \wedge \alpha \nvdash \bot$, alors $\varphi \circ \alpha \equiv \varphi \wedge \alpha$
($\boldsymbol{R_3}$) si $\alpha \nvdash \bot$, alors $\varphi \circ \alpha$ est cohérent
($\boldsymbol{R_4}$) si $\varphi_1 \equiv \varphi_2$ et $\alpha \equiv \beta$ alors $\varphi_1 \circ \alpha \equiv \varphi_2 \circ \beta$
($\boldsymbol{R_5}$) $(\varphi \circ \alpha) \wedge \beta \vdash \varphi \circ (\alpha \wedge \beta)$
($\boldsymbol{R_6}$) si $(\varphi \circ \alpha) \wedge \beta \nvdash \bot$, alors $\varphi \circ (\alpha \wedge \beta) \vdash (\varphi \circ \alpha) \wedge \beta$

Il a également été montré qu'un opérateur de révision $\circ$ satisfaisant tous ces postulats est équivalent à chercher le minimum dans un préordre total spécifique obtenu par le biais d'un *assignement fidèle* de $\circ$.

Définition 1 (Assignement fidèle, [21])**.** *Une fonction assignant, à une formule φ, un préordre total $\preceq_\varphi$ sur Ω est un* assignement fidèle *ssi*

(1) si $\omega_1, \omega_2 \in [\![\varphi]\!]$ alors $\omega_1 \prec_\varphi \omega_2$ est faux
(2) si $\omega_1 \in [\![\varphi]\!]$ et $\omega_2 \notin [\![\varphi]\!]$ alors $\omega_1 \prec_\varphi \omega_2$ est vrai
(3) si $\varphi \equiv \psi$, alors $\preceq_\varphi = \preceq_\psi$

La première condition stipule que les modèles des croyances actuelles sont équivalents. La seconde requiert que tous les modèles soient préférés à tous les contre-modèles. La troisième condition exprime l'indépendance à la syntaxe.

Théorème 1 ([21])**.** *Un opérateur de révision $\circ$ satisfait les postulats ($\boldsymbol{R_1}$)–($\boldsymbol{R_6}$) ssi il existe un* assignement fidèle *qui fait correspondre φ et le préordre total $\preceq_\varphi$ tel que*

$$[\![\varphi \circ \mu]\!] = \mathrm{Min}\left([\![\mu]\!], \preceq_\varphi\right).$$

2.2.3 Révision d'informations partiellement ordonnées

L'utilisation des postulats KM implique que les informations puissent être représentées par un préordre total. Ce prérequis constitue une limite en termes d'expressivité lorsque la situation comporte des informations incomplètes. C'est pourquoi, deux postulats plus faibles ont été proposés dans [21] pour remplacer ($\boldsymbol{R_6}$), à savoir ($\boldsymbol{R_7}$) et ($\boldsymbol{R_8}$), afin de représenter des informations partiellement ordonnées. Néanmoins, cette proposition est insuffisante pour représenter tous les ordres partiels. Dès lors, il a été proposé dans [6] d'utiliser un affaiblissement de ($\boldsymbol{R_2}$), ($\boldsymbol{R'_2}$), pour pouvoir représenter l'ensemble des informations partiellement ordonnées.

($\boldsymbol{R'_2}$) $\varphi \circ \top \equiv \varphi$
($\boldsymbol{R_7}$) si $\varphi \circ \alpha \vdash \beta$ et $\varphi \circ \beta \vdash \alpha$ alors $\varphi \circ \alpha \equiv \varphi \circ \beta$
($\boldsymbol{R_8}$) si $(\varphi \circ \alpha) \wedge (\varphi \circ \beta) \vdash \varphi \circ (\alpha \vee \beta)$

Ils ont aussi défini la notion de *p-assignement fidèle* pour capturer la notion d'ordre partiel.

Définition 2 (p-assignement fidèle, [6])**.** *Une fonction assignant à une formule φ, un préordre partiel $\preceq_\varphi$ sur Ω est un* p-assignement fidèle *ssi :*

(1) si $\omega_1, \omega_2 \in [\![\varphi]\!]$ alors $\omega_1 \prec_\varphi \omega_2$ est faux
(2') si $\omega_2 \notin [\![\varphi]\!]$ alors il existe ω_1 tel que $\omega_1 \in [\![\varphi]\!]$ et $\omega_1 \prec_\varphi \omega_2$ soit vrai
(3) si $\varphi \equiv \psi$, alors $\preceq_\varphi = \preceq_\psi$

La première différence avec la définition de *assignement fidèle* réside dans le fait que le préordre n'est plus total. La seconde différence est la condition (2'). Relaxant la condition (2), elle requiert uniquement l'existence d'un modèle préféré pour chaque contre-modèle.

Finalement, le théorème de représentation suivant a été proposé.

Théorème 2 ([6])**.** *Un opérateur de révision $p \circ$ satisfait les postulats ($\boldsymbol{R_1}$), ($\boldsymbol{R'_2}$) ,($\boldsymbol{R_3}$)–($\boldsymbol{R_5}$), ($\boldsymbol{R_7}$), ($\boldsymbol{R_8}$) ssi il existe un* p-assignement fidèle *qui fait correspondre φ au préordre partiel $\preceq_\varphi$ tel que :*

$$[\![\varphi \circ \mu]\!] = \mathrm{Min}\left([\![\mu]\!], \preceq_\varphi\right).$$

3 Opérateurs de révision totalement informatifs

Par essence, l'un des objectifs des opérateurs de révision de croyance n'est pas seulement d'intégrer une information potentiellement incohérente avec les croyances initiales

de l'agent, mais aussi d'utiliser cette nouvelle information pour les raffiner. Par exemple, si la nouvelle information α est cohérente avec les croyances initiales de l'agent, en satisfaisant ($\boldsymbol{R_2}$), le résultat de la révision conduit toujours au raffinement des croyances initiales $\varphi \circ \alpha \equiv \varphi \wedge \alpha$. La situation représentée par $\varphi \circ \alpha$ est plus spécifique que la croyance initiale φ de l'agent, dans le sens qu'elle décide plus de formules. De façon équivalente, si nous nous focalisons sur les modèles, si φ et α ne sont pas incohérents entre eux, les mondes possibles pour l'agent, représentés par $[\![\varphi \circ \alpha]\!]$, forme un sous-ensemble des modèles de ses croyances initiales $[\![\varphi]\!]$.

L'objectif de cette section est d'aller encore plus loin dans cette idée en nous concentrant sur les opérateurs menant à une situation totalement informée, c'est-à-dire à une formule complète qui décide toute formule. De façon équivalente, en termes de modèles, il s'agit d'une formule n'admettant qu'un seul modèle.

Exemple 1.
Considérons les connaissances expertes d'un médecin concernant les patients atteints de la grippe (g) et de la toux (t). Les deux situations qui sont les plus plausibles pour eux sont $\neg t \wedge \neg g$ où le patient ne tousse pas et n'a pas la grippe, et $t \wedge g$ où le patient tousse et souffre de la grippe. En d'autres mots, $[\![\varphi]\!] = \{\{\neg t, \neg g\}, \{t, g\}\}$.

Étant donné la nouvelle information disant que le patient tousse, modélisée par $\alpha = t$, le résultat du processus de révision est $\varphi \circ \alpha \equiv t \wedge g$. Dans ce cas, la situation est dite totalement informée, les croyances du médecin n'ayant qu'un seul modèle, $\{t, g\}$. Le médecin peut alors prendre une décision se basant sur ses croyances en étant totalement informé à propos de toutes les variables du langage.

Nous formalisons le concept d'opérateur de révision totalement informatif dans la sous-section suivante, y compris si la nouvelle information est contradictoire avec les croyances actuelles de l'agent, et nous les capturons au travers de postulats de rationalité.

3.1 Formalisation

Un opérateur de révision $\circ$ est *totalement informatif* s'il satisfait la propriété suivante :

($\boldsymbol{TI}$) $\forall \alpha \in \mathcal{L}$, si $\alpha \nvdash \bot$, alors $\varphi \circ \alpha$ est complète.

Autrement dit, si la nouvelle information n'est pas incohérente, un opérateur TI mène à une situation totalement informée, c'est-à-dire à une formule ne possédant qu'un seul et unique modèle.

Un opérateur satisfaisant ($\boldsymbol{R_2}$) ne peut pas être un opérateur de révision totalement informatif.

Proposition 1. *Les postulats ($\boldsymbol{R_2}$) et ($\boldsymbol{TI}$) sont mutuellement inconsistants.*

Cette proposition nous amène à considérer l'affaiblissement suivant de ($\boldsymbol{R_2}$) :

($\boldsymbol{R_{2w}}$) si $\varphi \wedge \alpha \nvdash \bot$ alors $\varphi \circ \alpha \vdash \varphi \wedge \alpha$

Cet affaiblissement de ($\boldsymbol{R_2}$) peut se retrouver dans la littérature, par exemple dans [6], [31], ou dans [19] où ($\boldsymbol{R_{2w}}$) apparaît sous le nom de ($\boldsymbol{R_7}$). Il y est présenté comme le postulat de *vacuité* dans la formulation AGM, cela permet à l'agent d'ignorer une partie de l'information sur ses croyances même si celle-ci est cohérente avec la nouvelle information. Nous préférons l'interpréter ici comme la possibilité pour l'agent de sélectionner arbitrairement un sous-ensemble de ces mondes, ce qui peut être vu comme un gain d'information.

Nous proposons maintenant un postulat original, nommé ($\boldsymbol{R_L}$), pour caractériser les opérateurs de révision totalement informatifs. Celui-ci stipule que si α et β sont cohérents, mais ensemble incohérents, alors la disjonction de $\varphi \circ \alpha$ et $\varphi \circ \beta$ ne permet pas de déduire la révision par la disjonction de α et β.

($\boldsymbol{R_L}$) si $\alpha \nvdash \bot, \beta \nvdash \bot$, et $\alpha \wedge \beta \vdash \bot$ alors $(\varphi \circ \alpha) \vee (\varphi \circ \beta) \nvdash \varphi \circ (\alpha \vee \beta)$

Proposition 2. *Les postulats ($\boldsymbol{R_2}$) et ($\boldsymbol{R_L}$) sont mutuellement inconsistants.*

La proposition suivante stipule que, en présence de ($\boldsymbol{R_1}$), ($\boldsymbol{R_{2w}}$),($\boldsymbol{R_3}$)–($\boldsymbol{R_6}$), les postulats ($\boldsymbol{R_L}$) et ($\boldsymbol{TI}$) sont équivalents.

Proposition 3. *Considérons un opérateur de révision $\circ$ satisfaisant ($\boldsymbol{R_1}$), ($\boldsymbol{R_{2w}}$),($\boldsymbol{R_3}$)–($\boldsymbol{R_6}$). Alors :*

$\circ$ *satisfait* ($\boldsymbol{R_L}$) *ssi* $\circ$ *satisfait* ($\boldsymbol{TI}$).

Finalement, nous proposons un théorème de représentation pour les opérateurs de révision TI. Ce théorème se réfère à la notion de *TI-assignement fidèle* qui fait correspondre une formule et un ordre linéaire où tous les contre-modèles sont dominés par les modèles (1). Cet ordre est aussi indépendant à la syntaxe (condition (2)). La différence principale avec les assignements *fidèles* et *p-fidèles*, si nous ignorons le fait que l'ordre doit être linéaire, est que les modèles peuvent être *strictement* comparés.

Définition 3 (TI-assignement fidèle)**.** *Une fonction assignant, à une formule φ, un* ordre linéaire $\preceq_\varphi$ *sur Ω est un* assignement fidèle totalement informé *ssi*

(1) si $\omega_1 \in [\![\varphi]\!]$ et $\omega_2 \notin [\![\varphi]\!]$ alors $\omega_1 \prec_\varphi \omega_2$ est vrai

(2) si $\varphi \equiv \psi$, alors $\preceq_\varphi = \preceq_\psi$

Nous pouvons maintenant introduire le théorème de représentation suivant. Globalement, ce théorème dit que, étant donné un opérateur de révision TI, il est possible d'associer aux croyances d'un agent un ordre linéaire correspondant sur l'ensemble des interprétations.

Théorème 3. *Un opérateur de révision TI ∘ satisfait* ($\boldsymbol{R_1}$)*,* ($\boldsymbol{R_{2w}}$)*,* ($\boldsymbol{R_3}$)*–*($\boldsymbol{R_6}$) *et* ($\boldsymbol{R_L}$) *ssi il existe un* TI-assignement fidèle *qui fait correspondre* φ *et l'ordre linéaire* $\preceq_\varphi$ *tel que :*

$$[\![\varphi \circ \alpha]\!] = \text{Min}\left([\![\alpha]\!], \preceq_\varphi\right)$$

3.2 Un opérateur totalement informatif concret

Nous présentons dans cette section un opérateur concret démontrant la cohérence de notre ensemble de postulats. En pratique, un opérateur de révision totalement informatif peut être obtenu en couplant un opérateur de révision de Dalal [13] avec une fonction de tie-break. Un opérateur de révision de Dalal est un opérateur de révision basé sur la distance de Hamming entre les interprétations, où $d_H(\omega, \omega') = |\{x \in \mathcal{P} : \omega(x) \neq \omega'(x)\}|$. Cette distance peut directement être étendue à la distance entre une interprétation ω et une formule φ en considérant la distance minimale entre ω et les modèles de φ, c'est-à-dire $d_H(\varphi, \omega) = \min\{d_H(\omega', \omega) : \omega' \in [\![\varphi]\!]\}$.

Exemple 2. *Soit* $\mathcal{P} = \{a, b, c, d\}$*. La distance de Hamming entre* $\omega = \{a, b\}$ *et* φ *telle que* $[\![\varphi]\!] = \{\{\neg a, \neg b\}, \{a, \neg b\}\}$*, est alors* $d_H(\varphi, \omega) = \min\{d_H(\{\neg a, \neg b\}, \{a, b\}), d_H(\{a, \neg b\}, \{a, b\})\} = 1$.

Étant donné une formule φ, il est possible d'ordonner les interprétations en fonction de leur distance de Hamming à la formule φ :

$$\omega \preceq_\varphi^{d_H} \omega' \text{ ssi } d_H(\varphi, \omega) \leq d(\varphi, \omega').$$

Le résultat de la révision de Dalal est équivalent à l'ensemble des éléments minimaux pour ce préordre :

$$[\![\varphi \circ_{d_H} \alpha]\!] = Min([\![\alpha]\!], \preceq_\varphi^{d_H}).$$

Nous proposons d'associer à $\circ_{d_H}$ une fonction de départage. Ce type de fonction est d'habitude utilisé dans la prise de décision collective et dans le choix social (computationnel) ; par exemple [12, 16]. Étant donné un ensemble d'alternatives, une fonction de départage est capable de sélectionner exactement un élément préféré. Pour ce papier, nous ne considérons uniquement que les fonctions de départage déterministes $\mathcal{T}$ sur Ω satisfaisant la propriété d'indépendance suivante (noté α dans [32]).

$\forall E, E' \in 2^\Omega$ t.q. $\mathcal{T}(E) \in E' \subseteq E$, nous avons $\mathcal{T}(E') = \mathcal{T}(E)$.

En considérant d_H et une fonction de départage $\mathcal{T}$, nous pouvons dériver, d'une formule φ, un ordre linéaire $\preceq_\varphi^{d_H,\mathcal{T}}$ tel que : $\omega \preceq_\varphi^{d_H,\mathcal{T}} \omega'$ ssi $\omega \prec_\varphi^{d_H} \omega'$ ou ($\omega \approx_\varphi^{d_H} \omega'$ et $\mathcal{T}(\{\omega, \omega'\}) = \omega$).

Définition 4 (Opérateur de révision totalement informatif de Dalal)**.** *Étant donné une fonction de départage* $\mathcal{T}$*, une formule* φ *et une nouvelle information* α*,* $\varphi \circ_{d_H,\mathcal{T}} \alpha$ *est tel que :*

$$[\![\varphi \circ_{d_H,\mathcal{T}} \alpha]\!] = Min\left([\![\alpha]\!], \preceq_\varphi^{d_H,\mathcal{T}}\right).$$

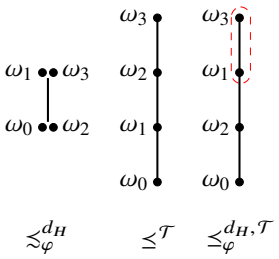


FIGURE 1 – Diagramme de Hasse [1] représentant φ de l'Exemple 3 avec $\preceq_\varphi^{d_H}$ étant le préordre obtenu avec les distances de Hamming, $\preceq^{\mathcal{T}}$ l'ordre linéaire dérivé de $\mathcal{T}$, et $\preceq_\varphi^{d_H,\mathcal{T}}$ l'ordre obtenu en appliquant la fonction de tie-break.

À partir d'une fonction de départage $\mathcal{T}$ satisfaisant la propriété d'indépendance, il est possible de construire un ordre linéaire $\preceq^{\mathcal{T}}$ sur l'ensemble des interprétations : l'élément préféré est $\mathcal{T}(\Omega)$, le second est $\mathcal{T}(\Omega \setminus \{\mathcal{T}(\Omega)\})$, etc. L'ordre linéaire $\preceq_\varphi^{d_H,\mathcal{T}}$ défini ci-dessus est équivalent au *raffinement lexicographique* du préordre $\preceq_\varphi^{d_H}$ par l'ordre linéaire $\preceq^{\mathcal{T}}$.

Un tel raffinement lexicographique peut être retrouvé dans la littérature, par exemple dans [5] où les états épistémiques sont révisés par d'autres états épistémiques. Il peut aussi être trouvé dans [2] qui se focalise sur les propriétés rationnelles de la combinaison de relations de préférence qui conduisent à une combinaison lexicographique. Cette idée est la pierre angulaire des modèles de décision non compensatoire (voir par exemple [11] pour trier des alternatives dans de multiples catégories ou [30] pour les classer) obtenue par raffinement successif via des préordres partiels bivalués [10]. Cette idée de raffiner les modèles de la nouvelle information, même s'ils ne sont pas aussi drastiques que nous le sommes, peut être trouvée chez [19].

Exemple 3. *Soit* $\mathcal{P} = \{a, b\}$*, et* $\Omega = \{\omega_0, \omega_1, \omega_2, \omega_3\}$ *tel que* $\omega_0 = \{\neg a, \neg b\}$*,* $\omega_1 = \{\neg a, b\}$*,* $\omega_2 = \{a, \neg b\}$*,* $\omega_3 = \{a, b\}$*. Considérons* $\varphi = \neg b$*, donc* $[\![\varphi]\!] = \{\omega_0, \omega_2\}$*. Comme fonction de départage* $\mathcal{T}$*, nous prenons l'ordre lexicographique sur les interprétations induit par l'ordre suivant sur les variables propositionnelles falsifiées* $a \prec b$*. La Figure 1 représente, dans un diagramme de Hasse , φ premièrement, comme le préordre de Dalal* $\preceq_\varphi^{d_H}$ *correspondant uniquement au calcul des distances de Hamming, puis comme un ordre* $\preceq_\varphi^{d_H,\mathcal{T}}$ *obtenu en appliquant la fonction de départage sur ce préordre. Considérons maintenant que* φ *est révisé par* $\alpha = b$*. Alors* $[\![\alpha]\!] = \{\omega_1, \omega_3\}$*, et nous avons* $[\![\varphi \circ \alpha]\!] = \{\omega_1\}$.

Comme attendu, l'opérateur de révision $\circ_{d_H,\mathcal{T}}$ est un opérateur de révision totalement informatif.

1. Les diagrammes de Hasse sont une représentation graphique des préordres où ni la réflexivité, ni la transitivité ne sont représentées et où les éléments minimaux sont au bas de la figure.

Proposition 4. *L'opérateur de révision $\circ_{d_H,\mathcal{T}}$ satisfait ($\boldsymbol{R_1}$), ($\boldsymbol{R_{2w}}$), ($\boldsymbol{R_3}$)–($\boldsymbol{R_6}$), ($\boldsymbol{R_L}$).*

4 Ordres linéaires partitionnés

En révision de croyance, des structures ordonnées peuvent être employées pour modéliser les croyances et la plausibilité qu'un agent associe à chaque monde. Pour ces structures ordonnées, la propriété de *totalité* est à la fois très exigeante quand il s'agit d'alimenter le modèle en information et d'une portée sans limites, ce qui peut parfois paraître abusif quand nous nous intéressons à la sémantique des modèles dans le monde réel. C'est pourquoi nous introduisons une relaxation de cette propriété, nommée *comparabilité transitive*, ainsi que la catégorie d'ordres la satisfaisant, appelée ici *ordres linéaires partitionnés* (OLP).

4.1 Définition

Nous introduisons la catégorie des *ordres linéaires partitionnés* via un détour avec la notion de *comparabilité* : étant donné une relation binaire $\mathcal{R}$ sur un ensemble X, on dit que deux éléments $x, y \in X$ sont comparables quand $x\ \mathcal{R}\ y$ ou $y\ \mathcal{R}\ x$. La relation de comparabilité est structurellement symétrique. La propriété de *totalité* d'une relation $\mathcal{R}$ peut alors se réécrire comme la complétude de la relation de comparabilité, c'est-à-dire la situation où toute paire d'éléments est comparable. Nous relâchons cette propriété globale pour n'imposer que la comparabilité soit seulement *transitive*, dans le but qu'elle se propage par contagion. En effet, quand n'importe quelle paire d'éléments est en relation avec un troisième, ils sont considérés comme faisant partie de la relation. Appliquée à un poset $(X, \preccurlyeq)$, cette notion conduit à une partition de X en classe d'équivalence sur la relation de comparabilité associée à $\preccurlyeq$, où deux éléments de classes différentes sont incomparables. La relation binaire induite par $\preccurlyeq$ sur chaque classe est alors un préordre total. Appliquée à un ensemble fini X ordonné par $\preceq$, la partition est finie et la relation induite par chaque classe est un ordre linéaire.

Définition 5 (Ordre linéaire partitionné)**.** *La relation binaire sur X, notée $\trianglelefteq$, est un ordre linéaire partitionné ssi $\exists m \in \mathbf{N}^*$ tel que $X_1, ..., X_m$ est une partition de X (c'est-à-dire $\bigcup_{i=1..m} X_i = X$ et $\forall i, j \in \{1, ..., m\}$ tel que $i \neq j$, $X_i \cap X_j = \emptyset$) et $\exists \prec_i$, un ordre linéaire sur X_i, $\forall i \in \{1, ..., m\}$ tel que $(X, \trianglelefteq) = \bigcup_{i=1..m} (X_i, \prec_i)$.*

Exemple 4. *Soit l'ensemble $X = \{a, b, c, d, e, f, g, h\}$ et l'ordre linéaire partitionné $\trianglelefteq = \{(c,c), (c,b), (c,a), (b,b), (b,a), (a,a), (f,f), (f,e), (f,d), (e,e), (e,d), (d,d), (h,h), (h,g), (g,g)\}$. La partition de X suivante, $X_1 = \{a, b, c\}$, $X_2 = \{d, e, f\}$ et $X_3 = \{g, h\}$ associée respectivement aux ordres linéaires $\preceq_1 = \{(c,c), (c,b), (c,a), (b,b), (b,a), (a,a)\}$, $\preceq_2 = \{(f,f), (f,e), (f,d), (e,e), (e,d), (d,d)\}$ et $\preceq_3 = \{(h,h), (h,g), (g,g)\}$ conduit aux trois ensembles linéairement ordonnés $(X_1, \preceq_1)$, $(X_2, \preceq_2)$, et $(X_3, \preceq_3)$ tels que $(X, \trianglelefteq) = \bigcup_{i\in\{1,2,3\}} (X_i, \preceq_i)$. La Figure 2 représente ces ordres.*

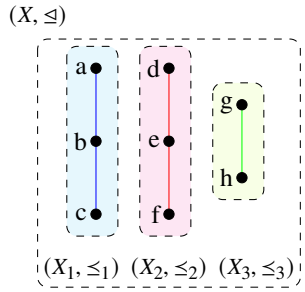


FIGURE 2 – OLP $\trianglelefteq$ sur X.

4.2 Représentations

Il est d'usage de représenter les ordres partiels comme une intersection d'ordres linéaires, ou encore à l'aide de fonctions numériques. Par exemple, n'importe quel préordre total $\precsim$ sur un ensemble fini X peut être représenté par une fonction de *score* $f : X \to \mathbb{N}$ telle que $x \precsim y \iff f(x) \geq f(y)$. La relation $\precsim$ est un ordre linéaire ssi f est injective. Nous proposons ces deux types de représentation pour les ordres linéaires partitionnés.

Soit $\trianglelefteq$ un OLP sur un ensemble fini X menant à une partition $X = \bigcup_{i=1}^{n} X_i$ en classes de comparabilités. Soit $\mathcal{R}^>$ et $\mathcal{R}^<$ les relations binaires définies sur X comme suivent :

$$\forall x, y \in X,\ x\ \mathcal{R}^>\ y \text{ ssi } x \trianglelefteq y \text{ ou } x \in X_i, y \in X_j \text{ et } i > j \quad (1)$$

$$\forall x, y \in X,\ x\ \mathcal{R}^<\ y \text{ ssi } x \trianglelefteq y \text{ ou } x \in X_i, y \in X_j \text{ et } i < j \quad (2)$$

Proposition 5. *$R^>$ et $R^<$ sont des ordres linéaires sur X qui raffine $\trianglelefteq$ et tels que $\trianglelefteq$ est le produit d'ordre de $\mathcal{R}^>$ et $\mathcal{R}^<$ (c'est-à-dire $x \trianglelefteq y$ ssi $x\ \mathcal{R}^>\ y$ et $x\ \mathcal{R}^<\ y$).*

Nous pouvons remarquer que les deux relations $\mathcal{R}^>$ et $\mathcal{R}^<$ dépendent de l'ordre arbitraire sur les classes de comparabilité de $(X, \trianglelefteq)$, mais que leur produit d'ordre non. Comme nous le verrons dans la Section 4.3, la Proposition 5 positionne les ordres linéaires partitionnés comme une sous-catégorie des ordres bilinéaires. De plus, il est possible de produire trois représentations numériques des ordres linéaires partitionnés.

Représentation en produit Un OLP sur un ensemble fini X peut être représenté avec deux fonctions de score $f, g : X \to \mathbb{N}$ telles que $x \trianglelefteq y$ ssi $f(x) < f(y)$ et $g(x) < g(y)$ comme illustré par la Figure 3.

$x \in X$	$f(x)$	$g(x)$	$f'(x)$	$g'(x)$
a	1	6	1	14
b	2	7	2	15
c	3	8	3	16
d	4	3	4	11
e	5	4	5	12
f	6	5	6	13
g	7	1	7	9
h	8	2	8	10

TABLE 1 – $f, g, f', g' : X \longrightarrow \mathbb{R}$, deux paires de fonctions de score, (f, g) offrant une représentation en produit et (f', g') une représentation en intervalles de $\trianglelefteq$.

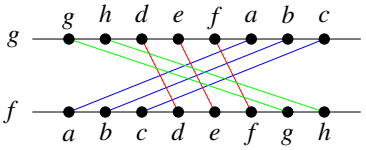


FIGURE 3 – Représentation graphique des fonctions f et g.

Représentation en intervalle Il peut être montré que la condition $\forall x \in X,\ f(x) > g(x)$ peut être imposée sans perte de généralité à la représentation en produit présentée ci-dessus. Alors, chaque élément de l'ensemble X peut être représenté par un intervalle non vide $[f(x), g(x)]$ [3, 28]. L'incomparabilité d'une paire d'éléments est représentée par l'inclusion/le fait d'être contenu de leurs intervalles respectifs, comme le montre la Figure 4. Cela implique que la relation de comparabilité stricte, c'est-à-dire $x \prec y$, est encodée par $f(x) < f(y)$ et $g(x) < g(y)$ comme présenté ci-dessus, ce qui diverge de la définition des ordres d'intervalles, où celle-ci est encodée par $g(x) < f(y)$.

Ordre linéaire divisé Un OLP sur un ensemble fini avec n classes de comparabilité peut être représenté via une fonction de score injective $f : X \longrightarrow \mathbb{N}$ et un ensemble de $n-1$ seuils $\tau_1, \ldots, \tau_{n-1}$ de la façon suivante : $x \trianglelefteq y$ ssi $f(x) \leq f(y)$ et $\nexists i \in \{1, \ldots, n-1\}\ f(x) < \tau_i < f(y)$.

Exemple 5. *La Table 1 contient une représentation numérique de $(X, \trianglelefteq)$ en tant que produit d'ordres de deux fonctions de score f et g. La représentation graphique de f et g est disponible Figure 3. Les éléments de X sont représentés par des segments tels que les segments représentant deux objets se croissent ssi ces objets sont incomparables. La représentation graphique de f' et g' est disponible Figure 4. Les éléments de X sont représentés par des intervalles tels qu'un intervalle en contient un autre ssi les deux éléments correspondants sont incomparables.*

4.3 Modèles interdits

Nous situons maintenant la catégorie des ordres linéaires partitionnés au sein de la cartographie de sous-catégories

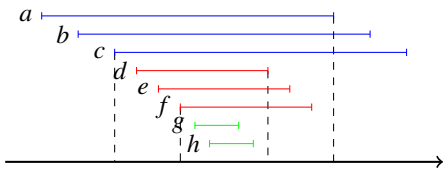


FIGURE 4 – Représentation graphique de $[f', g']$.

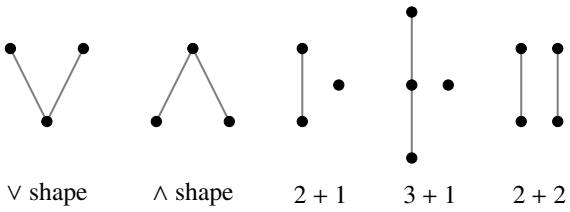


FIGURE 5 – Modèles interdits.

d'ordres partiels proposée dans la littérature. La Proposition 5 montre que les OLP sont une sous-catégorie des *ordres bilinéaires*, qui sont définis comme le produit d'ordres de deux ordres linéaires

Une autre façon de définir une sous-catégorie d'ordres partiels consiste à exclure certains sous-motifs du diagramme de Hasse, c'est-à-dire ce que cette catégorie est incapable de représenter.

Nous rappelons la définition d'*ordres d'intervalles*, de *semi-ordres* et d'*ordres forts* en termes de modèles interdits comme définis dans [15], ces modèles sont représentés Figure 5 :

— Ordres d'intervalles : pas de 2+2

— Semi-ordre : pas de 1+3 ni de 2+2

— Ordres forts : pas de 1+2

La catégorie des ordres linéaires partitionnés peut être définie au moyen de deux modèles interdits. Soit $(X, \precsim)$ un poset et $x, y, z \in X$ un triplet d'éléments de X.

— x, y, z forment un modèle en $\vee$ quand $z \precsim x$, $z \precsim y$ et ni $x \precsim y$, ni $y \precsim x$;

— x, y, z forment un modèle en $\wedge$ quand $x \precsim z$, $y \precsim z$ et ni $x \precsim y$, ni $y \precsim x$.

Tous ces modèles sont représentés sur la Figure 5.

Proposition 6. *Un ordre partiel $\preceq$ sur un ensemble fini X est un ordre linéaire partitionné ssi il n'y a aucun triplet $x, y, z \in X$ formant un modèle en $\vee$ ou un modèle en $\wedge$.*

Pour conclure cette section, la Figure 6 insère les ordres linéaires partitionnés au sein du diagramme d'inclusions propres des parties asymétriques de certains posets proposé

2. Les ordres bilinéaires correspondent au cas $n = 2$ des ordres n-linéaires, définis comme la catégorie obtenue par produit d'ordres de n ordres linéaires. Cette catégorie se trouve presque au plus bas de la hiérarchie, juste au-dessus de celle des ordres linéaires.

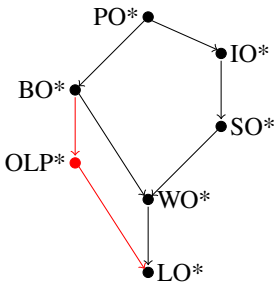


FIGURE 6 – Diagramme d'inclusions propres des parties asymétriques de structures ordonnées proposées dans [15]. PO = ordre partiel ; IO = ordre d'intervalles ; SO = semi-ordre ; BO = ordre bilinéaire ; OLP = ordre linéaire partitionné ; WO = ordre fort ; et LO = ordre linéaire. R* étant la partie asymétrique de la relation R.

dans [15]. Sur ce diagramme, les classes non connectées ne sont pas incluses l'une dans l'autre et parmi deux classes connectées, la plus basse est strictement incluse dans la plus haute.

5 Révision et OLP

Comme les OLP semblent être des structures plus réalistes que les ordres linéaires pour représenter la révision de croyances, nous proposons un jeu de postulats capturant la notion de *assignement fidèle* à un OLP.

Exemple 6.

Considérons à nouveau l'exemple du médecin, la grippe (g) et la toux (t). En général, il peut ordonner la plausibilité de chaque situation de la façon suivante : $\{\neg t \wedge \neg g\} \preceq \{t \wedge g\} \preceq \{t \wedge \neg g\} \preceq \{\neg t \wedge g\}$. *En effet, il semble que le plus réaliste soit que les personnes sont en bonne santé, mais, si un patient tousse, alors il doit avoir la grippe. La situation la moins plausible est lorsque le patient a la grippe, mais ne tousse pas.*

Si maintenant, nous considérons une nouvelle variable h, qui représente "c'est l'hiver", il peut être beaucoup plus simple pour le médecin de considérer deux ordres linéaires locaux dépendant du contexte h et $\neg h$ *:* $\{\neg h, \neg t, \neg g\} \preceq_1 \{\neg h, t, \neg g\} \preceq_1 \{\neg h, t, g\} \preceq_1 \{\neg h, \neg t, g\}$ *et* $\{h, \neg t, \neg g\} \preceq_2 \{h, t, g\} \preceq_2 \{h, t, \neg g\} \preceq_2 \{h, \neg t, g\}$.

Le premier ordre linéaire exprime le fait que, lorsque ce n'est pas l'hiver, il est plus plausible pour un patient de tousser sans avoir la grippe (par exemple à cause des allergies), au contraire, quand c'est l'hiver, il est plus plausible pour qu'un patient tousse à cause de la grippe. L'union de ces deux ordres conduit à un OLP.

Les structures OLP peuvent aussi se rencontrer dans la fusion de croyance [24, 23], en particulier en cas d'incommensurabilité des plausibilités fournies par les agents [7]. Ce problème est également intimement lié à celui des comparaisons interpersonnelles d'utilité [33, 8].

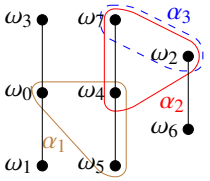


FIGURE 7 – Contre-exemple d'OLP pour $(\boldsymbol{R_2})$ et $(\boldsymbol{R_6})$.

5.1 Un nouveau postulat de rationalité

Tout d'abord, nous pouvons remarquer que les postulats KM ne permettent pas de capturer la structure des OLP et ne pouvaient pas non plus être utilisés pour les opérateurs de révision totalement informatifs. Par exemple, considérons les croyances φ d'un agent tel que $[\![\varphi]\!] = \{\omega_1, \omega_5, \omega_6\}$ et l'opérateur de révision $\circ_{OLP}$ menant à l'OLP représenté sur la Figure 7. Considérons maintenant une nouvelle information α_1 telle que $[\![\alpha_1]\!] = \{\omega_0, \omega_4, \omega_5\}$. Si $\circ_{OLP}$ satisfait $(\boldsymbol{R_2})$, comme $[\![\varphi \wedge \alpha]\!] = \{\omega_5\}$, le résultat, de la révision devrait être $[\![\varphi \circ_{OLP} \alpha_1]\!] = \{\omega_5\}$, alors qu'il n'y a aucune raison de rejeter ω_0 qui n'est plus dominée. De la même manière, un opérateur de révision OLP ne devrait pas satisfaire $(\boldsymbol{R_6})$, en considérant par exemple α_2 tel que $[\![\alpha_2]\!] = \{\omega_2, \omega_4, \omega_7\}$ et α_3 tel que $[\![\alpha_3]\!] = \{\omega_2, \omega_7\}$. Alors, $[\![\varphi \circ_{OLP} (\alpha_2 \wedge \alpha_3)]\!] = \{\omega_2, \omega_7\}$, alors que $[\![\varphi \circ_{OLP} \alpha_2]\!] = \{\omega_2, \omega_4\}$ et $[\![(\varphi \circ_{OLP} \alpha_2) \wedge \alpha_3]\!] = \{\omega_2\}$.

Cette incompatibilité avec $(\boldsymbol{R_2})$ et $(\boldsymbol{R_6})$ n'est pas une surprise puisque les OLP sont des ordres partiels, ce qui permet des situations d'incomparabilité que les opérateurs KM ne capturent pas. De notre côté, nous nous plaçons dans le cadre de [6] auquel nous proposons d'ajouter le postulat $(\boldsymbol{R_{OLP}})$.

$(\boldsymbol{R_{OLP}})$ Si $\alpha \wedge \beta \vdash \bot$ et $(\varphi \circ \alpha) \vee (\varphi \circ \beta) \vdash \varphi \circ (\alpha \vee \beta)$, alors pour tout formule *complète* γ, on a $(\varphi \circ \alpha) \vee (\varphi \circ \gamma) \vdash \varphi \circ (\alpha \vee \gamma)$ ou $(\varphi \circ \beta) \vee (\varphi \circ \gamma) \vdash \varphi \circ (\beta \vee \gamma)$

La pierre angulaire du postulat $(\boldsymbol{R_{OLP}})$ est la déduction $(\varphi \circ \alpha) \vee (\varphi \circ \beta) \vdash \varphi \circ (\alpha \vee \beta)$. Une fois ramenée aux structures ordonnées, cette déduction conduit à l'incomparabilité entre les deux éléments α et β quand α et β sont complets. C'est pourquoi, $(\boldsymbol{R_{OLP}})$ exprime le fait que pour tout triplet d'éléments où une paire est incomparable, il y a forcément au moins une autre paire incomparable. Imposer $(\boldsymbol{R_{OLP}})$ à une structure transitive est équivalent à exclure les modèles en $\vee$ et en $\wedge$.

Comme attendu, le théorème de représentation suivant montre qu'en ajoutant $(\boldsymbol{R_{OLP}})$ aux postulats de la révision avec des ordres partiels $((\boldsymbol{R_1}), (\boldsymbol{R_2'}), (\boldsymbol{R_3}), (\boldsymbol{R_4}), (\boldsymbol{R_5}), (\boldsymbol{R_7})$, and $(\boldsymbol{R_8}))$, il est possible de construire des opérateurs de révisions OLP.

Théorème 4. *Un opérateur de révision OLP $\circ$ satisfait les postulats $(\boldsymbol{R_1})$, $(\boldsymbol{R'_2})$, $(\boldsymbol{R_3})$–$(\boldsymbol{R_5})$, $(\boldsymbol{R_7})$, $(\boldsymbol{R_8})$, $(\boldsymbol{R_{OLP}})$ ssi il existe un p-assignement fidèle qui fait correspondre φ et le* ***OLP*** *$\preceq_\varphi$ tel que :*

$$[\![\varphi \circ \mu]\!] = \text{Min}\left([\![\mu]\!], \precsim_\varphi\right).$$

La prochaine section présente un exemple d'opérateur de révision OLP basé sur la distance de Hamming et sur les diagrammes de Voronoï non ambigus.

5.2 Un opérateur de révision OLP concret

Nous proposons un opérateur de révision OLP concret. Étant donné une formule φ, notre opérateur procède en 2 étapes pour construire un OLP. Pour cela, nous utilisons deux fonctions de départage différentes $\mathcal{T}_1$ et $\mathcal{T}_2$. En premier, nous construisons une partition de Ω autour des modèles de φ en utilisant la distance de Hamming, où chaque interprétation est associée au modèle le plus proche. En cas d'égalité, $\mathcal{T}_1$ est utilisé. Cette partition en cellule peut être vue comme diagramme de Voronoï non ambigu. La deuxième étape consiste à ordonner totalement les interprétations au sein de chaque cellule en utilisant pour cela la distance au centre de chaque cellule, les égalités sont résolues par $\mathcal{T}_2$.

Définition 6 (Diagramme de Voronoï non ambigu)**.** *Étant donné φ tel que $[\![\varphi]\!] = \{\omega_1, ..., \omega_m\}$ et une fonction de tie-break $\mathcal{T}$, alors $\Delta_\varphi^{\mathcal{T}} = \{V_{\omega_1}, ..., V_{\omega_m}\}$ est une partition Ω telle que :*

$$\omega \in V_{\omega^\star} \text{ ssi } \mathcal{T}(\text{Argmin}_{\omega' \in [\![\varphi]\!]} d_H(\omega', \omega)) = \omega^\star.$$

Il peut être remarqué que cette partition peut être construite de façon équivalente en utilisant les opérateurs de dilatation [9].

Définition 7 (Opérateur de révision de Voronoï non ambigu)**.** *Étant donné une formule φ, une nouvelle information α et deux fonctions de tie-break $\mathcal{T}_1$ et $\mathcal{T}_2$, nous définissons la relation binaire $\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$ sur Ω par $\omega \trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2} \omega'$ lorsque $\{\omega, \omega'\} \subseteq V_{\omega^\star}$ et soit $d_H(\omega^\star, \omega) < d_H(\omega^\star, \omega')$ soit $d_H(\omega^\star, \omega) = d_H(\omega^\star, \omega')$ et $\mathcal{T}_2(\{\omega, \omega'\}) = \omega$, où $V_{\omega^\star} \in \Delta_\varphi^{\mathcal{T}_1}$ est une cellule du diagramme de Voronoï non ambigu de φ.*

L'opérateur de révision $\circ_V$ associé est défini par :

$$[\![\varphi \circ_V \alpha]\!] = Min([\![\alpha]\!], \trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2})$$

$\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$ est un ordre linéaire partitionné où les classes de comparabilités sont les cellules du diagramme de Voronoï non ambigu $\Delta_\varphi^{\mathcal{T}_1} = \{V_{\omega_1}, \ldots, V_{\omega_m}\}$.

L'exemple suivant permet d'illustrer ces définitions. Nous prenons $\mathcal{T}_1 = \mathcal{T}_2$, un simple ordre lexicographique sur les variables.

Exemple 7. *Soit les croyances actuelles de notre agent φ telles que $[\![\varphi]\!] = \{\omega_1, \omega_5, \omega_6\}$.*

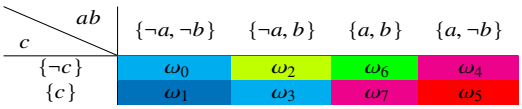
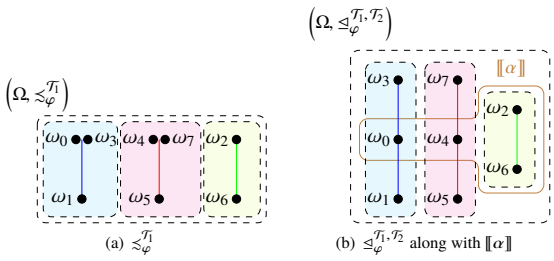


FIGURE 8 – Table de Karnaugh représentant le diagramme de Voronoï non ambigu de φ de l'Exemple 7.



(a) $\precsim_\varphi^{\mathcal{T}_1}$ (b) $\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$ along with $[\![\alpha]\!]$

FIGURE 9 – $\precsim_\varphi^{\mathcal{T}_1}$, le préordre partiel induit par la distance de Hamming aux modèles de φ de l'Exemple 7 au sein des cellules de Voronoï non ambiguë de la Figure 8 et $\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$, l'OLP obtenu après application du second tie-break avec $[\![\alpha]\!]$.

Comme fonctions de départage, nous prenons $\mathcal{T}_1$ et $\mathcal{T}_2$, l'ordre lexicographique sur les interprétations obtenu par l'ordre suivant sur les variables propositionnelles falsifiées : $a \prec b \prec c$.

La Figure 8 montre les cellules du diagramme de Voronoï non ambigu de φ sur une table de Karnaugh où V_{ω_1} est en bleu et V_{ω_5} est en rouge et V_{ω_6} est en vert. Sur cette figure, nous pouvons voir que bien que ω_4 et ω_7 sont à une distance de Hamming de 1 avec à la fois ω_5 et ω_6, ils sont associés à ω_5 via $\mathcal{T}_1$. Cette figure peut être représentée par le préordre partiel $\precsim_\varphi^{\mathcal{T}_1}$ disponible Figure 9(a).

Appliquée une nouvelle fois, la fonction de tie-break résout les indifférences conduisant à l'OLP $\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$ représenté sur la Figure 9(b).

Maintenant, supposons que notre agent soit confronté à la nouvelle information $\alpha = \neg c$, nous pouvons calculer $\varphi \circ \alpha$ en utilisant $\trianglelefteq_\varphi^{\mathcal{T}_1,\mathcal{T}_2}$. Nous avons alors $[\![\varphi \circ \alpha]\!] = \text{Min}\left([\![\alpha]\!], \trianglelefteq_\varphi\right) = \{\omega_0, \omega_6, \omega_4\}$ comme le montre la Figure 9(b).

Comme attendu, la proposition suivante dit que $\circ_V$ est un opérateur de révision OLP.

Proposition 7. *L'opérateur de révision $\circ_V$ satisfait $(\boldsymbol{R_1})$, $(\boldsymbol{R'_2})$, $(\boldsymbol{R_3})$–$(\boldsymbol{R_5}$,$)$, $(\boldsymbol{R_7})$, $(\boldsymbol{R_8})$, et $(\boldsymbol{R_{OLP}})$.*

À première vue, notre opérateur peut être confondu avec un opérateur de mise à jour [22, 20], comme l'opérateur PMA [35] très connu. Dans les deux cas, une famille

3. En utilisant une table de Karnaugh, les cellules directement voisines sont à une distance de Hamming de 1.

Référence	Structure ordonnée
[21]	préordre total
[21]	min-préordre partiel
[6]	préordre partiel
[29]	min-semi-ordre
[26]	semi-ordre
ici	ordre linéaire
ici	ordre linéaire partitionné

TABLE 2 – Un bref panorama des théorèmes de représentation de la littérature.

d'ordres est définie, chacun d'entre eux étant issu d'un modèle de la croyance initial φ. La différence la plus importante est que chaque ordre que nous produisons n'est pas sur l'ensemble des interprétations Ω mais seulement sur les éléments d'une partition. Une autre différence est que nous obtenons uniquement des comparabilités strictes. Cependant, une étude plus approfondie sur les liens entre la mise à jour et les opérateurs OLP doit être réalisée.

6 Discussion en rapport avec d'autres travaux

Les structures ordonnées jouent un rôle central dans la révision de croyance, à la fois dans sa version classique [1, 17] mais aussi dans sa version itérée [14]. Au moyen de théorèmes de représentation, ils offrent une meilleure compréhension des postulats de rationalité et confèrent une sémantique aux opérateurs de révisions. La Table 2 résume et situe notre contribution dans le paysage de la révision de croyance. Elle complète celle proposée dans [26]. Tous les articles mentionnés comme références dans la colonne de gauche donnent les théorèmes de représentations conduisant aux structures ordonnées données dans la colonne de droite.

Tout d'abord, nous présentons à nouveau la famille des opérateurs KM qui couvrent les opérateurs de révision pouvant être traduits par des préordre totaux. Un opérateur de révision totalement informatif quelconque est beaucoup plus spécifique qu'un opérateur de KM quelconque puisqu'il retourne toujours une formule complète. Les opérateurs de révision OLP sont significativement moins spécifiques puisqu'ils ne traitent que des situations *localement* totalement informées et de fait réintroduisent de l'incomparabilité.

Une première famille manipulant l'incomparabilité est celle proposée dans [21] obtenue en remplaçant $(\boldsymbol{R_6})$ par $(\boldsymbol{R_7})$ et $(\boldsymbol{R_8})$. Cette famille a été créée pour gérer un type spécifique d'informations partiellement ordonnées où les modèles des croyances actuelles sont préférés à toutes les autres interprétations. Afin de capturer l'ensemble des ordres partiels, même ceux ne disposant pas de minima globaux, il a été introduit dans [6] une nouvelle famille d'opérateur, déjà présentée dans la Section 2. Les opérateurs de révisions OLP sont une sous-classe de cette deuxième famille d'opérateurs de révision.

Une famille d'opérateur de révision pouvant être représentée par des semi-ordres avec des minima globaux a été proposée dans [29]. Pour ce faire, ils remplacent $(\boldsymbol{R_6})$ par $(\boldsymbol{R_8})$, $(\boldsymbol{R_9})$ et $(\boldsymbol{R_{10}})$.

$(\boldsymbol{R_9})$ Si $(\varphi \circ \alpha) \wedge \beta \nvdash \varphi \circ \beta$ alors $(\varphi \circ \beta) \wedge \alpha \vdash \varphi \circ \alpha$

$(\boldsymbol{R_{10}})$ Si $(\varphi \circ \alpha) \wedge \beta \vdash \bot$ et $(\varphi \circ \alpha) \wedge \gamma \nvdash \bot$ alors $(\varphi \wedge \gamma) \wedge (\alpha \wedge \beta) \vdash \varphi \circ (\alpha \wedge \beta)$

Les semi-ordres sont un type de structure ordonnée très connu dans le monde de l'économie pour être plus précautionneux vis-à-vis de l'indifférence. Cette structure a la capacité de rendre l'indifférence non transitive, ce qui est analogue avec l'exemple classique du sucre dans le café de [27]. Dans le but de pouvoir manipuler l'ensemble des semi-ordres, une autre famille d'opérateurs basés sur ceux de [29] mais en remplaçant $(\boldsymbol{R_2})$ par $(\boldsymbol{R_2'})$ a été proposée dans [26]. Aucune de ces familles ne peut gérer les situations localement totalement informées, cela peut se voir facilement via les modèles interdits des structures qu'elles représentent.

Notre famille d'opérateurs totalement informatifs permet à un agent de raffiner leur jugement entre les modèles qu'il ne pouvait pas distinguer avant de recevoir de nouvelles informations. Nous parvenons à ce résultat en affaiblissant $(\boldsymbol{R_2})$ en $(\boldsymbol{R_{2w}})$. Cette idée d'affaiblir $(\boldsymbol{R_2})$ de plusieurs manières a été étudiée en détail dans [19] au côté de l'étude du comportement de ce type d'opérateurs lorsqu'ils sont confrontés à la même information de façon répétée, conduisant au développement d'une notion de stabilité.

7 Conclusion et perspectives

Nous définissons dans cet article la notion d'opérateur totalement informatif $(\boldsymbol{TI})$ et apportons un nouveau postulat $(\boldsymbol{R_L})$. Nous proposons un jeu de postulats basé sur ceux de KM (à l'exception d'un affaiblissement de $(\boldsymbol{R_2})$) ainsi qu'un théorème de représentation conduisant aux ordres linéaires. Dans ce contexte, $(\boldsymbol{R_L})$ s'avère être équivalent à $(\boldsymbol{TI})$. Nous donnons un exemple d'opérateur totalement informatif, basé sur l'opérateur de Dalal couplé à une fonction de départage. Nous proposons une nouvelle structure, plus générale que les ordres linéaires, les ordres linéaires partitionnés OLP, dont nous étudions les propriétés en termes d'expressivité, de représentations graphiques et numériques, et de modèles interdits. Nous positionnons les OLP dans une constellation de structures ordonnées. Cette structure ne peut pas être capturée par les ordres d'intervalles ni par les ordres forts. Ensuite, nous proposons un nouveau postulat $(\boldsymbol{R_{OLP}})$, qui, couplé aux postulats de la révision partiellement ordonnée, conduit au théorème de représentation capturant les OLP. Enfin, nous apportons un opérateur satisfaisant tous ces postulats.

Ces résultats ouvrent nombre de nouvelles perspectives.

D'emblée, il semble intuitif de les appliquer à la révision de croyances itérée [14]. En particulier, une question intéressante est de savoir comment rester dans le même fragment de structures ordonnées tout au long du processus itéré. Il semble aussi naturel d'étudier les ordres forts partitionnés, qui constituent une généralisation directe des OLP. La création d'opérateurs utiles en pratique constitue une autre de ces questions. Ceux proposés dans cet article ont été réalisés avec un objectif d'illustration et afin de prouver la consistance de nos ensembles de postulats. La complexité computationnelle de ces opérateurs de révision reste une question théorique et pratique ouverte et les OLP sont de potentiels candidats pour efficacement représenter des modèles non représentables par des préordres totaux ou des ordres d'intervalles. Comme mentionné précédemment, nos travaux sont reliés à la fusion de croyances et partage des similarités avec la mise à jour. Il pourrait aussi être intéressant de se pencher sur ces autres facettes de la dynamique de croyances et il en va de même concernant le raisonnement non monotone [25].

Références

[1] Alchourrón, Carlos E., Peter Gärdenfors et David Makinson: *On the Logic of Theory Change : Partial Meet Functions for Contraction and Revision*. Journal of Symbolic Logic, 50 :510–530, 1985.

[2] Andrèka, Hajnal, Mark Ryan et Pierre Yves Schobbens: *Operators and Laws for Combining Preference Relations*. Journal of Logic and Computation, 12(1) :13–53, 2002.

[3] Arrow, K. J. et L. Hurwicz: *An optimality criterion for decision-making under ignorance*. Uncertainty and Expectations in Economics, 1979.

[4] Belahcène, Khaled, Vincent Mousseau, Wassila Ouerdane, Marc Pirlot et Olivier Sobrie: *Multiple criteria sorting models and methods—Part I : survey of the literature*. 4OR, pages 1–46, 2023.

[5] Benferhat, Salem, Sébastien Konieczny, Odile Papini et Ramón A. Pino Pérez: *Iterated Revision by Epistemic States : Axioms, Semantics and Syntax*. Dans *Proceedings of European Conference on Artificial Intelligence (ECAI'2000)*, pages 13–17, 2000.

[6] Benferhat, Salem, Sylvain Lagrue et Odile Papini: *Revision of Partially Ordered Information : Axiomatization, Semantics and Iteration*. Dans *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 376–381, 2005.

[7] Benferhat, Salem, Sylvain Lagrue et Julien Rossit: *Max-based Prioritized Information Fusion without Commensurability*. Journal of Logic and Computation, 19(6) :1577–1610, 2009.

[8] Binmore, Kenneth G.: *Interpersonal Comparison of Utility*. Oxford Handbook of Philosophy of Economic Science, page 200–254, 2007.

[9] Bloch, Isabelle, Jérôme Lang, Ramón Pino Pérez et Carlos Uzcátegui: *Morphologic for knowledge dynamics : revision, fusion, abduction*. CoRR, abs/1802.05142, 2018. `http://arxiv.org/abs/1802.05142`.

[10] Bouyssou, Denis et Thierry Marchant: *An axiomatic approach to noncompensatory sorting methods in MCDM, I : The case of two categories*. European Journal of Operational Research, 178(1) :217–245, 2007.

[11] Bouyssou, Denis et Thierry Marchant: *An axiomatic approach to noncompensatory sorting methods in MCDM, II : More than two categories*. European Journal of Operational Research, 178(1) :246–276, 2007.

[12] Bubboloni, Daniela et Michele Gori: *Breaking ties in collective decision-making*. Decisions in Economics and Finance, 44 :411–457, 2021.

[13] Dalal, Mukesh: *Investigations into a Theory of Knowledge Base Revision*. Dans *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[14] Darwiche, Adnan et Judea Pearl: *On the logic of iterated belief revision*. Artificial intelligence, 89 :1–29, 1997.

[15] Fishburn, Peter C.: *Generalizations of Semiorders : A Review Note*. Journal of Mathematical Psychology, 41(4) :357–366, 1997, ISSN 0022-2496.

[16] Freeman, Rupert, Markus Brill et Vincent Conitzer: *General Tiebreaking Schemes for Computational Social Choice*. Dans *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015*, pages 1401–1409, 2015.

[17] Gärdenfors, Peter: *Knowledge in flux : modeling the dynamics of epistemic states*. Bradford Books. MIT Press, Cambridge, 1988.

[18] Gärdenfors, Peter (rédacteur): *Belief Revision*. Cambridge University Press, 1992.

[19] Haret, Adrian et Stefan Woltran: *Belief Revision Operators with Varying Attitudes Towards Initial Beliefs*. Dans *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 1726–1733, 2019.

[20] Herzig, Andreas et Omar Rifi: *Propositional belief base update and minimal change*. Artificial Intelligence, 115(1), 1998.

[21] Katsuno, Hirofumi et Alberto O. Mendelzon: *Propositional Knowledge Base Revision and Minimal Change*. Artificial Intelligence, 52(3) :263–294, 1991.

[22] Katsuno, Hirofumi et Alberto O. Mendelzon: *On the Difference between Updating a Knowledge Base and Revising it*. Dans *[18]*, pages 183–203, 1992.

[23] Konieczny, Sébastien, Jérôme Lang et Pierre Marquis: *DA^2 Merging Operators*. Artificial Intelligence, 157(1-2) :49–79, 2004.

[24] Konieczny, Sébastien et Ramón Pino Pérez: *Merging Information Under Constraints : a Logical Framework*. Journal of Logic and Computation, 12(5) :773–808, 2002.

[25] Kraus, Sarit, Daniel Lehmann et Menachem Magidor: *Nonmonotonic reasoning, preferential models and cumulative logics*. Artificial Intelligence, 44 :167–207, 1990.

[26] León, María Victoria et Ramon Pino Pérez: *Orders and Belief Revision*. Dans *Proceedings of the 19th International Workshop on Non-Monotonic Reasoning (NMR21)*, pages 11–20, 2021.

[27] Luce, R. Duncan: *Semiorders and a Theory of Utility Discrimination*. Econometrica, 24(2) :178–191, 1956.

[28] Öztürk, Meltem, Marc Pirlot et Alexis Tsoukiàs: *Representing preferences using intervals*. Artificial Intelligence, 175(7-8) :1194–1222, 2011.

[29] Peppas, Pavlos et Mary-Anne Williams: *Belief Change and Semiorders*. Dans *Principles of Knowledge Representation and Reasoning : Proceedings of the Fourteenth International Conference, KR 2014*, 2014.

[30] Rolland, Antoine: *Reference-based preferences aggregation procedures in multi-criteria decision making*. Eur. J. Oper. Res., 225(3) :479–486, 2013.

[31] Ryan, Mark: *Belief Revision and Ordered Theory Presentations*. Logic, Action, and Information, page 129–151, 1996.

[32] Sen, Amartya K: *Choice functions and revealed preference*. The Review of Economic Studies, 38(3) :307–317, 1971.

[33] Sen, Amartya K.: *Handbook of Mathematical Economics*, tome 3. North-Holland, 1982.

[34] Smets, Philippe: *Decision making in the TBM : the necessity of the pignistic transformation*. International journal of approximate reasoning, 38(2) :133–147, 2005.

[35] Winslett, M: *Reasoning about action using a possible models approach*. Dans *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'88)*, pages 89–93, 1988.

Actes JIAF-JFPDA 2023

S&F: Évaluation de la fiabilité des sources et des faits

Quentin Elsaesser[1] **Patricia Everaere**[2] **Sébastien Konieczny**[1]

[1] CRIL, CNRS - Université d'Artois, France
[2] CRIStAL, Université de Lille, France

elsaesser@cril.fr
patricia.everaere-caillier@univ-lille.fr
konieczny@cril.fr

Résumé

Dans ce travail, nous proposons une famille de méthodes qui permettent de calculer conjointement la fiabilité d'un ensemble de sources d'information et la confiance des faits sur un ensemble d'objets, en confrontant les points de vue des sources. Nous utilisons une méthode de vote pour l'évaluation de la confiance des sources, en utilisant les arguments du théorème du jury de Condorcet afin d'identifier la vérité et les sources fiables. Nous discutons des propriétés théoriques générales que de tels opérateurs devraient satisfaire, et nous étudions quelles sont les propriétés satisfaites par nos méthodes. Nous proposons une étude expérimentale qui montre que nous sommes plus performants que les méthodes de l'état de l'art pour trouver la vérité parmi les faits possibles. Nous montrons que nous pouvons également évaluer de manière adéquate la fiabilité des sources d'information.

Abstract

In this work we propose a family of methods that allow to conjointly compute the reliability of a set of information sources and the confidence of the facts on a set of objects, by confronting the sources points of view. We use a (scoring-based) voting method for the evaluation of the trust of the sources, using Condorcet's Jury Theorem arguments in order to identify the truth and the reliable sources. We discuss general theoretical properties that such operators should satisfy, and we study what are the properties satisfied by our methods. We provide an experimental study that shows that we perform better than state of the art methods on the task of finding the truth among the possible facts. We show that we can also adequately evaluate the reliability of the sources of information.

1 Introduction

Il existe de nombreuses applications où l'on reçoit des informations (généralement contradictoires) de différentes sources. À l'aide de ces informations, on doit se forger une opinion. Dans cette situation, une façon standard de résoudre les conflits est de faire confiance aux sources les plus fiables. Nous proposons une définition de la fiabilité basée sur les informations disponibles. Cette définition peut être utile pour évaluer la fiabilité d'un agent dans un système multi-agents ou sur un réseau social, mais aussi d'une source sur le web, dans un journal, etc.

Plus précisément, nous considérons un ensemble de sources qui nous fournissent des informations (que nous appellerons *faits*) sur différentes questions (que nous appellerons *objets*). Notre objectif est d'évaluer à la fois la fiabilité des sources et la fiabilité des faits, ce qui nous permet ensuite de trouver les réponses correctes aux différentes questions (objets).

Il existe des travaux antérieurs qui utilisent la même structure (sources/faits/objets), mais leur objectif est uniquement de trouver la vérité parmi les faits [33, 28].

Pour trouver cette vraie information, nous nous appuyons sur l'idée du théorème du jury de Condorcet [6], selon lequel il est plus probable que la majorité des individus choisissent la bonne solution. L'intuition est la suivante : supposons que parmi 10 sources de fiabilité égale, 8 vous disent que la *Capitale de l'Australie* est *Canberra*, et 2 vous disent que c'est *Sydney*. Suivre ce que dit la majorité est le moyen le plus sûr de trouver la vérité. Le théorème du jury de Condorcet requiert un grand nombre d'hypothèses (toutes les sources ont la même fiabilité, elles sont toutes fiables (c'est-à-dire qu'elles ont plus de 50% de chances de trouver la vérité), elles sont indépendantes et le choix

ne porte que sur deux possibilités). Cependant, toutes ces hypothèses peuvent être plus ou moins assouplies [17, 35, 23, 4, 10, 12, 2, 5, 3, 19, 21, 34, 14, 15, 9].

Dans cet article, nous supposons qu'au départ, nous n'avons aucune information sur la fiabilité des sources, et nous définissons une procédure itérative pour déterminer leur fiabilité. Au début, nous attribuons la même fiabilité à toutes les sources, puis nous comparons les réponses aux différentes questions et nous utilisons cet argument du "théorème du jury de Condorcet" pour récompenser les sources qui fournissent des informations (*faits*) confirmées par d'autres, et qui sont donc plus susceptibles d'être vraies. Ensuite, nous itérons le processus avec la nouvelle fiabilité des sources jusqu'à convergence.

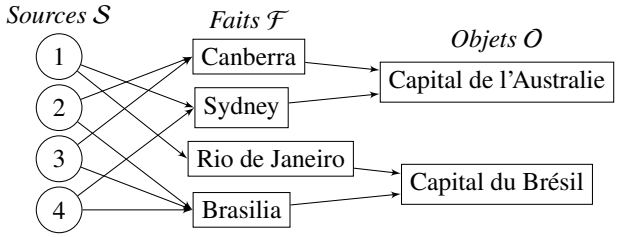


FIGURE 1 – Sources, faits et objets

Pour illustrer ce processus, prenons l'exemple de la figure 1, où quatre sources donnent des informations sur deux objets : *Capitale du Brésil* et *Capitale de l'Australie*. Notons qu'il y a d'abord égalité pour *Capitale de l'Australie*, deux sources donnant *Canberra* et deux sources donnant *Sydney*. Mais nous pouvons utiliser le résultat de l'autre objet. Il y a une majorité pour *Brasilia*, donc *Brasilia* est considéré comme le bon fait, et les sources qui donnent ce fait sont favorisées par rapport à celles qui donnent *Rio de Janeiro*. Et, à la prochaine itération, nous pourrons les départager sur *Capitale de l'Australie* puisque des sources plus fiables affirment *Canberra*.

Plus précisément, à chaque itération, les sources donnent une certaine force aux faits qu'elles affirment sur les différents objets. Cette force est la fiabilité (actuelle) de la source. Ainsi, on peut classer pour chaque objet les faits correspondants du plus fiable au moins fiable, en utilisant simplement la somme des forces obtenues. Ensuite, pour chaque objet, un vote est organisé pour récompenser les sources. Nous utilisons des méthodes de vote par scores (*scoring-voting rules*) afin d'associer un nombre à chaque rang de faits. La plus simple est la règle de la majorité simple, où seuls les faits les plus fiables rapportent un score de 1 aux sources correspondantes, et tous les autres n'obtiennent rien (0). La nouvelle fiabilité de chaque source est calculée en combinant tous ces scores. Nous devons faire un choix entre deux normalisations pour cette étape : une qui favorise les sources qui fournissent le plus d'informations, et une qui favorise les sources qui sont plus prudentes et qui font peu d'erreurs. Ensuite, une nouvelle itération commence avec la fiabilité actualisée de chaque source.

L'évaluation de la fiabilité peut être utilisée pour décider de faire confiance ou non à une source. Il s'agit d'un élément important pour évaluer la confiance (*trust*) directe (la confiance peut être divisée en confiance directe – obtenue par interaction directe avec un agent, et en confiance indirecte – évaluation obtenue par un tiers [20]). De nombreux travaux s'intéressent à la prise en compte de la confiance et de la réputation dans les décisions et les processus de raisonnement [31, 1, 26, 29, 24, 30, 11]. Mais il existe peu de travaux sur la manière d'évaluer cette confiance directe à partir de preuves, et notre travail peut être très utile à cet égard.

Après avoir présenté nos méthodes S&F (pour *Sources & Faits*), nous discutons des propriétés logiques pour caractériser les méthodes intéressantes qui visent à évaluer la fiabilité des sources et des faits. Nous passons en revue les propriétés qui ont été proposées par [28], et nous expliquons pourquoi certaines d'entre elles ne sont pas adaptées à ce contexte. Nous proposons également de nouvelles propriétés requises pour toutes les méthodes, ainsi que certaines propriétés qui caractérisent des sous-classes intéressantes. Nous vérifions ensuite quelles propriétés sont satisfaites par nos méthodes.

Outre cette évaluation formelle, nous proposons également une étude expérimentale. L'idée est de tester si nous pouvons atteindre cet objectif d'évaluation de la fiabilité des sources et des faits en pratique. Il n'y a pas beaucoup de jeux de données réels qui peuvent être utilisés pour cette tâche, mais nous testons nos méthodes sur deux de ces jeux de données. Ensuite, nous testons également nos méthodes sur des jeux de données que nous avons générés, ce qui nous permet de tester beaucoup plus de paramètres.

Et les résultats sont bons. Nous montrons que pour les tâches liées à la recherche des faits réels, nous sommes meilleurs que les méthodes existantes. Mais, contrairement aux méthodes existantes, nous pouvons également donner une bonne évaluation de la fiabilité des sources.

2 Préliminaires

Nous considérons trois ensembles $\mathcal{S}$, $\mathcal{F}$ et $\mathcal{O}$ respectivement appelés *Sources*, *Faits* et *Objets*. Les *Sources* représentent les agents (humains ou artificiels) qui fournissent les informations. Les *Objets* sont les questions sur lesquelles nous aimerions obtenir des informations et les *Faits* sont les réponses possibles. Sur chaque objet, les faits sont distincts et exclusifs : chaque source ne peut affirmer qu'un seul fait par objet.

Ces objets+faits peuvent donc être considérés comme des questions+réponses ou comme des variables+valeurs. Il s'agit ici d'une question de vocabulaire. Nous utilisons celui utilisé dans les travaux précédents [33, 28].

Définition 1 *Soit $G = (V, E)$ un graphe orienté avec $V = \mathcal{S} \cup \mathcal{F} \cup \mathcal{O}$ et $E \subseteq (\mathcal{S} \times \mathcal{F}) \cup (\mathcal{F} \times \mathcal{O})$, t.q. :*

- *Pour chaque fait $f \in \mathcal{F}$ il y a un objet unique $o \in \mathcal{O}$ avec $(f, o) \in E$.*
- *Une source $s \in \mathcal{S}$ peut affirmer au plus un fait par objet $o \in \mathcal{O}$ (c'est-à-dire $\forall s \in \mathcal{S}$ il n'y a pas de $f_1 \in \mathcal{F}, f_2 \in \mathcal{F}$ t.q. $\{(f_1, o), (f_2, o), (s, f_1), (s, f_2)\} \subseteq E$).*

$(s, f) \in E$ signifie que la source s affirme que le fait f est la réponse correcte pour l'objet correspondant. Il est possible qu'un fait ne soit affirmé par aucune source.

Pour une meilleure lisibilité, nous utiliserons les notations suivantes : $src(f) = \{s \in \mathcal{S} : (s, f) \in E\}$, $fct(s) = \{f \in \mathcal{F} : (s, f) \in E\}$, $fct(o) = \{f \in \mathcal{F} : (f, o) \in E\}$, $obj(f) = \{o \in \mathcal{O} : (f, o) \in E\}$.

Lorsque ce n'est pas évident (par exemple lorsque nous avons plus d'un graphe), nous pouvons spécifier le graphe, et nous écrivons, pour le graphe G, $src_G(f)$, $fct_G(s)$, $fct_G(o)$ et $obj_G(f)$ à la place des notations ci-dessus.

Nous notons $r_G(s) \in [0, 1]$ la fiabilité d'une source s dans le graphe G. Nous notons $c_G(f) \in \mathbb{R}^+$ la confiance d'un fait f dans le graphe G. [1]

3 Travaux Connexes

Il existe dans la littérature des algorithmes de *découverte de la vérité* qui visent à identifier les faits réels.

Truth Finder[33] est un algorithme itératif qui met à jour le score des sources et des faits à chaque itération. Cette méthode se concentre sur la confiance des faits pour trouver la vérité. Avec *Truth Finder*, la fiabilité d'une source est la confiance moyenne dans les faits affirmés par cette source. Pour la confiance d'un fait, les auteurs supposent que les faits peuvent se soutenir mutuellement, auquel cas la confiance augmente, mais diminue si les faits se contredisent. *Truth Finder* permet également de prendre en compte la similiraté entre les faits. Nous n'examinons pas cette possibilité dans ce travail, nous laissons donc cet aspect de côté (voir [32] pour une discussion sur les méthodes de découverte de la vérité, où il y a d'autres paramètres qui peuvent être pris en compte pour calculer la fiabilité).

Hubs and Authorities [13] est une méthode définie pour classer les pages web mais peut aussi être utilisée pour la recherche de la vérité. Il s'agit également d'une méthode itérative, qui définit deux scores différents pour une page. *Hub* (qui correspond aux sources) favorise les pages qui pointent vers de nombreuses autres pages, et *authority* (qui correspond aux faits) favorise les pages qui sont pointées par de nombreux hubs différents.

Sums [22] est basé sur *Hubs and Authorities*. La principale différence réside dans la manière dont la fiabilité des sources et des faits est normalisée. Avec *Sums*, la fiabilité d'une source est la somme de la confiance des faits qu'elle affirme ($r(s) = \Sigma_{f \in fct(s)} c(f)$), et la confiance d'un fait f est obtenue comme la somme de la fiabilité des sources qui l'affirment ($c(f) = \Sigma_{s \in src(f)} r(s)$). Ces résultats sont ensuite normalisés par la valeur maximale obtenue (par $max(r(s)) \in \mathcal{S}$ pour $r(s)$ et par $max(c(f)) \in \mathcal{F}$ pour $c(f)$).

Booth et Singleton ont été les premiers à proposer une approche axiomatique du problème de la *découverte de la vérité* dans [27] et [28]. Ils proposent également une nouvelle méthode, appelée *Unbounded-Sums*, qui est basée sur *Sums*, mais où ils ne normalisent pas le score.

Nous comparerons les résultats expérimentaux de nos méthodes aux résultats des algorithmes *Hubs and Authorities*, *Truth Finder*, *Sums* et *Unbounded-Sums*. Nous nous comparerons également avec *Voting*, la méthode naïve qui choisit le fait avec le plus de soutien (arêtes dans le graphe) sur chaque objet.

4 Méthodes S&F

Dans cette section, nous présentons nos méthodes, composées de deux étapes. Tout d'abord, la confiance des faits est calculée. Ensuite, une règle de vote est utilisée pour évaluer la fiabilité des sources. Nous utilisons une méthode itérative pour calculer ces deux étapes.

À chaque itération, la confiance des faits (evaluate-facts($\mathcal{F}$) – voir section 4.1) est évaluée et un classement des faits de chaque objet est obtenu. La méthode de vote est ensuite utilisée pour récompenser les faits par rapport au classement (vote($\mathcal{F}$) – voir section 4.2). La fiabilité des sources est déduite des récompenses des faits qu'elles affirment (evaluate-sources($\mathcal{S}$) – voir l'équation 2) et est normalisée (normalize($\mathcal{S}$) – voir la section 4.3).

L'algorithme s'arrête lorsque le processus converge, c'est-à-dire lorsque la distance euclidienne entre la fiabilité des sources de la dernière itération et celle de l'itération actuelle est inférieure à ϵ avec $\epsilon = 0,001$.

Une méthode de cette famille est caractérisée par le choix d'une règle de vote et d'une fonction de normalisation.

4.1 Confiance des faits

La confiance dans un fait f est simplement calculée en additionnant la fiabilité des sources qui l'ont affirmé. Plus les sources qui l'affirment sont fiables, plus la confiance en ce fait est grande.

$$c(f) = \sum_{s \in src(f)} r(s) \quad (1)$$

1. Nous notons simplement $r(s)$ et $c(f)$, sans l'indice, lorsqu'il n'y a aucune ambiguïté sur le graphe utilisé.

Algorithm 1 S&F Algorithme

Input : Un graphe $G = (\mathcal{S} \cup \mathcal{F} \cup \mathcal{O}, E)$.
Output : La fiabilité des sources $r(s)$ et la fiabilité des faits $c(f)$.

1: $r(s) = 1\ \forall s \in \mathcal{S}$ #fiabilité initiale des sources
2: $c(f) = 0\ \forall f \in \mathcal{F}$ #fiabilité initiale des faits
#ts (resp. ts^{-1}) est le vecteur de fiabilité des sources pendant l'itération actuelle (resp. dernière) t.q. $ts = \langle r(s) : \forall s \in \mathcal{S} \rangle$.
3: **while** not convergence(ts, ts^{-1}) **do**
4: evaluate-facts($\mathcal{F}$)
5: vote($\mathcal{F}$)
6: evaluate-sources($\mathcal{S}$)
7: normalize($\mathcal{S}$)
8: **end while**

4.2 Fiabilité des sources

L'évaluation de la fiabilité des sources est obtenue par un vote des objets. Chaque objet récompense les sources qui affirment les faits les plus plausibles. Nous utiliserons des règles de vote par scores pour calculer cette fiabilité.

Pour chaque objet, nous classons les faits correspondants du plus fiable au moins fiable, et la règle de vote par scores associe un score à chaque rang.

La nouvelle fiabilité des sources est alors la somme de ces scores, comme il est d'usage dans les règles de vote par scores.

Mais nous devons procéder à trois ajustements : Le premier est que les règles de vote par scores sont définies pour des ordres linéaires, alors que nous obtenons des pré-ordres totaux : certains faits peuvent avoir le même rang. Nous utiliserons donc la moyenne des scores des faits avec le même classement. Le second est que le nombre d'options (faits) n'est pas le même pour tous les objets, de sorte que nous devons choisir comment normaliser ces scores sur différentes échelles. Le troisième est que, lorsque les scores sont reçus par les sources, nous les normalisons afin d'avoir un résultat compris dans $[0, 1]$.

Définition 2 *Soit M un entier et e une suite d'entiers non décroissants avec $e_1 \geq e_2 \geq \ldots \geq e_M$ t.q. $e_1 > e_M$. Une règle de vote par scores v est une fonction qui, à chaque ordre linéaire $>$ sur un ensemble d'au plus M faits et à chaque fait f, associe un entier positif t.q. si le fait f est classé à la ième position dans l'ordre linéaire $>$, alors $v(>, f) = e_i$.*

Lorsque $e_1 = 1$ et $e_2 = e_M = 0$, la règle est appelée majorité simple. Lorsque $e_1 = M - 1, e_2 = M - 2, \ldots, e_M = 0$, il s'agit de la règle de Borda.

Pour les procédures de vote standard, les électeurs votent sur un ensemble fixe de candidats, et M est le nombre de candidats. Dans notre cas, les objets sont liés à différents nombres de faits. Nous devons donc procéder à une (première) normalisation en fonction du nombre maximal de faits. Nous définissons donc $M = best\_score(\mathcal{F}) = max(|fct(o)|), \forall o \in \mathcal{O}$. Cela signifie que pour tous les objets, le fait d'être le plus plausible donne toujours le même score (e_1), quel que soit le nombre de faits liés à l'objet.

De plus, contrairement aux hypothèses classiques des règles de vote standard, le classement associé à un objet n'est pas un ordre linéaire, mais un pré-ordre total (certains faits peuvent avoir la même confiance). Nous devons ajuster les règles de vote par scores pour pouvoir gérer de manière adéquate les égalités possibles. Dans ce cas, nous donnons la note moyenne, c'est-à-dire la moyenne des notes qu'ils étaient censés recevoir, comme dans [25].

Un pré-ordre total (une relation réflexive, transitive et totale) [2] $\geq$ peut être considéré comme un ensemble de strates. Un élément x appartient à une strate $T_{\geq}$ composée d'un ensemble d'éléments équivalents $\{y | x \simeq y\}$. $T^i_{\geq}$ est la i^{th} strate du pré-ordre si $\exists x_1, \ldots, x_{i-1}$ t.q. $x_1 > \ldots > x_{i-1} > y$ avec $y \in T^i_{\geq}$. S'il n'existe pas de $x_1 > y$ avec $y \in T^i_{\geq}$ alors $i = 1$.

Un ordre linéaire $\succ$ est dit compatible avec un pré-ordre $\geq$ si $\forall x \in T^i, \forall y \in T^j, i < j \Rightarrow x \succ y$.

Définition 3 *Pour chaque fait f, considérons l'objet correspondant o. Soit $\mathcal{P}(o)$ le pré-ordre donné par la confiance des faits (c'est-à-dire $f_1 \geq_{\mathcal{P}(o)} f_2$ ssi $c(f_1) \geq c(f_2)$) et m le nombre de strates dans $\mathcal{P}(o)$. On a $\mathcal{P}(o) = \{T^1(o), T^2(o), \ldots, T^m(o)\}$, où $T^k(o)$ est la k^{th} strate dans $\mathcal{P}(o)$. Le score attribué à f pour le pré-ordre $\mathcal{P}(o)$ et la règle de pondération v est défini comme suit (où $\succ_o$ est tout ordre linéaire compatible avec $\mathcal{P}(o)$) :*

$$V_v(f) = \frac{\sum_{\{y \in T^i(o) | f \in T^i(o)\}} v(\succ_o, y)}{|T^i(o)|}$$

Exemple 1 *Soit G un graphe avec deux objets. Soit v la règle de Borda. Le premier objet $o1$ et le second objet $o2$ ont respectivement 6 et 9 faits qui leur sont liés. On a $best\_score(\mathcal{F}) = 9$. Supposons que $\mathcal{P}(o1) = \{T^1(o1) = \{f_1, f_2\}, T^2(o1) = \{f_3\}, T^3(o1) = \{f_4, f_5, f_6\}\}$, c'est-à-dire que $o1$ a classé 2 faits en premier (dans la première strate), 1 fait en deuxième et 3 faits en troisième. Le score de f_1 et f_2 est donc $V_{Borda}(f_i) = \frac{(9-1)+(9-2)}{2} = 7.5$, le score de f_3 est $V_{Borda}(f_3) = (9 - 3) = 6$ et le score de f_4, f_5 et f_6 est $V_{Borda}(f_i) = \frac{(9-4)+(9-5)+(9-6)}{3} = 4$.*

Et, comme d'habitude pour les règles de vote par scores, le score d'une source est simplement la somme des scores donnés par chaque votant (objet) :

Définition 4 *La fiabilité initiale d'une source (avant normalisation) est :*

2. À partir de tout pré-ordre total $\geq$, nous définissons l'ordre strict correspondant $>$ comme $x > y$ ssi $x \geq y$ et $y \not\geq x$, et la relation d'équivalence correspondante $\simeq$ comme $x \simeq y$ ssi $x \geq y$ et $y \simeq x$.

$$r^I(s) = \sum_{f \in fct(s)} V_v(f) \quad (2)$$

Voyons maintenant le dernier ajustement nécessaire des scores.

4.3 Normalisations *A* and *C*

Nous souhaitons donner une estimation de la fiabilité d'une source, c'est-à-dire de la probabilité qu'étant donné un objet, cette source trouve le fait réel correspondant. Nous devons donc normaliser la fiabilité des sources pour nous assurer que cette fiabilité soit comprise entre 0 et 1. Il existe (au moins) deux façons raisonnables de normaliser la fiabilité. La première favorise les sources qui fournissent beaucoup d'informations plausibles. La seconde se concentre sur la qualité et ensuite sur la proportion des informations fournies par la source.

Nous appelons la première normalisation *A* (pour *All objects*). La fiabilité des sources est divisée par le nombre d'objets dans le graphe. Si une source a un score proche de 1, nous savons que la source s'exprime correctement pour presque tous les objets. Si la fiabilité d'une source est faible, cela signifie soit que la source commet beaucoup d'erreurs et perd des votes, soit que la source ne s'exprime pas beaucoup.

Définition 5 *La fiabilité d'une source après la normalisation A est :*

$$r^A(s) = \frac{r^I(s)}{best\_score(\mathcal{F}) * |O|} \quad (3)$$

La seconde normalisation est appelée *C* (pour *Claimed facts*). La fiabilité d'une source est divisée par le nombre d'objets sur lesquels elle affirme un fait. Contrairement à la normalisation précédente, si une source a un score proche de 1, nous savons que la source s'exprime correctement mais nous n'avons pas idée du nombre d'objets sur laquelle elle s'exprime.

Définition 6 *La fiabilité d'une source après la normalisation C est :*

$$r^C(s) = \frac{r^I(s)}{best\_score(\mathcal{F}) * |obj(s)|} \quad (4)$$

Où $obj(s) = \{o \in O \ : \ \exists f \in \mathcal{F} \ : \ (s, f), (f, o) \in E\}$.

Cette normalisation favorise donc les sources qui s'expriment correctement, tandis que la précédente favorise les sources qui s'expriment beaucoup (et de manière correcte).

Notons que le score le plus élevé qu'une source puisse obtenir est $best\_score(\mathcal{F})$, il faut donc multiplier le dénominateur par cette valeur. Dans le cas d'un graphe complet, les deux normalisations sont identiques. On note $r(s)$ la fiabilité normalisée d'une source lorsqu'il n'y a pas d'ambiguïté sur la normalisation utilisée.

4.4 Exemple

Nous allons maintenant voir un exemple afin de montrer les différences entre les deux normalisations pour la majorité simple.

L'impact de la normalisation est illustré dans la Table 1 et dans la Table 2 : la source 1 est l'une des meilleures sources avec la normalisation *C* mais l'une des pires avec la normalisation *A*. De même, le fait a_2 devient l'un des meilleurs sur l'objet *A* avec la normalisation *C*, alors qu'il est l'un des pires avec la normalisation *A*. En changeant la normalisation, nous avons des changements significatifs dans les résultats de l'algorithme, comme la source 1 et le fait a_2 dans l'exemple. Les itérations de l'algorithme sont détaillées pour la méthode avec la majorité simple et la normalisation *A* dans la Table 1 et dans la Table 2 pour la normalisation *C*.

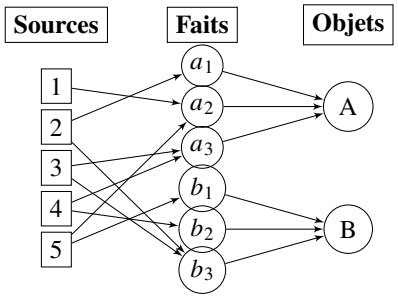


FIGURE 2 – Graphe

sources		**1**	**2**	**3**	**4**	**5**	**/**
It1	fiabilité	1	1	1	1	1	/
It2	fiabilité	0.5	0.5	1	0.5	0.5	/
It3	fiabilité	0	0.5	1	0.5	0	/
faits		**a1**	**a2**	**a3**	**b1**	**b2**	**b3**
It1	confiance	1	2	2	1	1	2
	$V_f(v)$	0	1	1	0	0	1
It2	confiance	0.5	1	1.5	0.5	0.5	1.5
	$V_f(v)$	0	0	1	0	0	1
It3	confiance	0.5	0	1.5	0	0.5	1.5
	$V_f(v)$	0	0	1	0	0	1

TABLE 1 – Itérations avec majorité simple *A*

sources		**1**	**2**	**3**	**4**	**5**	**/**
It1	fiabilité	1	1	1	1	1	/
It2	fiabilité	1	0.5	1	0.5	0.5	/
faits		**a1**	**a2**	**a3**	**b1**	**b2**	**b3**
It1	confiance	1	2	2	1	1	2
	$V_f(v)$	0	1	1	0	0	1
It2	confiance	0.5	1.5	1.5	0.5	0.5	1.5
	$V_f(v)$	0	1	1	0	0	1

TABLE 2 – Itérations avec majorité simple *C*

Si nous entrons dans les détails, avec la normalisation *A*,

nous pouvons voir à la première itération que la confiance des faits a_2 et a_3 est la même, puisqu'ils sont tous deux affirmés par deux sources. Mais la fiabilité de ces sources est ajustée à la fin de cette itération, et pour la deuxième itération, a_3 devient plus crédible. Notez qu'à la troisième itération, a_2 devient encore moins crédible que a_1 puisqu'il n'est affirmé que par des sources non fiables à cette itération.

Notons également que la méthode avec la normalisation C s'arrête en deux itérations mais que la normalisation A a besoin d'une itération supplémentaire pour s'arrêter.

5 Propriétés

Cette section comporte deux volets. Tout d'abord, nous souhaitons abstraire le problème et nous demander quelles propriétés devraient être satisfaites par les méthodes qui visent à évaluer la fiabilité des sources et la confiance dans les faits. Nous rappelons certaines propriétés proposées dans [28, 27] et les discutons, et nous en proposons de nouvelles. En particulier, nous proposons un ensemble de propriétés (les propriétés de base) que toute méthode devrait satisfaire, ainsi que d'autres propriétés intéressantes pour caractériser des sous-classes intéressantes de méthodes. Le second objectif de cette section est d'évaluer nos méthodes par rapport à cet ensemble de propriétés.

Commençons par donner quelques définitions utilisées par les propriétés.

Définition 7 *Nous notons $B(o)$ l'ensemble des faits classés premiers pour l'objet o : $B(o) = \{f \in fct(o) \mid \forall f' \in fct(o), c(f) \geq c(f')\}$.*

Définition 8 *Nous notons $B(\mathcal{F})$ l'ensemble des faits classés premiers pour un objet du graphe : $B(\mathcal{F}) = \{f \in \mathcal{F} \mid \exists o \in \mathcal{O}\ ,\ f \in B(o)\}$*

Nous avons besoin d'une définition venant de [28] pour la notion de *"est moins plausible que"* utilisée dans leurs propriétés.

Définition 9 *Soit $Y, Y' \subseteq \mathcal{F}$, Y est moins plausible que Y' s'il y a une bijection $\phi : Y \longrightarrow Y'$ t.q. $c(f) \leq c(\phi(f))$ pour chaque $f \in Y$ et $c(f') < c(\phi(f'))$ pour au moins un fait $f' \in Y$. Soit $X, X' \subseteq \mathcal{S}$, X est moins fiable que X' est défini de façon similaire.*

5.1 Propriétés de base

Nous présentons les propriétés que toutes méthodes qui souhaitent estimer correctement la fiabilité des sources et trouver la vérité parmi les faits doivent satisfaire.

Si la fiabilité d'une source est égale à 1 (le score le plus élevé pour une source), cela signifie que tous ses faits sont les plus plausibles, c'est-à-dire qu'ils ont la plus grande confiance sur leur objet :

P1 (Best) *Soit $s \in \mathcal{S}$, si $r(s) = 1$ alors $fct(s) \subseteq B(\mathcal{F})$.*

Si une source n'affirme aucun fait, sa fiabilité doit être la plus faible :

P2 (Null Player) *Soit $s \in \mathcal{S}$, si $fct(s) = \emptyset$ alors $r(s) = 0$.*

Nous rappelons quatre propriétés et une définition de [28].

Si un fait n'est affirmé par aucune source, sa confiance est inférieure ou égale à celle de tous les autres faits :

P3 (Groundedness) *Nous supposons que $src(f) = \emptyset$ pour $f \in \mathcal{F}$. Alors pour tous les autres $g \in \mathcal{F}$, $c(f) \leq c(g)$.*

Si un fait est affirmé par toutes les sources, sa confiance sera la plus élevée :

P4 (Unanimity) *Nous supposons que $src(f) = \mathcal{S}$ pour $f \in \mathcal{F}$. Alors pour tous les autres $g \in \mathcal{F}$, $c(f) \geq c(g)$.*

Définition 10 *Deux graphes G et G' sont équivalents s'il existe un graphe isomorphe π entre eux qui préserve les sources, faits et objets t.q. $\pi(s) \in \mathcal{S}', \pi(f) \in \mathcal{F}'$ et $\pi(o) \in \mathcal{O}'$ pour tous les $s \in \mathcal{S}$, $f \in \mathcal{F}$ et $o \in \mathcal{O}$.*

Les valeurs calculées pour la fiabilité des sources et la confiance des faits dépendent uniquement du graphe et non de leur nom. Cette propriété indique donc que toutes les sources et tous les faits sont traités de la même manière (le nom original de cette propriété est "*Symmetry*" dans [28]) :

P5 (Neutrality) *Si G et $G' = \pi(G)$ sont des graphes équivalent, alors ($r_G(s1) \geq r_G(s2)$ ssi $r_{G'}(\pi(s1)) \geq r_{G'}(\pi(s2))$) et ($c_G(f1) \geq c_G(f2)$ ssi $c_{G'}(\pi(f1)) \geq c_{G'}(\pi(f2))$).*

Le classement des éléments d'une composante connexe n'est pas influencé par les éléments extérieurs à la composante (le nom original de cette propriété est "*PCI*" dans [28]) :

Définition 11 *Soit $G = (V, E)$ et $G' = (V', E')$ deux graphes. On dit que G et G' sont indépendants lorsqu'il n'y a aucun liens qui relient les éléments du graphe G et les éléments du graphe G', i.e. $V \cap V' = \emptyset$.*

P6 (Independence) *Soit $G = (V, E)$, $G1 = (V1, E1)$, $G2 = (V2, E2)$ trois graphes t.q. G et Gi ($i \in \{1, 2\}$) sont des graphes indépendants. Alors le classement des sources et des faits du graphe G doit être le même pour $G \cup G1$ et pour $G \cup G2$: $\forall s1, s2 \in \mathcal{S}_G$ nous avons $r_{G \cup G1}(s1) \geq r_{G \cup G1}(s2)$ ssi $r_{G \cup G2}(s1) \geq r_{G \cup G2}(s2)$. Et $\forall f1, f2 \in \mathcal{F}_G$ nous avons $c_{G \cup G1}(f1) \geq c_{G \cup G1}(f2)$ ssi $c_{G \cup G2}(f1) \geq c_{G \cup G2}(f2)$.*

Les faits affirmés par des sources moins fiables sont forcément moins crédibles.

P7 (Fact Coherence) *Si* $src(f1)$ *est moins fiable que* $src(f2)$ *alors* $c(f1) < c(f2)$.

Certaines propriétés supplémentaires semblent également souhaitables.

Définition 12 *Soit* $G = (V, E)$ *un graphe. Nous notons* $dupS(G, s, n)$ *le graphe avec lequel l'on copie la source* s *ainsi que tous ses liens n fois.* $dupS(G, s, n) = (V', E')$ *où* $V' = V \cup \{s_1, s_2, ...s_n\}$ *et* $E' = E \cup \{(s_i, f) | f \in fct_G(s), i = 1, ..., n\}$.

Si une opinion est suffisamment populaire, elle doit être considérée comme la vérité. Ainsi, si une source est dupliquée suffisamment de fois, ses faits doivent devenir les plus plausibles.

P8 (Majority) *Soit* $G = (V, E)$ *un graphe et* $s \in \mathcal{S}_G$. $\exists n > 0$ *t.q.* $fct_{G'}(s) \subseteq B_{G'}(\mathcal{F})$ *avec* $G' = dupS(G, s, n)$.

Nous allons maintenant voir un cas particulier où un graphe n'a qu'un seul objet.

Définition 13 *Notons* $\mathcal{O}_1$ *le graphe* $G = (\mathcal{S} \cup \mathcal{F} \cup \mathcal{O}, E)$ *avec un seul objet (t.q.* $|\mathcal{O}| = 1$*).*

Lorsqu'il n'y a qu'un seul objet dans un graphe, un fait qui est affirmé par un plus grand nombre de sources aura une meilleure confiance qu'un autre fait étant moins affirmé. Cette propriété est importante, car elle indique que la force de base d'un fait est donnée par le nombre d'affirmations. Mais avec plus d'un objet, les informations recueillies sur d'autres objets peuvent être utilisées pour prendre de meilleures décisions. Cette propriété n'est donc pas souhaitable pour plus d'un objet, puisque dans ce cas, nous voulons prendre en compte à la fois le nombre d'affirmations et la performance des sources sur d'autres objets.

P9 (Claims) *Si* $c_{\mathcal{O}_1}(f) > c_{\mathcal{O}_1}(f')$ *alors* $|src_{\mathcal{O}_1}(f)| > |src_{\mathcal{O}_1}(f')|$

5.2 Propriétés supplémentaires

Les propriétés présentées dans la section précédente sont celles que toute méthode doit satisfaire. Dans cette section, nous donnons des propriétés supplémentaires, qui ne sont pas nécessaires pour toutes les méthodes, mais qui caractérisent des comportements intéressants de certaines méthodes.

Les deux propriétés suivantes sont liées à la propriété (Best). Une source doit (correctement) affirmer tous les faits si elle veut obtenir le score le plus élevé :

P10 (Best *A*) *Soit* $s \in \mathcal{S}$, $r(s) = 1$ *ssi* $fct(s) = B(\mathcal{F})$.

Une alternative consiste à considérer qu'une source est la plus fiable (fiabilité égale à 1) si elle trouve toujours le fait le plus plausible (sans avoir à s'exprimer sur tous les objets) :

P11 (Best *C*) *Soit* $s \in \mathcal{S}$, $r(s) = 1$ *ssi* $fct(s) \subseteq B(\mathcal{F})$.

Notons que (Best *A*) et (Best *C*) impliquent tous deux la propriété (Best).

Si la fiabilité d'une source est égale à 0 (le score le plus bas), cela signifie qu'aucun de ses faits n'est plausible (sur l'objet correspondant) :

P12 (Worst) *Soit* $s \in \mathcal{S}$, $r(s) = 0$ *ssi* $fct(s) \subseteq \mathcal{F} \setminus B(\mathcal{F})$.

Si une source affirme des faits plus crédibles qu'une autre source, la fiabilité de la première source sera meilleure :

P13 (Source Dominance) *Soit deux sources* s *et* s'*, si* $|B(\mathcal{F}) \cap fct(s)| > |B(\mathcal{F}) \cap fct(s')|$ *alors* $r(s) > r(s')$.

Lorsqu'une source s affirme un fait avec une confiance supérieure à celle d'une autre source s' pour chaque objet, alors la fiabilité de s doit être meilleure :

P14 (Pareto) *Soit* $G = (V, E)$ *un graphe complet et* $s, s' \in \mathcal{S}$. *Si* $c(f) > c(f')$ *avec* $f \neq f'$, $f \in fct(s)$, $f' \in fct(s')$ *et* $obj(f) \cap obj(f') = \{o\}$ $\forall o \in \mathcal{O}$ *alors* $r(s) > r(s')$.

5.3 Propriétés discutables

Nous présentons dans cette section certaines propriétés de [28] que nous considérons comme discutables pour toutes les méthodes et nous expliquons pourquoi nous pensons qu'elles ne sont pas satisfaisantes.

La première propriété stipule que les sources qui affirment des faits plus crédibles doivent être plus fiables :

P15 (Source Coherence) *Si* $fct(s1)$ *est moins plausible que* $fct(s2)$ *alors* $r(s1) < r(s2)$.

Notons que la notion de "*moins plausible*" ne nécessite pas que les faits portent sur les mêmes objets. Le problème de cette propriété est que nous comparons des faits qui concernent (potentiellement) des objets différents, alors que l'évaluation des faits est faite pour chaque objet. Par exemple, deux faits peuvent avoir la même confiance, mais l'un est le plus plausible pour son objet, tandis que l'autre est le moins plausible pour un autre objet.

La deuxième propriété stipule que, lorsqu'un fait reçoit un nouveau soutien, son classement doit être strictement meilleur :

P16 (Monotonicity) *Soit G un graphe,* $s \in \mathcal{S}, f \in \mathcal{F} \setminus fct(s)$. *Nous écrivons E pour les arêtes de G, et G' un graphe avec les arêtes* $E' = \{(s, f)\} \cup E \setminus \{(s, g) : g \neq f, obj(g) = obj(f)\}$. *Alors pour tout* $g \neq f, c_G(g) \leq c_G(f)$ *implique* $c_{G'}(g) < c_{G'}(f)$.

Cette propriété ne tient pas compte du reste du graphe et des changements qui peuvent survenir lorsqu'une arête est modifiée. Cette propriété semble être associée à une vision locale du problème, où l'évaluation des faits correspondant à un objet est indépendante des autres objets. Mais

il est important de connaître les performances des sources sur d'autres objets afin de prendre une décision sur un objet donné, et la modification d'une arête sur un objet peut changer la crédibilité des faits sur d'autres objets et la fiabilité de nombreuses sources. Enfin, l'évaluation de l'objet sur lequel le changement a été effectué donnera un résultat différent.

Une autre propriété discutable stipule que la confiance des faits ne doit dépendre que de l'objet auquel ils sont liés. Notons que les auteurs [28] classent également cette propriété comme discutable :

P17 (POI) *Soit G, G' deux graphes et $o \in O$. Nous supposons que $fct_G(o) = fct_{G'}(o)$ et $src_G(f) = src_{G'}(f)$ pour chaque $f \in fct_G(o)$. Alors $c_G(f1) \leq c_G(f2)$ ssi $c_{G'}(f1) \leq c_{G'}(f2)$ pour tous $f1, f2 \in fct_G(o)$.*

Cette propriété pose un problème similaire à la précédente. Il est important d'évaluer les performances des sources sur d'autres objets afin de prendre une décision sur un objet donné, comme illustré dans l'exemple de l'introduction, où l'évaluation des performances sur *Capitale du Brésil* nous aide à prendre une décision sur *Capitale de l'Australie*.

5.4 Propriétés des méthodes S&F

Dans les tableaux et les figures, Pl*A* et Pl*C* correspondent respectivement à la méthode S&F avec la majorité simple et à la normalisation *A* ou *C*. Bo*A* et Bo*C* correspondent aux méthodes avec la règle de Borda et les normalisation *A* et *C*.

Vérifions quelles sont les propriétés satisfaites par nos méthodes. Nous nous concentrons sur les deux normalisations (*C* et *A*), ainsi que sur la majorité simple et la règle de Borda.

Proposition 1 *Pl*A *satisfait (P1-P9), (P10), (P12), (P13) et (P14). Elle ne satisfait pas (P11), (P15-P17).*

Proposition 2 *Pl*C *satisfait (P1-P9), (P11), et (P12). Elle ne satisfait pas (P10), (P13), (P14) et (P15-P17).*

Proposition 3 *Bo*A *satisfait (P1-P9), (P10) et (P14). Elle ne satisfait pas (P11-P13) et (P15-P17).*

Proposition 4 *Bo*C *satisfait (P1-P9), (P11) et (P14). Elle ne satisfait pas (P10), (P12), (P13) et (P15-P17).*

Les résultats sont résumés dans la Table 3 [3]. Tout d'abord, il est important de noter que nos méthodes satisfont toutes les propriétés de base, c'est-à-dire les propriétés attendues pour toutes les méthodes. Il est intéressant de discuter des propriétés qui ne sont satisfaites que par certaines méthodes, afin d'illustrer la différence dans leurs comportements. Tout

3. Les lignes grises de la Table 3 correspondent aux propriétés de [28].

	Pl*A*	Pl*C*	Borda*A*	Borda*C*	
P1 Best	✔	✔	✔	✔	Nécessaire
P2 Null Player	✔	✔	✔	✔	
P3 Groundedness	✔	✔	✔	✔	
P4 Unanimity	✔	✔	✔	✔	
P5 Neutrality	✔	✔	✔	✔	
P6 Independence	✔	✔	✔	✔	
P7 Fact Coherence	✔	✔	✔	✔	
P8 Majority	✔	✔	✔	✔	
P9 Claims	✔	✔	✔	✔	
P10 Best A	✔	✗	✔	✗	Optionnelle
P11 Best C	✗	✔	✗	✔	
P12 Worst	✔	✔	✗	✗	
P13 Source Dominance	✔	✗	✗	✗	
P14 Pareto	✗	✗	✔	✔	
P15 Source Coherence	✗	✗	✗	✗	Indésirable
P16 Monotonicity	✗	✗	✗	✗	
P17 POI	✗	✗	✗	✗	

TABLE 3 – Propriétés satisfaites par les méthodes S&F

d'abord, notons que *Best C* implique d'utiliser la normalisation *C* pour nos méthodes, alors que *Best A* correspond à la normalisation *A*. La propriété *Worst* correspond au comportement de la majorité simple, avec d'autres règles de pondération, elle ne sera pas satisfaite. Inversement, la propriété *Pareto* est liée à la règle de Borda, et n'est pas satisfaite par la majorité simple, qui effectue une évaluation plus drastique des faits. Enfin, *Source dominance* n'est satisfaite que par la majorité simple et la normalisation *A*.

6 Étude Expérimentale

Outre l'évaluation théorique de nos méthodes, nous avons également procédé à une évaluation expérimentale de leurs performances en matière d'identification des faits réels et d'évaluation de la fiabilité des sources. Nous avons mené des expériences à la fois sur des ensembles de données réelles et des ensembles de données synthétiques.

6.1 Données réelles

Nous évaluons nos méthodes sur deux ensembles de données provenant de `http://lunadong.com/fusionDataSets.htm`, à savoir l'ensemble de données *Book* [8, 33] et le jeu de données *Flight* [16, 8].

Nous abrégeons TF pour Truth Finder ([33]), H&A pour Hubs and Authorities ([13]), Usums pour Unbounded-Sums ([28]) et Sums ([22]). P représente la métrique *Precision*, A pour *Accuracy*, R pour *Recall* et C pour *CSI* (Critical Success Index), voir [18, 7] pour plus de détails sur ces mesures.

Book. La difficulté avec cet ensemble de données a été de créer le graphe, car les données nécessitent un traitement de texte. Après le nettoyage des données, le graphe se compose

de 876 sources, 5685 faits et 1263 objets. Le *ground truth* est composé de 100 objets avec un fait réel connu. Nous voyons dans la Table 4 que la méthode S&F avec la majorité simple et la normalisation A est la meilleure méthode avec cet ensemble de données.

	PlA	PlC	BoA	BoC	TF	H&A	Sums	Usums
P	**78.00**	76.00	71.00	76.00	72.00	74.00	74.00	72.00
A	**90.98**	90.16	88.11	90.16	88.52	89.34	89.34	88.52
R	**78.00**	76.00	71.00	76.00	72.00	74.00	74.00	72.00
C	**63.93**	61.29	55.04	61.29	56.25	58.73	58.73	56.25

TABLE 4 – Résultat pour le jeu de données *Book*

	PlA	PlC	BoA	BoC	TF	H&A	Sums	Usums
P	**91.35**	82.34	83.82	81.91	80.36	82.21	82.21	82.79
A	**91.49**	82.61	84.06	82.18	81.72	82.48	82.48	83.05
R	**91,35**	82.34	83.82	81.91	83.22	82.21	82.21	82.79
C	**84,08**	69.98	72.14	69.36	69.15	69.80	69.80	70.63

TABLE 5 – Résultat pour le jeu de données *Flight*

Flight. Pour nettoyer cet ensemble de données, nous avons mis toutes les dates et heures dans le même format. Nous avons supprimé le terminal de la porte car il n'apparaît que quelques fois. Après le nettoyage des données, le graphe se compose de 38 sources, 399 506 faits et 207 912 objets. Le *ground truth* est composé de 16 089 objets avec un fait vrai connu. Nous voyons dans la Table 5 que la méthode avec la majorité simple et la normalisation A est également la meilleure méthode avec cet ensemble de données. Notre méthode surpasse les autres méthodes parce qu'elle parvient à trouver la vérité sur les objets même lorsque la majorité des sources n'affirment pas le vrai fait.

Nous voyons donc sur ces deux ensembles de données réelles que notre méthode S&F surpasse toutes les méthodes existantes de la littérature pour trouver les faits réels pour toutes les mesures de performance (P, A, R, C).

6.2 Données synthétiques

Le nombre limité d'ensembles de données réelles disponibles ne nous permet pas d'évaluer les performances des méthodes dans de nombreuses situations différentes. Nous avons généré des ensembles de données synthétiques pour pouvoir effectuer cette évaluation plus précise.

Tous les graphes générés sont composés de 10 objets et de 4 faits par objet. Pour chaque objet, nous choisissons au hasard l'un des quatre faits comme étant la vérité pour cet objet. Ce sera notre *ground truth* pour évaluer nos méthodes avec les métriques.

Pour chaque source, nous choisissons aléatoirement un nombre d'objets entre 1 et $|O|$ sur lequel cette source affirmera un fait. Pour générer les liens entre les sources et les faits, nous attribuons à chaque source une probabilité p (entre 0.1 et 0.9) de choisir un vrai fait sur chaque objet. Les faux faits ont la probabilité $1 - p$, uniformément distribuée, d'être choisis. Les graphes générés peuvent ne pas être complets, c'est-à-dire que les sources peuvent ne pas affirmer un fait sur chaque objet. Après la génération, nous connaissons la probabilité *a posteriori* de choisir un vrai fait pour toutes les sources. Cette valeur représente la fiabilité réelle de ces sources.

Dans les tests, nous classons les expériences en fonction de la fiabilité moyenne des sources. Nous pouvons voir ce qui se passe lorsque les sources sont globalement plus ou moins fiables. Dans les graphiques, une fiabilité moyenne de x% signifie qu'il y a x% de liens entre les sources et les vrais faits (et $(100 - x)$% de liens entre les sources et les faux faits). Chaque point sur les graphiques correspond à la moyenne obtenue avec la génération de 1000 graphes.

Nous comparons les résultats de nos méthodes aux méthodes de la littérature (Truth Finder, Hubs and Authorities, Sums, Unbounded sums) et Voting. [4]

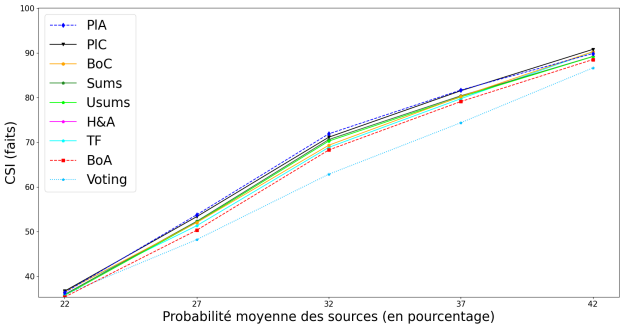


FIGURE 3 – CSI - 10 sources

Confiance des faits - Truth Discovery. Nous voyons (figure 3) que les méthodes S&F avec la majorité simple, et les deux normalisations, sont meilleures pour la métrique *CSI* que les autres méthodes de la littérature lorsque la fiabilité moyenne est supérieure à 27%. Par rapport aux autres méthodes, la méthode de la majorité simple trouve la vérité plus souvent lorsqu'il y a un nombre égal de sources affirmant le vrai et le faux fait. Elle trouve également la vérité lorsqu'une minorité de sources affirme le vrai fait. Il est intéressant de noter que les méthodes donnent de très bons résultats même lorsque la fiabilité moyenne est faible. Nos méthodes obtiennent de bons résultats avec les deux normalisations.

Toutes les méthodes trouvent la vérité lorsque la fiabilité moyenne est supérieure à 57%. Entre 42 et 57%, les résultats sont pratiquement les mêmes pour toutes les méthodes, c'est pourquoi nous ne montrons pas l'ensemble des résultats pour une meilleure lisibilité sur le graphique.

Fiabilité des sources.

Nous avons effectué des expériences avec plusieurs mesures (nombre de swaps, distance euclidienne, etc.), avec des résultats très convaincants, mais nous n'avons pas assez d'espace pour décrire toutes ces mesures, nous nous

4. Nous ne mettons ici que les figures pour CSI, mais nous obtenons des résultats identique pour *Precision*.

s	Probabilité	PlA	BoA	Voting	Sums	TF
s1	0.11	**0.131**	0.35	0.197	0.28	0.75
s2	0.17	**0.187**	0.39	0.249	0.35	0.77
s3	0.21	**0.233**	0.42	0.288	0.41	0.78
s4	0.27	**0.296**	0.46	0.341	0.47	0.81
s5	0.33	**0.341**	0.49	0.386	0.52	0.82
s6	0.39	**0.403**	0.53	0.438	0.58	0.85
s7	0.47	**0.476**	0.58	0.503	0.66	0.86
s8	0.53	**0.528**	0.61	0.549	0.70	0.88
s9	0.57	**0.564**	0.63	0.58	0.74	0.89
s10	0.61	0.591	0.64	**0.603**	0.75	0.90

TABLE 6 – Fiabilité des sources - Fiabilité moyenne de 37%

concentrerons donc sur la différence moyenne : nous calculons la différence moyenne entre la fiabilité calculée et la probabilité (*a posteriori*) de choisir le vrai fait pour chaque objet. Cette distance mesure donc à quel point la fiabilité estimée des sources est proche de la vraie fiabilité (la probabilité *a posteriori*).

Sur la Table 6, nous comparons la fiabilité estimée obtenue avec la fiabilité réelle (probabilité *a posteriori*), pour le cas où la fiabilité moyenne est de 37%. On peut voir que les estimations fournies par la majorité simple sont très proches de la probabilité réelle (rappelons que les résultats sont une moyenne sur 1000 expériences).

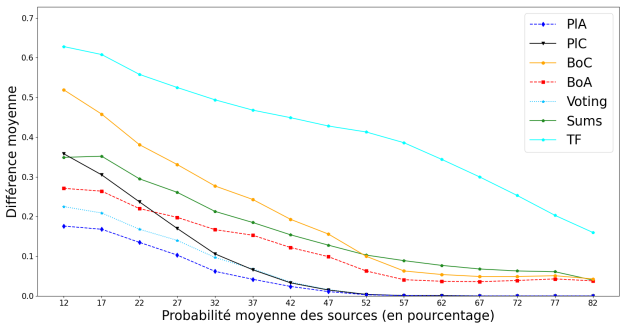


FIGURE 4 – Fiabilité des sources - Différence moyenne - 10 sources

Pour voir les résultats plus globalement, la figure 4 montre l'évolution de la différence moyenne avec différentes fiabilités moyennes pour les sources. La fiabilité estimée des sources est plus proche de la fiabilité réelle lorsque la fiabilité moyenne des sources augmente. Nous voyons que nous obtenons exactement la fiabilité réelle lorsque la fiabilité moyenne est meilleure que 57% pour la méthode utilisant la majorité simple. Avec la règle de Borda, on attribue des points à toutes les sources. C'est pourquoi la fiabilité n'est pas identique à la probabilité *a posteriori*, puisque les sources obtiendront également des points pour les fausses affirmations. Cependant, lorsque la fiabilité moyenne des sources augmente, la différence tend tout de même vers 0.

Nous comparons nos méthodes à celles de la littérature lorsque le score calculé en tant que *fiabilité des sources* est compris entre 0 et 1. Pour *Voting*, nous définissons la fiabilité d'une source comme la proportion d'objets pour lesquels la source affirme le choix de la majorité. Nous ne comparons pas ici le résultat à Unbounded-Sums car le score augmente toujours pour cette méthode. *Voting* donne de bons résultats lorsque les sources sont fiables (fiabilité moyenne supérieure à 57%), mais avant cela, notre méthode avec la majorité simple est meilleure. La méthode itérative permet de trouver les faits réels même lorsque les sources ne sont pas vraiment fiables par rapport à l'utilisation d'une méthode de vote basique.

6.3 Convergence

Nous n'avons pas de preuve de convergence de nos méthodes, mais lors de nos tests, nous avons généré des millions de graphes avec différents paramètres et notre algorithme s'est toujours arrêté rapidement. La Figure 5 donne le nombre maximal d'itérations pour différentes fiabilités moyennes. Ainsi, le nombre maximal d'itérations que nous avons obtenues est de 14, et en moyenne la convergence est obtenue autour de 4 itérations.

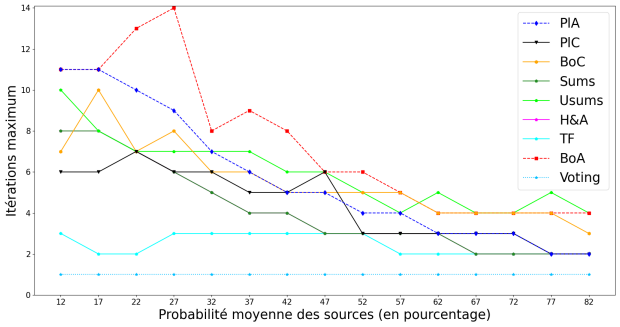


FIGURE 5 – Nombre d'itérations maximum - 10 sources

7 Conclusion

Dans cet article, nous avons présenté les méthodes S&F pour évaluer la fiabilité des sources conjointement à la confiance des faits dans un système multi-agents basé sur l'information. Nous avons proposé et discuté des propriétés que ces méthodes devraient ou pourraient satisfaire. Nous avons vérifié quelles propriétés sont satisfaites par nos méthodes. Nous avons également effectué quelques évaluations expérimentales. Tout d'abord, nous montrons que nos méthodes (en particulier avec la majorité simple) sont plus performantes que les méthodes de la littérature pour identifier les vrais faits sur des données réelles et générées. Mais nous montrons aussi que nos méthodes permettent d'estimer correctement la fiabilité des sources.

Il existe de nombreuses pistes pour les travaux à venir. Les plus directes sont de permettre une certaine similarité (ou dépendance) entre les objets, mais nous pourrions aussi utiliser des sujets (*topics* différents pour nos objets. On peut aussi tenter de prendre en compte des informations a priori sur la fiabilité des sources.

Remerciements

Ce travail a bénéficié du support de la Chaire IA BE4musIA (ANR-20-CHIA-0028).

Références

[1] Artz, Donovan et Yolanda Gil: *A survey of trust in computer science and the semantic web*. Journal of Web Semantics, 5(2) :58–71, 2007.

[2] Austen-Smith, David et Jeffrey Banks: *Information Aggregation, Rationality, and the Condorcet Jury Theorem*. American Political Science Review, 90 :34–45, 1996.

[3] Ben-Yashar, Ruth et Jacob Paroush: *A nonasymptotic Condorcet jury theorem*. Social Choice and Welfare, 17(2) :189–199, 2000.

[4] Ben-Yashar, Ruth et Mor Zahavi: *The Condorcet jury theorem and extension of the franchise with rationally ignorant voters*. Public Choice, 148(3/4) :435–443, 2011.

[5] Berend, Daniel et Jacob Paroush: *When is Condorcet's Jury Theorem valid ?* Social Choice and Welfare, 15(4) :481–488, 1998.

[6] Condorcet, Marquis de: *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie royale Paris, 1785.

[7] Donaldson, Ralph, Rosemary Dyer et Michael Kraus: *An objective evaluator of techniques for predicting severe weather events*. Dans *Preprints, Ninth Conf. on Severe Local Storms, Norman, OK, Amer. Meteor. Soc*, tome 321326, 1975.

[8] Dong, Xin Luna, Barna Saha et Divesh Srivastava: *Less is more : Selecting sources wisely for integration*. Proceedings of the VLDB Endowment, 6(2) :37–48, 2012.

[9] Estlund, David: *Opinion leaders, independence, and Condorcet's Jury Theorem*. Theory and Decision, 36(2) :131–162, 1994.

[10] Everaere, Patricia, Sébastien Konieczny et Pierre Marquis: *The epistemic view of belief merging : can we track the truth ?* Dans *Nineteenth European Conference on Artificial Intelligence (ECAI'10)*, pages 621–626, 2010.

[11] Hammer, Stephan, Michael Wissner et Elisabeth André: *Trust-based decision-making for smart and adaptive environments*. User Modeling and User-Adapted Interaction, 25 :267 – 293, 2015.

[12] Hummel, Patrick: *Jury theorems with multiple alternatives*. Social Choice and Welfare, 34(1) :65–103, 2010.

[13] Kleinberg, Jon M: *Authoritative sources in a hyperlinked environment*. Journal of the ACM (JACM), 46(5) :604–632, 1999.

[14] Ladha, Krishna K.: *The Condorcet Jury Theorem, Free Speech, and Correlated Votes*. American Journal of Political Science, 36(3) :617–634, 1992.

[15] Ladha, Krishna K: *Information pooling through majority-rule voting : Condorcet's jury theorem with correlated votes*. Journal of Economic Behavior & Organization, 26(3) :353 – 372, 1995.

[16] Li, Xian, Xin Luna Dong, Kenneth Lyons, Weiyi Meng et Divesh Srivastava: *Truth finding on the deep web : Is the problem solved ?* Proceedings of the VLDB Endowment, 6(2) :97–108, 2012.

[17] List, Christian et Robert E. Goodin: *Epistemic Democracy : Generalizing the Condorcet Jury Theorem*. Journal of Political Philosophy, 9(3) :277–306, 2001.

[18] Olson, David L et Dursun Delen: *Advanced data mining techniques*. Springer Science & Business Media, 2008, ISBN 978-3-540-76916-3.

[19] Owen, Guillermo, Bernard Grofman et Scott L. Feld: *Proving a distribution-free generalization of the Condorcet Jury Theorem*. Mathematical Social Sciences, 17(1) :1 – 16, 1989.

[20] Parhizkar, Elham, Mohammad Hossein Nikravan, Robert C. Holte et Sandra Zilles: *Combining Direct Trust and Indirect Trust in Multi-Agent Systems*. Dans *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI'20)*, pages 311–317, 2020.

[21] Paroush, Jacob: *Stay away from fair coins : A Condorcet jury theorem*. Social Choice and Welfare, 15(1) :15–20, 1998.

[22] Pasternack, Jeff et Dan Roth: *Knowing what to believe (when you already know something)*. Dans *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 877–885, 2010.

[23] Peleg, Bezalel et Shmuel Zamir: *Extending the Condorcet Jury Theorem to a general dependent jury*. Social Choice and Welfare, 39(1) :91–125, 2012.

[24] Pinyol, Isaac et Jordi Sabater-Mir: *Computational trust and reputation models for open multi-agent systems : a review*. Artificial Intelligence Review, 40 :1–25, 2013.

[25] Rast, Erich: *Theory of Value Structure : From Values to Decisions*. Lexington Books, 2022, ISBN 9781793616951. `https://books.google.fr/books?id=BGRoEAAAQBAJ`.

[26] Sabater, Jordi et Carles Sierra: *Review on computational trust and reputation models*. Artificial intelligence review, 24(1) :33–60, 2005.

[27] Singleton, Joseph et Richard Booth: *An axiomatic approach to truth discovery*. Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, pages 2011–2013, 2020.

[28] Singleton, Joseph et Richard Booth: *Towards an axiomatic approach to truth discovery*. Journal of Autonomous Agents and Multi-Agent Systems, 36(2) :1–49, 2022.

[29] Siqueira Braga, Diego de, Marco Niemann, Bernd Hellingrath et Fernando Buarque de Lima-Neto: *Survey on Computational Trust and Reputation Models*. ACM Computing Surveys (CSUR), 51 :1 – 40, 2018.

[30] Škorić, Boris, Sebastiaan de Hoogh et Nicola Zannone: *Flow-based reputation with uncertainty : evidence-based subjective logic*. International Journal of Information Security, 15 :381–402, 2014.

[31] Vu, Quang Anh Nguyen, Salima Hassas, Frederic Armetta, Benoit Gaudou et Richard Canal: *Combining trust and self-organization for robust maintaining of information coherence in disturbed MAS*. Dans *IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems(SASO'11)*, pages 178–187, 2011.

[32] Waguih, Dalia Attia et Laure Berti-Equille: *Truth discovery algorithms : An experimental evaluation*. arXiv preprint arXiv :1409.6428, 2014.

[33] Yin, Xiaoxin, Jiawei Han et Philip S. Yu: *Truth Discovery with Multiple Conflicting Information Providers on the Web*. IEEE Transactions on Knowledge and Data Engineering, 20(6) :796–808, 2008.

[34] Young, Hobart Peyton: *Condorcet's Theory of Voting*. The American Political Science Review, 82(4) :1231–1244, 1988.

[35] Young, Hobart Peyton et Arthur Levenglick: *A Consistent Extension of Condorcet's Election Principle*. SIAM Journal on Applied Mathematics, 35(2) :285–300, 1978.

Actes JIAF-JFPDA 2023

Encodeur hybride pour la détection automatique de désinformation

Géraud Faye[1,2] **Wassila Ouerdane**[2] **Sylvain Gatepaille**[1] **Guillaume Gadek**[1]
Souhir Gahbiche[1]

[1] Airbus Defence and Space, Élancourt, France
[2] Université Paris-Saclay, CentraleSupélec, MICS, France

geraud.faye@centralesupelec.fr

Résumé

L'encodage de texte est aujourd'hui basé sur de larges modèles de langue entièrement neuronaux, utilisés comme des *boîtes noires*. Afin de répondre au besoin d'explicabilité, nous proposons **CATS** [1] (Cognitive Attention To Syntax), une couche pouvant être utilisée dans les réseaux de neurones qui permet d'introduire du raisonnement syntaxique pour l'encodage et la classification de textes.

Abstract

Today, text encoding relies mostly on foundation models, used as *black boxes*. To bring more transparency, we propose **CATS** (Cognitive Attention To Syntax), a layer that incorporates syntactic reasoning in neural networks for text classification.

1 CATS, un encodeur neurosymbolique

La détection de désinformation à partir de textes uniquement a fait l'objet du développement de nombreux modèles. Les réseaux convolutionnels [3] et les modèles basés sur l'attention [5] ont montré à travers leurs résultats que les caractéristiques stylistiques du document sont discriminantes pour identifier la désinformation. Ces caractéristiques semblent même être partagées par plusieurs types de désinformation (rumeur, fake news, ...) [4]. Afin de créer un modèle plus transparent, nous proposons une approche neurosymbolique qui introduit du raisonnement syntaxique pour l'encodage de la phrase, en s'inspirant du principe de compositionnalité.

Dans un premier temps, les phrases sont analysées syntaxiquement, produisant un arbre semblable à celui de la Figure 1.

1. Une version étendue a été publiée dans les actes de EGC 2023 [2]

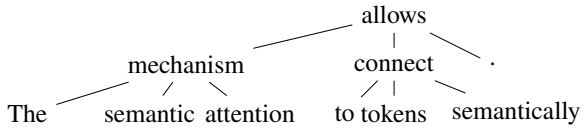


FIGURE 1 – Arbre syntaxique calculé par SpaCy

Cet arbre met en relation les mots entre eux en fonction de leur importance relative. Il est ensuite utilisé pour calculer une matrice d'attention qui relie les mots entre eux du bas vers le haut en fonction de leur distance dans l'arbre syntaxique, donnant la matrice présentée en Figure 2. Cette matrice est ensuite utilisée dans le mécanisme d'attention classique [7]. En retirant les projections vers les espaces *Query*, *Key* et *Value* , nous avons une couche neuronale basée sur du raisonnement symbolique qui permet la propagation des gradients, tout en n'ayant aucun poids entraînable.

2 Évaluation et explicabilité

Cette couche neurosymbolique a été comparée à son équivalent neuronal (noté Standard) dans un modèle utilisant les plongements de fastText [1], suivis de la couche d'attention standard ou de CATS, avant une couche de classification. Ces modèles ont été évalués sur PolitiFact et GossipCop [6] (détection de *fake news* et de rumeurs). Les résultats sont consignés dans la Table 1.

Les modèles basés sur CATS ont des performances légèrement supérieures aux modèles uniquement neuronaux. Mais le grand avantage de CATS est la réduction du besoin en données annotées qui a été mesuré avec des entraîne-

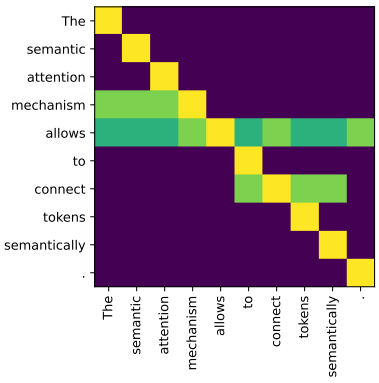


FIGURE 2 – Matrice d'attention correspondant à l'arbre Figure 1. Plus la case est claire, plus le poids correspondant est élevé.

	PolitiFact		GossipCop	
	Fiabilité	F1	Fiabilité	F1
Standard	0.889	0.902	0.727	0. 758
CATS	**0.916**	**0.929**	**0.732**	**0.762**

TABLE 1 – Résultats des différents modèles sur le jeu de données de test.

ments sur un dataset réduit (voir Figure 3). Le modèle neurosymbolique est capable de généraliser avec seulement 50 articles alors que le modèle neuronal n'apprend rien avec si peu de données.

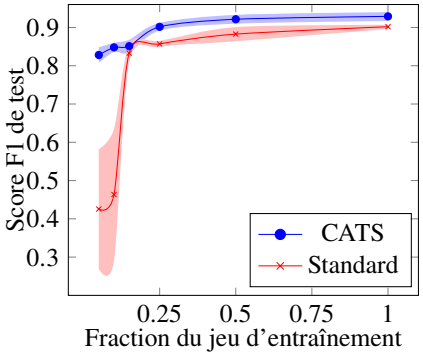


FIGURE 3 – Évolution du score F1 d'un apprentissage sur une portion du jeu de données PolitiFact.

Enfin, car notre matrice d'attention est inversible par construction, on peut calculer la contribution de chaque jeton dans la prédiction. Cela nous a permis de développer un outil identifiant les phrases les plus suspectes dans un texte, afin de faciliter la vérification manuelle des faits ou de mettre en garde les utilisateurs contre les marqueurs probables de désinformation, comme illustré dans la Figure 4.

Le modèle n'identifie pas les faits factuellement faux, mais les expressions et tournures de phrases caractéristiques de la désinformation.

Twelve More Hurricanes Headed Towards US . Twelve More Hurricanes Headed Towards US This is a satirical website . Don ' t take it Seriously . It ' s a joke . Wednesday 06 July 2059 4332 Shares The National Hurricane Center has issued twelve more hurricane warnings for the east coast of the US . " Regardless of which coast you live on , be prepared to evacuate at least twelve times " the National Weather Services said Thursday , not ruling out the possibility of a thirteenth hurricane by the end of the year . This is a satirical website . Don ' t take it Seriously . It ' s a joke . / ! \ Report Abuse loading Biewty

FIGURE 4 – Les indicateurs explicites de satire sont mis en évidence (rouge) pour le lecteur trop hâtif. Les phrases en vert ne contribuent pas à la classe désinformation.

Il serait intéressant par la suite d'incorporer un mécanisme de coréférence, ce qui permettrait de lier les phrases entre elles à partir de leurs entités communes.

Références

[1] Bojanowski, Piotr, Edouard Grave, Armand Joulin et Tomás Mikolov: *Enriching word vectors with subword information*. CoRR, abs/1607.04606, 2016.

[2] Faye, Géraud, Sylvain Gatepaille, Guillaume Gadek et Souhir Gahbiche: *Encodeur hybride pour la détection automatique de désinformation*. Revue des Nouvelles Technologies de l'Information, Extraction et Gestion des Connaissances, RNTI-E-39 :91–102, 2023.

[3] Gadek, Guillaume et Paul Guélorget: *An interpretable model to measure fakeness and emotion in news*. Procedia Computer Science, 176:78–87, 2020.

[4] Lee, Nayeon, Belinda Z. Li, Sinong Wang, Pascale Fung, Hao Ma, Wen tau Yih et Madian Khabsa: *On Unifying Misinformation Detection*, avril 2021.

[5] Pelrine, Kellin, Jacob Danovitch et Reihaneh Rabbany: *The Surprising Performance of Simple Baselines for Misinformation Detection*, avril 2021.

[6] Shu, Kai, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee et Huan Liu: *Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media*. CoRR, abs/1809.01286, 2018.

[7] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser et Illia Polosukhin: *Attention is all you need*. CoRR, abs/1706.03762, 2017.

Actes JIAF-JFPDA 2023

Sourcer, dater et mémoriser les informations d'origine verbale - proposition de modèle symbolique et opérationnel de l'interface langage/mémoire

Raoul Blin

CNRS-CRLAO, France

blin@ehess.fr

Résumé

Nous proposons un modèle symbolique, opérationnel, capable de rendre compte de la façon dont un allocutaire, dans une interaction verbale, exploite les données qui lui sont fournies pour construire ses propres connaissances, et ses croyances sur les connaissances du locuteur («théorie de l'esprit»). Le modèle est formulé en logique classique. Nous illustrons son fonctionnement en prenant exemple sur la psychologie (fausse croyance), l'intelligence artificielle (révision des connaissances) et la linguistique (interprétation de phrases simple et avec verbes épistémiques).

Abstract

We present a symbolic, operational model capable of accounting for the way in which a speaker, in a verbal interaction, exploits the data provided to him to construct his own knowledge, and his beliefs about the speaker's knowledge («theory of mind»). We describe the main features of the model by dealing with cases that concern psychology (false belief), artificial intelligence (knowledge revision) and linguistics (sentence interpretation with epistemic verbs).

1 Introduction

Dans le cadre d'une étude linguistique sur l'interprétation des expressions épistémiques verbale (verbes *savoir*, *croire*, *apprendre que*, etc.), nous recherchions un modèle symbolique capable de rendre compte des états mentaux décrits par ces expressions. Ce modèle devait pouvoir rendre compte au minimum de la distinction entre connaissances propres et connaissances attribuées (théorie de l'esprit [12] et mémoire de source [5]) et de la temporalité des connaissances. Par exemple

(1) Hier encore, Carole ne savait pas [qu'il avait plu avant-hier]$_P$.

indique explicitement à l'allocutaire que d'après le locuteur, la donnée P n'était pas une connaissance de Carole. Il indique implicitement que c'est une connaissance du locuteur. Bien sûr, l'énoncé ne dit rien de l'état des connaissances de l'allocutaire. Par contre, l'énoncé provoquera des modifications des connaissances de l'allocutaire. Temporellement, la date de l'événement décrit par P est distincte de la date de l'énonciation d'une part, et de la période à laquelle le locuteur estime que Carole ignore P. Le locuteur dit explicitement que cette période d'ignorance inclut hier, mais il ne dit pas si cette ignorance est toujours en cours au moment de l'énonciation. Il indique aussi implicitement que P est une donnée que lui-même tient présentement pour valide. Ces inférences ne constituent qu'une partie de tout ce qui peut être inféré par l'allocutaire sur les connaissances du locuteur. Nous en verrons bien d'autres.

Nous nous sommes naturellement tourné vers les logiques modales mais celles-ci se sont révélées frustrantes. Leur spécialisation (épistémique, déontique, temporelle, etc) ne les destine pas *a priori* à traiter en même temps deux phénomènes aussi éloignés que la théorie de l'esprit d'une part et la temporalité d'autre part. Cela ne signifie pas que c'est infaisable. Mais pour l'instant, à notre connaissance, personne n'a proposé de tel modèle modal et montré l'intérêt de l'approche modale par rapport à d'autres logiques pour traiter cette combinaison de phénomènes. Nous nous sommes plus particulièrement intéressé à la logique épistémique mais elle est à bien des égards peu réaliste d'un point de vue cognitif. Nous mentionnerons quelques problèmes dans cette présentation (ce qui ne nous empêche pas par ailleurs d'en reprendre certaines idées).

Nous avons aussi cherché du côté des modèles en intelligence artificielle symbolique. Les recherches

proposent des architectures cognitives (une revue entre autres dans [11], [3], etc.) mais pas de modèles qui manipulent ensemble les données que nous traitons, à savoir théorie de l'esprit et temporalité. Quelques études (par exemple [13]) se sont intéressées au langage mais sont difficiles à exploiter pour une étude linguistique fine.

Finalement, ne trouvant pas de modèle prêt à l'emploi, nous avons décidé de créer notre propre modèle avec le cahier des charges suivant. Le modèle doit être exploitable aussi bien en linguistique qu'en psychologie. Il doit donc couvrir une vaste plage de phénomènes et être révisable en vue d'en intégrer de nouveaux. Il doit aussi pouvoir être intégré dans des modèles de portée plus vaste encore. Pour répondre à ces contraintes, nous avons décidé de décrire le modèle à l'aide d'un langage simplement typé. Le fonctionnement du modèle, les données et les règles sont explicitement données formulées. Elles ne sont pas tenues pour des axiomes d'un «nouveau» système d'inférences et ne requiert pas de nouveau modèle sémantique.

Le présent texte décrit le modèle issu de nos observations. Tout d'abord, nous introduisons une première version du modèle, capable de distinguer les connaissances propres d'un agent et ses croyances sur les connaissances d'autrui. A l'aide de ce modèle, nous simulons l'état de fausse croyance, phénomène très étudié en psychologie. Puis nous introduisons la temporalité. Nous pouvons alors simuler l'évolution des connaissances chez un individu (enrichissement, modification). Le modèle permet aussi de décrire l'acquisition de connaissances d'origine verbale. Nous l'utiliserons pour décrire le traitement «mémoriel» d'énoncés simples avec et sans expression verbale épistémique (*savoir que*).

Nous conviendrons de désigner par «expression» toute chaîne (phrase, syntagme, mot, etc) bien formée du langage naturel. Dans les exemples les expressions entre crochets ⟨...⟩ sont données à titre indicatif, pour faciliter la compréhension. Deux points d'interrogations ?? indiquent que l'expression est peu naturelle ou difficile à interpréter. Un «terme» désigne toute formule logique bien formée. Dans les termes, les symboles non quantifiés explicitement sont des constantes. Les connecteurs logiques sont associatifs à droite.

Nous utiliserons les termes de «mémoire» en prenant peut-être quelques libertés par rapport à l'usage qui peut en être fait dans les domaines de spécialités. Employé dans la discussion, il s'agit de l'ensemble des connaissances ou état mental d'un agent. Nous l'utiliserons aussi pour décrire la base de connaissances utilisée pour modéliser l'ensemble des connaissances d'un agent. Le terme «mémoriser» est employé dans le même esprit.

2 La structure générale du modèle

Pour interagir verbalement avec autrui (transmettre ou recevoir des informations, provoquer chez autrui une réaction), tout agent met en oeuvre ses propres connaissances, et celles qu'il attribue à autrui. Pour décrire l'état mental d'un agent au cours d'une interaction verbale, il faut se donner un modèle de son système de connaissances et des opérations qu'il peut faire sur ses connaissances, et y intégrer la distinction entre connaissances propres et attribuées. Dans cette section, nous jetons les bases du modèle du système de connaissances et montrons comment il permet de simuler la théorie de l'esprit et le cas particulier de fausse croyance. La principale caractéristique du modèle est d'utiliser une base de connaissances stratifiée de façon récursive.

2.1 Les éléments de base

De façon très classique, nous simulons l'état des connaissances (nous simplifierons parfois en parlant de «mémoire») d'un individu, à l'aide d'une base de connaissances.

Avec le prédicat BC, on signifie qu'une donnée P (représentée par une proposition p) figure parmi les connaissances d'un agent A (représenté par la constante individuelle a), ou bien que cette donnée est inférable à partir des connaissances de A : $\text{BC}(a, p)$. A est le «propriétaire» de la connaissance. La négation $\neg\text{BC}(a, p)$ signifie que l'agent ne possède pas l'information ni ne peut l'inférer. On prêtera attention au fait que $\text{BC}(a, \neg p)$ signifie que $\neg p$ est une connaissance de A. Cela ne signifie pas que A ne possède pas la connaissance $\neg p$.

Attribuer une connaissance P à un agent A et P' à un agent B se note simplement : $\text{BC}(a, p) \wedge \text{BC}(b, p')$, ou plus classiquement $\{\text{BC}(a, p), \text{BC}(b, p')\}$. Désormais les crochets seront omis. On glosera un terme $\text{BC}(a, p)$ «p est une connaissance de a» ou plus simplement, si aucune confusion entre glose et langue décrite n'est possible, «A sait que P».

On se dote de plusieurs règles rendant compte de raisonnements triviaux, à commencer par la «distributivité des connaissances pour la conjonction». Le dialogue suivant montre que les deux données d'une conjonction peuvent être séparées en deux connaissances.

(2) Lucie à Alain : Carole sait qu'[[il a neigé avant-hier]$_{P^1}$ et [plu hier]$_{P^2}$]
Eric à Alain : Carole sait-elle qu'[il a neigé avant-hier]$_{P^1}$?
Alain : oui
Eric à Alain : sait-elle qu'[il a plu hier]$_{P^2}$?
Alain : aussi oui

Un dialogue similaire montrerait qu'à partir des connaissances séparées des deux données, on peut inférer

la connaissance de la conjonction des deux données. Nous rendons compte de cette relation entre connaissances :

(R^1) Distributivité des connaissances pour la conjonction (version 1)
$\forall x \forall p \forall p'$
$\text{BC}(x, p \ \wedge \ p') \equiv \text{BC}(x, p) \ \wedge \ \text{BC}(x, p')$

Il est difficile de créer ce type de dialogue pour la disjonction, ce qui nous amène à poser que la distributivité n'est pas valide pour la disjonction.

La notation n'impose aucune contrainte sur le terme p. Il est possible d'exprimer aussi bien des faits que des règles :

(3) Alain sait qu'[il a plu hier].
$\text{BC}(a, pleuvoir(e) \ \wedge \ date(e) = hier)$
Alain sait que [quand il pleut, il ne neige pas].
$\text{BC}(a, \forall e^1 \ pleuvoir(e^1) \supset \neg \exists e^2 \ neiger(e^2))$

Pour signifier que deux agents A et B partagent une même donnée P, il suffit de noter : $\text{BC}(a, p) \wedge \text{BC}(b, p)$. On pourrait envisager une notation plus compacte comme $\text{BC}(\{a, b\}, p)$ mais ici nous n'abordons pas les questions d'optimisation de l'écriture. Pour des questions de place, nous n'abordons pas non plus le traitement des connaissances d'un groupe d'agents.

2.2 Théorie de l'esprit

En psychologie, la «théorie de l'esprit» désigne la capacité d'un individu à attribuer des états mentaux à autrui, parmi lesquels, des connaissances. C'est une compétence cognitive importante dans les interactions, y compris les interactions verbales ([9],[2]). Nous montrons comment notre modèle rend compte de l'attribution de connaissances.

Le fait qu'un «agent A sache qu'un agent B sait P» est rendu en jouant sur la récurrence :

$\text{BC}(a, \text{BC}(b, p))$

Nous utiliserons l'abréviation $\text{BC}^n(x^n, p)$ pour désigner le terme $\text{BC}(x^1, \text{BC}(x^2, ..., \text{BC}(x^n, p)...))$ lorsque x^1 désigne l'individu observé et qu'il n'est pas nécessaire de préciser l'identité de $x^2 ... x^{n-1}$. n est la «profondeur». Le n-uplet $(x^1, ..., x^n)$ est la «chaîne de propriétaires». Nous appellerons «sous-base» tout ensemble de données $\lambda p \ \text{BC}^n(x^n, p)$ pour une chaîne de propriétaires donnée.

Formellement, la notation ne restreint pas la profondeur. Il est cependant clair que dans les faits, les capacités cognitives d'un humain ne permettent pas une profondeur infinie. C'est perceptible à travers les limites imposées par la langue. Au delà de deux itérations, les phrases qui rendent compte de l'état des connaissances sur autrui deviennent difficiles à interpréter (k la constante correspondant à Carole) :

(4) ?? Alain : ⟨je sais que⟩ Lucie sait que Carole sait que [Martine sait qu'il a plu]$_P$.
$\text{BC}(a, \text{BC}(l, \text{BC}(k, \text{BC}(m, p))))$

Formellement, il n'y a pas non plus de restrictions sur les agents. Un agent peut apparaître plusieurs fois dans une chaîne de propriétaires. Ainsi, un agent A peut avoir des croyances sur les croyances d'autrui (B) sur lui-même (A) : « ⟨je (=A) sais que⟩ B sait que je (=A) sais que P>> :

$\text{BC}(a, \text{BC}(b, (\text{BC}, a, p)))$

La notation permet de retranscrire l'axiome **4** ($\Box P \supset \Box\Box P$), dit d'instrospection positive, de la logique épistémique.

«si X sait P, il sait qu'il sait P»
$\forall x \forall p \ \text{BC}(x, p) \supset \text{BC}(x, \text{BC}(x, p))$

Pour autant, nous n'inscrirons pas cet axiome dans notre propre propre ensemble de règles car on peut douter qu'il s'applique à tout individu. Peut-on vraiment considérer qu'il est valable pour un enfant en bas âge par exemple ? Par ailleurs, dans quelle mesure un adulte, même de développement typique, est systématiquement conscient de ce qu'il sait ? Beaucoup de connaissances sont utilisées «inconsciemment». Ce qui soulève d'ailleurs la question de conscience : M.Jourdain «sait» faire de la prose sans en être conscient. L'axiome de la logique épistémique n'est pas claire sur ce point. Pour notre part, la présence d'une donnée dans la base signifie que cette donnée est mémorisée, pas que son propriétaire est nécessairement conscient de posséder cette donnée. Il nous semble préférable de laisser à l'observateur le soin de décider d'inscrire ou non cet axiome dans sa propre axiomatique, selon les caractéristiques de l'agent à modéliser (agent adulte de développement typique, enfant, etc.)

Remarquons ici qu'en ne recourant pas à une logique modale donnée avec son axiomatique propre, le modèle est d'un usage plus souple. On peut modéliser le comportement de deux agents de capacités différentes sans avoir à adopter un système logique pour l'un (avec axiome **4** par exemple) et un autre système (sans l'axiome) pour l'autre agent. Dans notre cas, le système axiomatique est unique et commun aux agents. Le «réglage individuel» se fait par adjonction de règles faciles à cibler.

Bien que ce soit techniquement faisable, nous n'adoptons pas l'axiome **5**, dit d'introspection négative, de la logique épistémique.

«si X ne sait pas que P, il sait qu'il ne sait pas que P»
$\forall x \forall p \ \neg\text{BC}(x, p) \supset \text{BC}(x, \neg\text{BC}(x, p))$

Nous postulons que cet axiome n'est cognitivement pas réaliste. Prenons un exemple. Le concept «La physique quantique est hilarante» est absente de l'esprit d'un enfant de quatre ans. Un enfant de cet âge n'a jamais entendu parler

de physique quantique, il n'y a jamais pensé, et a fortiori il n'a jamais conçu l'idée que cette «chose» pouvait être hilarante. Il n'est à aucun moment en mesure de savoir qu'il ne sait pas.

Insistons enfin sur le cloisonnement des données. La théorie de l'esprit nécessite de bien séparer les connaissances propres et attribuées. Avec notre notation, les connaissances de A ($\lambda p\ \text{BC}(a,p)$) sont strictement séparées des connaissances que A attribue à B ($\lambda p\ \text{BC}(a,\text{BC}(b,p))$). Néanmoins, le partage de données entre sous-bases n'est pas interdit. Simplement, il ne peut se faire que lorsqu'il est explicitement autorisé. Ainsi, lorsqu'un agent A accorde du crédit à un autre agent B, il peut faire siennes les données de l'agent B. Nous parlerons «d'appropriation des connaissances». Si par exemple on voulait rendre compte du fait que les enfants adoptent les connaissances de leurs parents, nous pourrions écrire :

$$\forall x \forall y\ enfant(y) \ \wedge\ parentde(x,y) \\ \supset (\forall p\ \text{BC}(y,\text{BC}(x,p)) \supset \text{BC}(y,p))$$

La théorie de l'esprit ne concerne pas seulement des connaissances statiques. Elle touche aussi les raisonnements : un agent doué de théorie de l'esprit distingue ses propres raisonnements des raisonnements qu'il attribue à autrui. La règle suivante rend compte de cette faculté en permettant des raisonnements (modus ponens) à l'intérieur d'une sous-base. Cette règle correspond à l'axiome **K** dit «de distribution» de la logique épistémique.

(R^2) Règle de distribution (ou $\mathcal{MP}$), version 1

$$\frac{\text{BC}^n(x^n,\alpha) \qquad \text{BC}^n(x^n,\alpha\supset\beta)}{\text{BC}^n(x^n,\beta)}$$

3 Fausse croyance

Le phénomène de fausse croyance ([14]) joue un rôle essentiel dans les études sur la théorie de l'esprit, à travers les épreuves dites de fausse croyance ([1]).

La fausse croyance désigne une situation où un agent observé (A) considère qu'un agent (autre que lui-même) (B) a une connaissance erronée, c'est à dire une connaissance en contradiction avec ses (=A) propres connaissances. Ce que l'agent A pourrait décrire [1] « ⟨je sais que⟩ B sait que P tandis que moi-même je sais que non-P» :

$$\text{BC}(a,p) \ \wedge\ \text{BC}(a,\text{BC}(b,p))$$

Le phénomène peut avoir lieu à n'importe quel niveau d'itération, comme le montre la phrase suivante. Alain sait que Lucie attribue à Carole une fausse croyance, donc une connaissance qu'il sait erronée aux yeux de Lucie.

(5) Alain, Lucie sait que Carole croit à tord que P.
$\text{BC}(a,\text{BC}(l,\neg p)\ \wedge\ \text{BC}(l,\text{BC}(k,p)))$

1. Rappelons et insistons sur le fait que dans les gloses, «savoir que P» et «croire que P» signifient «P est une connaissance» ou «P est dans la base de connaissance».

Voici une formulation très générale, indépendante du niveau d'itération, de la situation où un agent x^n attribue une fausse croyance p à un agent x^{n+1} : Cette représentation ne dit pas si les autres agents de la chaîne de propriétaires (en particulier l'agent x^1) accordent ou non du crédit à la croyance P.

(Hyp) Fausse croyance attribuée par x^n à x^{n+1}
$\text{BC}^n(x^n,\text{BC}(x^{n+1},p)) \ \wedge\ \text{BC}^n(x^n,\neg p)$

Voici à titre indicatif la notation pour un cas particulier de fausse croyance où un agent A sait qu'un agent B lui (=A) attribue une croyance erronée, c'est à dire en contradiction ce que A sait :

(6) Alain^A : Lucie^B croit (à tord) que je^A crois que P (alors qu'en réalité, je sais que non-P)
$\text{BC}(a,\neg p)\ \wedge\ \text{BC}(a,\text{BC}(b,\text{BC}(a,p)))$

4 Détection d'un mensonge

On peut aussi modéliser la capacité à détecter un mensonge. Considérons le propos suivant :

(7) Lucie^B à Alain^A : Il a plu hier. (P)

Pour qu'un agent A considère P comme un mensonge, il faut qu'il sache lui-même que P n'est pas vrai (c^1 dans l'hypothèse ci-dessous). Cela ne suffit pas pour distinguer une simple erreur et un mensonge. Il faut en plus que l'allocutaire détecte une intentionnalité de la part du locuteur. «L'intentionnalité» n'est pas un concept aisé à définir et nous nous garderons de le faire. Par contre, on peut utiliser des indices. A peut suspecter le locuteur B de mentir si il sait que B tient P pour faux (c^2) (« ⟨je sais que⟩ B dit P alors qu'il sait non-P»). Enfin, on ment en général à quelqu'un qui ignore la vérité (c^3). L'allocutaire peut donc être alerté si il sait que le locuteur le croit ignorant («le locuteur croit que j'ignore la vérité, c'est-à-dire qu'il croit que je ne sais pas non-P, ou il croit que je crois P»).

En résumé :

(Hyp) Mensonge
Un agent A peut soupçonner un agent B de mentir si B tient le propos P et que A est dans l'état mental
$\text{BC}(a,\neg p)$ (c^1)
$\text{BC}(a,\text{BC}(b,\neg p))$ (c^2)
$\text{BC}(a,\text{BC}(b,\neg\text{BC}(a,\neg p))\ \vee\ \text{BC}(a,p)))$ (c^3)

5 Datation des connaissances

Les connaissances des individus évoluent avec le temps. Un individu de développement typique est capable de mémoriser une bonne partie de ses connaissances passées, sans les confondre avec celles présentes. Le langage naturel met d'ailleurs à disposition de nombreuses expressions pour rendre compte de l'état des connaissances passées :

(8) Hier je pensais qu'[avant hier il avait plu]$_P$ ⟨mais⟩ j'ai appris ce matin qu'en fait, [il avait neigé]$_{P'}$. Je sais donc désormais qu' [il n'avait pas plu]$_{non-P}$.

Cet énoncé peut se gloser «P était une de mes connaissances jusqu'à ce matin. Depuis ce matin, P n'est plus une connaissance valide mais elle ne disparaît pas de ma mémoire pour autant. Par ailleurs, ce matin, j'ai acquis une nouvelle connaissance P'. Cette connaissance est toujours valide présentement. Depuis l'instant où P' est devenue valide, je suis capable d'inférer non-P.»

Mémoriser la chronologie des connaissances étant nécessaire pour une cognition efficiente ([6]), il faut l'intégrer à notre modèle. Dans un premier temps, nous décrivons le dispositif formel. Puis nous montrons comment la datation offre une solution simple pour gérer les mises à jour des (sous-)bases de connaissances et pour rendre compte de l'évolution des connaissances.

5.1 Modéliser la datation des connaissances

On adopte une indexation des connaissances en s'inspirant de la notation néo-davidsonienne ([10]) utilisée en sémantique des langues. Cela nous permet d'utiliser un langage commun pour décrire les connaissances et les représentations sémantiques. L'interfaçage entre données linguistiques et cognition en est facilité.

Désormais, BC est un prédicat à trois arguments. Le terme $\text{BC}(e, a, p)$ signifie qu'une donnée P figure dans les connaissances de l'agent A. e est l'index de cette connaissance. L'itération reste bien sûr possible : $\text{BC}(e^1, x^1, \text{BC}(e^2, x^2, ...\text{BC}(e^n, x^n, p)))$. $(e^1, ..., e^n)$ est une «chaîne d'index».

L'index obéit à la règle d'unicité (règle R^3), classique dans la notation néo-davidsonienne. Nous la formulons informellement :

> (R^3) Unicité de l'index
> Chaque connaissance, quel que soit le niveau d'itération, dispose d'un index (symbole individuel e^i) qui lui est propre.

Cela signifie que pour une chaîne d'index, une chaîne de propriétaires et une donnée particulières, l'index est unique. A l'inverse, à un index ne peut correspondre qu'un seul triplet chaîne d'index, chaîne de propriétaires et donnée.

On impose à toute connaissance d'avoir une date ($date$) «d'activité» (terme que l'on préférera à celui de «validité», susceptible d'entraîner des confusions).

(R^4) Obligation de temporalité
$\forall e \forall x \forall p \ \text{BC}^n(e^n, x^n, p) \supset \exists t \ date(e) = t$

Toute date d'activité a un début ($deb(e)$) et une fin ($fin(e)$). La fonction $maint()$ décrit «l'instant présent». L'abréviation $actuel$ signifie qu'une connaissance est active à l'instant présent :

$actuel = \lambda e \ maint() \in date(e)$

Par exemple, rendons compte d'une connaissance de P que Lucie peut gloser comme suit, à la date $maint() = t$.

(9) Lucie : Je sais depuis ce matin qu'[il a neigé avant-hier]$_P$

$$\begin{array}{ll} neiger(e^1) \ \wedge \ date(e^1) = avthier & (p) \\ \text{BC}(l, e^2, p) \ \wedge \ deb(e^2) = cematin & (c^1) \\ actuel(e^2) & (c^2) \end{array}$$

Le terme c^1 rend compte de la présence de P dans les connaissances de Lucie et indique le début d'activité de cette connaissance. c^2 rend compte du fait que la connaissance est toujours active au moment présent. La règle R^4 dit que cette connaissance aura une fin mais celle-ci n'est pas spécifiée. Une règle pourrait identifier par défaut la fin de toute connaissance et la disparition (physique) du propriétaire. Mais ce genre de considération dépasse le cadre de notre étude.

Il n'existe pas de contraintes sur la relation temporelle entre les dates td «contenues» dans une donnée et la date d'activité ta de la connaissance de cette donnée. Ainsi, on peut acquérir aujourd'hui (ta) une donnée relative à un événement qui a eu lieu à la date antérieure td. C'est le cas de l'exemple ex.9 où l'événement décrit a lieu avant hier, et sa connaissance débute ce matin.

La datation permet de faire coexister chez un agent des connaissances contradictoires dès lors qu'elles sont actives à des dates différentes. Par exemple, (1) «jusqu'à hier soir, je croyais qu'[il avait neigé avant-hier]$_P$.» (nous reprenons la représentation précédente de P). Puis (2) «Et soudainement, hier soir, j'ai réalisé qu'[il n'avait pas du tout neigé]$_{non-P}$». Autrement-dit, jusqu'à hier soir, ma connaissance de P était active. Depuis hier soir, elle est désactivée, et c'est ma connaissance de non-P qui est active au moment ou je parle.

1) $\text{BC}(l, e^1, a, p) \ \wedge \ fin(e^1) \in hiersoir$
2) $\text{BC}(l, e^2, a, \neg p) \wedge deb(e^1) = fin(e^1) \wedge actuel(e^2)$

Il faut adapter les règles d'inférence pour prendre en compte la temporalité. Pour appliquer le modus ponens sur deux connaissances, il faut, comme auparavant, que ces connaissances aient la même profondeur (nombre d'itérations) et la même chaîne de propriétaires. Nous imposons en plus que les connaissances aient une date commune. La connaissance inférée sera active durant la date commune. Par exemple, je connais depuis mon enfance la règle que si il pleut il ne neige pas et vice versa (P). Par ailleurs, j'ai appris ce matin (P') qu'il avait plu avant hier. Je peux donc, depuis ce matin, c'est à dire dès que les deux connaissances sont actives en même temps, inférer (P") qu'il n'a pas neigé avant hier. Je n'aurai pas été capable de faire cette inférence avant l'acquisition de la connaissance de P'. On rend compte

formellement des contraintes temporelles en n'activant un modus ponens que pour l'intervalle d'activité commun aux deux connaissances manipulées. Automatiquement, le modus ponens sera empêché si l'intersection des dates est nulle. Cet empêchement est déclenché par la règle (R^4), qui impose à toute connaissance d'avoir une date d'activité. Voici la reformulation de la règle de distribution (R^2).

(R^5) Règle de distribution (ou $\mathcal{MP}$) (version 2)
$$\frac{\text{BC}^n(x^n,e^1,\alpha) \qquad \text{BC}^n(x^n,e^2,\alpha\supset\beta)}{\text{BC}^n(x^n,e^3,\beta) \wedge date(e^3)=date(e^1)\cap date(e^2)}$$

Il faut noter qu'avec cette règle, la date de réalisation d'un raisonnement est indépendante de la date d'activité des connaissances de ces données. On peut ainsi reconstituer des raisonnements passés, mais sans en tirer des connaissances actives au moment où est réalisé le raisonnement. Pour en rendre compte, il faut que les connaissances inférées soient activées à la même date que les connaissances utilisées, et non pas à la date de réalisation du raisonnement. Il n'y a donc pas de risque d'anachronisme. Dans l'exemple qui suit, la connaissance P^1 n'est désormais plus active. Le locuteur peut l'exploiter mais la période d'activité de sa conclusion P^3 sera identique à celle de la connaissance de P^1. Il n'y a pas de conflit entre P^3 et P^4 parce qu'au moment (actuel) où P^4 est active, P^3 ne l'est plus.

(10) Avant, [lorsque je voyais un sol mouillé, je pensais que c'était à cause de la pluie]$_{P^1}$. C'est pourquoi [en voyant ta terrasse mouillée]$_{P^2}$, j'avais pensé qu'[il avait plu]$_{P^3}$. Mais je sais désormais que [c'est mouillé par les arrosoirs]$_{P^4}$.

Il faut aussi adapter la règle de distributivité par rapport à la conjonction (R^6). A partir des deux connaissances suivantes, acquises à des dates différentes mais toujours en vigueur au moment de l'énonciation :

(11) Alain : J'ai su avant hier qu'[il était parti en mars]$_{P^1}$ et aujourd'hui qu'[il est revenu en février]$_{P^2}$.
$\text{BC}(a, e^1, p^1) \wedge \text{BC}(a, e^2, p^2)$
$\wedge\ deb(e^1) = avthier \wedge deb(e^2) = aujourdhui$
$\wedge\ actuel(e^1) \wedge actuel(e^2)$

on peut inférer

$\Rightarrow$ présentement, je sais [qu'il est parti en mars et revenu en février].
$\text{BC}(a, e^3, p^1 \wedge p^2) \wedge actuel(e^3)$

Pour réécrire la règle, il faut déterminer la date d'activité de la connaissance de la conjonction des données. On serait tenté d'identifier son début et celui de la connaissance acquise en dernier. Nous n'avons pas trouvé de tests capables de confirmer cette hypothèse. L'hypothèse la plus prudente nous semble au final d'identifier la date d'activité de la connaissance d'une conjonction de connaissances à l'intersection des dates des connaissances d'origine. Cela nous semble valide aussi pour la datation de la connaissance de la conjonction, inférée à partir des connaissances distinctes des données.

(R^6) Distributivité de la conjonction
par rapport aux connaissances (version 2)
$\forall x \forall p \forall p'\ (\exists e^1\ \text{BC}^n(x^n, e^1, p \wedge p'))$
$\equiv \exists e^2 \exists e^3\ \text{BC}^n(x^n, e^2, p) \wedge \text{BC}(x^n, e^3, p')$
$\wedge\ date(e^1) = date(e^2) \cap date(e^3)$

R^6 permet fusionner des informations redondantes. Lorsqu'une information est acquise deux fois ou plus (1), on ne peut pas pour autant considérer qu'elle génère deux connaissances distinctes. La règle permet de fusionner les données (2) :

(1) $\text{BC}(x, e^1, p) \wedge \text{BC}(x, e^2, p)$
(2) $(1), Ex.6 \vdash \exists e^3\ \text{BC}(x, e^3, p \wedge p)$

6 Mise à jour des connaissances

Avec la datation, mettre à jour les connaissances (expansion, révision) consiste à clôturer et/ou faire débuter la date d'activité d'une connaissance.

L'expansion consiste à ajouter une nouvelle connaissance et en faire débuter l'activité à la date d'acquisition. Par exemple, pour rendre compte du fait qu'à la date t, Lucie apprend P, on note :

$$\text{BC}(l, e, p) \wedge deb(e) = t$$

Un connaissance est révisée lorsqu'une donnée est changée en partie ou complètement. Ce qui est important, c'est que la connaissance de la donnée reste dans la base et qu'il sera toujours possible d'y faire référence («Autrefois, je croyais que ...»). Une révision complète consiste à clôturer l'ancienne connaissance et à insérer la nouvelle connaissance. Par exemple, Lucie (c^1) croyait que Martine et Alain étaient ses voisins. A la date t elle apprend que ce n'est pas le cas. On (c^2) clôture donc c^1 et ajoute (c^3) la nouvelle connaissance que l'on fait débuter à t.

$\text{BC}(l, e^1, voisins(l) = \{m, a\})$ (c^1)
$fin(e^1) = t$ (c^2)
$\text{BC}(l, e^2, \neg(voisin(l) = \{m, a\}) \wedge deb(e^2) = t$ (c^3)

L'exemple (ex.8) provoque aussi une révision complète.

Les révisions partielles sont traitées comme des révisions complètes. Par exemple, Lucie pensait jusqu'à la date t que ses voisins étaient Martine et Alain. Finalement elle apprend à la date t que ses voisins sont en fait Martine et Carole. La donnée a partiellement changé puisque Carole remplace Alain, mais Martine ne change pas. Comme précédemment, la connaissance antérieure est clôturée et la nouvelle connaissance est ajoutée et démarre au même moment.

$\text{BC}(l, e^1, voisins(l) = \{m, a\})$ (c^1)
$fin(e^1) = t$ (c^2)
$\text{BC}(l, e^2, \neg(voisin(l) = \{m, k\}) \wedge deb(e^2) = t$ (c^3)

En définitive, pour l'expansion comme pour la révision, il faut ajouter et démarrer une nouvelle connaissance. La révision nécessite en plus de clôturer une connaissance active et d'identifier la date de fin de l'ancienne connaissance avec la date de début de la nouvelle. Pour des raisons de clarté, nous avions volontairement distingué les deux modes de modification de base mais ce n'est peut-être pas pertinent. En effet, tout ajout de connaissance est susceptible d'entraîner une révision.

Même si la mise à jour de la base est grandement facilitée par le jeu sur les datations, l'opération n'est probablement pas simple à implémenter. La principale difficulté que nous voyons est de savoir quelles connaissances clôturer lors d'une révision. En effet, lorsqu'une nouvelle information est fournie, celle-ci n'est pas livrée avec la liste des informations à clôturer. Il faut donc disposer d'un mécanisme de détermination des connaissances à clôturer.

Se pose aussi la question de savoir à quel moment s'opère la clôture lors d'une révision : au moment de l'acquisition de la nouvelle connaissance, ou à mesure que l'on rencontre des données qui sont contredites par la nouvelle information. Dans la réalité, les conséquence d'une nouvelle connaissances peuvent apparaître bien après l'acquisition. Il paraît donc plus vraisemblable que la clôture s'étale dans le temps et qu'une distinction se fasse selon la «familiarité» des connaissances : une connaissance très sollicitée sera peut-être vérifiée plus rapidement que d'autres moins sollicitées. Une étude cognitive reste à réaliser avant d'implémenter une procédure dans le modèle.

Enfin, il est un cas que nous ne travaillerons pas ici, c'est l'oubli. Intuitivement, il s'agirait d'un cas de rétractation pure et simple. Nous postulons que la modélisation du phénomène d'oubli est différent de ce que nous venons de voir, puisque typiquement, on ne garde pas trace de la connaissance ancienne. Cela a certainement des conséquences sur les connaissances inférées. C'est là un sujet délicat dont l'analyse dépasse le présent projet de présentation du modèle. Nous laissons la modélisation de l'oubli à des études ultérieures.

7 Datation des connaissances et temps réel

Un agent est plongé dans le temps, et quand bien même il ne fait rien (ce qui en soi est quelque chose, que l'on peut même désigner verbalement (dans un langage relâché) : «je glande depuis 10 mn»), sa base de connaissances est mise à jour en permanence («je ne fais rien depuis 1mn, je ne fais rien depuis 2mn, ...»). Il faut supposer qu'à chaque unité de temps, la mémoire est mise à jour et que toutes les données qui n'ont pas été clôturées doivent être signalées comme actives.

Il y a là une difficulté notoire. Dans un modèle «à l'échelle», il est hors de question de passer en revue les milliards de connaissances et de notifier expressément la prolongation de leur activité pour chacune d'elles. On devine par ailleurs que c'est une procédure qui se fait naturellement chez l'humain. Mais avec la logique classique que nous nous sommes donnée, il n'y a pas moyen d'indiquer que «par défaut» les connaissances qui n'ont pas été clôturée sont toujours active. Nous ne voyons pas d'autre solution ici que de recourir à une technique ad hoc. Celle-ci consisterait à introduire de la non-monotonie, et à accepter un raisonnement par défaut :

(R^7) Si une fin d'activité pour une connaissance
n'est pas inférable, alors la connaissance est actuelle.

8 Langage

Jusqu'à présent, nous avons observé l'état des connaissances et le mécanisme de mise à jour. Nous nous intéressons maintenant à la relation entre langage, état des connaissances et opérations sur les connaissances. Il faut distinguer l'interprétation et la génération. Faute de place, nous nous en tiendrons ici à l'interprétation et à la gestion des connaissances de l'allocutaire.

En interprétation, le traitement d'un énoncé se déroule en deux temps : l'interprétation proprement dite, qui produit une représentation sémantique, puis le post-traitement. L'interprétation est classiquement modélisée par un parsage à l'aide d'une grammaire, comme la grammaire de Montague. Mais contrairement à ce qui se pratique avec ce type de grammaire l'objectif n'est pas de traduire un énoncé en une valeur de vérité. «Notre» grammaire s'en tient à produire une représentation sémantique. [2]

La représentation est ensuite traitée de différentes manières. Si l'énoncé est une interrogative, l'allocutaire peut être amené à déclencher une recherche d'informations dans ses connaissances et à formuler une réponse. L'allocutaire peut aussi modifier les connaissances qu'il attribue à autrui (ex : «si le locuteur me pose cette question, c'est qu'il n'a pas telle connaissance.»).

Avec des phrases simplement affirmatives comme celles que nous étudions ici, l'énoncé est mémorisé et les connaissances mises à jour. L'allocutaire réalise une expansion ou une révision de ses connaissances propres, et modifie les connaissances qu'il attribue à autrui, avec là encore expansion ou révision. Nous proposons ici

2. Nous considérons en effet que le «sens» de l'énoncé réside dans sa relation aux connaissances existantes de l'allocutaire. La valeur de vérité n'est qu'un des aspects du sens. Par exemple la formule «E=mc2» a beau être vraie, elle n'a aucun sens pour un agent qui n'a pas les moyens de la «raccrocher» à son réseau de connaissances (à quoi font référence les composants de la formule, que représentent les opérations décrites par la multiplication, qu'est-ce que cela implique, etc.).

plusieurs exemples en nous appuyant sur notre modèle des connaissances.

Dans les exemples de la section, nous présentons un énoncé, sa représentation sémantique, et les modifications déclenchées dans la base de connaissances. Pour signifier qu'un énoncé E entraîne une modification M dans les connaissances, nous écrirons $E \Rightarrow M$. Le symbole est a néanmoins la valeur $\supset$.

Jusqu'ici, nous avons utilisé le terme «savoir» comme l'abréviation de «posséder une donnée». Désormais, nous ne l'utilisons que comme objet linguistique, en tant que verbe épistémique dont on étudie le sens.

8.1 Mémorisation d'un énoncé simple (sans verbe épistémique)

Considérons un énoncé simple, sans verbe épistémique, dans un dialogue :

(12) Lucie à Alain : Il a plu avant hier (P)
$pleuvoir_i(e) \ \wedge \ date(e) = avthier \quad (p)$

Le fait que Lucie énonce P est considéré par l'allocutaire comme une preuve que P est une connaissance de Lucie («dire P revient à dire qu'on possède la connaissance de P»). Cela rejoint la règle «de nécessitation» de la logique épistémique [3] . À la date $maint()$ (qui vaut ici la date d'énonciation), l'allocutaire prend donc connaissance de la connaissance de la locutrice. Il n'est pas en mesure d'établir sa date d'activité. Il peut seulement dire que la connaissance est active présentement. L'activité de la connaissance de l'allocutaire (sur la connaissance de la locutrice) débute à la date de l'énonciation $deb(e^1) = maint()$. Après interprétation et mémorisation, l'état de la mémoire de l'allocutaire est la suivante.

$$\text{ex.12} \Rightarrow \text{BC}(a, e^1, \text{BC}(l, e^2, p) \ \wedge \ actuel(e^2)) \\ \wedge \ deb(e^1) = maint()$$

L'allocutaire met en plus à jour ses connaissances (voir section 6). Ce traitement s'applique à tout énoncé sans verbe épistémique, aussi bien pour la forme affirmative que négative, et quelle que soit la date. On peut donc se donner une règle de gestion d'un énoncé sans verbe épistémique ni expression modale (ex : «peut-être», etc.). $maint()$ vaut la date de l'énonciation.

(R^8) Attribution de croyances au locuteur :
pour tout énoncé simple p
$\exists e^1 \exists e^2 \ \text{BC}(alloc(), e^1, \text{BC}(loc(), e^2, p) \wedge actuel(e^2)) \\ \wedge \ deb(e^1) = maint()$

L'énonciation apporte aussi à l'allocutaire une information sur les connaissances que le locuteur lui attribue désormais à lui, l'allocutaire : «[Lucie sait désormais que [je sais qu'[elle sait P]$_{c^3}$]$_{c^2}$]$_{c^1}$.». L'ajout de cette connaissance étant systématique, il peut être déclenché par une règle. Pour plus de lisibilité, nous la décomposons.

(R^9) Auto-attribution de connaissances
Pour tout énoncé simple p
$\exists e \ \text{BC}(alloc(), e, c^1) \ \wedge \ deb(e) = maint()$
$\exists e^1 \ \text{BC}(loc(), e^1, c^2) \ \wedge \ deb(e^1) = maint() \quad (c^1)$
$\exists e^2 \ \text{BC}(alloc(), e^2, c^3) \ \wedge \ deb(e^2) = maint() \quad (c^2)$
$\exists e^3 \ \text{BC}(loc(), e^3, p) \ \wedge \ deb(e^3) = maint() \quad (c^3)$

Ajoutons pour finir que l'allocutaire peut s'approprier cette connaissance. Cela n'étant pas systématique (dépend du crédit accordé au locuteur ou encore à la compatibilité de la nouvelle information avec les connaissances propres existantes de l'allocutaire), ça n'est pas géré par une règle.

8.2 Énoncé direct simple avec verbe modal *savoir*

Simulons l'acquisition d'un énoncé direct simple [4] en *savoir que*.

(13) Lucie : Je sais qu'il a plu hier.
$pleuvoir(e) \ \wedge \ date(e) = hier \quad (p)$
$savoir(l, e^1, p) \ \wedge \ actuel(e^1)$

Les effets sur la mémorisation seront identiques à ceux provoqués par un énoncé simple sans verbe épistémique. La procédure peut être «automatisée» en réutilisant les règles de la section précédente. Il faut prêter attention au fait que la règle qui suit n'est valable que lorsque le sujet de *savoir* est le locuteur (première personne).

(R^{10}) Mémorisation d'un énoncé direct simple avec verbe épistémique
$\forall e \forall p \ savoir(loc(), e, p) \ \wedge \ actuel(e)$
$\Rightarrow$ appliquer R^8 et R^9 sur p.

8.3 Acquisition d'un énoncé oblique en savoir

Observons cette fois-ci une phrase où le sujet du verbe *savoir que* est autrui (*Carole*) :

(14) Lucie à Alain : Carole sait qu'[il a plu]$_P$.
$pleuvoir(e) \ \wedge \ date(e) < maint() \quad (p)$
$savoir(k, e^1, p) \ \wedge \ maint() \in date(e^1)$

L'énoncé rend compte de (c^2) l'attribution d'une connaissance c^1 (par la locutrice à autrui). Le tout bien sûr devient une connaissance (c^3) de l'allocutaire. Elle démarre au moment de l'énonciation. Par hypothèse, nous appliquons aux connaissances c^1 et c^2 la date du verbe *savoir* de l'énoncé :

3. $\frac{\alpha}{K_i(\alpha)}$

4. Le terme est emprunté à Gosselin [7] ; il désigne le présent à la première personne.

(ex.14) $\Rightarrow$
$\text{BC}(k, e^1, p) \ \wedge \ date(e^1) = date(e)$ (c^1)
$\text{BC}(l, e^2, c^1) \ \wedge \ date(e^2) = date(e^1)$ (c^2)
$\text{BC}(a, e^3, c^2) \ \wedge \ deb(e^3) = maint()$ (c^3)

Il faut ajouter à cela l'auto-attribution d'une croyance (R^9) : «la locutrice sait désormais que je sais qu'elle sait c^2 et c^3». On peut généraliser ces inférences pour tous les énoncés en *savoir que* où le sujet n'est pas le locuteur.

Il faut en plus rendre compte de la règle mentionnée dans la littérature ([8]), selon laquelle affirmer que «X (autre que moi) *sait que* P.» sous-entend que «moi (locuteur) aussi je *sais* que P.». C'est une différence notoire avec d'autres expressions en *penser que* ou *savoir si*. Bien sûr la date de la connaissance attribuée égale celle du verbe savoir. La connaissance implicite du locuteur est active présentement. Le tout doit être énoncé du point de vue de l'allocutaire. La connaissance de l'allocutaire débute à la date de l'énonciation. On obtient donc :

(R^{11}) Descente du savoir attribué vers le locuteur
$\forall p \forall x \forall e^1 \ savoir(x, e^1, p)$
$\Rightarrow \exists e^2 \exists e^3 \ \text{BC}(alloc(), e^2, \text{BC}(loc(), e^3, p)$
$\wedge \ date(e^3) = date(e^1))$
$\wedge \ deb(e^2) = main()$

8.4 *savoir*, première personne, négatif au passé

Nous ne nous sommes jusqu'ici intéressé qu'aux connaissances «qui existent». Le modèle permet de traiter aussi l'absence de connaissances

Considérons la phrase négative, au passé, à la première personne :

(15) Lucie à Alain : ⟨hier⟩ je ne savais pas qu'il avait plu avant hier
$pleuvoir(e) \ \wedge \ date(e) = avthier$ (p)
$\neg(\exists e^1 \ savoir(l, e^1, p) \ \wedge \ date(e^1) = hier)$

Une telle phrase à la forme négative déclenche une première inférence : «P n'a pas appartenu à la base de la locutrice hier» :

$\Rightarrow \neg \exists e^2 \text{BC}(l, e^2, p) \ \wedge \ date(e^2) = hier$

Une deuxième inférence est que désormais, Lucie sait qu'il a plu. Si ce n'était pas le cas, Lucie aurait par exemple utilisé la forme «je ne sais pas si P».

$\Rightarrow \text{BC}(l, e^3, p) \ \wedge \ actuel(e^3)$

Puisque ces inférences sont systématiquement déclenchées par une phrase épistémique négative avec le locuteur comme sujet, il est possible d'en tirer une règle. Il faut bien sûr adopter le point de vue de l'allocutaire : l'énoncé «je ne savais pas que P» déclenche «l'allocutaire a désormais connaissance que le locuteur a connaissance que P n'était pas dans ses (=locuteur) connaissances (c^1) mais qu'actuellement P y est (c^2). ».

(R^{12}) Mémorisation de phrases épistémiques négatives avec locuteur comme sujet :
$\forall p \forall x \forall e \ \neg savoir(loc(), e, p)$
$\Rightarrow \exists e^1 \ \text{BC}(alloc(), e^1, c^1 \ \wedge \ c^2)$
$\wedge \ deb(e^1) = maint()$
$\neg \exists e^2 \ \text{BC}(x, e^2, p) \ \wedge \ date(e^2) = date(e)$ (c^1)
$\exists e^3 \ \text{BC}(loc(), e^3, p) \ \wedge \ actuel(e^3)$ (c^2)
$\Rightarrow$ Appliquer R^9

Notons que l'ignorance du locuteur est délimitée dans le temps ($date(e)$). Cela n'empêche formellement pas d'avoir eu cette connaissance à une date antérieure. C'est possible dans le cas d'un oubli. Comme pour tout énoncé, «le locuteur sait désormais que l'allocutaire sait que le locuteur sait» (R^9). Enfin, une appropriation de la connaissance par l'allocutaire est possible.

9 Conclusion

La présentation avait pour objectif d'introduire un modèle symbolique du processus de mémorisation des données verbalement acquises. Ce modèle est destiné à «expliquer» le processus et non simplement produire un résultat identique. Il avait aussi pour contrainte d'être cognitivement réaliste et pour cela de prendre en compte la théorie de l'esprit et la temporalité des connaissances.

Le cahier des charges nous amène à nous démarquer des travaux des logiciens. En particulier, nous avons clairement pris le parti de rendre compte des processus de mémorisation du point de vue d'un agent (l'allocutaire dans un dialogue) et non pas d'un observateur omniscient. Cela nous amène à renoncer à certains partis pris de la logique modale, et notamment la logique épistémique. Nous constatons que la notation utilisée, basée sur la logique classique (théorie des types simples) suffit pour décrire les données (représentations sémantiques, état de la mémoire) et les opérations à réaliser sur les données pour les mettre en mémoire.

Nous avons montré sur des exemples concrets que ce modèle est exploitable dans plusieurs disciplines. En psychologie et cognition, il offre une description du phénomène de fausse croyance et peut aider à décrire les mécanismes de mémorisation des sources par exemple. Il peut aussi aider à décrire les déficiences. On peut ainsi réfléchir aux conséquences d'une oblitération pure et simple d'une connaissance. En gestion des données, le marquage temporel des connaissances est une solution simple pour les à jour une base de connaissances.

En linguistique, pour des raisons de place, nous avons limité nos observations aux phrases simples non modales et à la seule expression verbale épistémique *savoir que*. Mais nous savons, pour l'avoir déjà implémenté, que le

modèle permet de rendre compte de la distinction entre les expressions *savoir que* et *savoir si* ([4] etc.). Pour avoir commencé le travail dessus, nous avons de bonnes raisons de croire que le modèle est aussi efficace pour décrire les effets d'autres expressions épistémiques : verbales (*croire*, *penser que* etc.), adverbiales (*peut-être* ou autres). Il faut pour cela introduire des nouveaux prédicats qui prendront la place dans certains de BC.

Un point a été délibérément mis de côté, c'est l'efficience pratique, à l'échelle, de notre modèle. La théorie des types simples n'est pas le système de calcul le plus efficace qui soit et nombre de règles que nous avons produites sont certainement difficilement calculables. Cette dimension applicative reste à étudier.

Références

[1] Baron-Cohen, Simon, Alan M Leslie et Uta Frith : *Does the autistic child have a theory of mind?* Cognition, 21(1) :37–46, 1985.

[2] Blin, Raoul : *Résolution de l'ambiguïté sémantique des noms propres par utilisation des croyances sur les connaissances d'autrui - application au prénom*. Lingvisticae Investigationes, 40(2) :200 – 227, 2017.

[3] Chong, Hui Qing, Ah Hwee Tan et Gee Wah Ng : *Integrated cognitive architectures : a survey*. Artificial Intelligence Review, 28 :103–130, 2007.

[4] Egré, Paul : *Question-Embedding and Factivity*. Grazer Philosophische Studien, 77(1) :85–125, 2008.

[5] Ferchiou, Abdelaziz, Franck Schürhoff, E Bulzacka, M Mahbouli, Marion Leboyer et Andrei Szöke : *Mémoire de source–présentation générale et revue des études dans la schizophrénie*. L'Encéphale, 36(4) :326–333, 2010.

[6] Ferchiou, Abdelaziz, Franck Schürhoff, E Bulzacka, M Mahbouli, Marion Leboyer et Andrei Szöke : *Mémoire de source–présentation générale et revue des études dans la schizophrénie*. L'Encéphale, 36(4) :326–333, 2010.

[7] Gosselin, Laurent : *Marqueurs de modalité épistémique et calcul des valeurs modales : sémantique de savoir que*. 2013.

[8] Martin, Robert : *Langage et croyance*. Mardaga, Bruxelles, 2023.

[9] Norimatsu, H., R. Blin, K. Hashiya, Ch Sorsana et H. Kobayashi : *Understanding of others' knowledge in French and Japanese children : a comparative study with a disambiguation task on 16-38-month-olds*. Infant Behavior & Development, 37(4) :632–643, novembre 2014, ISSN 1934-8800.

[10] Parsons, Terence : *Events in the semantics of English*. Numéro 19 dans *Current in linguistics series*. ITMIT press édition, 1990.

[11] Pellier, Damien, Carole Adam, Wafa Johal, Humbert Fiorino et Sylvie Pesty : *Une architecture cognitive et affective orientée interaction*. Dans *Workshop on Affect, Compagnon Artificiel and Interaction*, Brest, France, juin 2016. `https://hal.science/hal-01365355`.

[12] Premack, David et Guy Woodruff : *Does the chimpanzee have a theory of mind?* Behavioral and Brain Sciences, 1(04) :515, 1978, ISSN 0140-525X, 1469-1825.

[13] Sabouret, nicolas et Manween Belkaid : *Un modèle logique de théorie de l'esprit pour un agent virtuel dans le contexte de simulation d'entretien d'embauche*. Dans *Proc. Workshop Affect, Compagnon Artificiel, Interaction*, Rouen, 2014.

[14] Wimmer, Heintz et Josef Perner : *Beliefs about beliefs : Representation and constraining function of wrong beliefs in young children's understanding of deception*. Cognition, 13(1) :103–128, jan 1983, ISSN 00100277.

Session 3 : Logique

Actes JIAF-JFPDA 2023

Morpho-logique d'un point de vue de la théorie des topos : application à l'IA symbolique

Marc Aiguier[1] **Isabelle Bloch**[2] **Salim Nibouche**[1] **Ramón Pino Pérez**[3]

[1] MICS, CentraleSupélec, Université Paris-Saclay, France
[2] Sorbonne Université, CNRS, LIP6, Paris, France
[3] CRIL-CNRS, Université d'Artois, France
{marc.aiguier,salim.nibouche}@centralesupelec.fr
isabelle.bloch@sorbonne-universite.fr
pinoperez@cril.fr

Résumé

La morphologie mathématique (MM) est une théorie non linéaire d'analyse de structures qui a été largement appliquée à l'analyse d'images. Les fondements mathématiques de la MM proviennent de l'algèbre, de la théorie des treillis complets ou encore de la topologie. Depuis une vingtaine d'années, de forts liens ont été établis entre la MM et la logique mathématique, et principalement la logique modale. Dans ce cadre, les modalités de nécessité $\Box$ et de possibilité $\Diamond$ sont interprétées par les deux opérations de base de la MM d'érosion et de dilatation. Il a alors été montré que cette interprétation facilitait les raisonnements logiques non classiques tels que la révision, la fusion, l'abduction ou encore le raisonnement spatial. Dans cet article, nous proposons d'étendre ce lien entre la MM et la logique modale au cadre de la théorie algébrique des topos élémentaires, i.e. une structure catégorielle généralisant la notion d'espace, et permettant de connecter dans un même cadre général la logique, la théorie des ensembles et la topologie. Nous montrons alors que la logique modale ainsi définie (appelée morpho-logique ici), est bien adaptée pour définir des opérateurs concrets et efficaces pour la révision, la fusion, et l'abduction de nouvelles connaissances, ou encore le raisonnement spatial.

Abstract

Mathematical morphology (MM) is a theory for non-linear analysis of structures, that was widely developed and applied in image analysis. Its mathematical bases rely on algebra, complete lattices, topology. Strong links have been established between MM and mathematical logics, mostly modal logics. Necessity $\Box$ and possibility $\Diamond$ modalities are then interpreted by the two basic MM operators, namely erosion and dilation. This interpretation allows for easy formulations of non-classical reasoning, including revision, merging, abduction, spatial reasoning. In this paper, we propose to extend this link between MM and modal logics in the setting of algebraic theory of topos, i.e. a categorical structure that generalizes the notion of space, and unifies in a same general framework logics, set theory and topology. We demonstrate that the modal logic we define (called morpho-logic) is well suited to define concrete and efficient operators for revision, merging (or fusion), abduction of new knowledge, as well as spatial reasoning.

1 Introduction

La morphologie mathématique (MM) [35, 36] est une théorie non linéaire d'analyse de structures qui a été largement appliquée à l'analyse d'images. Les fondements mathématiques de la MM, dans son cadre déterministe, proviennent de l'algèbre, de la théorie des treillis complets ou encore de la topologie. Depuis une vingtaine d'années, de forts liens ont été établis entre la MM et la logique mathématique, et principalement la logique modale [1, 11, 15]. Dans ce cadre, les modalités de nécessité $\Box$ et de possibilité $\Diamond$ sont interprétées par les deux opérations de base de la MM d'érosion et de dilatation. Il a alors été montré que cette interprétation facilitait les raisonnements logiques non classiques tels que la révision, la fusion, l'abduction [2, 3, 15, 22] ou encore le raisonnement spatial [1, 11].

L'érosion et la dilatation sont souvent définies à partir d'un élément structurant B utilisé pour sonder les structures spatiales, soit pour les éroder, soit pour les dilater. Plus formellement, dans le cas particulier des ensembles, pour E un espace euclidien (souvent $\mathbb{R}^d$ ou $\mathbb{Z}^d$ où d est la dimension de l'espace), et B (l'élément structurant) un sous-ensemble de E, si nous notons $B_x = \{x + b \mid b \in B\}$ sa translation

au point $x \in E$, alors la dilatation d'un ensemble X par B est définie par $\delta[B](X) = \{x \in E \mid \check{B}_x \cap X \neq \emptyset\}$ où $\check{B}$ est le symétrique de B par rapport à l'origine, et l'érosion de X par $\varepsilon[B](X) = \{x \in E \mid B_x \subseteq X\}$. Observons que B peut aussi être vu comme une relation binaire sur E : $B(x, y)$ ssi $y \in B_x$. Dans ce cadre ensembliste, nous avons les propriétés suivantes [1] :

— l'érosion commute avec l'intersection et préserve E,
— la dilatation commute avec la réunion et préserve $\emptyset$,
— l'érosion et la dilatation définies par un même élément structurant sont des opérateurs duaux.

Ainsi, le tuple $(\mathcal{P}(E), \cap, \cup, \_^c, \emptyset, E, \varepsilon[B], \delta[B])$ est une algèbre modale. Par cette interprétation, la logique modale devient un outil puissant pour parler de transformations d'espace, et c'est dans ce cadre que la morpho-logique a été appliquée efficacement à l'intelligence artificielle (IA) symbolique [4, 11, 12].

Jusqu'à présent, ce lien entre la logique modale et la MM a été étudié dans le cadre ensembliste (avec des extensions aux ensembles flous). Depuis, la MM a été étendue à une large famille de structures algébriques telles que les graphes [19, 20, 31, 37], les hypergraphes [13, 14], les complexes simpliciaux [21], etc. Toutes ces extensions se sont montrées très utiles pour la représentation des connaissances, prenant en compte un faible niveau d'information (points ou voisinage de points), des informations structurelles (par exemple fondées sur des relations spatiales entre régions ou objets), des aspects sémantiques, etc. Ce que nous constatons alors est que l'ensemble de ces extensions se définissent de façon générale par la notion de préfaisceau, i.e. un foncteur contravariant $F : C^{op} \rightarrow Set$ où la catégorie de base C est petite [2], et Set est la catégorie des ensembles. Il est connu que la catégorie des préfaisceaux sur une catégorie de base petite est un topos complet. Pour aller plus loin dans la généralisation, nous proposons alors d'approfondir ce lien entre la logique modale et la MM dans le cadre de la théorie des topos. Les topos constituent des structures catégorielles définies par A. Grothendieck au début des années 1960 [24], qui généralisent la notion d'espace. Ainsi, comme l'a remarqué O. Caramello dans [18], tout topos incarne un certain domaine de la réalité susceptible de devenir des objets de connaissances (i.e. les instanciations idéalisées de cette réalité sont alors les points de ce topos). Cette interprétation toposique permettra alors de donner une sémantique avec une "saveur" topologique aux modalités classiquement utilisées pour le raisonnement spatial, et qui ne peut pas s'obtenir directement à partir des érosions et dilatations, ni de leur composition (on parle alors d'ouverture et de fermeture). Dans [23], la MM à partir d'éléments structurants a été étendue à la notion de voisinage proche de la notion classique en topologie. Nous proposons dans cet article d'étendre ce travail défini dans un cadre ensembliste à celui des topos. Cette extension nous permettra de donner une sémantique toposique de voisinage à la logique modale constructive CS4 [8, 30, 38].

Les topos, et plus précisément les topos élémentaires de Lawvere et Tierney [28], sont introduits de façon succincte dans la section 2. Dans la section 3, la MM est étendue au cadre des topos. La section 4 est dédiée à l'extension de la notion d'élément structurant à celle de voisinage structurant. Dans la section 5, nous proposons alors une nouvelle façon d'interpréter les modalités de nécessité et de possibilité à partir des nouveaux opérateurs d'érosions et de dilatations définis sur les voisinages structurants utilisés comme relation d'accessibilité. Cela généralise les premiers travaux établissant un lien entre la MM et la logique modale [11] mais aussi étend aux topos la sémantique des voisinages habituellement définie dans le cadre ensembliste [32]. Enfin, dans la section 6, nous montrons que la logique modale ainsi définie est bien adaptée pour définir des opérateurs concrets et efficaces pour la révision, la fusion, et l'abduction de nouvelles connaissances, ou encore le raisonnement spatial.

Cet article est une version courte et écrite en français de l'article [6]. Entre autres, le lecteur ne trouvera aucune preuve des résultats énoncés. Nous renvoyons les lecteurs intéressés à l'article original pour trouver les preuves de ces derniers.

2 Préliminaires : topos

Nous supposons que le lecteur connaît les notions de base de la théorie des catégories (catégorie, foncteur, transformation naturelle, limite, colimite, cartésienne close, sous-objets). Dans le cas contraire, nous renvoyons le lecteur intéressé aux livres [10, 29].

2.1 Notation

Dans la suite, C désigne une catégorie générique, et X, Y, et Z des objets de C. Quand C est cartésienne close, X^Y désigne l'objet exponentiel de X et Y. Les symboles f, g, et h définissent des morphismes, et étant donné un morphisme $f : X \rightarrow Y$, nous notons $\mathrm{dom}(f) = X$ et $\mathrm{cod}(f) = Y$. Les foncteurs sont désignés par les lettres F, G, et H, et les transformations naturelles par les lettres greques $\alpha, \beta : F \Rightarrow G$. Le morphisme identité est noté Id, et les objets initiaux et terminaux (quand ils existent) sont notés $\emptyset$ et $\mathbb{1}$. Enfin, les monomorphismes m entre deux objets X et Y sont notés $m : X \rightarrowtail Y$.

1. Notons que, plus généralement en MM, les érosions et les dilatations algébriques sur les treillis complets sont définies comme des opérations qui commutent avec les bornes inférieures et supérieures, respectivement (i.e. l'intersection et la réunion dans $(\mathcal{P}(E), \subseteq)$). Cette forme plus générale de ces opérateurs ne fait alors plus référence à un élément structurant, et donc les deux premières propriétés sont plutôt des définitions dans ce cadre général.

2. Une catégorie est petite quand à la fois la collection des objets et celle des morphismes entre deux objets sont des ensembles.

2.2 Topos élémentaire

Un topos C est une catégorie cartésienne close finiment complète, et munie d'un classificateur de sous-objets Ω. Posséder un classificateur de sous-objets signifie qu'il existe un morphisme $true : \mathbb{1} \to \Omega$ tel que pour tout monomorphisme $m : Y \rightarrowtail X$ il existe un unique morphisme $\chi_m : X \to \Omega$ (morphisme caractéristique de m) tel que le diagramme suivant est un produit fibré (*pullback*) :

$$\begin{array}{ccc} Y & \xrightarrow{!} & \mathbb{1} \\ {\scriptstyle m}\downarrow & & \downarrow{\scriptstyle true} \\ X & \xrightarrow[\chi_m]{} & \Omega \end{array}$$

Soit $X \in |C|$ un objet de C. Son ensemble de sous-objets est :

$$\mathrm{Sub}(X) = \{[m] \mid \mathrm{cod}(m) = X \text{ et } m \text{ est un monomorphisme}\}$$

où $[m]$ est la classe d'équivalence de m pour la relation d'équivalence $m \simeq m'$ ssi $\mathrm{cod}(m) = \mathrm{cod}(m')$ et $\mathrm{dom}(m)$ et $\mathrm{dom}(m')$ sont isomorphes.

Soit la relation d'ordre $\preceq_X$ sur $\mathrm{Sub}(X)$ définie par : pour tout $f : Y \rightarrowtail X$ et tout $g : Z \rightarrowtail X$

$$[f] \preceq_X [g] \iff \exists h : Y \rightarrowtail Z, f = g \circ h$$

Il est connu que $\mathrm{Sub}(X)$ est une algèbre de Heyting [25], i.e. $(\mathrm{Sub}(X), \preceq_X)$ est un treillis borné et distributif avec $[Id_X]$ et $[\emptyset \rightarrowtail X]$ comme borne supérieure et borne inférieure, et qui possède une implication $\to$ adjointe à droite de $\wedge$ (quand l'algèbre de Heyting $(\mathrm{Sub}(X), \preceq_X)$ est vue comme une catégorie).

Comme C est finiment complète, nous avons le foncteur contravariant $\mathrm{Sub} : C^{op} \to Pos; X \mapsto Sub(X); f : X \to Y \mapsto ([Y' \rightarrowtail X'] \mapsto [Y \rightarrowtail X])$ [3], tel que le diagramme

$$\begin{array}{ccc} Y & \longrightarrow & Y' \\ \downarrow & & \downarrow \\ X & \longrightarrow & X' \end{array}$$

est un produit fibré.

Tout topos a les propriétés suivantes [9, 25] :

— Il a aussi toutes les colimites finies, et donc il a un objet intial $\emptyset$ et un objet terminal $\mathbb{1}$ qui sont, respectivement, la colimite et la limite du diagramme vide.

— Tout morphisme f se factorise de façon unique comme $m_f \circ e_f$ où e_f est un épimorphisme et m_f un monomorphisme (i.e. $(A \xrightarrow{f} B) = (A \xrightarrow{e_f} \mathrm{Im}(f) \overset{m_f}{\rightarrowtail} B)$).

— Tout objet $X \in |C|$ a un *objet puissance* Ω^X aussi noté PX. Comme objet puissance, il satisfait la propriété d'adjonction suivante :

$$\mathrm{Hom}_C(X \times Y, \Omega) \simeq \mathrm{Hom}_C(X, PY)$$

Étant donné un morphisme $f \in \mathrm{Hom}_C(X \times Y, \Omega)$ (respectivement $f \in \mathrm{Hom}_C(X, PY)$) nous notons $f^\#$ son transposé par la bijection précédente. En particulier, le transposé de l'identité $Id_{PX} : PX \to PX$ est le morphisme caractéristique du sous-objet $\in_X \rightarrowtail X \times PX$ tel que pour tout $Y \in |C|$ et tout morphisme $R \rightarrowtail X \times Y$, il existe un unique morphisme $R \to \in_X$ faisant du diagramme suivant un produit fibré :

$$\begin{array}{ccc} R & \longrightarrow & \in_X \\ \downarrow & & \downarrow \\ X \times Y & \xrightarrow[Id_X \times \chi_R^\#]{} & X \times PX \end{array}$$

Par ses propriétés, les topos ont un comportement qui se rapproche de celui des ensembles, et qui permet ainsi d'internaliser une logique avec laquelle on peut raisonner (de façon constructive) comme si on manipulait des ensembles, des fonctions et des prédicats. Pour des raisons de place, nous ne présentons pas cette logique interne ici. Néanmoins, nous utiliserons grandement cette dernière dans cet article, entre autres pour définir les différentes notions d'érosions et de dilatations, et leur extension aux voisinage structurants. Cette logique interne est détaillée dans [6]. Pour une étude plus approfondie de cette dernière, nous renvoyons le lecteur au chapitre D du livre [25]. Elle permet en particulier de donner des définitions explicites des érosions et dilations (section 3), et de la morpho-logique (section 5).

3 Morphologie mathématique dans les topos

Soit C un topos. Les objets structurants se définissent simplement par tout morphisme de la forme $b : X \to PX$. À partir de ces objets structurants, il est assez simple de définir les opérations d'érosion et de dilatation.

Définissons par $\check{b} : X \to PX$ le transposé du morphisme qui classifie l'image du morphisme $R_b \rightarrowtail X \times X \xrightarrow{\Delta_{X\times X}} (X \times X) \times (X \times X) \xrightarrow{p_2 \times p_1} X \times X$, où Δ est le morphisme diagonal, et p_i la projection sur le $i$ème argument. Dans le langage interne, cela s'écrit : $\check{b}(y) = \{x : X \mid y \in_X b(x)\}$.

Definition 3.1 (Erosion) *Soit $b : X \to PX$ un objet structurant. L'***érosion par** *b est le morphisme $\varepsilon[b] : PX \to PX$ dont le transposé classifie le morphisme $r : R \rightarrowtail PX \times X$ (i.e. $\varepsilon[b] = \chi_r^\#$) où R est le produit fibré du diagramme :*

$$\begin{array}{ccc} R & \longrightarrow & \geq_X \\ \downarrow & & \downarrow \\ PX \times X & \xrightarrow[Id \times b]{} & PX \times PX \end{array}$$

3. Pos est la catégorie des ensembles partiellements ordonnés (*posets*).

Dans la logique interne du topos C, cela s'exprime de façon équivalente par :

$$\varepsilon[b](Y) = \{x : X \mid b(x) \preceq_X Y\}$$

Definition 3.2 (Dilation) *Soit $b : X \to PX$ un objet structurant. La* **dilatation par** *b est le morphisme $\delta[b] : PX \to PX$ qui classifie l'image de $R \rightarrowtail X \times X \times PX \to PX \times X$ où R est le produit fibré du diagramme :*

$$\begin{array}{ccc} R & \longrightarrow & R_{\check{b}} \times \in_X \\ \downarrow & & \downarrow \\ X \times X \times PX & \xrightarrow[Id \times \Delta_X \times Id]{} & X \times X \times X \times PX \end{array}$$

Dans la logique interne, cela s'écrit :

$$\delta[b](Y) = \{x : X \mid \exists y.\ y \in_X \check{b}(x) \ \wedge \ y \in_X Y\}$$

Nous retrouvons l'ensemble des résultats classiques de la MM.

Proposition 3.3 *Les propriétés suivantes sont satisfaites :*

- Adjunction. $\forall Y.\ \forall Z.\ \delta[b](Y) \preceq_X Z \iff Y \preceq_X \varepsilon[b](Z)$.
- Monotonie. *Les érosions et les dilatations sont monotones pour $\preceq_X$.*
- Préservation.
  - $\varepsilon[b](X) = X$,
  - $\delta[b](\emptyset) = \emptyset$.
- Extensivité. *$\varepsilon[b]$ et $\delta[b]$ sont, respectivement, anti-extensive et extensive pour $\preceq_X$ ssi la formule $\forall x.\ x \in_X b(x)$ est valide dans la logique interne.*

4 Voisinage structurant : topologie interne

Donner une sémantique morphologique aux modalités avec des aspects topologiques a montré son importance dans la représentation des connaissances et les raisonnements associés (comme le raisonnement spatial). Dans [23], les objets structurants ont été étendus aux voisinages structurants, concept similaire à la notion topologique, dans le cadre ensembliste. Nous proposons ici de l'étendre à la théorie des topos.

4.1 Voisinage structurant

Soit $X \in |C|$ un objet.

Definition 4.1 (Filtre) *Un **filtre** sur X est tout sous-objet de PPX satisfaisant les axiomes suivants : soit $F : PPX$ une variable*

- *Fermeture par intersection finie :*

$$\forall A.\ \forall B.\ A \in_{PX} F \ \wedge \ B \in_{PX} F \ \Rightarrow \ A \wedge B \in_{PX} F$$

- *Fermeture par sur-ensemble :*

$$\forall A.\ \forall B.\ A \in_{PX} F \ \wedge \ A \preceq_X B \ \Rightarrow \ B \in_{PX} F$$

- *Non vide :*

$$X \in_X F$$

- *Stricte :*

$$\forall A.\ A \in_{PX} F \ \Rightarrow \ (\exists x.\ x \in_X A)$$

Cela définit un morphisme $\mathcal{F} : PPX \to \Omega$.

Definition 4.2 (Voisinage structurant) *Un* **voisinage structurant** *est un morphisme $N : X \to PPX$ qui valide les formules suivantes :*

1. $\forall x.\ \mathcal{F}(N(x))$,
2. $\forall x. \forall A. A \in_{PX} N(x) \Rightarrow x \in_X A$

Definition 4.3 (Érosion et dilatation) *Soit $N : X \to PPX$ un voisinage structurant. Définissons le morphisme $\varepsilon[N] : PX \to PX$ par la formule :*

$$\varepsilon[N](Y) = \{x : X \mid Y \in_{PX} N(x)\}$$

et le morphisme $\delta[N] : PX \to PX$ par :

$$\begin{aligned} \delta[N](Y) &= \{x : X \mid \exists F.\ \mathcal{F}(F) \ \wedge \ Y \in_{PX} F \ \wedge \ N(x) \preceq_{PX} F\} \\ &= \{x : X \mid \forall A.\ A \in_{PX} N(x) \ \Rightarrow \ \exists y.\ y \in_X A \wedge Y\} \end{aligned}$$

Étant donné un voisinage structurant $N : X \to PPX$, nous avons la transformation naturelle $\overline{\varepsilon[N]} : \mathrm{Sub} \Rightarrow \mathrm{Sub}$ définie par le diagramme commutatif :

$$\begin{array}{ccc} Hom(\mathbb{1}, PX) & \xrightarrow{\simeq} & \mathrm{Sub}(X) \\ {\scriptstyle Hom(Id_{\mathbb{1}}, \varepsilon[b])}\downarrow & & \downarrow{\scriptstyle \overline{\varepsilon[N]}_X} \\ Hom(\mathbb{1}, PX) & \xrightarrow{\simeq} & \mathrm{Sub}(X) \end{array}$$

De façon similaire, nous pouvons construire la transformation naturelle $\overline{\delta[N]} : \mathrm{Sub} \Rightarrow \mathrm{Sub}$.

Proposition 4.4 *$\varepsilon[N]$ et $\delta[N]$ sont monotones. De plus :*

- *$\varepsilon[N]$ satisfait :*
  - $\forall A.\ \forall B.\ \varepsilon[N](A \wedge B) = \varepsilon[N](A) \wedge \varepsilon[N](B)$.
  - $\varepsilon[N](X) = X$.
  - $\forall Y. \varepsilon[N](Y) \preceq_X Y$.
- *$\delta[N]$ satisfait :*
  - $\forall A.\ \forall B.\ \delta[N](A \vee B) \succeq_X \delta[N](A) \vee \delta[N](B)$.
  - $\delta[N](\emptyset) = \emptyset$.
  - $\forall Y. Y \preceq_X \delta[N](Y)$.
- $\forall Y.\ \varepsilon[N](\neg_X Y) \preceq_X \neg_X \delta[N](Y)$

Bien que $\varepsilon[N]$ soit une ouverture au sens toplogique du terme, $\delta[N]$ n'est pas une fermeture (elle n'est pas idempotente) ni une dilatation au sens strict (elle ne se distribue pas complètement sur $\vee$). Les voisinages structurants seront

suffisants pour donner une sémantique à la logique modale intuitioniste IT. Dans la section suivante, nous étendrons ces derniers aux voisinages topologiques pour permettre d'interpréter la logique modale constructive CS4.

Chaque objet structurant $b : X \to PX$ définit un voisinage structurant $N_b : X \to PPX : N_b(x) = \{Y : PX \mid U \geq_X b(x)\}$. Il n'est pas difficile de montrer que $\varepsilon[N_b] = \varepsilon[b]$ et $\delta[N_b] = \delta[\breve{b}]$.

4.2 Voisinage topologique

Definition 4.5 (Voisinage topologique) *Un* **voisinage topologique** *est un voisinage structurant $N : X \to PPX$ satisfaisant :*

$$\forall x.\ \forall A.\ A \in_{PX} N(x) \Rightarrow \left(\begin{array}{l} \exists B.\ B \in_{PX} N(x) \wedge \\ (\forall y.\ y \in_X B \Rightarrow A \in_{PX} N(y)) \end{array} \right)$$

Proposition 4.6 *Pour les voisinages topologiques N, $\varepsilon[N]$ est un opérateur intérieur interne sur PX (i.e. il est anti-extensif, préserve X, se distribue sur $\wedge$, et est idempotent). À l'inverse, $\delta[N]$ n'est qu'un opérateur de clôture interne sur PX (i.e. il est extensif, préserve l'objet initial, et est idempotent).*

5 Morpho-logique : interprétation des modalités dans les topos

Dans la section 4.1, nous avons montré que le tuple $(PX, \wedge, \vee, \neg_X, \emptyset, \varepsilon[N], \delta[N])$ est une agèbre modale interne pour un voisinage structurant N. Nous utilisons ce fait pour donner une sémantique par voisinage aux logiques modales constructives IT (voisinage structurant) et CS4 (voisinage topologique).

Syntaxe. Soit PV un ensemble dénombrable de **variables propositionelles** $p, q \ldots$. L'ensemble Φ des formules est défini par la grammaire :

$$\varphi, \psi ::= \top \mid \bot \mid p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \Rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi$$

où $p \in PV$.

Sémantique. Soit C un topos. Étant donné un ensemble de variables propositionnelles PV, un PV**-modèle** $\mathcal{M}$ est un triplet (X, N, ν) où :
- $X \in |C|$,
- $N : X \to PPX$ est un voisingae structurant,
- $\nu : PV \to \mathrm{Sub}(X)$ est une **évaluation**.

Pour donner une sémantique à la logique CS4, nous restreignons les modèles aux voisinages topologiques.
Notons Mod la classe des PV-modèles.

Comme il est d'usage en logique catégorielle, la sémantique des formules est définie par des sous-objets. Ainsi, étant donné un modèle $\mathcal{M} = (X, N, \nu)$, il donne lieu à une application $[\![\mathcal{M}]\!](\_) : \Phi \to \mathrm{Sub}(X)$ définie par induction structurelle sur les formules comme suit :
- $[\![\mathcal{M}]\!](\top) = [Id_X]$,
- $[\![\mathcal{M}]\!](\bot) = [\emptyset \rightarrowtail X]$
- $[\![\mathcal{M}]\!](p) = \nu(p)$,
- $[\![\mathcal{M}]\!](\neg\varphi) = [\![\mathcal{M}]\!](\varphi) \to [\![\mathcal{M}]\!](\bot)$
- $[\![\mathcal{M}]\!](\varphi \wedge \psi) = [\![\mathcal{M}]\!](\varphi) \wedge [\![\mathcal{M}]\!](\psi)$ (borne inférieure),
- $[\![\mathcal{M}]\!](\varphi \vee \psi) = [\![\mathcal{M}]\!](\varphi) \vee [\![\mathcal{M}]\!](\psi)$ (borne supérieure),
- $[\![\mathcal{M}]\!](\varphi \Rightarrow \psi) = [\![\mathcal{M}]\!](\varphi) \to [\![\mathcal{M}]\!](\psi)$
- $[\![\mathcal{M}]\!](\Box\varphi) = \overline{\varepsilon[N]}_X([\![\mathcal{M}]\!](\varphi))$
- $[\![\mathcal{M}]\!](\Diamond\varphi) = \overline{\delta[N]}_X([\![\mathcal{M}]\!](\varphi))$

Nous écrivons $\mathcal{M} \models \varphi$ si $[\![\mathcal{M}]\!](\varphi) = [Id_X]$, et donc pour tout $\iota \in \mathrm{Sub}(X)$, nous écrivons $\mathcal{M} \models_\iota \varphi$ pour signifier que $\iota \leq_X [\![\mathcal{M}]\!](\varphi)$. Notons $Mod(\varphi)$ la classe des modèles $\mathcal{M}$ qui valident φ. Enfin, étant donné un ensemble de formules Γ et une formule φ, nous écrivons $\Gamma \models \varphi$ pour dire que pour tout modèle $\mathcal{M}$ qui satisfait $\mathcal{M} \models \psi$ pour toute formule $\psi \in \Gamma$, nous avons $\mathcal{M} \models \varphi$.

Système d'inférence. Comme il est d'usage pour les logiques catégorielles, le système d'inférence est défini comme un système de séquents où un séquent est une expression de la forme $\varphi \vdash \psi$ avec $\varphi, \psi \in \Phi$. Étant donné un modèle $\mathcal{M} = (X, N, \nu)$, on dit que $\mathcal{M}$ valide $\varphi \vdash \psi$, noté $\varphi \models_{\mathcal{M}} \psi$, si $[\![\mathcal{M}]\!](\varphi) \leq_X [\![\mathcal{M}]\!](\psi)$, et ce séquent est valide, noté $\varphi \models \psi$, s'il est valide pour tous les modèles. Nous avons alors les règles suivantes [4] (où $\varphi \dashv\vdash \psi$ signifie que à la fois $\varphi \vdash \psi$ et $\psi \vdash \varphi$) :
- *Identité :*
$$\varphi \vdash \varphi$$
- *Axiomes :*
  - **Préservation.** $\Box\top \dashv\vdash \top$, et $\Diamond\bot \dashv\vdash \bot$
  - **Dualité.** $\Box\neg\varphi \vdash \neg\Diamond\varphi$
  - **Distributivité.** $\Box(\varphi \wedge \psi) \dashv\vdash \Box\varphi \wedge \Box\psi$ et $\Diamond\varphi \vee \Diamond\psi \vdash \Diamond(\varphi \vee \psi)$
  - **Axiome K.** $\Box(\varphi \Rightarrow \psi) \vdash \Box\varphi \Rightarrow \Box\psi$
  - **Axiome T.** $\Box\varphi \vdash \varphi$, et $\varphi \vdash \Diamond\varphi$
  - **Axiome S4.** $\Box\varphi \vdash \Box\Box\varphi$, et $\Diamond\Diamond\varphi \vdash \Diamond\varphi$ (valide pour les modèles restreints aux voisinages topologiques)
  - **Classique.** $\neg\neg\varphi \vdash \varphi$ (quand C est un topos booléen)
- *Inconsistance :* $\bot \vdash \psi$
- *Tautologie :* $\varphi \vdash \top$
- *Coupure :*
$$\frac{\varphi \vdash \psi \qquad \psi \vdash \chi}{\varphi \vdash \chi}$$

4. Comme il est d'usage, une règle se présentera sous la forme $\frac{\Gamma}{\sigma}$ et signifiera que le séquent σ peut être inféré de l'ensemble de séquents Γ. Toute règle avec une double ligne signifiera que chacun des séquents peut être inféré des autres.

— *Conjonction :* $\varphi \wedge \psi \vdash \varphi \quad \varphi \wedge \psi \vdash \psi \quad \varphi \wedge \varphi \vdash \varphi \quad \varphi \wedge \psi \vdash \psi \wedge \varphi$

$$\frac{\varphi \vdash \psi \qquad \varphi \vdash \chi}{\varphi \vdash \psi \wedge \chi}$$

— *Disjonction :* $\varphi \vdash \varphi \vee \psi \quad \psi \vdash \varphi \vee \psi \quad \varphi \vee \psi \vdash \psi \vee \varphi$

$$\frac{\varphi \vdash \chi \qquad \psi \vdash \chi}{\varphi \vee \psi \vdash \chi}$$

— *Distributivité :* $\varphi \wedge (\psi \vee \chi) \vdash (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$

— *Implication :*

$$\frac{\varphi \wedge \psi \vdash \chi}{\varphi \vdash \psi \Rightarrow \chi}$$

— *Négation :* $\neg\varphi \vdash \varphi \Rightarrow \bot$

— *Modalités :*

$$\frac{\varphi \vdash \psi}{\Box\varphi \vdash \Box\psi} \qquad \frac{\varphi \vdash \psi}{\Diamond\varphi \vdash \Diamond\psi}$$

Dans [6], nous avons montré que ce système est correct et complet. Pour montrer la complétude, nous avons utilisé une méthode proche de la méthode de Henkin. Cela nous a assuré un résultat de complétude indépendant d'un topos donné.

6 Application à l'IA symbolique

Que ce soit en logique mathématique, en philosophie ou en intelligence artificielle, on est souvent confronté à faire évoluer nos croyances ou connaissances à la lumière de nouvelles observations (révision de croyances), à savoir comment extraire une information cohérente de plusieurs sources éventuellement contradictoires (fusion), ou encore comment une observation donnée peut s'expliquer à partir de connaissances acquises (abduction).

Il a été montré dans le cadre de la logique propositionnelle que l'application d'opérateurs issus de la MM se montrait efficace pour répondre à ce type de questions [16]. Profitant du fait que nos opérateurs modaux sont définis à partir d'extensions des opérateurs de base de la MM que sont les érosions et les dilatations, nous proposons d'étendre ces travaux au cadre de la morpho-logique définie dans cet article. La figure 1 illustre, dans le cas simple où les modèles des formules sont des ensembles, le principe de l'approche proposée.

6.1 Révision

Nous supposons donc ici que les croyances des agents sont formalisées par des formules de notre morpho-logique. La révision de croyances a alors pour objectif de définir un opérateur $\circ$ entre deux formules φ et ψ qui définira comment transformer de façon minimale φ en une formule φ' telle que $\varphi' \wedge \psi$ soit cohérente [5]. Une axiomatisation de la révision s'est imposée dans le domaine, la théorie AGM [7], dont nous rappelons ici les axiomes :

— **(*G*1)** Si ψ est une formule cohérente, alors $\varphi \circ \psi$ l'est aussi ;
— **(*G*2)** $Mod(\varphi \circ \psi) \subseteq Mod(\psi)$;
— **(*G*3)** Si $\varphi \wedge \psi$ est cohérente, alors $\varphi \circ \psi = \varphi \wedge \psi$;
— **(*G*4)** Si $\varphi \equiv \varphi'$ et $\psi \equiv \psi'$, alors $Mod(\varphi \circ \psi) = Mod(\varphi' \circ \psi')$ ($\equiv$ signifie logiquement équivalent) ;
— **(*G*4′)** Si $\psi \equiv \psi'$, alors $Mod(\varphi \circ \psi) = Mod(\varphi \circ \psi')$;
— **(*G*5)** $Mod((\varphi \circ \psi) \wedge \chi) = Mod(\varphi \circ (\psi \wedge \chi))$ si $(\varphi \circ \psi) \wedge \chi$ est une formule cohérente.

L'axiome $(G4)$ exprime une complète indépendance de l'opérateur de révision par rapport à la syntaxe. Or, nous verrons dans la suite que, lorsque l'opérateur $\circ$ applique une transformation syntaxique sur la base de connaissances (ici représentée par la formule φ), cet axiome ne peut plus être assuré. Ce sera le cas pour notre opérateur de révision dédié à la logique CS4 (voir ci-dessous), d'où l'introduction de l'axiome $(G4')$.

Opérateur de révision fondé sur la dilatation. Par la façon dont les modalités sont interprétées, nous avons, pour toute formule $\varphi \in \Phi$, $Mod(\varphi) \subseteq Mod(\Diamond\varphi)$ (les dilatations sont extensives ici).

Suivant l'approche développée dans [12], l'idée consiste alors à dilater la classe des modèles de φ jusqu'à rencontrer la classe des modèles de ψ. Nous obtenons alors l'opérateur de révision $\circ$ suivant :

$$\varphi \circ \psi = \Diamond^n \varphi \wedge \psi$$

où $n = \min\{k \in \mathbb{N} \mid \Diamond^k\varphi \wedge \psi \text{ est cohérente}\}$ et $\Diamond^n\varphi = \underbrace{\Diamond \ldots \Diamond}_{n\ fois}\varphi$.

Proposition 6.1 *L'opérateur* $\circ$ *satisfait les axiomes* $(G1) - (G5)$.

L'approche ci-dessus n'est plus applicable si nous restreignons nos modèles aux voisinages topologiques. En effet, dans ce cas-là, le séquent $\Diamond\Diamond\varphi \vdash \Diamond\varphi$ est valide. Une autre façon de faire pour définir l'opérateur de révision $\circ$ est alors de changer les modalités de nécessité en possibilité. Il semble assez intuitif que si la formule n'est pas cohérente pour tous les états, elle peut l'être pour certains. Une approche similaire a été adoptée dans [2] pour définir les opérateurs de révision pour les logiques modales et du 1er ordre dans le cadre ensembliste. La définition de ces opérateurs profitait alors du raisonnement booléen qui assure l'existence de formules en forme normale sur lesquelles étaient définies les opérateurs de révison. Dans un cadre

5. φ sera souvent la conjonction d'un ensemble fini de croyances elles-mêmes définies par des formules [26].

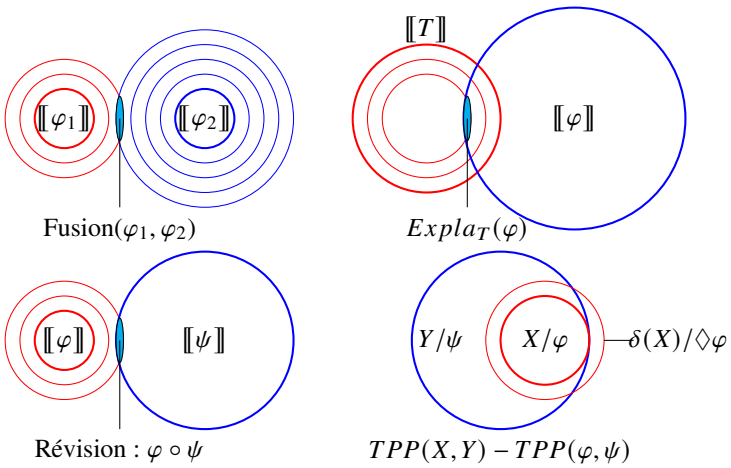


FIGURE 1 – Illustration du principe proposé : fusion, révision, abduction, relation TPP. Les modèles des formules sont représentés par des ensembles. La fusion de φ_1 et φ_2 résulte de dilatations successives de l'ensemble des modèles de chacune des formules, jusqu'à ce qu'elles soient cohérentes. La révision de φ par ψ est obtenue en dilatant l'ensemble des modèles de φ jusqu'à rencontrer ceux de ψ, et ce sont donc les modèles de ψ les plus proches de ceux de φ qui sont retenus. Pour les explications, il s'agit au contraire de réduire l'ensemble des modèles de T, par des rétractions, tout en préservant la cohérence avec φ. Les explications sont alors les modèles de φ qui sont aussi les plus centraux possibles de la théorie. La relation TPP, une des relations de la théorie RCC-8, entre φ et ψ (expressions logiques d'entités spatiales) peut-être exprimée par l'inclusion de X (représentant les modèles de φ) dans Y (modèles de ψ) mais pas celle de la dilatation de X dans Y (dès que l'on dilate un peu X, on rencontre le complémentaire de Y). Les propositions de cet article permettent de généraliser ces opérations dans le cadre des topos.

intuitioniste, une telle forme normale pour les formules n'existe pas, et donc, quand le topos $\mathcal{C}$ n'est pas booléen, nous proposons la définition suivante. Tout d'abord, définissons deux applications $\rho, \kappa : \Phi \to \Phi$ sur les formules (ces dernières seront aussi utiles pour traiter l'abduction) :

- $\rho(\top) = \top$ et $\kappa(\top) = \top$
- $\rho(\bot) = \bot$ et $\kappa(\bot) = \bot$
- $\rho(p) = p$ et $\kappa(p) = p$ pour $p \in PV$.
- $\rho(\varphi \Rightarrow \psi) = (\kappa(\varphi) \Rightarrow \psi) \vee (\varphi \Rightarrow \rho(\psi))$ et $\kappa(\varphi \Rightarrow \psi) = (\varphi \Rightarrow \kappa(\psi)) \vee (\rho(\varphi) \Rightarrow \psi)$
- $\rho(\varphi \mathbin{@} \psi) = (\rho(\varphi) \mathbin{@} \psi) \vee (\varphi \mathbin{@} \rho(\psi))$ et $\kappa(\varphi \mathbin{@} \psi) = (\kappa(\varphi) \mathbin{@} \psi) \vee (\varphi \mathbin{@} \kappa(\psi))$ avec $@ \in \{\wedge, \vee\}$
- $\rho(\neg\varphi) = \neg\kappa(\varphi)$ et $\kappa(\neg\varphi) = \neg\rho(\varphi)$
- $\rho(\Box\varphi) = \Diamond\varphi$ et $\kappa(\Box\varphi) = \Box\kappa(\varphi)$
- $\rho(\Diamond\varphi) = \Diamond\rho(\varphi)$ et $\kappa(\Diamond\varphi) = \Box\varphi$

Proposition 6.2 *Pour toute formule φ et tout modèle $\mathcal{M} = (X, N, \nu)$, nous avons $[\![M]\!](\varphi) \preceq_X [\![M]\!](\rho(\varphi))$ et $[\![M]\!](\kappa(\varphi)) \preceq_X [\![M]\!](\varphi)$.*

Définissons alors l'application $\tau : \Phi \to \Phi$ qui sera à la base de notre nouvel opérateur de révision. Soit χ une tautologie.

$$\tau : \begin{cases} \Phi & \to & \Phi \\ \varphi & \mapsto & \begin{cases} \chi & \text{si } \rho(\varphi) = \varphi \\ \rho(\varphi) & \text{sinon} \end{cases} \end{cases} \quad (1)$$

Proposition 6.3 *L'application τ satisfait pour toute formule cohérente $\varphi \in \Phi$:*

- **Extensivité.** *Pour toute formule φ, nous avons : $Mod(\varphi) \subseteq Mod(\tau(\varphi))$.*
- **Exhaustivité.** *Il existe $k \in \mathbb{N}$ tel que $Mod(\tau^k(\varphi)) = Mod$.*

Dans [2], les applications satisfaisant de telles conditions sont appelées des relaxations.

Nous obtenons alors l'opérateur de révision $\circ$:

$$\varphi \circ \psi = \tau^n(\varphi) \wedge \psi$$

où $n = \min\{k \in \mathbb{N} \mid \tau^k(\varphi) \wedge \psi \text{ est cohérente}\}$ et $\tau^n(\varphi) = \underbrace{\tau(\ldots\tau(\varphi)\ldots)}_{n\ fois}$.

Proposition 6.4 *L'opérateur $\circ$ satisfait les axiomes $(G1)$ – $(G3)$, $(G4')$, et $(G5)$.*

En s'appuyant sur des résultats généraux établis dans [2], nous montrons aussi dans [6] que nos opérateurs de révision induisent des transformations minimales au sens du

théorème de représentation de [26]. Comme illustré sur la figure 1, les dilatations successives induisent un ordre sur les modèles, et les modèles minimaux de ψ au sens de cet ordre sont retenus (les plus proches de φ).

6.2 Fusion

Quand les croyances jouent des rôles symétriques, un autre problème largement abordé est celui de la fusion de croyances. Soient m formules $\varphi_1, \ldots, \varphi_m$ définissant des croyances d'agents. Comme pour la révison, leur fusion peut se définir simplement à partir d'une succession de dilatations. Dans notre cadre, cela s'écrit :

$$Fusion(\varphi_1, \ldots, \varphi_m) = \tau^n(\varphi_1) \wedge \ldots \wedge \tau^n(\varphi_m)$$

où $n = \min\{k \in \mathbb{N} \mid \bigwedge_{i=1}^{m} \tau^k(\varphi_i) \text{ est cohérente}\}$, et τ est défini par l'équation 1. Il a été montré dans [16, 17] que les fusions de cette forme sont équivalentes aux opérateurs de fusion définis à partir de fonctions d'agrégation et de distances spécifiques, et satisfont l'ensemble des postultats de rationalité introduits dans [27].

6.3 Abduction

L'abduction est le procédé qui consiste, étant données une théorie T et une observation φ, à trouver la meilleure explication ψ telle que $T \cup \{\psi\} \models \varphi$. Les explications possibles de φ selon T sont multiples (voir en nombre infini). Dans une approche logique, suivant nos précédents travaux [3] où nous avons étudié l'abduction indépendamment d'une logique donnée, nous étudions aussi ici l'abduction comme un procédé d'inférence. Intuitivement, trouver une explication à φ selon T consiste à couper dans la classe des modèles de T tout en restant cohérent avec φ. Ainsi, l'abduction peut être vue comme une l'érosion, et donc ce procédé consistera à éroder $Mod(T)$ autant que possible tout en conservant des proriétés de minimalité. Nous proposons alors d'instancier l'approche développée dans [3] à notre cadre [6]. Nous définissons l'application $\zeta : \Phi \to \Phi$, χ étant une antilogie, par :

$$\zeta : \left\{ \begin{array}{ccc} \Phi & \to & \Phi \\ \varphi & \mapsto & \left\{ \begin{array}{ll} \chi & \text{si } \kappa(\varphi) = \varphi \\ \kappa(\varphi) & \text{sinon} \end{array} \right. \end{array} \right.$$

où κ est l'application précédemment définie dans la section 6.1.

Proposition 6.5 *L'application ζ satisfait pour toute formule cohérente $\varphi \in \Phi$:*

— **Anti-extensivité.** $Mod(\zeta(\varphi)) \subseteq Mod(\varphi)$.

— **Vacuum.** *Il existe $k \in \mathbb{N}$ tel que $Mod(\zeta^k(\varphi)) = \emptyset$.*

6. Dans [3], l'approche a déjà été appliquée à la logique modale mais dans le cadre ensembliste. Ici, comme précédemment, nous devons adapter l'instanciation au raisonnement intuitioniste.

Dans [3], de telles applications sont appelées des rétractions.

Suivant [3], à partir de ζ nous définissons deux familles de classes de modèles C_{lcr} et C_{lnr} de la façon suivante [7], T étant un ensemble fini de formules et φ une formule tels que $T \cup \{\varphi\}$ est cohérent :

$$C_{lcr}^{\varphi} = \{Mod(\zeta^k(\bigwedge T) \wedge \varphi) \mid k \in \mathbb{N}, Mod(\zeta^k(\bigwedge T) \wedge \varphi) \neq \emptyset\}$$

$$C_{lnr}^{\varphi} = \{Mod(\zeta^k(\bigwedge T \wedge \varphi)) \mid k \in \mathbb{N}, Mod(\zeta^k(\bigwedge T \wedge \varphi)) \neq \emptyset\}$$

où $\bigwedge T = \varphi_1 \wedge \ldots \wedge \varphi_n$ si $T = \{\varphi_1, \ldots, \varphi_n\}$.

Nous pouvons montrer assez simplement que C_{lcr}^{φ} et C_{lnr}^{φ} sont fermés par inclusion, et sont des ensembles bien fondés. Dans [3], de telles sous-familles de modèles s'appellent des coupures.

Ces deux coupures permettent de définir les deux relations d'explicabilité suivantes :

$$\varphi \rhd_{C_{lcr}} \psi \iff \left\{ \begin{array}{l} Mod(T \cup \{\psi\}) \neq \emptyset, \text{ et} \\ Mod(T \cup \{\psi\}) \subseteq Mod(\zeta^n(T) \cup \{\varphi\}) \end{array} \right.$$

où $n = \sup\{k \in \mathbb{N} \mid Mod(\zeta^k(T) \cup \{\varphi\}) \neq \emptyset\}$;

$$\varphi \rhd_{C_{lnr}} \psi \iff \left\{ \begin{array}{l} Mod(T \cup \{\psi\}) \neq \emptyset, \text{ et} \\ Mod(T \cup \{\psi\}) \subseteq Mod(\zeta^m(T \cup \{\varphi\})) \end{array} \right.$$

où $m = \sup\{k \in \mathbb{N} \mid Mod(\zeta^k(T \cup \{\varphi\})) \neq \emptyset\}$.

À partir directement des théorèmes 2, 3 et 4 dans [3], nous obtenons que ces relations satisfont tout ou partie des postulats de rationalité définis dans [33] que l'on résume dans la table 1. Rappelons ces postualts de rationalité (plus de détails peuvent être touvés dans [3]) :

LLE :	si $\vdash_T \alpha \leftrightarrow \alpha'$ et $\alpha \rhd \gamma$ alors $\alpha' \rhd \gamma$.
RLE :	si $\vdash_T \gamma \leftrightarrow \gamma'$ et $\alpha \rhd \gamma$ alors $\alpha \rhd \gamma'$.
E-CM :	si $\alpha \rhd \gamma$ et $\gamma \vdash_T \beta$ alors $(\alpha \wedge \beta) \rhd \gamma$.
E-C-Cut :	si $\forall \delta$ $[\alpha \rhd \delta \implies \delta \vdash_T \beta]$ et $(\alpha \wedge \beta) \rhd \gamma$ alors $\alpha \rhd \gamma$.
RS :	si $\alpha \rhd \gamma$, $\gamma' \vdash_T \gamma$ et $\gamma' \nvdash_T \bot$ alors $\alpha \rhd \gamma'$.
ROR :	si $\alpha \rhd \gamma$ et $\alpha \rhd \delta$ alors $\alpha \rhd (\gamma \vee \delta)$.
E-Reflexivity :	si $\alpha \rhd \gamma$ alors $\gamma \rhd \gamma$.
E-Con :	$\nvdash_T \neg\alpha$ ssi il existe γ tel que $\alpha \rhd \gamma$.

6.4 Raisonnement spatial

Les approches topologiques appliquées au raisonnement spatial qualitatif décrivent des relations entre régions spatiales. Ici, nous proposons d'appliquer notre morphologique au domaine de la méréotopologie, plus spécifiquement le modèle RCC-8 [34]. Cette théorie permet de définir plusieurs relations topologiques à partir d'un prédicat de base C de connectivité. C'est pourquoi dans la littérature, RCC-8 a reçu une axiomatisation dans la logique du premier ordre. Rappelons les huit relations de RCC-8 :

7. *lcr* pour *last consistent retraction* et *lnr* pour *last non-trivial retraction*

Postulats de rationalité	$\rhd_{C_{lcr}}$	$\rhd_{C_{lnr}}$
LLE et RLE	√	√
RS	√	√
E-Con	√	√
ROR	√	√
E-Reflexivity	√	√
E-CM	√	
E-C-Cut	√	

TABLE 1 – Liens entre les postulats de rationalité de [33] et les propriétés satisfaites par $\rhd_{C_{lcr}}$ et $\rhd_{C_{lnr}}$.

— **Déconnexion** DC**.** $DC(X,Y)$ signifie que la région X est déconnectée de la région Y ;
— **Connexion externe** EC**.** $EC(X,Y)$ signifie que X est connectée de façon externe à Y ;
— **Chevauchement partiel** PO**.** $PO(X,Y)$ signifie que X et Y ont une intersection commune qui ne recouvre pas X ni Y ;
— **Partie propre tangentielle (resp. inverse)** TPP **(resp.** TPP_i**).** $TPP(X,Y)$ (resp. $TPP_i(X,Y)$) signifie que X (resp. Y) est une partie tangentielle propre de Y (resp. de X) ;
— **Partie propre non-tangentielle (resp. inverse)** $NTPP$ **(resp.** $NTPP_i$**).** $NTPP(X,Y)$ (resp. $NTPP_i(X,Y)$) signifie que X (resp. Y) est une partie non-tangentielle de Y (resp. de X) ;
— **Égalité** EQ**.** $EQ(X,Y)$ signifie que X et Y sont des régions identiques.

Comme il a été montré dans [4, 11, 12], la morpho-logique telle que définie dans cet article permet une axiomatisation plus simple de certaines de ces relations.

Soit $\mathcal{M} = (X, N, \nu)$ un modèle. Les sous-objets de X sont des entités spatiales (i.e. des régions), et les formules sont des combinaisons de telles entités. De là, on a la formalisation suivante des relations RCC-8 :

— $C(X,Y) : \varphi \wedge \psi$;
— $DC(X,Y) : \neg(\varphi \wedge \psi)$;
— $EC(X,Y) : \neg(\varphi \wedge \psi)$ et $\Diamond\varphi \wedge \psi$ et $\varphi \wedge \Diamond\psi$;
— $PO(X,Y) : \varphi \wedge \psi$ et $\varphi \wedge \neg\psi$ et $\neg\varphi \wedge \psi$;
— $TPP(X,Y) : \varphi \Rightarrow \psi$ et $\Diamond\varphi \wedge \neg\psi$;
— $NTPP(X,Y) : \varphi \Rightarrow \psi$ et $\varphi \Rightarrow \Box\psi$;
— $EQ(X,Y) : \varphi \Leftrightarrow \psi$.

où φ et ψ sont des formules qui définissent respectivement les régions X et Y. Ainsi, les formules suivantes traduisent :

— Pour EC. Les deux régions X et Y ne s'intersectent pas mais dès que l'une des deux est dilatée (par l'opérateur $\Diamond$) alors l'intersection devient non vide.
— Pour TPP. X est inclus dans Y mais la dilatation de X (représentée par $\Diamond\varphi$) ne l'est plus.
— Pour $NTPP$. X est inclus dans Y et le reste même si l'on érode la région Y.

Les autres relations sont naturelles dans leur traduction.

7 Conclusion

Les contributions de cet article sont les suivantes :
— extension de la MM aux voisinages structurants, et ce dans le cadre de la théorie des topos élémentaires ;
— sémantique toposique par voisinage aux logiques modales constructives IT et CS4 ;
— défintion d'un calcul correct et complet ;
— définition d'opérateurs de révision, de fusion et d'abduction dédiés à la logique modale toposique, la morpho-logique ;
— application de la morpho-logique au domaine de la méréotopologie, plus spécifiquement le modèle RCC-8.

Plusieurs perspectives s'offrent à nous. Par la bijection $Hom(X, PY) \simeq \mathrm{Sub}(X \times Y)$, les objets et les voisinages structurants peuvent se voir comme des applications qui associent à chaque élément x les éléments en relation avec lui. Une généralisation de cela peut être obtenue au travers de la notion de co-algèbre principalement étudiée dans le cadre ensembliste. Dans ce cadre, la notion de *predicate lifting* a émergé comme concept sous-jacent à la sémantique des opérateurs modaux. Il serait alors intéressant de généraliser cette notion à notre cadre toposique, comme nous avons commencé à le faire dans [5].

Une autre extension serait aussi de voir comment étendre la morpho-logique au cadre flou. En effet, il est habituel d'introduire de l'incertitude dans le raisonnement.

Références

[1] Aiello, M. et B. Ottens: *The Mathematical morphological view on reasoning about space*. Dans *International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 205–211, 2007.

[2] Aiguier, M., J. Atif, I. Bloch et C. Hudelot: *Belief Revision, Minimal Change and Relaxation : A General Framework based on Satisfaction Systems, and Applications to Description Logics*. Artificial Intelligence, 256 :160–180, 2018.

[3] Aiguier, M., J. Atif, I. Bloch et R. Pino Pérez: *Explanatory relations in arbitrary logics based on satisfaction systems, cutting and retraction*. International Journal of Approximate Reasoning, 102 :1–20, 2018.

[4] Aiguier, M. et I. Bloch: *Logical Dual Concepts based on Mathematical Morphology in Stratified Institutions : Applications to Spatial Reasoning*. Journal of Applied Non-Classical Logics, 29(4) :392–429, 2019.

[5] Aiguier, M. et I. Bloch: *Abstract Categorical Logic*. Logica Universalis, 2022.

[6] Aiguier, M., I. Bloch, S. Nibouche et R. Pino-Pérez: *Morpho-logic from a topos perspective : Application to symbolic AI*. Rapport technique arXiv 2303.04895, arXiv cs.AI, 2023.

[7] Alchourron, C., P. Gardenfors et D. Makinson: *On the logic of theory change*. Journal of Symbolic Logic, 50(2) :510–530, 1985.

[8] Alechina, N., M. Mendler, V. de Paiva et E. Ritter: *Categorical and Kripke Semantics for Constructive S4 Modal Logic*. Dans *15th International Workshop on Computer Science Logic*, tome LNCS 2142. Springer-Verlag, 2003.

[9] Barr, M. et C. Wells: *Toposes, triples and theories*. Springer-Verlag, 1985.

[10] Barr, M. et C. Wells: *Category Theory for Computing Science*. Prentice-Hall, 1990.

[11] Bloch, I.: *Modal logics based on mathematical morphology for qualitative spatial reasoning*. Journal of Applied Non-Classical Logics, 12(3-4) :399–423, 2002.

[12] Bloch, I., S. Blusseau, R. Pino Pérez, E. Puybareau et G. Tochon: *On Some Associations Between Mathematical Morphology and Artificial Intelligence*. Dans *International Conference on Discrete Geometry and Mathematical Morphology*, tome LNCS 12708, pages 457–469. Springer-Verlag, 2021.

[13] Bloch, I. et A. Bretto: *Mathematical morphology on hypergraphs, application to similarity and positive kernel*. Computer Vision and Image Understanding, 117 :342–354, 2013.

[14] Bloch, I., A. Bretto et A. Leborgne: *Robust Similarity between Hypergraphs based on Valuations and Mathematical Morphology Operators*. Discrete Applied Mathematics, 183 :2–19, 2015.

[15] Bloch, I. et J. Lang: *Towards Mathematical Morpho-Logics*. Dans Bouchon-Meunier, B., J. Gutierrez-Rios, L. Magdalena et R. Yager (rédacteurs) : *Technologies for Constructing Intelligent Systems*, pages 367–380. Springer-Verlag, 2002.

[16] Bloch, I., J. Lang, R. Pino Pérez et C. Uzcátegui: *Morphologic for knowledge dynamics : revision, fusion, abduction*. Rapport technique arXiv 1802.05142, arXiv cs.AI, 2018.

[17] Bloch, I., R. Pino-Pérez et C. Uzcategui: *A unified Treatment of Knowledge Dynamics*. Dans *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 329–337. AAAI Press, 2004.

[18] Caramello, O. et L. Lafforgue: *Ontologies, knowledge representations and Grothendieck toposes*. Dans *Invited talk to Semantics Workshop, Lagrange Center, Huawei*, 2022.

[19] Cousty, J., L. Najman, F. Dias et J. Serra: *Morphological filtering on graphs*. Computer Vision and Image Understanding, 117 :370–385, 2013.

[20] Cousty, J., L. Najman et J. Serra: *Some morphological operators in graph spaces*. Dans Wilkinson, M. et J. Roerdink (rédacteurs) : *International Symposium on Mathematical Morphology (ISSM09)*, tome LNCS 5720, pages 149–160. Springer-Verlag, 2009.

[21] Dias, F., J. Cousty et L. Najman: *Some morphological operators on simplicial complex spaces*. Dans *Discrete Geometry for Computer Imagery (DGCI11)*, tome LNCS 6607, pages 441–452. Springer-Verlag, 2011.

[22] Gorogiannis, N. et A. Hunter: *Merging First-Order Knowledge using Dilation Operators*. Dans *Fifth International Symposium on Foundations of Information and Knowledge Systems, FoIKS'08*, tome LNCS 4932, pages 132–150, 2008.

[23] Goy, A., M. Aiguier et I. Bloch: *From Structuring Elements to Structuring Neighborhood Systems*. Dans *Mathematical Morphology and Its Applications to Signal and Image Processing - 14th International Symposium, ISMM 2019*, tome LNCS 11564, pages 16–28, 2019.

[24] Grothendieck, A.: *Sur quelques points d'algèbre homologique*. Tohoku Mathematical Journal, 9 :119–221, 1957.

[25] Johnstone, P.: *Sketches of an Elephant : A Topos Theory Compendium. Vol1. and Vol.2*. Oxford University Press, 2002.

[26] Katsuno, H. et A. O. Mendelzon: *Propositional knowledge base revision and minimal change*. Artificial Intelligence, 52 :263–294, 1991.

[27] Konieczny, S. et R. Pino Pérez: *Logic based merging*. Journal of Philosophical Logic, 40(2) :239–270, 2011, ISSN 0022-3611.

[28] Lawvere, F. W.: *Introduction to Toposes, Algebraic Geometry and Logic*. Dans *Halifax Conference*, tome 274 de *Lecture Notes in Mathematics*, pages 1–12. Springer, 1972.

[29] MacLane, S.: *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[30] Menni, M. et C. Smith: *Modes of Adjointness*. Journal of Philosophical Logic, 42(1), 2013.

[31] Meyer, F. et J. Stawiaski: *Morphology on graphs and minimum spanning trees*. Dans Wilkinson, M. et J. Roerdink (rédacteurs) : *International Symposium on Mathematical Morphology (ISSM09)*, tome LNCS 5720, pages 161–170. Springer-Verlag, 2009.

[32] Pacuit, E.: *Neighborhood semantics for modal logic*. Springer, 2017.

[33] Pino-Pérez, R. et C. Uzcátegui: *Jumping to explanation versus jumping to conclusions*. Artificial Intelligence, 111(2) :131–169, 1999.

[34] Randell, D., Z. Cui et A. Cohn: *A Spatial Logic based on Regions and Connection*. Dans Nebel, B., C. Rich et W. Swartout (rédacteurs) : *Principles of Knowledge Representation and Reasoning KR'92*, pages 165–176, San Mateo, CA, 1992. Kaufmann.

[35] Serra, J.: *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[36] Serra, J.: *Image Analysis and Mathematical Morphology. Part II : Theoretical Advances*. Academic Press, 1988.

[37] Vincent, L.: *Graphs and Mathematical morphology*. Signal Processing, 16(4) :365–388, 1989.

[38] Wijesekera, D.: *Constructive modal logic I*. Annals of Pure and Applied Logic, 50, 1990.

Actes JIAF-JFPDA 2023

Décrire et quantifier la contradiction entre des éléments de preuve via la logique de Belnap-Dunn et la théorie de Dempster-Shafer

Marta Bílková[1] **Sabine Frittella**[2] **Daniil Kozhemiachenko**[2]
Ondrej Majer[3] **Krishna Manoorkar**[4]

[1] The Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic
[2] INSA Centre Val de Loire, Univ. Orléans, LIFO EA 4022, France
[3] The Czech Academy of Sciences, Institute of Philosophy, Prague, Czech Republic
[4] Vrije University, Amsterdam, Netherlands

bilkova@cs.cas.cz sabine.frittella@insa-cvl.fr
daniil.kozhemiachenko@insa-cvl.fr majer@flu.cas.cz
k.b.manoorkar@vu.nl

Résumé

Les fonctions de croyance sont une généralisation des fonctions de probabilité qui permettent de coder une incertitude sur la probabilité d'un événement en fournissant des bornes inférieure et supérieure sur sa probabilité. La théorie des fonctions de croyance (aussi appelée théorie de Dempster-Shafer) permet d'encoder des éléments de preuve via des fonctions de croyance et de les combiner. La Logique de Belnap-Dunn (LBD) est une logique à quatre valeurs introduite pour modéliser le raisonnement avec des informations incomplètes ou contradictoires. Dans ce travail, nous montrons comment la théorie de Dempster-Shafer peut être utilisée avec LBD afin de formaliser un raisonnement avec des éléments de preuve incomplets et/ou contradictoires.

Abstract

Belief functions are a generalisation of probability functions that allow encoding uncertainty on the probability of an event by providing a lower and an upper bound on its probability. Belief function theory (also called Dempster-Shafer theory) proposes ways to encode pieces of evidence via belief functions and to combine them. Belnap-Dunn Logic (LBD) is a four-valued logic introduced to model reasoning with incomplete or contradictory information. In this work, we show how Dempster-Shafer theory can be used over LBD in order to formalise reasoning with incomplete and/or contradictory pieces of evidence.

Nous présentons d'abord un bref état de l'art sur la théorie de Dempster-Shafer, la logique de Belnap-Dunn et les probabilités paraconsistantes. Puis nous discutons nos axes de recherche.

Combinaison de preuves et gestion du conflit dans la théorie de Dempster-Shafer. Dans la théorie de Dempster-Shafer (DS), l'information fournie par une source est encodée via une fonction de croyance. La combinaison de l'information provenant de sources conflictuelles ou contradictoires est un sujet d'étude majeur [6, 8, 7, 3]. Pour appliquer la règle de combinaison originale de Dempster (règle DS) [6], on suppose que les sources sont complètement fiables, et donc tout conflit entre elles est considéré comme impossible. Zadeh [8] donne un exemple montrant que la règle DS peut conduire à des résultats contre-intuitifs lorsqu'elle est utilisée pour agréger des informations qui ne sont pas entièrement fiables et avec un degré important de conflit entre eux.

Plusieurs modifications de la règle DS ont été proposées et étudiées dans la littérature pour agréger des informations provenant à la fois de sources peu fiables et fortement contradictoires. [6] décrit la méthode d'*affaiblissement* pour gérer les conflits. Dans cette méthode, lorsque les sources ont un conflit entre elles, l'analyste affaiblie l'information transmise par les sources en fonction de leur fiabilité avant d'utiliser la règle DS. [7] propose une règle de combinaison similaire à la règle DS mais la masse attachée aux informations contradictoires est assignée à l'ensemble du cadre de discernement. Autrement dit, avoir des preuves contradictoires est considéré comme équivalent à ne pas avoir d'informations. [3] propose de considérer que, si deux sources attachent des masses aux ensembles A et

B, avec $A \cap B = \emptyset$, alors, lors de la combinaison, la masse $m(A) \cdot m(B)$ est attachée à l'ensemble $A \cup B$. Intuitivement, cela correspond à l'idée que si les sources sont contradictoires, alors l'analyste conclut qu'au moins l'une d'entre elles est correcte.

Dans ce travail, nous utilisons une extension de la Logique de Benlap Dunn (LBD) pour représenter et combiner des preuves contradictoires.

Logique de Belnap-Dunn. LBD a été introduit pour raisonner sur l'information disponible au sujet d'un énoncé plutôt que sur la vérité de cet énoncé [1]. En logique classique, un énoncé p est soit *vrai* soit *faux*, ce qui signifie que p est *vrai* (resp. faux) ssi l'énoncé p est vrai (resp. faux) dans le monde réel. Dans LBD, un énoncé p est soit "étayé par les informations disponibles", soit "contredit par les informations disponibles", soit "ni étayé ni contredit par les informations disponibles", soit "à la fois étayé et contredit par les informations disponibles". Ces quatre valeurs de vérité sont respectivement notées **T** (*true*), **F** (*false*), **N** (*neither*), **B** (*both*) et sont interprétées sur l'algèbre de De Morgan à **4** éléments (Figure 1).

Les quatre éléments ordonnés de bas en haut définissent le *treillis de la vérité*. En passant de **F** à **T**, on passe d'une situation où l'information disponible soutient pleinement la fausseté de l'énoncé, à une situation où l'information disponible soutient pleinement la véracité de l'énoncé. Les quatre éléments ordonnés de gauche à droite définissent le *treillis de l'information*. En passant de **N** à **B**, on passe d'une situation où il n'y a pas d'information sur l'énoncé, à une situation où l'information est contradictoire.

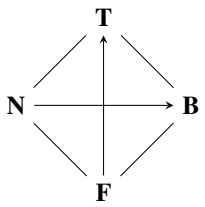


FIGURE 1 – Carré de Belnap-Dunn

Une logique où $p \vee \neg p$ (resp. $p \wedge \neg p$) n'est pas un axiome est appelée *paracomplète* (resp. *paraconsistante*). LBD est un affaiblissement de la logique classique propositionnelle qui est à la fois paracomplète et paraconsistante.

Probabilités paraconsistantes. Dans le cas classique, $\mathrm{p}(\phi)$ (resp. $\mathrm{p}(\neg\phi)$) encode la probabilité que ϕ soit vrai (resp. faux). [4] introduit des probabilités paraconsistantes qui décrivent l'information disponible sur ϕ via quatre nombres (b, d, u, c). Ils encodent le degré de croyance b, d'incrédulité d, d'incertitude u (ignorance) et de conflit c (contradiction) à propos de ϕ. [5] présente une extension probabiliste de LBD avec une axiomatisation correcte et complète.

Projet en cours. Dans [2], nous introduisons des fonctions de croyance sur des modèles de Belnap-Dunn et présentons des logiques pour raisonner à la fois avec des probabilités et des fonctions de croyance sur LBD. Ceci est un premier pas vers la compréhension des probabilités (imprécises) dans un cadre paracomplet et paraconsistant.

Dans ce travail, nous montrons comment des situations, où des informations hautement contradictoires sont disponibles, peuvent être formalisées en utilisant la théorie de Dempster-Shafer et la logique LBD. Tout d'abord, nous expliquons comment encoder des preuves via des fonctions de masse sur des modèlesde Belnap-Dunn et comment interpréter les fonctions de croyance et de plausibilité qui en résultent. Ensuite, nous discutons d'une variation de la règle DS sur les modèles de Belnap-Dunn et de son interprétation dans LBD. Enfin, nous introduisons différentes notions de support d'un énoncé qui induisent différentes fonctions de croyance sur des formules de la logique LBD, et nous montrons que certaines d'entre elles permettent de déduire des ensembles de probabilités classiques basés sur des fonctions de masse sur des modèles de Belnap-Dunn.

Références

[1] Belnap, N.D.: *How a Computer Should Think*. Dans Omori, H. et H. Wansing (rédacteurs) : *New Essays on Belnap-Dunn Logic*, tome 418 de *Synthese Library (Studies in Epistemology, Logic, Methodology, and Philosophy of Science)*. Springer, Cham, 2019.

[2] Bílková, M., S. Frittella, D. Kozhemiachenko, O. Majer et S. Nazari: *Reasoning with belief functions over Belnap–Dunn logic*. `https://arxiv.org/abs/2203.01060`, 2022.

[3] Dubois, D. et H. Prade: *Representation and combination of uncertainty with belief functions and possibility measures*. Computational intelligence, 4 :244–264, 1988.

[4] Dunn, J.M.: *Contradictory information : Too much of a good thing*. Journal of Philosophical Logic, 39 :425–452, 2010.

[5] Klein, D., O. Majer et S. Rafiee Rad: *Probabilities with Gaps and Gluts*. Journal of Philosophical Logic, 50(5) :1107–1141, 2021.

[6] Shafer, G.: *A mathematical theory of evidence*. Princeton university press, 1976.

[7] Yager, R.R.: *On the Dempster-Shafer framework and new combination rules*. Information sciences, 41(2) :93–137, 1987.

[8] Zadeh, L.A.: *On the validity of Dempster's rule of combination of evidence*. Infinite Study, 1979.

Actes JIAF-JFPDA 2023

Une logique pour représenter des variations propositionnelles *

Nicolas François[1] **Thomas Laure**[2] **Jean Lieber**[1]

1 Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
2 DIENS, École Normale Supérieure, CNRS, Université PSL, 75005 Paris, France
nicolas.francois@loria.fr thomas.laure@ens.psl.eu jean.lieber@loria.fr

Résumé

La logique propositionnelle (comme d'autres logiques) peut être vue comme une façon de représenter des ensembles d'états du monde (les interprétations). La représentation de variations d'un ensemble d'états du monde à un autre est motivée par des travaux sur le raisonnement à partir de cas : la comparaison entre deux problèmes et le passage d'une solution à une autre peuvent être vus comme des variations d'un ensemble d'états à un autre. Cela a conduit à une notation pour représenter ces variations (une syntaxe) et cet article étudie comment associer à cette notation une sémantique en théorie des modèles dans laquelle une interprétation est un couple d'interprétations en logique propositionnelle (un des états et l'autre). L'article entame une étude classique de cette logique (syntaxe, sémantique, équivalences, NP-complétude de la satisfiabilité, etc.) et de façons d'associer à un couple de formules propositionnelles une formule de cette logique. L'article se termine par une discussion envisageant une poursuite de cette étude et des applications potentielles.

Abstract

Propositional logic (as other logics) can be seen as a way to represent a set of states of the world (the interpretations). Representation of variations from a set of states of the world to another is motivated by studies on case-based reasoning: the comparison of two problems and the way a solution is transformed into another solution can be seen as variations from a set of states to another. This has led to a notation for representing these variations (a syntax) and this article studies how to associate to this notation a semantics in model theory in which an interpretation is an ordered pair of interpretations in propositional logic (one of the states and the other one). The article initiates a classical study of this logic (syntax, semantics, equivalences, NP-completeness of satisfiability, etc.) and a way to associate to an ordered pair of propositional formulas a formula of this logic. The article ends with a discussion presenting directions of future work and a presentation of potential applications.

*Les auteurs tiennent à remercier les relecteurs de cet article pour leurs remarques constructives et encourageantes ainsi qu'Henri Prade pour plusieurs conseils judicieux qui ont été utiles à cet article.

1 Introduction et motivations

Le raisonnement à partir de cas (RàPC [15]) est un raisonnement s'appuyant sur une base de cas $\mathtt{BC}$ où un cas est, en général, la donnée d'un couple $(\mathtt{x}, \mathtt{y})$ où $\mathtt{x}$ représente un problème (du domaine d'application considéré) et $\mathtt{y}$ est une solution de ce problème. Une session de RàPC a en entrée un problème à résoudre $\mathtt{x}^{\mathtt{cible}}$ (le « problème cible ») et est souvent constituée de deux étapes : la remémoration et l'adaptation. La remémoration consiste à chercher un cas $(\mathtt{x}^s, \mathtt{y}^s) \in \mathtt{BC}$ (cas *source*) tel que $\mathtt{x}^s$ est jugé similaire à $\mathtt{x}^{\mathtt{cible}}$. L'adaptation consiste à modifier $\mathtt{y}^s$ dans l'optique de la résolution du problème cible. Un principe souvent utilisé pour l'adaptation s'appuie sur la notion intuitive de *variations* entre problèmes et entre solutions :

(1) Calcul de la variation entre $\mathtt{x}^s$ et $\mathtt{x}^{\mathtt{cible}}$, notée $\Delta\mathtt{x}$;

(2) Calcul de $\Delta\mathtt{y}$, la variation entre $\mathtt{y}^s$ et la solution cherchée, sur la base de $\Delta\mathtt{x}$;

(3) Calcul de $\mathtt{y}^{\mathtt{cible}}$, solution plausible [1] de $\mathtt{x}^{\mathtt{cible}}$.

L'étape (2) s'appuie typiquement sur des connaissances d'adaptation, souvent représentées par des règles d'adaptation. En général, la prémisse d'une telle règle contient une variation entre problèmes et sa conclusion, une variation entre solutions : « Si deux problèmes varient selon $\Delta\mathtt{x}$ alors leurs solutions varient selon $\Delta\mathtt{y}$. » De telles règles peuvent être apprises à partir de la base de cas, selon un principe introduit dans [10] et qui a été appelé l'heuristique des variations entre cas (*case difference heuristics*, voir par exemple [13] [2]). L'idée est d'appliquer un processus d'apprentissage artificiel avec comme jeu d'apprentissage des couples $(\Delta\mathtt{x}^{ij}, \Delta\mathtt{y}^{ij})$ où $(\mathtt{x}^i, \mathtt{y}^i), (\mathtt{x}^j, \mathtt{y}^j) \in \mathtt{BC}$ et $\Delta\mathtt{x}^{ij}$

1. En général, le RàPC est un raisonnement hypothétique : la solution proposée résout de façon plausible le problème cible.

2. Les auteurs ont hésité entre le terme « différence » et le terme « variation ». L'argument qui a fait préférer le second est qu'on s'attend à ce que la différence entre un objet et lui-même soit indépendante de cet objet, ce qu'on attend moins d'une variation.

(resp., $\Delta \mathbf{y}^{i,j}$) est la variation entre $\mathbf{x}^i$ et $\mathbf{x}^j$ (resp., entre $\mathbf{y}^i$ et $\mathbf{y}^j$).

Dans certaines applications du RàPC, les problèmes et les solutions peuvent être représentés (ou traduits) sous la forme de propriétés booléennes, qu'on peut coder sous la forme de conjonctions de variables booléennes. Par exemple, on peut considérer les deux problèmes suivants :

$$\mathbf{x}^i = a \wedge \neg b \wedge \neg c \wedge d$$
$$\mathbf{x}^j = a \wedge \neg b \wedge c \wedge \neg d$$

La variation entre ces problèmes a été *notée* de la façon suivante dans certains travaux sur l'apprentissage de connaissances d'adaptation s'appuyant sur cette heuristique des variations :

$$\Delta \mathbf{x}^{i,j} = a^{=\mathtt{v}} \wedge b^{=\mathtt{f}} \wedge c^{+} \wedge d^{-}$$

où a^v signifie que la propriété booléenne varie selon v où $v = \mathtt{=v}$ (resp. $v = \mathtt{=f}$) signifie « reste à $\mathtt{v}$ (resp. à $\mathtt{f}$) » et $v = \mathtt{+}$ (resp. $v = \mathtt{-}$) signifie « change de $\mathtt{f}$ à $\mathtt{v}$ » (resp. de $\mathtt{v}$ à $\mathtt{f}$).

À titre d'exemple, l'apprentissage de règles d'adaptation a été étudiée pour un système de RàPC culinaire [5]. Supposons qu'on trouve dans un livre de recettes (constituant la base de cas) des couples de recettes de desserts similaires dont la première contient des pommes et de la cannelle et la seconde, des poires et du chocolat (mais pas de poires ni de chocolat dans la première et pas de pommes ni de cannelle dans la seconde). À partir de ces couples, on peut apprendre la règle d'adaptation suivante :

> Dans les recettes de desserts avec des pommes et de la cannelle, on peut remplacer ces deux ingrédients par des poires et du chocolat.

Cette règle d'adaptation peut s'écrire par l'expression suivante :

$$R = \mathtt{rDessert}^{=\mathtt{v}} \wedge \mathtt{iPomme}^{-} \wedge \mathtt{iCannelle}^{-} \wedge \mathtt{iPoire}^{+} \wedge \mathtt{iChocolat}^{+} \quad (1)$$

(où $\mathtt{rDessert}$ est la propriété d'être une recette de dessert, $\mathtt{iPomme}$, celle d'être une recette avec des pommes, etc.).

Les expressions telles que celles présentées ci-dessus ne constituent à ce stade qu'une *notation*, pas un formalisme logique, puisqu'on dispose d'une syntaxe mais pas de sémantique permettant de définir des inférences. L'objectif de cet article est de définir une telle sémantique et d'entamer son étude. Partant de la logique propositionnelle finie $(\mathcal{LP}, \models)$, sera définie la logique $(\Delta\mathcal{LP}, \models)$ dont la syntaxe contient les formules telles que celle de la notation introduite ci-dessus et la sémantique sera définie en théorie des modèles.

La section 2 rappelle des notions et notations relatives à la logique propositionnelle finie. La section 3 définit la logique $(\Delta\mathcal{LP}, \models)$ et présente quelques résultats élémentaires à son sujet. Étant donné deux formules propositionnelles α et β, comment définir leurs variations par une formule de $(\Delta\mathcal{LP}, \models)$? La section 4 étudie cette question. L'article se termine par une discussion (section 5) qui montre que les directions de recherche pour poursuivre ce travail sont nombreuses.

Remarque. Les résultats de cet article ont tous été démontrés mais certaines preuves ne sont pas incluses dans cet article, soit parce qu'elles sont jugées suffisamment simples soit, à l'inverse, parce qu'elles demandent une preuve détaillée. Le rapport [8] est une version étendue de cet article et comprend ces preuves.

2 Rappels sur la logique propositionnelle

Soit $\mathcal{V}$ un ensemble fini et non vide de symboles, appelés variables. Une formule est soit une variable (*atome* de cette logique) soit une expression d'une des formes suivantes : $\neg\alpha$, $\alpha_1 \wedge \alpha_2$ et $\alpha_1 \vee \alpha_2$, où α, α_1 et α_2 sont des formules. On étend cette syntaxe avec d'autres connecteurs considérés comme des abréviations (où a est une variable) : $\bot = a \wedge \neg a$, $\top = \neg\bot$, $\alpha_1 \rightarrow \alpha_2 = \neg\alpha_1 \vee \alpha_2$, $\alpha_1 \leftrightarrow \alpha_2 = (\alpha_1 \rightarrow \alpha_2) \wedge (\alpha_2 \rightarrow \alpha_1)$ et $\alpha_1 \oplus \alpha_2 = \neg(\alpha_1 \leftrightarrow \alpha_2)$. $\mathcal{LP}$ est l'ensemble des formules de cette logique. Pour éviter d'écrire trop de parenthèses, on utilise la priorité des connecteurs usuelle : $\neg$ avant les autres ; $\wedge$ et $\vee$ avant les autres connecteurs sauf $\neg$. Ainsi $\neg a \vee b \rightarrow c \wedge \neg d$ se lira $((\neg a) \vee b) \rightarrow (c \wedge (\neg d))$. Un littéral est une formule de la forme a ou de la forme $\neg a$ où $a \in \mathcal{V}$. La taille d'une formule α, $|\alpha|$, est le nombre d'occurrences de connecteurs de α.

Une interprétation $\mathcal{I}$ est une fonction de $\mathcal{V}$ dans $\{\mathtt{f}, \mathtt{v}\}$ où $\mathtt{f}$ et $\mathtt{v}$ dénotent respectivement les valeurs booléennes « faux » et « vrai ». L'ensemble des interprétations est dénoté par Ω. Pour $\mathcal{I} \in \Omega$ et $\alpha \in \mathcal{LP}$, on définit $\mathcal{I} \models \alpha$ – « $\mathcal{I}$ satisfait α » – de la façon suivante (pour $a \in \mathcal{V}$ et $\alpha, \alpha_1, \alpha_2 \in \mathcal{LP}$) :

- $\mathcal{I} \models a$ si $\mathcal{I}(a) = \mathtt{v}$;
- $\mathcal{I} \models \neg\alpha$ si $\mathcal{I} \not\models \alpha$;
- $\mathcal{I} \models \alpha_1 \wedge \alpha_2$ si $\mathcal{I} \models \alpha_1$ et $\mathcal{I} \models \alpha_2$.
- $\mathcal{I} \models \alpha_1 \vee \alpha_2$ si $\mathcal{I} \models \alpha_1$ ou $\mathcal{I} \models \alpha_2$ (ou les deux).

Un modèle d'une formule α est une interprétation $\mathcal{I}$ qui satisfait α et l'ensemble des modèles de α est dénoté par $\mathcal{M}(\alpha)$. Une formule α_1 entraîne une formule α_2, noté $\alpha_1 \models \alpha_2$, si $\mathcal{M}(\alpha_1) \subseteq \mathcal{M}(\alpha_2)$. Les formules α_1 et α_2 sont équivalentes, noté $\alpha_1 \equiv \alpha_2$, si $\mathcal{M}(\alpha_1) = \mathcal{M}(\alpha_2)$. Une formule α est une *tautologie* (noté $\models \alpha$) si $\mathcal{M}(\alpha) = \Omega$. Une formule α est *satisfiable* si $\mathcal{M}(\alpha) \neq \emptyset$.

Une façon équivalente de considérer la logique propositionnelle consiste à considérer chaque formule $\alpha \in \mathcal{LP}$ comme une représentation d'un sous-ensemble $\mathcal{M}(\alpha)$ de Ω. La logique propositionnelle peut alors être vue comme l'étude de 2^Ω, l'ensemble des parties de Ω via l'utilisation

d'intersections, d'unions et de complémentaires (l'algèbre de Boole finie $(2^\Omega, \cap, \cup, \overline{\cdot})$).

Soit $\mathcal{B} = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ un ensemble fini de formules. $\bigwedge \mathcal{B}$ dénote $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_p$. $\bigvee \mathcal{B}$ dénote $\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_p$. On écrira $\mathcal{B} \models \beta$ si $\bigwedge \mathcal{B} \models \beta$, pour $\beta \in \mathcal{LP}$.

3 La logique des variations propositionnelles

Cette section définit la logique $(\Delta\mathcal{LP}, \models)$ sur la base de la logique propositionnelle $(\mathcal{LP}, \models)$.

3.1 Syntaxe

Soit $\mathcal{D}$ l'ensemble de symboles suivant :

$$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$$
$$\text{où } \mathcal{D}_0 = \{\texttt{=f}, \texttt{=v}, \texttt{+}, \texttt{-}\} \text{ et } \mathcal{D}_1 = \{\texttt{=}, \texttt{≠}, \texttt{f•}, \texttt{•f}, \texttt{v•}, \texttt{•v}\}$$

Un $v \in \mathcal{D}$ est appelé *symbole de variation* ; si $v \in \mathcal{D}_0$, v est un symbole de variation *primitif*.

$\Delta\mathcal{LP}$ est l'ensemble des expressions obtenues en substituant dans les formules propositionnelles les variables par des expressions a^v où $a \in \mathcal{V}$ et $v \in \mathcal{D}$. Un tel a^v est appelé *atome* de cette logique. Par exemple, si $a, b, c \in \mathcal{V}$, $a^{\texttt{=v}} \vee \neg(b^{\texttt{+}} \wedge c^{\texttt{-}}) \in \Delta\mathcal{LP}$.

Une formule de $(\Delta\mathcal{LP}, \models)$ est un élément de $\Delta\mathcal{LP}$. On définit donc formellement une formule de cette logique comme étant soit un atome, soit d'une des formes suivantes : $\neg\varphi$, $\varphi_1 \wedge \varphi_2$ et $\varphi_1 \vee \varphi_2$. Les connecteurs $\top$, $\bot$, $\rightarrow$ et $\leftrightarrow$ sont utilisés en tant qu'abréviations, comme en logique propositionnelle (par exemple, $\varphi_1 \rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$). La taille d'une formule $\varphi \in \Delta\mathcal{LP}$, notée $|\varphi|$, est le nombre d'occurrences de connecteurs de φ.

3.2 Sémantique

Une interprétation en logique propositionnelle représente un état du monde particulier. Dans la logique des variations, on considérera qu'une interprétation représente un changement d'un état du monde à un autre. Ainsi, on définit une interprétation pour la sémantique de cette logique par un couple d'interprétations pour la sémantique de $(\mathcal{LP}, \models)$. Formellement, soit $\Delta\Omega = \Omega \times \Omega$. On notera un élément $(\mathcal{I}, \mathcal{J}) \in \Delta\Omega$ par $\mathcal{I}\mathcal{J}$ (sans parenthèse ni virgule, pour simplifier).

Étant donné $\mathcal{I}\mathcal{J} \in \Delta\Omega$ et $\varphi \in \Delta\mathcal{LP}$, il reste à définir la relation $\mathcal{I}\mathcal{J} \models \varphi$ (« $\mathcal{I}\mathcal{J}$ satisfait φ »). On commence par le faire sur les atomes (pour $a \in \mathcal{V}$) :

- Avec symboles de variations primitifs :
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{=f}}$ si $\mathcal{I} \not\models a$ et $\mathcal{J} \not\models a$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{=v}}$ si $\mathcal{I} \models a$ et $\mathcal{J} \models a$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{+}}$ si $\mathcal{I} \not\models a$ et $\mathcal{J} \models a$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{-}}$ si $\mathcal{I} \models a$ et $\mathcal{J} \not\models a$;
- Avec symboles de variations non primitifs :
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{=}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{=f}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{=v}}$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{≠}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{+}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{-}}$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{f•}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{=f}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{+}}$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{•f}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{=f}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{-}}$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{v•}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{=v}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{-}}$;
  - $\mathcal{I}\mathcal{J} \models a^{\texttt{•v}}$ si $\mathcal{I}\mathcal{J} \models a^{\texttt{=v}}$ ou $\mathcal{I}\mathcal{J} \models a^{\texttt{+}}$.

La sémantique des connecteurs est définie de façon similaire à ce qu'elle était en logique propositionnelle (pour $\varphi, \varphi_1, \varphi_2 \in \Delta\mathcal{LP}$) :

- $\mathcal{I}\mathcal{J} \models \neg\varphi$ si $\mathcal{I}\mathcal{J} \not\models \varphi$;
- $\mathcal{I}\mathcal{J} \models \varphi_1 \wedge \varphi_2$ si $\mathcal{I}\mathcal{J} \models \varphi_1$ et $\mathcal{I}\mathcal{J} \models \varphi_2$;
- $\mathcal{I}\mathcal{J} \models \varphi_1 \vee \varphi_2$ si $\mathcal{I}\mathcal{J} \models \varphi_1$ ou $\mathcal{I}\mathcal{J} \models \varphi_2$.

Les notions introduites en logique propositionnelle relatives à la sémantique se transposent aisément : notion de modèle, $\mathcal{M}(\varphi) = \{\mathcal{I}\mathcal{J} \in \Delta\Omega \mid \mathcal{I}\mathcal{J} \models \varphi\}$, conséquence logique $\varphi_1 \models \varphi_2$, tautologie, satisfiabilité d'une formule, etc.

Une façon équivalente de considérer la sémantique de cette logique consiste à considérer chaque formule $\varphi \in \Delta\mathcal{LP}$ comme une représentation du sous-ensemble $\mathcal{M}(\varphi)$ de $\Delta\Omega = \Omega \times \Omega$, i.e. d'une relation binaire sur Ω.

3.3 Propriétés

Représentabilité d'un ensemble d'interprétations par une formule. Certaines définitions de formules dans la suite de l'article seront données par leurs ensembles de modèles, donc à l'équivalence logique près. Ce genre de définition est acceptable car tout sous-ensemble de $\Delta\Omega$ est représentable dans $\Delta\mathcal{LP}$:

$$\text{pour tout } A \subseteq \Delta\Omega, \text{ il existe } \varphi \in \Delta\mathcal{LP} \text{ telle que } \mathcal{M}(\varphi) = A \quad (2)$$

Preuve de (2). Soit $\mathcal{I}\mathcal{J} \in \Delta\Omega$, et soit la formule

$$\psi_{\mathcal{I}\mathcal{J}} = \bigwedge \left\{ a^{\texttt{var}(\mathcal{I}(a), \mathcal{J}(a))} \;\middle|\; a \in \mathcal{V} \right\}$$
$$\text{où } \texttt{var} : (\texttt{f}, \texttt{f}) \mapsto \texttt{=f} \quad (\texttt{v}, \texttt{v}) \mapsto \texttt{=v}$$
$$(\texttt{f}, \texttt{v}) \mapsto \texttt{+} \quad (\texttt{v}, \texttt{f}) \mapsto \texttt{-}$$

On démontre d'abord que, pour toute $a \in \mathcal{V}$, $\mathcal{I}\mathcal{J} \models a^{\texttt{var}(\mathcal{I}(a), \mathcal{J}(a))}$ (il suffit de considérer les quatre cas pour s'en convaincre). Inversement, si quelle que soit $a \in \mathcal{V}$, $\mathcal{I}_1\mathcal{J}_1 \models a^{\texttt{var}(\mathcal{I}(a), \mathcal{J}(a))}$ entraîne $\mathcal{I}_1(a) = \mathcal{I}(a)$ et $\mathcal{J}_1(a) = \mathcal{J}(a)$, alors $\mathcal{I}_1\mathcal{J}_1 = \mathcal{I}\mathcal{J}$. Par conséquent, $\mathcal{M}(\psi_{\mathcal{I}\mathcal{J}}) = \{\mathcal{I}\mathcal{J}\}$. Soit alors $\varphi = \bigvee\{\psi_{\mathcal{I}\mathcal{J}} \mid \mathcal{I}\mathcal{J} \in A\}$. On en déduit que $\mathcal{M}(\varphi) = A$. ∎

Équivalences et mise sous forme normale. Les résultats suivants (et leurs preuves) se transposent aisément de résultats de la logique propositionnelle (les formules considérées ci-dessous sont toutes éléments de $\Delta\mathcal{LP}$) :

- Théorème de la déduction et son corollaire :

$$\varphi_1 \models \varphi_2 \quad \text{ssi} \quad \models \varphi_1 \rightarrow \varphi_2$$
$$\varphi_1 \equiv \varphi_2 \quad \text{ssi} \quad \models \varphi_1 \leftrightarrow \varphi_2$$

— Lien entre tautologies et formules satisfiables :

$$\models \varphi \quad \text{ssi} \quad \neg\varphi \text{ est insatisfiable}$$

— Lois de De Morgan.

— Involutivité de $\neg$, commutativité de $\wedge$ et $\vee$, associativité de $\wedge$ et $\vee$, distributivité de $\wedge$ sur $\vee$, distributivité de $\vee$ sur $\wedge$, etc. (toutes ces propriétés étant vérifiées modulo l'équivalence logique).

Les résultats suivants découlent des définitions :

$$\left.\begin{array}{lll} a^{=} \equiv a^{=\mathtt{f}} \vee a^{=\mathtt{v}} & a^{\neq} \equiv a^{+} \vee a^{-} & a^{\mathtt{f}\bullet} \equiv a^{=\mathtt{f}} \vee a^{+} \\ a^{\bullet\mathtt{f}} \equiv a^{=\mathtt{f}} \vee a^{-} & a^{\mathtt{v}\bullet} \equiv a^{=\mathtt{v}} \vee a^{-} & a^{\bullet\mathtt{v}} \equiv a^{=\mathtt{v}} \vee a^{+} \end{array}\right| \quad (3)$$

On peut faire les transformations suivantes sur une formule de $(\Delta\mathcal{LP}, \models)$, transformations qui conservent l'équivalence logique. Dans un premier temps, on peut supprimer tous les atomes a^v où $v \in \mathcal{D}_1$ en utilisant les équivalences (3). Dans un deuxième temps, on peut utiliser une transformation mettant sous une forme normale négative (où $\neg$ n'apparaît que devant les atomes) comme on le fait en logique propositionnelle (en utilisant les lois de De Morgan et l'involutivité de $\neg$ modulo $\equiv$). Enfin, on peut appliquer l'équivalence suivante (de gauche à droite), pour $a \in \mathcal{V}$ et $v \in \mathcal{D}_0$:

$$\neg a^v \equiv \bigvee_{w \in \mathcal{D}_0 \setminus \{v\}} a^w \quad (4)$$

De cette façon, on peut écrire toute formule de $(\Delta\mathcal{LP}, \models)$ comme une formule n'utilisant pas le connecteur $\neg$ ni les symboles de variation non primitifs. L'idée d'une telle transformation est qu'elle pourrait servir de prétraitement dans une procédure de test de satisfiabilité s'appuyant sur la méthode des tableaux sémantiques [16], pour laquelle un conflit serait n'importe quelle paire d'atomes $\{a^v, a^w\}$ où $a \in \mathcal{V}$, $v, w \in \mathcal{D}_0$ et $v \neq w$. Cette idée sera évoquée à nouveau à la section 5.3.

Preuve de (4). Soit $\mathtt{var}^{-1}$ la fonction inverse de la fonction $\mathtt{var}$ introduite dans la preuve de (2). Pour $\mathcal{IJ} \in \Delta\Omega$ on a les équivalences

$$\begin{aligned} \mathcal{IJ} \models \neg a^v \text{ ssi } & \mathcal{IJ} \not\models a^v \\ \text{ssi } & (\mathcal{I}(a), \mathcal{J}(a)) \neq \mathtt{var}^{-1}(v) \\ \text{ssi } & (\mathcal{I}(a), \mathcal{J}(a)) \in \{\mathtt{f}, \mathtt{v}\}^2 \setminus \{\mathtt{var}^{-1}(v)\} \\ \text{ssi } & \mathcal{IJ} \models a^w \text{ pour un } w \in \mathcal{D}_0 \setminus \{v\} \\ \text{ssi } & \mathcal{IJ} \models \bigvee_{w \in \mathcal{D}_0 \setminus \{v\}} a^w \end{aligned}$$

Ce qui permet de conclure. ■

On peut noter que le choix, parmi les symboles de variations, de $\{\mathtt{=f}, \mathtt{=v}, \mathtt{+}, \mathtt{-}\}$ comme ensemble de variations primitives pourrait être changé : il existe d'autres sous-ensembles de $\mathcal{D}$ permettant de générer $\mathcal{D}$ par des connecteurs logiques. Par exemple, c'est le cas de $\{\mathtt{f}\bullet, \bullet\mathtt{f}, \mathtt{v}\bullet, \bullet\mathtt{v}\}$ puisqu'on a les équivalences suivantes :

$$\begin{array}{ll} a^{=\mathtt{f}} \equiv a^{\mathtt{f}\bullet} \wedge a^{\bullet\mathtt{f}} & a^{=\mathtt{v}} \equiv a^{\mathtt{v}\bullet} \wedge a^{\bullet\mathtt{v}} \\ a^{+} \equiv a^{\mathtt{f}\bullet} \wedge a^{\bullet\mathtt{v}} & a^{-} \equiv a^{\mathtt{v}\bullet} \wedge a^{\bullet\mathtt{f}} \end{array}$$

Satisfiabilité et plongement. Soit SAT_Δ le problème de décision qui, étant donné une formule φ de $(\Delta\mathcal{LP}, \models)$ détermine si φ est satisfiable. Le problème SAT_Δ est NP-complet. On peut prouver cela en montrant que SAT_Δ est dans NP puis en montrant que SAT_Δ est NP-difficile.

On peut prouver que SAT_Δ est dans NP en exhibant un algorithme polynomial non déterministe répondant à ce problème : il suffit de construire un algorithme avec un oracle qui choisit une interprétation $\mathcal{IJ}$ qui satisfait φ si φ est satisfiable, et de tester que c'est effectivement le cas.

On peut prouver que SAT_Δ est NP-difficile en montrant que le problème SAT peut être réduit en temps polynomial à SAT_Δ. Étant donné $\alpha \in \mathcal{LP}$, soit $\widehat{\alpha} \in \Delta\mathcal{LP}$ défini en remplaçant toutes les occurrences de variables propositionnelles a dans α par $a^{=\mathtt{v}}$. On appelle $\widehat{\alpha}$ le *plongement* de α. Par exemple, si $\alpha = a \wedge \neg(b \vee c)$ alors, $\widehat{\alpha} = a^{=\mathtt{v}} \wedge \neg(b^{=\mathtt{v}} \vee c^{=\mathtt{v}})$. On peut montrer le résultat suivant, pour $\mathcal{I} \in \Omega$ et $\alpha \in \mathcal{LP}$:

$$\mathcal{I} \models \alpha \quad \text{ssi} \quad \mathcal{II} \models \widehat{\alpha} \quad (5)$$

Cela peut se prouver par récurrence sur $|\alpha|$. On peut ensuite montrer l'équivalence (pour $\alpha \in \mathcal{LP}$) suivante :

$$\alpha \text{ est satisfiable} \quad \text{ssi} \quad \widehat{\alpha} \text{ est satisfiable} \quad (6)$$

Par conséquent, si on dispose d'un algorithme pour SAT_Δ, en utilisant cette équivalence, on peut construire un algorithme pour SAT avec un pont polynomial (le calcul $\alpha \mapsto \widehat{\alpha}$ étant linéaire en la taille de la formule). Donc, SAT_Δ est NP-difficile : le problème NP-complet SAT peut être réduit en temps polynomial à SAT_Δ.

Preuve de (6). L'implication de gauche à droite est une conséquence directe de (5).

La réciproque peut se montrer par la contraposée : on suppose que α est insatisfiable et on va en déduire que $\widehat{\alpha}$ est insatisfiable. On peut mettre α sous FND (forme normale disjonctive [3]) en appliquant un processus systématique (utilisant l'involutivité de $\neg$, la distributivité de $\vee$ sur $\wedge$, etc.) pour aboutir à une formule α_{FND}. En utilisant ce *même processus* sur $\widehat{\alpha}$ on aboutit à une formule $(\widehat{\alpha})_{\mathrm{FND}} = \widehat{\alpha_{\mathrm{FND}}}$. La mise sous FND préserve l'équivalence dans les deux logiques, donc $\alpha_{\mathrm{FND}} \equiv \alpha$ et $\widehat{\alpha_{\mathrm{FND}}} \equiv \widehat{\alpha}$. Or α est insatisfiable, donc α_{FND} l'est également ce qui signifie que chaque terme de la disjonction α_{FND} est une conjonction insatisfiable de littéraux. Dans une telle conjonction, on a nécessairement une variable a apparaissant dans un littéral positif et dans un littéral négatif (sinon, la conjonction de littéraux serait satisfiable). Par conséquent, $\widehat{\alpha_{\mathrm{FND}}}$ est une disjonction de conjonctions de littéraux, chacune de ces conjonctions contenant un terme $a^{\equiv\mathtt{v}}$ et un terme $\neg a^{\equiv\mathtt{v}}$. Or, $a^{\equiv\mathtt{v}} \wedge \neg a^{\equiv\mathtt{v}}$ est insatisfiable dans $(\Delta\mathcal{LP}, \models)$. Donc, chaque conjonction constituant $\widehat{\alpha_{\mathrm{FND}}}$ est insatisfiable et donc $\widehat{\alpha_{\mathrm{FND}}}$ est également

3. Pour rappel, une formule sous FND est une disjonction de conjonctions de littéraux où un littéral est une formule d'une des formes a (littéral positif) et $\neg a$ (littéral négatif) avec $a \in \mathcal{V}$.

insatisfiable. Comme $\widehat{\alpha_{\text{FND}}} \equiv \widehat{\alpha}$, on en conclut que $\widehat{\alpha}$ est insatisfiable et c'est ce qu'il fallait démontrer. ■

Un corollaire de (6) est le suivant, pour $\mathcal{B}$, un ensemble fini de formules propositionnelles et $\alpha \in \mathcal{LP}$:

$$\mathcal{B} \models \alpha \quad \text{ssi} \quad \widehat{\mathcal{B}} \models \widehat{\alpha} \tag{7}$$

où $\widehat{\mathcal{B}} = \left\{\widehat{\beta} \mid \beta \in \mathcal{B}\right\}$. Cela permet de justifier *a posteriori* le terme de plongement : on peut considérer que l'injection $\alpha \mapsto \widehat{\alpha}$ permet de représenter (dans un sens cohérent avec la relation de conséquence logique) la logique $(\mathcal{LP}, \models)$ au sein de la logique $(\Delta\mathcal{LP}, \models)$. On pourrait aussi se servir de ce plongement en étendant la syntaxe de $(\Delta\mathcal{LP}, \models)$ par celle de $(\mathcal{LP}, \models)$, considérant, par exemple, que $a \wedge b^{+}$ est une notation pour $a^{=\mathrm{v}} \wedge b^{+}$. Nous éviterons néanmoins de le faire dans cet article.

Preuve de (7). Soit $\gamma = \neg(\bigwedge \mathcal{B} \wedge \neg\alpha)$: les assertions « $\mathcal{B} \models \alpha$ » et « γ est satisfiable » sont équivalentes entre elles, de même que les assertions « $\widehat{\mathcal{B}} \models \widehat{\alpha}$ » et « $\widehat{\gamma}$ est satisfiable ». En appliquant l'équivalence (6) sur γ cela permet de conclure. ■

4 Variation d'une formule propositionnelle à une autre

Soit $\alpha, \beta \in \mathcal{LP}$, on cherche à exprimer la variation de α à β, qu'on notera $\alpha \triangleright \beta$ et qui sera une formule de $(\Delta\mathcal{LP}, \models)$. Plusieurs définitions non équivalentes de l'opérateur $\triangleright$ peuvent *a priori* être envisagées. Dans cette section, celle qui nous semble la plus simple est étudiée : elle est définie à la section 4.1. Elle permet aussi d'introduire une extension de la syntaxe de $(\Delta\mathcal{LP}, \models)$ ne modifiant pas sa sémantique (section 4.2). Une étude des propriétés de cet opérateur de variation est présentée à la section 4.3. Enfin, d'autres définitions de la variation entre formules propositionnelles sont présentées brièvement, leur étude détaillée étant une perspective (section 4.4).

4.1 Définition d'un opérateur de variation

L'opérateur $\triangleright$ qu'on cherche à définir doit *a minima* correspondre à l'exemple donné en introduction : on s'attend à ce que $\mathtt{x}^i \triangleright \mathtt{x}^j \equiv \Delta\mathtt{x}^{ij}$. La définition proposée ci-dessous vérifie cela et est donnée à l'équivalence logique près, par l'ensemble de ses modèles :

$$\mathcal{M}(\alpha \triangleright \beta) = \mathcal{M}(\alpha) \times \mathcal{M}(\beta) \tag{8}$$

En d'autres termes, on considère les variations entre tout modèle de α et tout modèle de β.

4.2 Une extension de la syntaxe de la logique des variations propositionnelles

Les atomes a^v où $a \in \mathcal{V}$ et $v \in \mathcal{D}_0$ peuvent s'exprimer à l'aide de l'opérateur $\triangleright$ défini ci-dessus :

$$a^{=\mathrm{f}} \equiv \neg a \triangleright \neg a \qquad a^{=\mathrm{v}} \equiv a \triangleright a$$
$$a^{+} \equiv \neg a \triangleright a \qquad a^{-} \equiv a \triangleright \neg a$$

On peut généraliser cela en introduisant la notation α^v pour toute formule propositionnelle α et tout $v \in \mathcal{D}_0$:

$$\alpha^{=\mathrm{f}} = \neg\alpha \triangleright \neg\alpha \qquad \alpha^{=\mathrm{v}} = \alpha \triangleright \alpha$$
$$\alpha^{+} = \neg\alpha \triangleright \alpha \qquad \alpha^{-} = \alpha \triangleright \neg\alpha$$

Et on peut étendre cela aux symboles de variation non primitifs :

$$\alpha^{=} = \alpha^{=\mathrm{f}} \vee \alpha^{=\mathrm{v}}$$
$$\alpha^{\neq} = \alpha^{+} \vee \alpha^{-}$$
$$\alpha^{\mathrm{f}\bullet} = \alpha^{=\mathrm{f}} \vee \alpha^{+} \equiv \neg\alpha \triangleright \top$$
$$\alpha^{\bullet\mathrm{f}} = \alpha^{=\mathrm{f}} \vee \alpha^{-} \equiv \top \triangleright \neg\alpha$$
$$\alpha^{\mathrm{v}\bullet} = \alpha^{=\mathrm{v}} \vee \alpha^{-} \equiv \alpha \triangleright \top$$
$$\alpha^{\bullet\mathrm{v}} = \alpha^{=\mathrm{v}} \vee \alpha^{+} \equiv \top \triangleright \alpha$$

Cette extension de la syntaxe ne change pas la sémantique : comme on peut exprimer $\alpha \triangleright \beta$ dans la syntaxe de la logique avant extension, on peut également considérer les α^v (pour α une formule propositionnelle qui n'est pas une variable) comme une abréviation pratique (qui pourrait aussi avoir un intérêt en terme de temps de calculs).

Cette syntaxe étendue se justifie par le fait que la définition de la satisfaction d'un atome a^v par une interprétation $\mathcal{I}\mathcal{J}$ (cf. section 3.2) s'étend aux formules α^v, pour tout $\alpha \in \mathcal{LP}$. Ainsi, $\mathcal{I}\mathcal{J} \models \alpha^{=\mathrm{f}}$ ssi $\mathcal{I} \not\models \alpha$ et $\mathcal{J} \not\models \alpha$; $\mathcal{I}\mathcal{J} \models \alpha^{+}$ ssi $\mathcal{I} \not\models \alpha$ et $\mathcal{J} \models \alpha$, etc.

Les α^v, pour $\alpha \in \mathcal{LP}$ et $v \in \mathcal{D}_0$ sont deux à deux incohérents et leur disjonction est une tautologie :

$$\text{si } w, x \in \mathcal{D}_0 \text{ et } w \neq x,\ \alpha^w \wedge \alpha^x \models \bot \tag{9}$$
$$\models \alpha^{=\mathrm{f}} \vee \alpha^{=\mathrm{v}} \vee \alpha^{+} \vee \alpha^{-} \tag{10}$$

Donc, si $\alpha \in \mathcal{LP}$ est satisfiable et n'est pas une tautologie alors $\left\{\mathcal{M}\left(\alpha^{=\mathrm{f}}\right), \mathcal{M}\left(\alpha^{=\mathrm{v}}\right), \mathcal{M}\left(\alpha^{+}\right), \mathcal{M}\left(\alpha^{-}\right)\right\}$ est une partition de $\Delta\Omega$ [4].

Dans la suite de l'article, on considérera qu'une formule $\varphi \in \Delta\mathcal{LP}$ est n'importe quelle expression respectant cette syntaxe étendue.

On définit la substitution d'une variable a par une formule propositionnelle α dans une formule $\varphi \in \Delta\mathcal{LP}$ (notation : $\varphi[a\backslash\alpha]$) de façon classique : $a^v[a\backslash\alpha] =$

4. La condition α est satisfiable et n'est pas une tautologie n'est là que parce que l'ensemble vide ne peut pas être élément d'une partition : si $\alpha \models \bot$ (resp. $\models \alpha$) alors $\mathcal{M}\left(\alpha^{=\mathrm{v}}\right) = \emptyset$ (resp. $\mathcal{M}\left(\alpha^{=\mathrm{f}}\right) = \emptyset$).

α^v, $x^v[a\backslash\alpha] = x^v$ pour $x \in \mathcal{V} \setminus \{a\}$, $(\neg\psi)[a\backslash\alpha] = \neg(\psi[a\backslash\alpha])$, $(\varphi_1 \wedge \varphi_2)[a\backslash\alpha] = (\varphi_1[a\backslash\alpha]) \wedge (\varphi_2[a\backslash\alpha])$ et $(\varphi_1 \vee \varphi_2)[a\backslash\alpha] = (\varphi_1[a\backslash\alpha]) \vee (\varphi_2[a\backslash\alpha])$. La substitution d'une variable par une formule propositionnelle dans une tautologie de $(\Delta\mathcal{LP}, \models)$ est une tautologie. Formellement, avec $a \in \mathcal{V}$, $\alpha \in \mathcal{LP}$ et $\varphi \in \Delta\mathcal{LP}$:

$$\text{si } \models \varphi \text{ alors } \models \varphi[a\backslash\alpha] \tag{11}$$

Preuve de (11). Supposons que $\models \varphi$. Soit φ_{FNC}, une formule équivalente à φ de la forme suivante : $\varphi_{\text{FNC}} = \bigwedge\{\varphi_k \mid k \in \{1, 2, \ldots, p\}\}$ où chaque φ_k est une disjonction d'atomes a^v où $v \in \mathcal{D}_0$: pour obtenir φ_{FNC}, on effectue une série de transformations partant de φ, d'abord en revenant à la syntaxe de départ (dans laquelle on n'a une occurrence de α^v que si $\alpha \in \mathcal{V}$), puis on se débarrasse des symboles de variations non primitifs, puis, on utilise les mêmes opérations que pour la mise sous forme normale conjonctive en logique propositionnelle et enfin, on applique de gauche à droite l'équivalence (4) autant de fois que nécessaire. Comme $\varphi_{\text{FNC}} \equiv \varphi$ et que φ est une tautologie, chacun des φ_k est une tautologie. On montre alors que, pour un φ_k donné, il existe une variable a telle que φ_k contient les quatre atomes a^v pour tout $v \in \mathcal{D}_0$. La substitution $\varphi_k[a\backslash\alpha]$ est donc une disjonction contenant quatre termes deux à deux différents α^v ($v \in \mathcal{D}_0$) et on peut en déduire que $\varphi_k[a\backslash\alpha]$ est une tautologie et, partant, que $\varphi_{\text{FNC}}[a\backslash\alpha]$ est une tautologie. Or, $\varphi_{\text{FNC}}[a\backslash\alpha] \equiv \varphi[a\backslash\alpha]$ (on peut le démontrer en appliquant sur $\varphi[a\backslash\alpha]$ la séquence d'opérations qui a permis de passer de φ à φ_{FNC}). Donc $\varphi[a\backslash\alpha]$ est une tautologie, ce qui conclut la preuve. ■

4.3 Étude de l'opérateur ▹

On notera $\mathcal{LP}^{\text{sat}}$ (resp. $\Delta\mathcal{LP}^{\text{sat}}$) l'ensemble des formules $\alpha \in \mathcal{LP}$ (resp. $\varphi \in \Delta\mathcal{LP}$) qui sont satisfiables.

Articulation de ▹ avec les connecteurs. L'opération ▹ est « distributive » sur $\wedge$ et sur $\vee$ modulo l'équivalence [5]. Formellement, pour $\alpha, \alpha_1, \alpha_2, \beta, \beta_1, \beta_2 \in \mathcal{LP}$:

$$\begin{aligned}
\alpha \triangleright (\beta_1 \wedge \beta_2) &\equiv (\alpha \triangleright \beta_1) \wedge (\alpha \triangleright \beta_2)\\
\alpha \triangleright (\beta_1 \vee \beta_2) &\equiv (\alpha \triangleright \beta_1) \vee (\alpha \triangleright \beta_2)\\
(\alpha_1 \wedge \alpha_2) \triangleright \beta &\equiv (\alpha_1 \triangleright \beta) \wedge (\alpha_2 \triangleright \beta)\\
(\alpha_1 \vee \alpha_2) \triangleright \beta &\equiv (\alpha_1 \triangleright \beta) \vee (\alpha_2 \triangleright \beta)
\end{aligned}$$

Par ailleurs, pour $\alpha, \beta \in \mathcal{LP}$, on a :

$$\alpha \triangleright \beta \equiv (\alpha \triangleright \top) \wedge (\top \triangleright \beta) \tag{12}$$
$$\neg(\alpha \triangleright \beta) \equiv (\neg\alpha \triangleright \beta) \vee (\alpha \triangleright \neg\beta) \vee (\neg\alpha \triangleright \neg\beta) \equiv \alpha^{\mathtt{f}\bullet} \vee \beta^{\bullet\mathtt{f}}$$

5. Les guillemets sont justifiés d'une part par le fait que la distributivité est définie d'habitude sur des opérations internes (alors que ▹ est externe), d'autre part, par le fait que $\wedge$ et $\vee$ sont des connecteurs (pas des opérations).

Le résultat suivant permet de calculer $\Delta\mathbf{x}^{ij}$ en fonction de $\mathbf{x}^i$ et $\mathbf{x}^j$ (cf. introduction). Soit $\alpha, \beta \in \mathcal{LP}$ équivalents à une conjonction de littéraux. Alors, $\alpha \triangleright \beta$ est une conjonction d'atomes de $(\Delta\mathcal{LP}, \models)$. Plus précisément :

$$\begin{aligned}
&\text{si } \alpha \equiv \bigwedge_{a\in\mathcal{V}} \ell(a) \text{ et } \beta \equiv \bigwedge_{a\in\mathcal{V}} m(a)\\
&\qquad \text{avec } \ell(a), m(a) \in \{a, \neg a, \top\}\\
&\text{alors } \alpha \triangleright \beta \equiv \bigwedge_{a\in\mathcal{V}} \ell(a) \triangleright m(a)
\end{aligned} \tag{13}$$

et $\ell(a) \triangleright m(a)$ est équivalent à un atome ou à $\top$, selon le tableau suivant :

	$m(a) = a$	$m(a) = \neg a$	$m(a) = \top$
$\ell(a) = a$	$a^{=\mathtt{v}}$	$a^{\text{-}}$	$a^{\mathtt{v}\bullet}$
$\ell(a) = \neg a$	$a^{\text{+}}$	$a^{=\mathtt{f}}$	$a^{\mathtt{f}\bullet}$
$\ell(a) = \top$	$a^{\bullet\mathtt{v}}$	$a^{\bullet\mathtt{f}}$	$\top$

Preuve de (13). La preuve se fait en appliquant (12) de gauche à droite sur $\alpha \triangleright \beta$, en utilisant la distributivité à gauche et à droite de ▹ sur $\wedge$, pour obtenir la conjonction sur $a \in \mathcal{V}$ de $(\ell(a) \triangleright \top) \wedge (\top \triangleright m(a))$. En appliquant (12) de droite à gauche, on obtient le résultat désiré. ■

Étude fonctionnelle de ▹. L'opérateur ▹ n'est pas injectif, en particulier parce que si α est insatisfiable ou β est insatisfiable, alors $\alpha \triangleright \beta \equiv \bot$. En revanche, la restriction de ▹ à $\mathcal{LP}^{\text{sat}} \times \mathcal{LP}^{\text{sat}}$ est injective :

$$\begin{aligned}
&\textbf{si } \alpha_1, \beta_1, \alpha_2, \beta_2 \in \mathcal{LP}^{\text{sat}} \text{ et } \alpha_1 \triangleright \beta_1 \equiv \alpha_2 \triangleright \beta_2\\
&\textbf{alors } \alpha_1 \equiv \alpha_2 \text{ et } \beta_1 \equiv \beta_2
\end{aligned} \tag{14}$$

L'opérateur ▹ n'est pas non plus surjectif, mais son ensemble image est intéressant à étudier. Soit les opérations $\mathsf{G} : \Delta\mathcal{LP} \to \mathcal{LP}$ et $\mathsf{D} : \Delta\mathcal{LP} \to \mathcal{LP}$ définies, à la syntaxe près, pour $\varphi \in \Delta\mathcal{LP}$, par :

$$\begin{aligned}
\mathcal{M}(\mathsf{G}(\varphi)) &= \{\mathcal{I} \in \Omega \mid \text{il existe } \mathcal{J} \in \Omega \text{ telle que } \mathcal{I}\mathcal{J} \models \varphi\}\\
\mathcal{M}(\mathsf{D}(\varphi)) &= \{\mathcal{J} \in \Omega \mid \text{il existe } \mathcal{I} \in \Omega \text{ telle que } \mathcal{I}\mathcal{J} \models \varphi\}
\end{aligned}$$

$\mathsf{G}(\varphi)$ (resp. $\mathsf{D}(\varphi)$) peut être compris comme une « projection » à gauche (resp. à droite) de φ. Soit alors

$$\mathsf{F}(\varphi) = \mathsf{G}(\varphi) \triangleright \mathsf{D}(\varphi)$$

F est un opérateur de fermeture sur $(\Delta\mathcal{LP}, \models)$ (modulo $\equiv$) :

$$\begin{aligned}
\varphi &\models \mathsf{F}(\varphi)\\
\mathsf{F}(\mathsf{F}(\varphi)) &\equiv \mathsf{F}(\varphi)\\
\text{si } \varphi_1 \models \varphi_2 &\text{ alors } \mathsf{F}(\varphi_1) \models \mathsf{F}(\varphi_2)
\end{aligned}$$

pour $\varphi, \varphi_1, \varphi_2 \in \Delta\mathcal{LP}$. Si $\varphi \equiv \mathsf{F}(\varphi)$, on dira que φ est *fermée* pour F. L'image de $\mathcal{LP}^2$ par ▹ est l'ensemble des formules de $\Delta\mathcal{LP}$ fermées pour F.

Cette notion de fermeture est proche de celle qu'on trouve en analyse formelle de concepts (AFC [9]) et l'AFC pourrait être utilisée pour une représentation compacte de formules de $(\Delta\mathcal{LP}, \models)$ de la façon suivante. Soit $\varphi \in \Delta\mathcal{LP}$. Comme $\triangleright$ n'est pas surjective, φ ne peut pas nécessairement s'écrire sous la forme $\alpha \triangleright \beta$, mais on va l'écrire sous la forme d'une disjonction de telles formules, il restera alors à écrire chaque α et chaque β de façon compacte (ce qui est un problème de logique propositionnelle). Pour ce faire, on considère un tableau dont les lignes sont indexées par les $\mathcal{I} \in \mathsf{G}(\varphi)$ et les colonnes, par les $\mathcal{J} \in \mathsf{D}(\varphi)$ et tel qu'il y ait une incidence sur la case de ligne $\mathcal{I}$ et de colonne $\mathcal{J}$ ssi $\mathcal{I}\mathcal{J} \models \varphi$. En appliquant un algorithme d'AFC, on cherche l'ensemble $\{(A_k, B_k)\}_{k\in\{1,2,\ldots,p\}}$ des rectangles maximaux de ce tableau [6]. De cette façon, pour tout $k \in \{1, 2, \ldots, p\}$, en introduisant $\alpha_k, \beta_k \in \mathcal{LP}^{\text{sat}}$ tels que $\mathcal{M}(\alpha_k) = A_k$ et $\mathcal{M}(\beta_k) = B_k$, on peut montrer que $\varphi \equiv \bigvee_{k\in\{1,2,\ldots,p\}} \alpha_k \triangleright \beta_k$.

Inversion des variations. Pour $\varphi \in \Delta\mathcal{LP}$, on définit $\mathtt{inv}(\varphi) \in \Delta\mathcal{LP}$ obtenue en remplaçant chaque occurrence dans φ d'un $v \in \mathcal{D}$ par $\mathtt{inv}(v)$ de la façon suivante :

$$\begin{array}{rclcrcl} \mathtt{inv}(v) & = & v & \text{pour } v \in \{\mathtt{=f}, \mathtt{=v}, \mathtt{=}, \neq\} & & & \\ \mathtt{inv(+)} & = & \mathtt{-} & & \mathtt{inv(-)} & = & \mathtt{+} \\ \mathtt{inv(f\bullet)} & = & \mathtt{\bullet f} & & \mathtt{inv(\bullet f)} & = & \mathtt{f\bullet} \\ \mathtt{inv(v\bullet)} & = & \mathtt{\bullet v} & & \mathtt{inv(\bullet v)} & = & \mathtt{v\bullet} \end{array}$$

On peut montrer alors qu'on a, pour $\alpha, \beta \in \mathcal{LP}$:

$$\beta \triangleright \alpha \equiv \mathtt{inv}(\alpha \triangleright \beta) \tag{15}$$

Pour prouver ce résultat, on peut d'abord prouver par récurrence sur $|\varphi|$ que $\mathcal{M}(\mathtt{inv}(\varphi)) = \{\mathcal{J}\mathcal{I} \mid \mathcal{I}\mathcal{J} \in \mathcal{M}(\varphi)\}$. Ensuite, en appliquant ce résultat à $\varphi = \alpha \triangleright \beta$, on obtient :

$$\begin{aligned} \mathcal{M}(\mathtt{inv}(\alpha \triangleright \beta)) &= \{\mathcal{J}\mathcal{I} \mid \mathcal{I}\mathcal{J} \in \mathcal{M}(\alpha \triangleright \beta)\} \\ &= \{\mathcal{J}\mathcal{I} \mid \mathcal{I} \in \mathcal{M}(\alpha) \text{ et } \mathcal{J} \in \mathcal{M}(\beta)\} \\ &= \mathcal{M}(\beta) \times \mathcal{M}(\alpha) = \mathcal{M}(\beta \triangleright \alpha) \end{aligned}$$

4.4 D'autres opérateurs de variations entre formules

On peut envisager d'autres opérateurs de variations que $\triangleright$. L'un d'entre eux est considéré en fin d'article (section 5.6.3) et cet opérateur est une généralisation de l'opérateur $\triangleright$ au sens où tout modèle de $\alpha \triangleright \beta$ est un modèle de la variation de α à β au sens de cet opérateur. On va considérer dans cette section des opérateurs plus spécifiques que $\triangleright$: l'idée est que prendre tous les couples $\mathcal{I}\mathcal{J}$ tels que $\mathcal{I} \models \alpha$ et $\mathcal{J} \models \beta$ (comme c'est le cas avec $\alpha \triangleright \beta$) peut être considéré comme insuffisamment restrictif.

La question qui se pose alors est celle des critères de restriction sur ces couples. Pour ce faire, une idée inspirée d'opérateurs de révision des croyances fondés sur des distances $\mathtt{dist}$ sur Ω consiste à ne garder dans $\mathcal{M}(\alpha \triangleright \beta)$ que les $\mathcal{I}\mathcal{J}$ qui minimisent $\mathtt{dist}(\mathcal{I}, \mathcal{J})$. Formellement, on définit, à l'équivalence près, l'opérateur $\triangleright^{\mathtt{dist}}$ (pour $\alpha, \beta \in \mathcal{LP}$) par

$$\mathcal{M}(\alpha \triangleright^{\mathtt{dist}} \beta) = \{\mathcal{I}\mathcal{J} \in \mathcal{M}(\alpha \triangleright \beta) \mid \mathtt{dist}(\mathcal{I}, \mathcal{J}) = \mathtt{dist}^*\}$$
$$\text{où } \mathtt{dist}^* = \mathtt{dist}(\mathcal{M}(\alpha), \mathcal{M}(\beta))$$

La révision de α par β au sens d'un opérateur de révision $\circ^{\mathtt{dist}}$ paramétré par une distance peut être définie par $\mathcal{J} \models \alpha \circ^{\mathtt{dist}} \beta$ si $\mathtt{dist}(\mathcal{M}(\alpha), \mathcal{J}) = \mathtt{dist}^*$ [12] [7]. On a alors immédiatement $\mathsf{D}(\alpha \triangleright^{\mathtt{dist}} \beta) \equiv \alpha \circ^{\mathtt{dist}} \beta$ et aussi $\mathsf{G}(\alpha \triangleright^{\mathtt{dist}} \beta) \equiv \beta \circ^{\mathtt{dist}} \alpha$ (la seconde propriété est liée au fait que $\mathtt{dist}$ est symétrique). Il devrait être possible (mais c'est laissé en perspective) d'étudier les opérateurs de variations entre formules propositionnelles associées à d'autres opérateurs de révision (pas seulement ceux paramétrés par des distances) voire à d'autres opérateurs du domaine des changements de croyances (tels que les opérateurs de mise à jour).

5 Discussion

Cet article a présenté un début d'étude d'une logique des variations propositionnelles. Une tentative de construire des liens avec des travaux proches est présentée en section 5.1. Une brève présentation d'un système formel correct et complet pour cette logique est donnée en section 5.2. Cette étude peut se poursuivre de plusieurs façons, en particulier la conception d'algorithmes d'inférences (§5.3), l'étude d'applications de ce formalisme (§5.4) et celle d'autres logiques des variations, s'appuyant sur d'autres logiques que la logique propositionnelle finie (§5.5). Cette discussion se termine par la présentation de plusieurs (autres) questions ouvertes (section 5.6).

5.1 Des travaux proches

Assez étonnamment, nous n'avons pas trouvé de travaux vraiment proches de ce travail, ce qui signifie potentiellement qu'un tel travail très proche nous a échappé (nous pensons avoir cherché sérieusement). Néanmoins, on peut tenter de faire le lien avec différents travaux.

La notion de variation propositionnelle peut évoquer la notion de différence entre fonctions booléennes (une formule de $(\mathcal{LP}, \models)$ pouvant être vue comme une représentation d'une fonction de $\{\mathtt{f}, \mathtt{v}\}^{|\mathcal{V}|}$ dans $\{\mathtt{f}, \mathtt{v}\}$), ce qui est

6. On rappelle qu'un rectangle d'un tel tableau (appelé contexte en AFC) est un $A \times B$ où $A \neq \emptyset$ est un ensemble de lignes et $B \neq \emptyset$, un ensemble de colonnes, tel qu'il y a une incidence sur toute $(\mathcal{I}, \mathcal{J}) \in A \times B$. Un rectangle $A \times B$ est maximal s'il n'existe aucun rectangle le contenant strictement.

7. Techniquement, dans [12], l'opérateur $\circ^{\mathtt{dist}}$ est défini pour $\mathtt{dist}$ étant la distance de Hamming (ce qui fait qu'il coïncide avec l'opérateur de Dalal [6]). Cependant, la généralisation à toute distance sur Ω est immédiate et satisfait les postulats AGM [1].

étudié dans les travaux d'André Thayse [17]. Dans ces travaux, la différence entre deux fonctions booléennes est calculée par un ou exclusif et ces travaux mènent à un calcul différentiel sur les fonctions booléennes inspiré des travaux en analyse sur les nombres réels (développements de Taylor, etc.). Néanmoins, si on reprend l'objectif initial de ce travail, à savoir représenter des variations utiles, en particulier, pour le RàPC, le simple usage du ou exclusif s'avère insuffisant. Par exemple, la symétrie de $\oplus$ ($\alpha \oplus \beta \equiv \beta \oplus \alpha$) ne permet pas d'exprimer des variations orientées, i.e. de distinguer le passage de $\mathtt{f}$ à $\mathtt{v}$ du passage de $\mathtt{v}$ à $\mathtt{f}$.

La donnée d'une interprétation $\mathcal{I}\mathcal{J} \in \Delta\Omega$ équivaut à la donnée de la fonction qui à $a \in \mathcal{V}$ associe $(\mathcal{I}(a), \mathcal{J}(a)) \in \{\mathtt{f}, \mathtt{v}\}^2$ (correspondant aux quatre éléments de $\mathcal{D}_0$). Il est donc légitime de s'interroger sur les liens entre $(\Delta\mathcal{LP}, \models)$ et une logique avec quatre valeurs de vérité, comme c'est le cas de la logique de Belnap-Dunn [3], d'autant que les valeurs de vérité de cette logique sont parfois représentées par des couples de booléens. D'un point de vue sémantique, le lien n'est pas direct, puisque $(\Delta\mathcal{LP}, \models)$ est une logique avec deux valeurs de vérité. Par ailleurs, le nombre 4 de valeurs de vérité pour la logique de Belnap-Dunn est un fondement de cette logique alors que pour la logique des variations, il n'apparaît que comme une conséquence du choix de la logique propositionnelle comme point de départ pour construire cette logique : d'autres choix sont envisageables comme ce sera évoqué à la section 5.5 et peuvent conduire à des ensembles $\mathcal{D}_0$ de symboles de variation primitifs de cardinaux différents de 4. On peut néanmoins s'interroger sur une relation potentielle entre ces deux logiques, par exemple au niveau algorithmique. Ce travail n'a pas été fait et est une perspective potentielle.

Un lien peut être établi entre $(\Delta\mathcal{LP}, \models)$ et les logiques modales. Syntaxiquement, on pourrait considérer les symboles de variation comme autant de modalités. Sémantiquement, on pourrait définir $\models$ par la sémantique de Kripke en associant à chaque $\mathcal{I}\mathcal{J} \in \Delta\Omega$ un ensemble de deux mondes, w_1 et w_2, étiquetés respectivement par $\mathcal{I}$ et $\mathcal{J}$ et une relation d'accessibilité réduite à $\{(w_1, w_2)\}$. Ce lien pourrait être détaillé et étudié, même s'il apparaît à première vue comme excessif d'utiliser des logiques modales pour ne considérer que deux états (alors qu'en général, dans la sémantique de Kripke, l'ensemble des mondes et la relation d'accessibilité change d'une interprétation à une autre) et, de plus, qu'il n'y a pas d'équivalent dans la logique étudiée dans cet article à un emboîtement des modalités (quelque chose comme par un exemple un $(\alpha^{\mathtt{t}})^{=\mathtt{f}}$).

5.2 Un système formel pour $(\Delta\mathcal{LP}, \models)$

Dans le rapport [8], un système formel pour la logique des variations propositionnelles est présenté et il est montré qu'il est correct et complet. Cette section explique comment a été construit ce système formel.

Le point de départ a été S_H, le système formel de Hilbert, qui est correct et complet pour la logique propositionnelle. Cela passe par une réduction aux connecteurs $\neg$ et $\rightarrow$ (sans perte d'expressivité, les autres connecteurs pouvant être définis comme des abréviations utilisant ces deux connecteurs) et à l'usage d'un ensemble réduit de symboles de variations : les symboles de variations primitifs sont suffisants et nous les avons tous considérés. Les schémas d'axiomes de S_H ont été repris et d'autres ont été ajoutés qui concernent les symboles de variations. Le fait que $\{\mathcal{M}(\alpha^v) \mid v \in \mathcal{D}_0\}$ soit une partition de $\Delta\Omega$ (pour α satisfiable et non tautologique) nous a semblé nécessaire à exprimer pour assurer la complétude et il s'est avéré suffisant. Cela se traduit ainsi :

— D'après (9), pour tout $w, x \in \mathcal{D}_0$ avec $w \neq x$, $\models \neg(\alpha^w \wedge \alpha^x)$, ce qui se traduit sous la forme de ce schéma d'axiomes :

$$\alpha^w \rightarrow \neg\alpha^x$$
(pour $\alpha \in \mathcal{LP}$ et $w, x \in \mathcal{D}_0$ avec $w \neq x$)

— La relation (10) donne une tautologie qui peut être transformée en ce schéma d'axiomes :

$$\neg\alpha^w \rightarrow (\neg\alpha^x \rightarrow (\neg\alpha^y \rightarrow \alpha^z))$$
(pour $\alpha \in \mathcal{LP}$ et $w, x, y, z \in \mathcal{D}_0$ deux à deux distincts)

En utilisant le *modus ponens* (seule règle d'inférence de S_H), on obtient un système formel dont il est facile de montrer qu'il est correct pour $(\Delta\mathcal{LP}, \models)$, en particulier puisque tous ses axiomes sont des tautologies, mais la preuve de sa complétude a été plus complexe à établir.

Une perspective serait la définition d'un autre système formel qui soit plus pratique à utiliser en pratique, de la même façon que la déduction naturelle est plus facile à utiliser que S_H.

5.3 Des algorithmes d'inférences

Les problèmes de décision associés à $(\Delta\mathcal{LP}, \models)$ se ramènent au problème SAT_Δ de la même manière que ceux associés à $(\mathcal{LP}, \models)$ se ramènent à SAT.

La méthode la plus simple à implanter pour SAT_Δ consiste en une énumération des $\mathcal{I}\mathcal{J} \in \Delta\Omega$, mais comme $|\Delta\Omega| = 4^{|\mathcal{V}|}$, c'est une méthode très coûteuse. Plusieurs pistes sont envisageables pour trouver un algorithme plus efficace en pratique.

L'une d'elles a déjà été évoquée à la section 3.3 : elle consiste à chercher un algorithme appliquant la méthode des tableaux sémantique pour SAT_Δ. Un tel algorithme a été spécifié et implanté et ses correction et complétude ont été prouvées (voir [8]).

Une autre serait de chercher une traduction du problème SAT_Δ en un problème SAT et de profiter des algorithmes efficaces en pratique pour SAT. On sait qu'il existe des traductions polynomiales entre ces deux problèmes, puisqu'ils sont tous les deux NP-complet. Il s'agirait alors de trouver une traduction concrète qui soit efficace.

5.4 Applications potentielles

5.4.1 Au RàPC

Comme cela a été détaillé dans l'introduction, des travaux sur le RàPC, notamment pour l'apprentissage de connaissances d'adaptation (dans des travaux déjà anciens [7], plus récents [14] ou en cours), ont conduit à la syntaxe de la logique des variations, mais ne produisent des expressions qui n'étaient pas des formules logiques, avant qu'une sémantique y soit associée (et c'est l'objet principal de cet article).

À titre d'exemple, la règle d'adaptation R donnée en introduction (équation (1)) permet de résoudre le problème d'adaptation donné par le cas $(\mathtt{x}^s, \mathtt{y}^s)$ (représentation abstraite d'une recette de tarte aux pommes) et le problème $\mathtt{x}^{\mathtt{cible}}$ (requête « Je veux une recette de dessert avec des poires. ») suivants :

$$\begin{aligned}\mathtt{x}^s \wedge \mathtt{y}^s &= \mathtt{rDessert} \wedge \mathtt{iPomme} \wedge \mathtt{iCannelle}\\ &\quad \wedge \mathtt{iPâteBrisée} \wedge \neg\mathtt{iPoire} \wedge \neg\mathtt{iChocolat}\\ \mathtt{x}^{\mathtt{cible}} &= \mathtt{rDessert} \wedge \mathtt{iPoire}\end{aligned}$$

Le résultat attendu de cette adaptation est $\mathtt{y}^{\mathtt{cible}} \in \mathcal{LP}$ telle que

$$\begin{aligned}\mathtt{x}^{\mathtt{cible}} \wedge \mathtt{y}^{\mathtt{cible}} \equiv\ & \mathtt{rDessert} \wedge \mathtt{iPoire} \wedge \mathtt{iChocolat}\\ & \wedge \mathtt{iPâteBrisée} \wedge \neg\mathtt{iPomme} \wedge \neg\mathtt{iCannelle}\end{aligned}$$

Une question encore ouverte est comment spécifier une telle adaptation s'appuyant sur R. Une façon de faire qui coïncide avec le résultat attendu dans cet exemple, mais mérite d'être examinée, est la suivante. D'abord, on vérifie que R est applicable sur le problème d'adaptation en testant la satisfiabilité de $(\mathtt{x}^s \wedge \mathtt{y}^s \triangleright \mathtt{x}^{\mathtt{cible}}) \wedge R$ (qui est satisfiable dans l'exemple). Puis, on définit l'ensemble des solutions candidates :

$$\mathtt{Y} = \left\{ y \in \mathcal{LP} \mid \mathtt{x}^s \wedge \mathtt{y}^s \triangleright \mathtt{x}^{\mathtt{cible}} \wedge y \models R \right\}$$

Un $y \in \mathtt{Y}$ n'entraîne pas nécessairement `iPâteBrisée` dans l'exemple : cette variable n'apparaît pas dans R. Il faut un autre critère de choix et celui de la conservation maximale du cas source (disant qu'on ne fait une modification que quand elle est nécessaire) permet d'avoir le terme `iPâteBrisée`.

5.4.2 À la révision des croyances

Comme le RàPC, le domaine du changement des croyances est un domaine dans lequel on considère à la fois des assertions sur ce qui est vérifié à un moment donné (les croyances d'un agent donné à un instant donné) et sur ce qui change ou est susceptible de changer. Pour cette raison, on peut envisager d'utiliser le formalisme de représentation des variations présenté ici pour exprimer partiellement des connaissances sur le changement de croyances (partiellement, car ce formalisme exprime des variations mais ne permet pas d'exprimer des préférences entre variations). Pour explorer plus avant cette idée, on peut s'appuyer sur les liens entre l'adaptation en RàPC et la révision des croyances (voir par exemple [4]).

Une autre idée fait suite à ce qui a été présenté en section 4.4, avec l'introduction d'un opérateur $\triangleright^{\mathtt{dist}}$ qui est inspiré de la révision $\circ^{\mathtt{dist}}$, puisque $\alpha \triangleright^{\mathtt{dist}} \beta$ permet d'exprimer dans $(\Delta\mathcal{LP}, \models)$ à la fois la révision $\alpha \circ^{\mathtt{dist}} \beta$ et la révision $\beta \circ^{\mathtt{dist}} \alpha$ (la seconde exprimant les modèles de α les plus proches des modèles de β et intervenant dans la minimisation de la distance). Ainsi, $\alpha \triangleright^{\mathtt{dist}} \beta$ serait une façon de réifier le changement de croyances de la révision de α par β et pas uniquement son résultat.

5.5 Vers d'autres logiques des différences

Le principe de construction de $(\Delta\mathcal{LP}, \models)$ à partir de $(\mathcal{LP}, \models)$ peut se généraliser à la construction d'une logique $(\Delta\mathcal{L}, \models)$ partant d'une logique $(\mathcal{L}, \models)$ dont la relation $\models$ est définie en théorie des modèles :

- Syntaxiquement, il suffit de remplacer les atomes a de $\mathcal{L}$ par des atomes a^v ($v \in \mathcal{D}$) de $\Delta\mathcal{L}$;
- Sémantiquement, on définira une interprétation de $(\Delta\mathcal{L}, \models)$ comme étant un couple d'interprétations de $(\mathcal{L}, \models)$ et le reste suit le schéma de définition de $(\Delta\mathcal{LP}, \models)$.

De cette façon, on peut définir une logique des variations à partir du calcul des prédicats du premier ordre, par exemple, ou d'un de ses fragments décidables. On peut aussi appliquer ce principe sur $(\Delta\mathcal{LP}, \models)$ pour obtenir une logique des variations de variations propositionnelles $(\Delta\Delta\mathcal{LP}, \models)$, dont on pourrait imaginer (au moins théoriquement) qu'elle puisse s'appliquer à un système de RàPC dont l'étape d'adaptation serait un système de RàPC (travaillant sur des cas d'adaptation, voir, p. ex., [11]).

Une autre façon de généraliser ce travail concerne l'extension vers d'autres symboles de variations, pour des logiques dans lesquelles on aurait des variables non propositionnelles, par exemple des variables s'interprétant comme des nombres ou des valeurs nominales (et non des booléens). À titre d'illustration, considérons la variable $\hat{\mathtt{age}}$, représentant l'âge d'une personne, pour comparer deux personnes, une de 150 ans et l'autre de 200 ans. On pourrait noter cette variation de la première personne à la seconde par $\hat{\mathtt{age}}^{\mathtt{ajouter(50)}}$ et en déduire (en s'appuyant sur des connaissances sur les variations) qu'on aura $\hat{\mathtt{age}}^{<}$ (ajouter 50 ans à un âge c'est faire croître cet âge). Ce genre de symboles de variation a été introduit dans [2] qui cite, à titre d'autres exemples, les relations de l'algèbre de Allen utilisées comme symboles de variations entre deux intervalles.

5.6 Autres questions ouvertes

Cet article présente plusieurs perspectives, souvent légèrement entamées. Cette dernière section en ajoute plusieurs qui sont pratiquement intactes.

5.6.1 Composition des variations

Soit φ et ψ, deux formules de $(\Delta\mathcal{LP}, \models)$, représentant donc des variations propositionnelles. L'idée de les composer semble naturelle : on peut définir $\psi \circ \varphi$, la composition de φ par ψ, de la même façon qu'on compose les relations binaires $\mathcal{M}(\varphi)$ et $\mathcal{M}(\psi)$ sur Ω, i.e. pour $\mathcal{I}, \mathcal{K} \in \Omega$: $\mathcal{IK} \models \psi \circ \varphi$ s'il existe $\mathcal{J} \in \Omega$ telle que $\mathcal{IJ} \models \varphi$ et $\mathcal{JK} \models \psi$. En particulier, on peut montrer, avec cette définition, que, pour $\alpha, \beta, \gamma \in \mathcal{LP}^{\text{sat}}$, $(\beta \triangleright \gamma) \circ (\alpha \triangleright \beta) \equiv \alpha \triangleright \gamma$. L'étude de cet opérateur de composition reste à faire, en particulier pour exprimer la composition dans $(\Delta\mathcal{LP}, \models)$. Par exemple, on peut montrer qu'on a $\alpha^{+} \circ \alpha^{-} \equiv \alpha^{=\mathtt{v}}$, $\alpha^{+} \circ \alpha^{+} \equiv \bot$ etc. (pour $\alpha \in \mathcal{LP}$), mais il reste à étudier l'articulation entre la composition et les connecteurs.

Cette étude de la composition est motivée par l'étude de la composition des règles d'adaptation pour le raisonnement à partir de cas. En particulier, on peut s'intéresser à la question de la recherche d'une famille génératrice pour la composition de règles d'adaptation : cette problématique est encore peu étudiée (voir [18]) et mériterait de l'être dans l'optique de l'apprentissage de connaissances d'adaptation (par exemple pour réduire le nombre de règles d'adaptation apprises à valider par un expert).

5.6.2 Relation de conséquence dans $(\Delta\mathcal{LP}, \models)$ modulo une base de $(\mathcal{LP}, \models)$

Soit $\mathcal{B}$, une base de connaissances de la logique propositionnelle. Une perspective de ce travail serait d'étudier la relation $\models_{\mathcal{B}}$ sur la logique des variations propositionnelles définie, pour $\varphi, \psi \in \Delta\mathcal{LP}$ par $\varphi \models_{\mathcal{B}} \psi$ si, pour toute $\mathcal{IJ} \in \mathcal{M}(\mathcal{B}) \times \mathcal{M}(\mathcal{B})$, $\mathcal{IJ} \models \varphi$ entraîne $\mathcal{IJ} \models \psi$. En particulier, les relations $\models$ et $\models_{\emptyset}$ coïncident. L'étude de cette relation devrait être utile en particulier pour l'organisation des règles d'adaptation d'un système de RàPC, étant donné les connaissances du domaine $\mathcal{B}$ de ce système.

5.6.3 Un fragment de $(\Delta\mathcal{LP}, \models)$

La logique des variations propositionnelles permet d'exprimer à la fois des changements – via les symboles de variations `+` et `-` – et des persistances – via les symboles de variations `=f` et `=v`. On peut s'intéresser à ne retenir que les changements, ce qui conduit à définir un fragment $(\Delta^{\pm}\mathcal{LP}, \models)$ de la logique des variations, pour lesquels les seuls symboles de variations permis soient `+` et `-`. Ce fragment est strict : il existe des formules de $\Delta\mathcal{LP}$ qui ne sont équivalentes à aucune formule de $\Delta^{\pm}\mathcal{LP}$, comme par exemple $a^{=\mathtt{v}}$. On peut prouver cela en montrant, pour $\varphi \in \Delta^{\pm}\mathcal{LP}$, l'équivalence suivante pour toute $a \in \mathcal{V}$: $\varphi \wedge a^{=\mathtt{v}}$ est satisfiable ssi $\varphi \wedge a^{=\mathtt{f}}$ est satisfiable. Comme $a^{=\mathtt{v}} \wedge a^{=\mathtt{v}}$ est satisfiable mais que $a^{=\mathtt{v}} \wedge a^{=\mathtt{f}}$ ne l'est pas, on en conclut que $a^{=\mathtt{v}}$ n'est pas expressible dans $(\Delta^{\pm}\mathcal{LP}, \models)$.

Un intérêt de ce fragment serait de « forcer » à exprimer des variations plus générales. Par exemple, si on prend $\alpha, \beta \in \mathcal{LP}^{\text{sat}}$, la variation $\alpha \triangleright \beta$ ne réalise pas de généralisation puisqu'elle permet de « retrouver » α et β (cf. (14)). Supposons qu'on trouve un moyen d'associer à toute formule $\varphi \in \Delta\mathcal{LP}$ une formule $\varphi^{\pm} \in \Delta^{\pm}\mathcal{LP}$ qui soit une généralisation minimale de φ : $\varphi \models \varphi^{\pm}$ et, pour toute formule $\psi \in \Delta^{\pm}\mathcal{LP}$ telle que $\varphi \models \psi$, on aurait $\varphi^{\pm} \models \psi$. L'existence pour toute formule $\varphi \in \Delta\mathcal{LP}$ d'une telle formule $\varphi^{\pm}$ est une question ouverte (son unicité, à l'équivalence près, est facile à montrer). Dans le cas où une telle formule existerait (ou, du moins, existerait pour les formules fermées), on pourrait définir $(\alpha \triangleright \beta)^{\pm}$ qui serait une façon plus générale que $\alpha \triangleright \beta$ de traiter de la variation de α vers β.

La conjecture suivante sur $\varphi^{\pm}$ est proposée. Soit $\sim$ la relation d'équivalence sur $\Delta\Omega$ définie de la façon suivante (pour $\mathcal{I}_1\mathcal{J}_1, \mathcal{I}_2\mathcal{J}_2 \in \Delta\Omega$) : $\mathcal{I}_1\mathcal{J}_1 \sim \mathcal{I}_2\mathcal{J}_2$ si pour toute $a \in \mathcal{V}$, $\mathcal{I}_1\mathcal{J}_1 \models a^{-}$ ssi $\mathcal{I}_2\mathcal{J}_2 \models a^{-}$ et $\mathcal{I}_1\mathcal{J}_1 \models a^{+}$ ssi $\mathcal{I}_2\mathcal{J}_2 \models a^{+}$. Soit alors $C\ell_{\sim}(\mathcal{IJ})$ la classe d'équivalence d'une $\mathcal{IJ} \in \Delta\Omega$. Nous pensons que le résultat suivant est correct :

$$\mathcal{M}(\varphi^{\pm}) = \bigcup \{C\ell_{\sim}(\mathcal{IJ}) \mid \mathcal{IJ} \in \mathcal{M}(\varphi)\} \quad \text{(conjecture)}$$

pour toute formule φ de la logique des variations propositionnelles.

Références

[1] Alchourrón, C. E., P. Gärdenfors et D. Makinson: *On the logic of theory change : partial meet functions for contraction and revision*. J. Symbolic Logic, 50 :510–530, 1985.

[2] Badra, F. et J. Lieber: *Une approche pour représenter les variations entre cas — Vers une application à l'extraction de connaissances d'adaptation*. Dans Cordier, A. (rédacteur) : *15ème atelier sur le raisonnement à partir de cas*, pages 47–56, Grenoble, 2007.

[3] Belnap, N. D.: *How a computer should think*. New essays on Belnap-Dunn logic, pages 35–53, 2019.

[4] Cojan, J. et J. Lieber: *Applying belief revision to case-based reasoning*. Dans Prade, H. et G. Richard (rédacteurs) : *Computational Approaches to Analogical Reasoning : Current Trends*, tome 548 de *Studies in Computational Intelligence*, pages 133–161. Springer, 2014.

[5] Cordier, A., V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P.

Molli, A. Napoli et H. Skaf-Molli: *Taaable : a Case-Based System for personalized Cooking*. Dans Montani, S. et L. C. Jain (rédacteurs) : *Successful Case-based Reasoning Applications-2*, tome 494 de *Studies in Computational Intelligence*, pages 121–162. Springer, 2014.

[6] Dalal, M.: *Investigations into a theory of Knowledge Base Revision : Preliminary Report*. Dans *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI)*, pages 475–479, 1988.

[7] d'Aquin, M., F. Badra, S. Lafrogne, J. Lieber, A. Napoli et L. Szathmary: *Case base mining for adaptation knowledge acquisition*. Dans Veloso, M. M. (rédacteur) : *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 750–755. Morgan Kaufmann, Inc., 2007.

[8] François, N., Th. Laure et J. Lieber: *Une logique pour représenter des variations propositionnelles (version étendue)*. Rapport de recherche du LORIA, accessible à partir de l'adresse `https://k.loria.fr/publications/`, 2023.

[9] Ganter, B. et R. Wille: *Formal concept analysis : mathematical foundations*. Springer Science & Business Media, 2012.

[10] Hanney, K. et M. T. Keane: *Learning adaptation rules from a case-base*. Dans Smith, I. et B. Faltings (rédacteurs) : *Advances in Case-Based Reasoning – Proc. of the Third Eur. Workshop, EWCBR'96*, LNAI 1168, pages 179–192. Springer Verlag, Berlin, 1996.

[11] Jarmulak, J., S. Craw et R. Rowe: *Using case-base data to learn adaptation knowledge for design*. Dans *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, pages 1011–1016. Morgan Kaufmann, Inc., 2001.

[12] Katsuno, H. et A. Mendelzon: *Propositional knowledge base revision and minimal change*. Artificial Intelligence, 52(3) :263–294, 1991.

[13] Leake, D., X. Ye et D. J Crandall: *Supporting Case-Based Reasoning with Neural Networks : An Illustration for Case Adaptation*. Dans *AAAI Spring Symposium : Combining Machine Learning with Knowledge Engineering*, tome 2, 2021.

[14] Lieber, J. et E. Nauer: *Adapation knowledge discovery using positive and negative cases*. Dans *ICCBR 2021 - 29th International Conference on Case-Based Reasoning*, Salamanca (Virtual), Spain, 2021.

[15] Riesbeck, C. K. et R. C. Schank: *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989. Available on line.

[16] Smullyan, R. M.: *First-order logic*. Courier Corporation, 1995.

[17] Thayse, A.: *Boolean calculus of differences*. Springer, 1981.

[18] Tixier, M., F. Badra et J. Lieber: *Familles génératrices de règles d'adaptation pour assister leur acquisition semi-automatique*. Dans 19èmes *Journées Francophones d'Ingénierie des Connaissances (IC 2008)*, pages 225–236, Nancy, France, juin 2008. `https://hal.science/hal-00416700`.

Actes JIAF-JFPDA 2023

Premiers pas vers une logique des paires ordonnées

Henri Prade[1] **Gilles Richard**[1]

[1] IRIT, CNRS et Université Paul Sabatier, France

henri.prade@irit.fr, gilles.richard@irit.fr

Résumé

Les proportions logiques sont des connecteurs propositionnels liant quatre variables sous la forme d'une formule codant la conjonction de deux équivalences entre des indicateurs de similarité ou de dissimilarité relatifs d'une part part à un couple (a, b) et d'autre part à un couple (c, d). La proportion analogique "a est à b comme c est à d" est un exemple de proportion logique. L'article se place dans ce cadre pour dégager des éléments d'une logique manipulant des paires ordonnées. La construction de cette logique s'appuie sur un parallèle avec la logique des évènements conditionnels (qui est à la base du raisonnement non monotone), l'équivalence entre deux évènements conditionnels étant un autre exemple de proportion logique. La logique obtenue semble pouvoir offrir un cadre pour une logique de la "créativité", où des paires de vecteurs Booléens décrivent des transformations réalisables entre objets et où à partir d'un objet donné on peut induire un autre objet sur la base de telles transformations.

Abstract

Logical proportions are propositional connectors linking four variables in the form of a formula encoding the conjunction of two equivalences between indicators of similarity or dissimilarity relative on the one hand to a pair (a, b) and on the other hand to a pair (c, d). The analogical proportion "a is to b as c is to d is an example of a logical proportion. The article places itself in this framework to find elements of a logic manipulating ordered pairs. The construction of this logic is based on a parallel with the logic of conditional events (which is the basis of non-monotonic reasoning), the equivalence between two conditional events being another example of logical proportion. The logic obtained seems to be able to offer a framework for a logic of "creativity", where pairs of Boolean vectors describe feasible transformations between objects and where from a given object one can induce another object on the basis of such transformations.

1 Introduction

La comparaison d'objets ou de situations est certainement une opération cognitive de base. Il n'y a pas pour autant véritablement de logique de la comparaison, ni même de raisonnement de comparaison, si on excepte les proportions analogiques, qui sont des énoncés de la forme "a est à b comme c est à d", qui manifestement posent un parallèle entre les paires ordonnées (a, b) et (c, d), dont les éléments sont rapportés l'un à l'autre.

Pourquoi s'intéresser à des paires ? Il y a au moins deux exemples de paires ordonnées qui font sens du point de vue du raisonnement : i) les paires <condition(s), conclusion> correspondant à des règles "si $\cdots$ alors" ; ii) les paires comparatives entre deux items. On s'occupera principalement de ces dernières dans la suite, même si on rencontrera aussi les premières.

Dans la mesure où il s'agit notamment de définir une relation de conséquence entre paires ordonnées, cette relation une fois symétrisée doit donner naissance à une relation d'équivalence entre paires, qui doit donc être réflexive, symétrique et transitive. Dans un cadre booléen, cette relation correspond donc à un connecteur logique entre quatre variables (deux par paire).

Les proportions logiques [10] offrent précisément un cadre, en logique propositionnelle, de connecteurs quaternaires exprimant des relations entre paires. C'est dans ce cadre, dont nous rappelons l'essentiel maintenant, que nous démarrons les investigations [1].

2 Proportions logiques

De façon générale, l'idée de proportion est associée à la comparaison de paires (ordonnées) dont chaque élément

1. Tous les résultats énoncés dans ce document peuvent être testés sur le site https ://www.irit.fr/ Gilles.Richard/analogy/logic/.

d'une paire est rapporté à l'autre élément de la paire. [2] C'est une comparaison de comparaisons, comme le suggère l'énoncé de la proportion analogique "a est à b comme c est à d".

Dans le cadre booléen, nous avons quatre indicateurs de comparaison pour comparer a à b :

— Deux expriment la *similarité*, soit *positivement* comme $a \wedge b$ (qui est vrai si a et b sont vrais), soit *négativement* comme $\neg a \wedge \neg b$ (qui est vrai si a et b sont faux).

— Les deux autres sont des indicateurs de *dissimilarité* $\neg a \wedge b$ (qui est vrai si a est faux et b est vrai) et $a \wedge \neg b$ (qui est vrai si a est vrai et b est faux).

Les proportions logiques [10, 11] connectent quatre variables booléennes par la conjonction de deux équivalences entre indicateurs de similarité ou de dissimilarité se rapportant respectivement à deux paires (a, b) ordonnées et (c, d). Plus formellement,

Definition 1 *Une proportion logique $T(a, b, c, d)$ est la conjonction de deux équivalences entre un indicateur pour (a, b) d'un côté et un indicateur pour (c, d) de l'autre.*

L'expression

$$((a \wedge \neg b) \equiv (c \wedge \neg d)) \wedge ((a \wedge b) \equiv (c \wedge d))$$

fournit un exemple de proportion logique, où un même opérateur de similarité et un même opérateur de dissimilatité sont appliqués aux deux paires. Comme on peut le voir, elle exprime que "a diffère de b comme c diffère de d" et que "a est similaire à b comme c est similaire à d". Elle semble se rapporter à la comparaison des éléments à l'intérieur de chaque paire, mais on verra qu'il ne s'agit pas d'une proportion analogique.

Il a été établi [10] qu'il existe 120 proportions logiques syntaxiquement et sémantiquement distinctes. Toutes ces proportions partagent une propriété remarquable : elles sont vraies pour exactement 6 valuations de $abcd$ parmi $2^4 = 16$ possibles. Ainsi, l'exemple ci-dessus est vrai pour 0000, 1111, 1010, 0101, 0001, et 0100. Le lecteur intéressé est invité à consulter [10, 11] pour des études approfondies des différents types de proportions logiques.

Dans ce qui suit on ne s'intéressera qu'à des proportions logiques *symétriques* pour la raison indiquée dans l'introduction. Cette propriété indique que l'on peut échanger la paire (a, b) avec la paire (c, d) dans la proportion logique T, i.e., $T(a, b, c, d) \rightarrow T(c, d, a, b)$. De telles proportions logiques sont assez rares :

Proposition 1 *[10] Il n'existe que 12 proportions satisfaisant la symétrie : 4 proportions homogènes, 4 proportions conditionnelles et 4 proportions hybrides.*

2. Dans le cadre numérique, cela correspond notamment aux proportions aritmétiques $a - b = c - d$ et aux proportions géométriques $\frac{a}{b} = \frac{c}{d}$ qui égalisent des différences et des rapports respectivement.

Les proportions homogènes ne mélangent pas différents types d'indicateurs dans leurs équivalences (elles n'utilisent que des indicateurs de similarité ou que des indicateurs de dissimilarité). L'expression des proportions conditionnelles est constituée de la conjonction d'une équivalence entre des indicateurs de similarité et d'une équivalence entre des indicateurs de dissimilarité (la raison de leur dénomination apparaitra plus tard). Les proportions hybrides sont caractérisées par des équivalences entre des indicateurs de similarité et des indicateurs de dissimilarité dans leurs définitions.

Les expressions des 12 proportions symétriques sont données dans [10]. Nous redonnerons dans la suite que les expressions de celles qui nous intéressent ici (ce qui excluera les proportions hybrides, aucune n'étant transitive).

Commençons par les proportions homogènes, elles sont au nombre de 4, toutes symétriques. Elles incluent la proportion analogique et 3 autres proportions.

La proportion analogique "a est à b comme c est à d" énonce formellement que "a diffère de b comme c diffère de d et que b diffère de a comme d diffère de c". Cela s'exprime logiquement [8] par le connecteur quaternaire A :

$$\mathrm{A}(a, b, c, d) \triangleq ((a \wedge \neg b) \equiv (c \wedge \neg d)) \wedge ((\neg a \wedge b) \equiv (\neg c \wedge d)) \quad (1)$$

Les noms et les expressions des 3 autres proportions homogènes sont donnés ci-après :

— *paralogie* : $\mathrm{P}(a, b, c, d) \triangleq$

$$((a \wedge b) \equiv (c \wedge d)) \wedge ((\neg a \wedge \neg b) \equiv (\neg c \wedge \neg d)).$$

Elle exprime que "ce que a et b ont en commun (positivement ou négativement), c et d l'ont aussi, et inversement". On peut montrer que
$\mathrm{P}(a, b, c, d)) \Leftrightarrow \mathrm{A}(c, b, a, d)$.

— *analogie renversée* : $\mathrm{R}(a, b, c, d) \triangleq$

$$((\neg a \wedge b) \equiv (c \wedge \neg d)) \wedge ((a \wedge \neg b) \equiv (\neg c \wedge d)).$$

L'analogie renversée exprime que "b est à a comme c est à d". De fait, $\mathrm{R}(a, b, c, d) \Leftrightarrow \mathrm{A}(b, a, c, d)$.

— *paralogie inversée* : $\mathrm{I}(a, b, c, d) \triangleq$

$$((a \wedge b) \equiv (\neg c \wedge \neg d)) \wedge ((\neg a \wedge \neg b) \equiv (c \wedge d))$$

Cette expression est obtenue en échangeant les indicateurs de similarité positifs et négatifs relatifs à la paire (c, d) dans la définition de la paralogie. $\mathrm{I}(a, b, c, d)$ indique que "ce que a et b ont en commun, c et d ne l'ont pas, et inversement". Cela exprime une sorte d'"orthogonalité" entre les paires (a, b) et (c, d).

A				P				R				I			
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	1
1	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1	0	1	1	0	0	1	0	1
1	0	1	0	1	0	1	0	1	0	0	1	1	0	1	0

TABLE 1 – Valuations qui rendent vrai A, P, R, I

La Table 1 donne les 6 valuations booléennes (quadruplets de valeurs) qui rendent vrai A, P, R et I.

Notons également dans la Table 1 que les 6 valuations qui rendent les quatre proportions vraies appartiennent à un ensemble de 8 valuations. Cet ensemble de 8 motifs est caractérisé par la formule logique $K(a,b,c,d) \triangleq (a \equiv b) \equiv (c \equiv d)$, qui correspond à un connecteur de type analogique proposé par S. Klein [5], en relation avec des matériaux anthropologiques (dans le cadre d'une approche structuraliste).

De manière assez remarquable, on peut vérifier que :
— A et I sont les seules proportions homogènes qui satisfont les permutations centrales et externes, à savoir, $T(a,b,c,d) \rightarrow T(a,c,b,d)$ et $T(a,b,c,d) \rightarrow T(d,b,c,a)$;
— P et I sont les seules proportions homogènes qui satisfont les permutations $T(a,b,c,d) \rightarrow T(b,a,c,d)$ et $T(a,b,c,d) \rightarrow T(a,b,d,c)$;
— R et I sont les seules proportions homogènes qui satisfont les permutations $T(a,b,c,d) \rightarrow T(c,b,a,d)$ et $T(a,b,c,d) \rightarrow T(a,d,c,b)$.

La permutation centrale est considérée de très longue date comme une propriété caractéristique de la proportion analogique A, sans doute par mimétisme avec les proportions numériques. La parologie inverse I est extrêmement remarquable car elle est la seule des 120 proportions logiques à être stable sous toutes les permutations de variables deux à deux. [9].

Si on a à l'esprit que la proportion analogique décrit une sorte d'égalité entre paires qui étend l'idée de proportions arithmétiques ou géométriques, il est naturel de s'attendre à une forme de propriété de *transitivité* pour l'analogie A et plus généralement pour d'autres proportions T, ce qui s'exprime comme suit :

$$T(a,b,c,d) \wedge T(c,d,e,f) \rightarrow T(a,b,e,f)$$

On peut vérifier que la proportion analogique A, et la paralogie P sont transitives au sens ci-dessus (mais ni l'analogie renversée R, ni la paralogie inversée I ne sont transitives).

Le résultat suivant indique quelles sont les proportions logiques transitives :

Proposition 2 *[10] Il y a 54 proportions logiques qui sont transitives : 2 homogènes A et P, 4 proportions logiques conditionnelles (sur les 16 existantes), à savoir*
$((a \wedge b) \equiv (c \wedge d)) \wedge ((a \wedge \neg b) \equiv (c \wedge \neg d))$;
$((a \wedge b) \equiv (c \wedge d)) \wedge ((\neg a \wedge b) \equiv (\neg c \wedge d))$;
$((a \wedge \neg b) \equiv (c \wedge \neg d)) \wedge ((\neg a \wedge \neg b) \equiv (\neg c \wedge \neg d))$;
$((\neg a \wedge b) \equiv (\neg c \wedge d)) \wedge ((\neg a \wedge \neg b) \equiv (\neg c \wedge \neg d))$,
et les 48 proportions dites dégénérées.

Dans une proportion dite dégénérée, deux des quatre indicateurs de similarité ou de dissimilarité de la proportion logique sont identiques. Nous renvoyons le lecteur à [10] pour plus de détails, car ces proportions ne sont jamais symétriques.

Les 4 proportions logiques conditionnelles de la Proposition 2, sont symétriques, ce sont celles dont il est question à la Proposition 1.

Remarquons qu'une proportion logique T peut être *réflexive*, c'est-à-dire que $T(a,b,a,b)$ est vrai pour tout a, tout b, et que donc T est vrai pour les valuations $(0,0,0,0)$, $(0,1,0,1)$, $(1,0,1,0)$, et $(1,1,1,1)$. On a le résultat suivant :

Proposition 3 *[10] Il y a 6 proportions logiques qui sont réflexives : A, P, et les 4 proportions logiques conditionnelles de la Proposition 2.*

Quand on considère les paires ordonnées (a,b) comme des objets atomiques, A, P et les 4 proportions conditionnelles sont des relations d'équivalence sur l'univers des paires booléennes. On peut alors énoncer le résultat :

Proposition 4 *Les proportions logiques A, P, et les 4 proportions logiques conditionnelles de la Proposition 2 sont les seules relations d'équivalence entre paires.*

Venons-en aux 4 proportions logiques conditionnelles qui, comme on va le voir, sont en relation avec notre propos. Expliquons le terme "conditionnel". Il vient du fait que ces proportions expriment des équivalences entre des énoncés conditionnels. En effet, il a été souligné dans [4] qu'une règle "si a alors b" peut être considérée comme une entité à trois valeurs qui est appelée "objet conditionnel", ou "évènement conditionnel", et dénotée $b|a$. Cette entité est *trivaluée* [3] ; elle est :
— vraie si $a \wedge b$ est vrai. Les éléments qui rendent vrai $a \wedge b$ sont les *exemples* de la règle "si a alors b",
— fausse si $a \wedge \neg b$ est vrai. Les éléments qui rendent vrai $a \wedge \neg b$ sont les *contre-exemples* de la règle "si a alors b",
— indéfinie si $\neg a$ est vrai. La règle "si a alors b" n'est alors pas applicable.

Considérons la proportion conditionnelle, qui apparait dans la Proposition 2, et qui a été notre premier exemple de proposition logique :

$$((a \wedge b) \equiv (c \wedge d)) \wedge ((a \wedge \neg b) \equiv (c \wedge \neg d))$$

La proportion logique ci-dessus peut être notée $b|a :: d|c$ en combinant la notation des objets conditionnels et celle de la proportion analogique. En effet, la proportion $b|a :: d|c$ exprime une équivalence sémantique entre les deux règles "si a alors b" et "si c alors d" en énonçant qu'elles ont les mêmes exemples, c'est-à-dire $(a \wedge b) \equiv (c \wedge d)$), et les mêmes contre-exemples, c'est-à-dire $(a \wedge \neg b) \equiv (c \wedge \neg d)$.

La relation de conséquence logique (encore notée $\models$) entre deux objets conditionnels $b|a \models d|c$, se définit à partir de la conséquence logique booléenne usuelle $\models$ de la manière suivante :

$$a \wedge b \models c \wedge d \text{ et } (c \wedge \neg d) \models a \wedge \neg b \qquad (2)$$

qui exprime que les exemples de $b|a$ sont des exemples de $d|c$ et que les contre-exemples de $d|c$ sont des contre-exemples de $b|a$. Cette relation de conséquence logique est naturellement associée à la proportion conditionelle $b|a :: d|c$, puisque $b|a :: d|c$ est équivalent à :

$$b|a \models d|c \text{ et } d|c \models b|a.$$

La transitivité des 4 proportions conditionnelles de la Proposition 2 reflète le fait qu'elles expriment des équivalences entre objets conditionnels (et donc entre règles), à savoir respectivement $b|a :: d|c$, $a|b :: c|d$, $a|\neg b :: c|\neg d$, et $b|\neg a :: d|\neg c$.

L'objet conditionnel $b|a$ doit donc être pensé comme une règle "si a alors b". Une règle peut avoir des exceptions. C'est-à-dire, qu'on peut avoir en même temps "si a alors b" et une règle "si $(a \wedge c)$ alors $\neg b$". Les deux objets conditionnels $b|a$ et $\neg b|a \wedge c$ ne conduisent pas à une contradiction en présence des faits a et c (à la différence d'une modélisation des règles par l'implication matérielle), dans le cadre d'une logique tri-valuée où la conjonction & est définie par

$$b|a \;\&\; d|c \triangleq (a \rightarrow b) \wedge (c \rightarrow d)|(a \vee c)$$

avec pour sémantique

$$val(o_1 \;\&\; o_2) = \min(val(o_1), val(o_2))$$

où indéfini > vrai > faux. [3]

On montre [4] que cette quasi-conjonction & (c'est son nom) est associative. Elle exprime que l'ensemble constitué par les deux règles "si a alors b" et "si c alors d" est déclenchable si a ou c est vrai, et dans ce cas la règle déclenchée se comporte comme l'implication matérielle. Cette logique constitue la sémantique la plus simple [1] du système P d'inférence non monotone de Kraus, Lehmann, et Magidor [6]. Le lecteur pourra consulter [1] pour plus de détails.

Comme on vient de le voir, dans ce calcul la règle "si a alors b" est assimilée à une paire ordonnée (a, b) (<condition>, <conclusion>) et possède une sémantique tri-valuée. Dans la suite on s'intéresse de la même façon à une logique de paires, basée sur l'idée de comparaison, en relation avec les équivalences sémantiques exprimées par A et par P.

3 Eléments de logique de paires ordonnées

Dans cette section, nous nous efforçons de dégager quelques éléments d'une logique comparative de paires ordonnées. Les objets, ou items, de la comparaison sont décrits par des vecteurs de valeurs d'attributs (ici booléens).

3.1 Comparer les éléments d'une paire

Soient $\vec{a} = (a_1, ..., a_n)$, $\vec{b} = (b_1, ..., b_n)$, etc. des items décrits au moyen de n attributs booléens.

Les proportions logiques s'étendent à des vecteurs de variables booléennes, en les appliquant composante par composante, sous la forme :

Definition 2

$T(\vec{a}, \vec{b}, \vec{c}, \vec{d})$ *si et seulement si* $\forall i \in \{1, ..., n\}$, $T(a_i, b_i, c_i, d_i)$

Etant donné deux vecteurs $\vec{a}$, $\vec{b}$, leur comparaison amène à considérer les sous-ensembles d'attributs où ils sont égaux (à 1 ou à 0), et les sous-ensembles d'attributs où ils diffèrent, passant de 0 à 1 ou de 1 à 0, quand on va de $\vec{a}$ à $\vec{b}$. Ce qui conduit à poser :

$Equ^0(\vec{a}, \vec{b}) = \{i \mid a_i = b_i = 0\}$,
$Equ^1(\vec{a}, \vec{b}) = \{i \mid a_i = b_i = 1\}$,
$Equ(\vec{a}, \vec{b}) = \{i \mid a_i = b_i\} = Equ^0(\vec{a}, \vec{b}) \cup Equ^1(\vec{a}, \vec{b})$,
et
$Dif^{10}(\vec{a}, \vec{b}) = \{i \mid a_i = 1, b_i = 0\}$,
$Dif^{01}(\vec{a}, \vec{b}) = \{i \mid a_i = 0, b_i = 1\}$;
$Dif(\vec{a}, \vec{b}) = \{i \mid a_i \neq b_i\} = Dif^{01}(\vec{a}, \vec{b}) \cup Dif^{10}(\vec{a}, \vec{b})$.

Ceci nous permet d'énoncer le résultat suivant :

$$A(\vec{a}, \vec{b}, \vec{c}, \vec{d}) \text{si et seulement si } \begin{cases} Equ(\vec{a}, \vec{b}) = Equ(\vec{c}, \vec{d}) \\ Dif^{10}(\vec{a}, \vec{b}) = Dif^{10}(\vec{c}, \vec{d}) \\ Dif^{01}(\vec{a}, \vec{b}) = Dif^{01}(\vec{c}, \vec{d}) \end{cases}$$

On voit que ce qui importe dans une analogie c'est l'orientation des différences, alors que peu importe la valeur avec laquelle l'égalité est réalisée. La Table 2 met en évidence la structure d'une proportion analogique, en trois sous-ensembles d'attribut(s), un où les 4 items sont égaux, un

3. La négation est définie par $\neg(b|a) = (\neg b|a)$. Donc $\neg(b|a)$ est indéfinie si et seulement si $b|a$ l'est.

où ils sont égaux à l'intérieur des paires, mais pas de la même manière, et enfin le sous-ensemble d'attribut(s) dont la/les valeur(s) change(nt), dans le même sens, en passant de $\vec{a}$ à $\vec{b}$ et de $\vec{c}$ à $\vec{d}$.

items	*Tous égaux*	*Egal. par paires*	*Chang.*
$\vec{a}$	1 0	1 0	1 0
$\vec{b}$	1 0	1 0	0 1
$\vec{c}$	1 0	0 1	1 0
$\vec{d}$	1 0	0 1	0 1

TABLE 2 – Les 3 parties d'une proportion analogique et les valuations associées

Comme on peut le voir, la permutation centrale de $\vec{b}$ et de $\vec{c}$ échange les sous-ensembles d'"Egalité par paires" et de "Changement". Aucun de ces deux sous-ensembles ne doit être vide si on veut que la proportion analogique soit non triviale, c'est-à-dire que $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ soient distincts (pour $n = 2$, $\vec{a} = (1,1)$, $\vec{b} = (1,0)$, $\vec{c} = (0,1)$, $\vec{d} = (0,0)$ réalisent une proportion analogique avec des vecteurs distincts). Par contre, le sous-ensemble d'attribut(s) "Tous égaux" peut être vide. Si le sous-ensemble "Egalité par paires" ou bien le sous-ensemble "Changement" est vide, alors $\vec{a} = \vec{c}$ et $\vec{b} = \vec{d}$ ou bien $\vec{a} = \vec{b}$ et $\vec{c} = \vec{d}$ respectivement.

Etant donnés 4 vecteurs distincts, ils constituent deux paires ordonnées $(\vec{a}, \vec{b})$ et $(\vec{c}, \vec{d})$ dans la même classe d'équivalence de A (rappelons que A est réflexive, symétrique et transitive) si et seulement si [4] :

1. $Dif(\vec{a}, \vec{b}) = Dif(\vec{c}, \vec{d})$;
2. $\forall j \in Dif(\vec{a}, \vec{b})\ a_j = c_j$ et $b_j = d_j$.

La condition 1 assure que les changements concernent les mêmes attributs dans les deux paires, la condition 2 qu'ils s'appliquent dans le même sens pour les deux paires. Il est clair que deux paires quelconques prises dans la même classe d'"équivalence forment ensemble une proportion analogique. Cette notion de classe d'"équivalence rejoint l'idée de "cluster analogique" introduite par [7] dans un contexte de linguistique computationnelle.

Tandis que la proportion analogique insiste sur l'identité des différences existant dans chaque paire, la paralogie exprime plutôt un parallèle entre les paires au plan des propriétés partagées, positivement ou négativement. C'est ce que traduit le résultat suivant, dual de celui pour l'analogie :

$$P(\vec{a}, \vec{b}, \vec{c}, \vec{d}) \text{ si et seulement si } \begin{cases} Dif(\vec{a}, \vec{b}) = Dif(\vec{c}, \vec{d}) \\ Equ^1(\vec{a}, \vec{b}) = Equ^1(\vec{c}, \vec{d}) \\ Equ^0(\vec{a}, \vec{b}) = Equ^0(\vec{c}, \vec{d}) \end{cases}$$

4. Si les vecteurs ne sont pas distincts, on doit ajouter la condition $Dif(\vec{a}, \vec{b}) \neq \emptyset$ et $\exists i\ a_i \neq c_i$.

3.2 Combiner des relations entre paires

Une forme de raisonnement entre paires est obtenue en étudiant les "combinaisons" de relations entre paires exprimées par les proportions homogènes, au sens suivant :

$$T(\vec{a}, \vec{b}, \vec{c}, \vec{d}) \wedge T'(\vec{c}, \vec{d}, \vec{e}, \vec{f}) \rightarrow T''(\vec{a}, \vec{b}, \vec{e}, \vec{f})$$

où $T, T', T'' \in \{A, P, I, R\}$.

Cette forme de "combinaisons" généralise l'idée de transitivité. On a déjà vu que A et P sont transitives.

	A	P	R	I
A	A	K	R	K
P	K	P	K	I
R	R	K	A	K
I	K	I	K	P

TABLE 3 – Combinaison de proportions homogènes

La Table 3 résume tous les résultats des combinaisons qu'on peut obtenir à partir de $\{A, P, R, I\}$. Ces combinaisons étant commutatives, la table est symétrique. Le résultat K indique l'opérateur de Klein rappelé en Section 2, un résultat donc trivial (rappelons que K n'est pas une proportion logique). En permutant les deux dernières lignes et les deux dernières colonnes, on peut faire apparaitre les résultats non triviaux sur les deux diagonales :

	A	P	I	R
A	A	K	K	R
P	K	P	I	K
I	K	I	P	K
R	R	K	K	A

En dehors des transitivités de A et P, les autres résultats sont conformes aux idées de "parallèlisme" pour P et d'"orthogonalité" pour I. En effet, P $\wedge$ I $\rightarrow$ I et I $\wedge$ I $\rightarrow$ P. Notons aussi que R $\wedge$ R $\rightarrow$ A, ce qui est conforme à l'idée que deux renversements successifs ramènent à l'endroit.

3.3 Relation de conséquence entre paires ordonnées

Les connecteurs logiques s'étendent à des vecteurs composante par composante. On a donc

— $\neg\vec{a} = (\neg a_1, ..., \neg a_n)$,
— $\vec{a} \wedge \vec{b} = (a_1 \wedge b_1, ..., a_n \wedge b_n)$,
— $\vec{a} \vee \vec{b} = (a_1 \vee b_1, ..., a_n \vee b_n)$.

En s'inspirant du cas des proportions conditionnelles, on est amené à définir composante par composante la relation de conséquence logique suivante (encore notée $\models$) entre

paires ordonnées $(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$ à partir de la définition de la proportion analogique de la manière suivante [5] :

$$\neg\vec{a} \wedge \vec{b} \models \neg\vec{c} \wedge \vec{d} \text{ et } \vec{c} \wedge \neg\vec{d} \models \vec{a} \wedge \neg\vec{b} \qquad (3)$$

Quand on considère des paires $(\vec{a},\vec{b})$, une valuation $(a_i, b_i) = (0,1)$ peut être interprétée comme le fait que l'on acquiert la propriété i quand on passe de $\vec{a}$ à $\vec{b}$. Donc le sens de la conséquence logique devient :

— Les propriétés acquises quand on passe de $\vec{a}$ à $\vec{b}$ restent acquises quand on passe de $\vec{c}$ à $\vec{d}$,
— De plus, si une propriété est perdue en passant de $\vec{c}$ à $\vec{d}$, c'était déjà le cas dans le passage de $\vec{a}$ à $\vec{b}$ [6].

Naturellement $(\vec{a},\vec{b}) \models (\vec{a},\vec{b})$, mais de plus :

Proposition 5 *On a l'équivalence suivante :*

$$(\vec{a},\vec{b}) \models (\vec{c},\vec{d}) \text{ et } (\vec{c},\vec{d}) \models (\vec{a},\vec{b}) \text{ ssi } A(\vec{a},\vec{b},\vec{c},\vec{d})$$

Preuve. Voyons le sens précis de cette définition pour les paires. Comme nous travaillons composante par composante, il suffit de considérer la conséquence de cette définition sur une composante. Deux cas sont à considérer :

— cas $a = b$ (qui représente 8 valuations parmi les 16 candidates for a, b, c, d). Puisque $\neg a \wedge b$ et $a \wedge \neg b$ sont égaux à 0, la seule contrainte est que $c \wedge \neg d = 0$ qui est satisfaite seulement si $(c, d) \neq (1, 0)$, ce qui élimine (0010) et (1110) comme valuations valides, laissant 6 valuations encore valides.
— cas $a \neq b$ (qui représente les 8 valuations restantes) : si $(a, b) = (1, 0)$, il n'y a pas de contrainte sur (c, d). Si $(a, b) = (0, 1)$, seulement $(c, d) = (0, 1)$ est valide, ce qui élimine 3 valuations parmi les 8 : (0100), (0110), (0111).

La conjonction $(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$ et $(\vec{c},\vec{d}) \models (\vec{a},\vec{b})$, conduit à la table de vérité de $A(a, b, c, d)$ avec exactement 6 valuations valides. □

Puisque quand $(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$, les 5 valuations $(0, 0, 1, 0)$, $(1, 1, 1, 0)$, $(0, 1, 0, 0)$, $(0, 1, 1, 0)$, $(0, 1, 1, 1)$ sont *interdites* pour chaque composante (a_i, b_i, c_i, d_i), cela signifie que

— $(a_i, b_i) = (0, 1) \Rightarrow (c_i, d_i) = (0, 1)$; (une propriété acquise en passant de $\vec{a}$ à $\vec{b}$ doit être aussi acquise en allant de $\vec{c}$ à $\vec{d}$) ;
— $a_i = b_i \Rightarrow (c_i, d_i) \neq (1, 0)$ (quand il n' y a pas acquisition de propriété ou qu'il y a une perte de propriété en allant de $\vec{a}$ à $\vec{b}$, il ne peut pas y avoir une perte en allant de $\vec{c}$ à $\vec{d}$).

De manière similaire, on a $(\vec{c},\vec{d}) \models (\vec{a},\vec{b}) \Leftrightarrow \begin{cases} (a_i, b_i) = (1, 0) \Rightarrow (c_i, d_i) = (1, 0) \\ a_i = b_i \Rightarrow (c_i, d_i) \neq (0, 1) \end{cases}$

qui interdit les 5 valuations $(1, 0, 0, 0)$, $(1, 0, 0, 1)$, $(1, 0, 1, 1)$, $(0, 0, 0, 1)$, $(1, 1, 0, 1)$.
On a donc, comme attendu :
$(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$ et $(\vec{c},\vec{d}) \models (\vec{a},\vec{b}) \Leftrightarrow A(\vec{a},\vec{b},\vec{c},\vec{d})$.

Remarque. *Preuve ensembliste*. Puis qu'on parle ci-dessus de propriétés (acquises), il peut être intéressant d'introduire explicitement les ensembles de propriétés qui caractérisent chaque item, et de faire une preuve ensembliste. Pour varier (un peu !), nous établissons ci-après la Proposition 5, *dans le cas où on utilise pour la conséquence logique la définition de la note en bas de page numéro* 6 qui privilégie la perte de propriété.

Pour ce faire on introduit $A = \{i \mid a_i = 1\}$, $B = \{i \mid b_i = 1\}$, $C = \{i \mid c_i = 1\}$, et $D = \{i \mid d_i = 1\}$. La definition $\models$ ci-dessus se traduit alors en

$$A \cap \overline{B} \subseteq C \cap \overline{D} \text{ et } \overline{C} \cap D \subseteq \overline{A} \cap B$$

Les deux conditions d'inclusion peuvent se réécrire

$A \cap \overline{B} \cap (\overline{C} \cup D) = \emptyset$ et $\overline{C} \cap D \cap (A \cup \overline{B}) = \emptyset$
$\Leftrightarrow$
$A \cap (\overline{B} \cap \overline{C}) = \emptyset$ et $A \cap \overline{B} \cap D = \emptyset$
et $\overline{C} \cap D \cap A = \emptyset$ et $\overline{C} \cap D \cap \overline{B} = \emptyset$
$\Leftrightarrow$
$(A \cup D) \cap \overline{B} \cap \overline{C} = \emptyset$ et $(\overline{B} \cup \overline{C}) \cap A \cap D = \emptyset$
$\Leftrightarrow$
$A \cap D \subseteq B \cap C$ et $A \cup D \subseteq B \cup C$.

En utilisant l'équivalence ci-dessus, on peut vérifier que quand lorsque $(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$, les 5 valuations $(1, 0, 0, 0)$, $(1, 0, 0, 1)$, $(1, 0, 1, 1)$, $(0, 0, 0, 1)$, $(1, 1, 0, 1)$ sont *interdites* pour chaque composante (a_i, b_i, c_i, d_i). Ceci signifie que

— $(a_i, b_i) = (1, 0) \Rightarrow (c_i, d_i) = (1, 0)$;
— $a_i = b_i \Rightarrow (c_i, d_i) \neq (0, 1)$

En d'autres termes, tout changement de 1 vers 0 dans $(\vec{a},\vec{b})$ existe aussi dans $(\vec{c},\vec{d})$, et sur les composantes où $\vec{a}$ et $\vec{b}$ sont égaux, $\vec{c}$ et $\vec{d}$ sont égaux ou présentent ce même changement de 1 vers 0. De même, $(\vec{c},\vec{d}) \models (\vec{a},\vec{b})$ est équivalent à :
$\begin{cases} (a_i, b_i) = (0, 1) \Rightarrow (c_i, d_i) = (0, 1) \\ a_i = b_i \Rightarrow (c_i, d_i) \neq (1, 0) \end{cases}$
qui interdit les 5 valuations $(0, 1, 0, 0)$, $(0, 1, 1, 0)$, $(0, 1, 1, 1)$, $(0, 0, 1, 0)$, $(1, 1, 1, 0)$. On a donc bien, comme attendu :
$(\vec{a},\vec{b}) \models (\vec{c},\vec{d})$ et $(\vec{c},\vec{d}) \models (\vec{a},\vec{b})$ ssi $A(\vec{a},\vec{b},\vec{c},\vec{d})$. □

3.4 Logiques trivaluées et connecteurs de paires

Une façon naturelle d'associer une tri-valuation à une paire ordonnée $(\vec{a},\vec{b})$, du point de vue de l'analogie, est de faire la différence des vecteurs pour obtenir

5. Remarquons que si on note $\vec{1}$ le vecteur dont toutes les composantes sont nulles, alors $(\vec{1},\vec{b}) \models (\vec{1},\vec{d})$ se réduit à $\vec{b} \models \vec{d}$, ce qui correspond à la relation de conséquence propositionnelle classique.

6. Le choix de la définition (3), plutôt que $(\vec{a},\vec{b}) \models (\vec{c},\vec{d}) \Leftrightarrow \vec{a} \wedge \neg\vec{b} \models \vec{c} \wedge \neg\vec{d}$ et $\neg\vec{c} \wedge \vec{d} \models \neg\vec{a} \wedge \vec{b}$, est gouverné par le besoin de privilégier l'acquisition de propriétés plutôt que leur perte.

un vecteur $val_A(\vec{a},\vec{b}) = (a_1-b_1, ..., a_n-b_n) \in \{-1,0,1\}^n$.

On peut alors vérifier que si $A(\vec{a},\vec{b},\vec{c},\vec{d})$ est vrai, on a

$$(\vec{a}\wedge\vec{c})-(\vec{b}\wedge\vec{d}) = val_A(\vec{a},\vec{b}) = val_A(\vec{c},\vec{d}) = (\vec{a}\vee\vec{c})-(\vec{b}\vee\vec{d}).$$

Cela suggère de définir (composante par composante) :

$$(\vec{a},\vec{b}) \wedge (\vec{c},\vec{d}) = (\vec{a}\wedge\vec{c},\vec{b}\wedge\vec{d})$$
$$(\vec{a},\vec{b}) \vee (\vec{c},\vec{d}) = (\vec{a}\vee\vec{c},\vec{b}\vee\vec{d})$$
$$\neg(\vec{a},\vec{b}) = (\neg\vec{a},\neg\vec{b})$$

En conséquence, on a

$$(\vec{a},\vec{b}) \wedge (\vec{a},\vec{b}) = (\vec{a},\vec{b}) = (\vec{a},\vec{b}) \vee (\vec{a},\vec{b})$$

Notons que $\neg(\vec{a},\vec{b}) \neq (\vec{b},\vec{a})$ en général.

Cependant on peut observer que

$$(\vec{a},\vec{b}) \wedge (\vec{c},\vec{d}) \not\models (\vec{a},\vec{b}) \not\models (\vec{a},\vec{b}) \vee (\vec{c},\vec{d}).$$

C'est simplement parce qu'une propriété acquise de $\vec{a}\wedge\vec{c}$ vers $\vec{b}\wedge\vec{d}$ peut ne pas l'être en passant de $\vec{a}$ à $\vec{b}$. Partant de l'exemple $(a_i,b_i,c_i,d_i) = (1,1,0,1)$, on obtient $(a_i\wedge c_i, b_i\wedge d_i) = (0,1)$, la propriété *i* est acquise dans le passage de $\vec{a}\wedge\vec{c}$ à $\vec{b}\wedge\vec{d}$, mais elle ne l'est pas dans le passage de $\vec{a}$ à $\vec{b}$: $(0,1) \not\models (1,1)$. [7]

En fait, la relation de conséquence logique $\models$ définie par (3) préserve les paires de la forme $(0,1)$, tandis que la conjonction des paires préserve $(0,1)$ si $(0,1)$ apparait des deux côtés de la conjonction, mais aussi quand une des paires est égale à $(1,1)$, pour la même propriété. Cela nous conduit à introduire un nouvel opérateur $\wedge\vee$ utilisant à la fois conjonction et disjonction :

$$(\vec{a},\vec{b}) \wedge\vee (\vec{c},\vec{d}) \triangleq (\vec{a}\wedge\vec{c},\vec{b}\vee\vec{d})$$

Si la notion de conséquence logique entre paires fait sens, l'intuition derrière cette conjonction / disjonction reste plus fragile. On peut noter que $(a_i\wedge c_i, b_i\vee d_i) = (1,0)$ seulement si $(a_i,b_i) = (c_i,d_i) = (1,0)$. Par contraste, si (a_i,b_i) or $(c_i,d_i) = (0,1)$, $(a_i\wedge c_i, b_i\vee d_i) = (0,1)$.

De manière duale, on peut définir :

$$(\vec{a},\vec{b}) \vee\wedge (\vec{c},\vec{d}) = (\vec{a}\vee\vec{c},\vec{b}\wedge\vec{d})$$

Notons que $(a_i\vee c_i, b_i\wedge d_i) = (0,1)$ seulement si $(a_i,b_i) = (c_i,d_i) = (0,1)$. Mais, si on a (a_i,b_i) ou $(c_i,d_i) = (1,0)$, alors $(a_i\vee c_i, b_i\wedge d_i) = (1,0)$. On peut alors vérifier que $\vee\wedge$ se comporte comme une conjonction et $\wedge\vee$ comme une disjonction, au sens où :

Proposition 6

$$(\vec{a},\vec{b}) \vee\wedge (\vec{c},\vec{d}) \models (\vec{a},\vec{b}) \models (\vec{a},\vec{b}) \wedge\vee (\vec{c},\vec{d})$$

où $\models$ est défini par (3).

Preuve. On doit d'abord montrer que $(a\vee c, b\wedge d) \models (a,b)$. Ce qui tient en effet puisque on a 1. $\neg(a\vee c)\wedge b\wedge d \models \neg a\wedge b$; 2. $a\wedge\neg b \models (a\vee c)\wedge\neg(b\wedge d)$.

Reste à vérifier que $(a,b) \models (a\wedge c, b\vee d)$. On a bien en effet 1. $\neg a\wedge b \models \neg(a\wedge c)\wedge(b\vee d)$; 2. $a\wedge c\wedge\neg(b\vee d) \models a\wedge\neg b$. □

Remarque On pourrait aussi définir une conséquence logique en partant de la paralogie, telle que $(\vec{a},\vec{b}) \models_P (\vec{c},\vec{d})$ ssi $\vec{a}\wedge\vec{b} \models \vec{c}\wedge\vec{d}$ et $\neg\vec{c}\wedge\neg\vec{d} \models \neg\vec{a}\wedge\neg\vec{b}$, ou alternativement $(\vec{a},\vec{b}) \models'_P (\vec{c},\vec{d})$ ssi $\neg\vec{a}\wedge\neg\vec{b} \models \neg\vec{c}\wedge\neg\vec{d}$ et $\vec{c}\wedge\vec{d} \models \vec{a}\wedge\vec{b}$. De plus, la tri-valuation naturellement associée avec une paire, du point de vue de la paralogie, serait $val_P(\vec{a},\vec{b}) = (a_1+b_1, ..., a_n+b_n) \in \{0,1,2\}^n$. L'étude de ces notions de conséquence logique et des logiques associées est laissée à plus tard.

3.5 Vers une logique de la créativité

Considérons un ensemble S un ensemble de profils décrivant des individus ou items existants et $P \subset S\times S$ un sous-ensemble représentatif de paires ordonnées qui illustrent des changements "intéressants" entre profils. P constitue ainsi une base de connaissance sur des changements réalisables. En d'autres termes, un vecteur est le profil d'un item existant, et chaque paire ordonnée de vecteurs peut être interprétée comme représentant des changements possibles/légitimes entre deux profils. Plus précisément, P est constitué de k paires $(\vec{a}^j,\vec{b}^j)$ avec $j = 1,k$, où chaque vecteur est une représentation booléenne d'une paire d'individus appartenant à un univers réel.

Etant donnée un profil $\vec{c} \in S$, on peut se demander si on pourrait obtenir de nouveaux profils plausibles $\vec{d} \notin S$ sur la base des changements possibles déjà observés sur l'ensemble P. Ces nouveaux profils seraient des représentants légitimes d'individus réalisables. La réponse pourrait être l'ensemble des solutions (si elles existent)

$$\vec{d} \in \{\vec{x^j} \mid A(\vec{a}^j,\vec{b}^j,\vec{c},\vec{x^j}) \text{ pour } j = 1,k\}$$

Dans le cas où aucune solution n'est trouvée, on pourrait élargir la base de connaissance initiale formée des paires d'éléments de S en calculant toute ou partie de la fermeture de l'opérateur $\wedge\vee$ tel que défini dans la section précédente.

7. Il y a deux autres cas de violation quand $(a_i,b_i) = (1,0)$: $(c_i,d_i) = (0,0)$ ou $(c_i,d_i) = (0,1)$, on obtient $(a_i\wedge c_i, b_i\wedge d_i) = (0,0)$, et $(0,0) \not\models (1,0)$. Enfin, $(\vec{a},\vec{b}) \not\models (\vec{a},\vec{b}) \vee (\vec{c},\vec{d})$ dans 3 situations possibles : i) $(a_i,b_i) = (0,0)$, $(c_i,d_i) = (1,0)$ et $(0,0) \not\models (1,0)$; ii) & iii) $(a_i,b_i) = (0,1)$, $(c_i,d_i) = (1,1)$ ou $(c_i,d_i) = (1,0)$, et $(0,1) \not\models (1,1)$.

Cette opération a le mérite de "*cumuler*" les acquisitions de propriétés. [8]

Cette manière de raisonner établit un parallèle avec le raisonnement non monotone sur des objets conditionels, où, à partir d'une base de règles par défaut "si a^j alors b^j" représentée par un ensemble d'objets conditionels $b^j|a^j$, on déduit un nouvel objet conditionel $d|c$, par la conséquence logique définie par (3) et la conjonction &, où c correspond à ce que l'on sait dans le contexte courant, et pour lequel on peut conclure d [4].

4 Premières expérimentations

Cette approche ne prend son sens pratique que lorsque l'on s'intéresse à des représentations booléennes de relativement grande dimension. En effet, dans ce contexte, les données dont on dispose sont en général peu nombreuses en regard de l'univers dans sa totalité. Par exemple pour des vecteurs de dimension 30, l'espace des profils possibles est de taille $2^{30} \sim 10^9$. Si on possède un échantillon S de taille 1000, il est naturel de s'intéresser à une extension "raisonnable" de l'échantillon. C'est là qu'intervient d'abord l'analogie avec l'extension analogique qui consiste à compléter l'ensemble des exemples de départ, comme par exemple dans [2]. Mais si l'analogie ne fournit pas suffisamment de nouveaux éléments, on pourrait alors mettre en oeuvre, dans un premier temps, la conséquence logique des paires, vu comme un moyen d'affaiblir la contrainte analogique de la manière suivante :

— Toute paire $(\vec{a}, \vec{b})$ de l'échantillon représente une variation possible des profils.
— Toute paire $(\vec{c}, \vec{d})$ telle que $(\vec{a}, \vec{b}) \models (\vec{c}, \vec{d})$ peut être considérée éventuellement comme la description d'une variation candidate des profils.

En l'absence d'un algorithme efficace, la tâche de générer des conséquences logiques peut s'avérer très complexe. On peut cependant d'abord s'interroger sur l'existence, au sein de l'échantillon P, de paires $(\vec{c}, \vec{d})$ qui sont conséquence logique d'une autre paire $(\vec{a}, \vec{b})$ dans P ici pris égal à $S \times S$. Afin de répondre à cette question, nous avons réalisé des expériences en dimension 10 et 30 (avec des temps d'exécution raisonnables) en variant la taille de l'échantillon S. Nous avons ensuite calculé en moyenne sur 10 tests, le nombre de paires $(\vec{c}, \vec{d})$ qui sont conséquence logique d'une autre paire.

On constate que le ratio $\frac{\#cons.log.}{\#paires}$ est toujours faible : donc il existe relativement peu de paires conséquence logique l'une de l'autre à l'intérieur de P.

8. Cumulatif veut dire ici que si $(a_i, b_i) = (0, 1)$ et $(c_j, d_j) = (0, 1)$ alors les composantes i et j de $(\vec{a}, \vec{b}) \wedge\!\vee (\vec{c}, \vec{d})$ sont aussi égales à $(0, 1)$. Notons cependant que $(0, 0) \wedge\!\vee (1, 1) = (1, 1) \wedge\!\vee (0, 0) = (0, 1)$ peut créer un changement non légitime ; dans ce cas, la paire générée ne doit pas être considérée dans la suite du processus.

Dim	Taille S	# paires	# tests	# cons. log.
10	50	1225	10	32
10	100	4950	10	120
30	100	4950	10	1
30	500	124750	10	2200

TABLE 4 – Nombre de paires conséquences logiques à l'intérieur de l'échantillon S

Une deuxième expérience consiste à résoudre l'équation $(\vec{a}, \vec{b}) \models (\vec{c}, \vec{d})$ où $\vec{a}, \vec{b}, \vec{c}$ sont dans S et on recherche au moins un $\vec{d}$ s'il existe, qui ne soit pas dans S. Là encore :

— Toute paire $(\vec{a}, \vec{b})$ de l'échantillon représente une variation possible des profils.
— Etant donné un autre profil $\vec{c}$ de S, un profil $\vec{d} \notin S$ tel que $(\vec{a}, \vec{b}) \models (\vec{c}, \vec{d})$ peut être considéré comme plausible et être ajouté à l'échantillon initial.

Nous avons expérimenté en dimension 10, 30 et 50 avec diverses tailles d'échantillon. Pour chaque paire $(\vec{a}, \vec{b})$, nous comptons le nombre total de $\vec{d}$ pour lesquels il existe un $\vec{c} \in S$ tel que $(\vec{a}, \vec{b}) \models (\vec{c}, \vec{d})$. Nous moyennons ce nombre à la fois sur le nombre de paires et le nombre de tests.

Dim.	Taille S	# paires	# tests	# $\vec{d} \in S$	# $\vec{d} \notin S$
10	50	1225	100	1	13
10	100	4950	100	3	25
30	100	4950	100	0	4
30	500	124750	100	0	9
50	100	4950	100	0	2
50	500	124750	100	0	2

TABLE 5 – Nombre de vecteurs $\vec{d}$ solutions de l'équation

On constate que, en dimension 50, la taille de S est très faible en regard de la taille de l'univers 2^{50} et que dans ce cas, en moyenne, une paire $(\vec{a}, \vec{b})$ sera susceptible de produire très peu de nouveaux vecteurs $\vec{d}$. Cela signifie que l'équation n'a pas de solution en général.

On peut enfin utiliser l'opérateur $\wedge\!\vee$ vu comme créateur de paires. On va compter dans la Table 6 combien de paires totalement nouvelles sont créées quand on applique l'opérateur $\wedge\!\vee$ à toutes les paires issues de l'échantillon S. A ce stade, on n'élimine pas les paires où apparaîtrait au moins une composante $(0, 0) \wedge\!\vee (1, 1)$ ou $(1, 1) \wedge\!\vee (0, 0)$. Cf. note de bas de page numéro 8.

Puisque les paires obtenues ne sont retenues que si les 2 vecteurs qui la constituent ne sont pas dans S, on a construit au moins $\#nouvellespaires$ nouveaux vecteurs (un nouveau vecteur pouvant apparaître dans plusieurs nouvelles paires). Cependant, le ratio $\frac{\#nouvellespaires}{\#paires}$ semble diminuer quand on augmente la taille de S.

Le raisonnement analogique ne conduit qu'à des conséquences plausible. Son usage pour la créativité n'échappe pas à cette règle. Il sera certainement utile en pratique de vérifier, d'une manière ou d'une autre, la réalisabilité de ces nouvelles paires.

Dim.	Taille S	# paires	# tests	# nouvelles paires
10	50	1225	10	334
10	100	4950	10	547
30	100	4950	10	9300
30	500	124750	10	Non Disponible
50	100	4950	10	9700
50	500	124750	10	Non Disponible

TABLE 6 – Nombre de paires déduites formées de vecteurs n'apparaissant pas dans S

5 Remarques de conclusion

Cette note a commencé à explorer l'idée que les proportions logiques en tant que connecteurs quaternaires pouvaient être aussi vues comme définissant des relations entre paires ordonnées, et que, de la même façon qu'une logique (tri-valuée) des objets conditionnels se trouvait associée à des proportions conditionelles, il était concevable d'explorer la possibilité d'une logique de paires en association avec des proportions logiques homogènes.

Cela a permis de mettre en évidence l'idée de classe d'équivalence de paires, pour les proportions analogiques (qui pourrait être aussi développée pour les proportions paralogiques). Quelques résultats ont été présentés sur la composition de relations entre paires, ainsi qu'une relation de conséquence logique entre paires.

Il est clair que nous n'en sommes qu'aux premiers balbutiements de la construction d'une logique de paires. Une question évidemment importante concerne l'usage pratique d'une telle logique. Comme les proportions logiques homogènes sont créatives au sens où à partir de 3 vecteurs distincts on peut produire un 4 ème vecteur différent des 3 premiers [9], on peut se demander comment elle pourrait contribuer à une logique de la créativité.

Remerciements

Le premier auteur remercie Jean Lieber pour avoir attiré à plusieurs reprises son attention sur l'intérêt potentiel de développer une logique de paires ordonnées. Mais ce n'est qu'en remarquant - enfin - que les proportions analogiques définissaient une relation d'équivalence entre deux paires ordonnées, tout comme la logique des objets conditionnels a pour point de départ l'équivalence de deux objets conditionnels (ce qui constitue un autre type de proportion logique), que le premier auteur a entrevu la possibilité de la logique des paires ordonnées présentée ici.

Cette recherche a bénéficié du soutien du projet ANR "Analogies : de la théorie aux outils et aux applications" (AT2TA), ANR-22-CE23-002.

9. Pourvu que les équations $T(a_i, b_i, c_i, x)$ aient des solutions.

Références

[1] Benferhat, S., D. Dubois et H. Prade: *Nonmonotonic reasoning, conditional objects and possibility theory*. Artificial Intelligence, 92(1-2) :259–276, 1997.

[2] Couceiro, M., N. Hug, H. Prade et G. Richard: *Analogy-preserving functions : A way to extend Boolean samples*. Dans Sierra, C. (rédacteur) : *Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17), Melbourne, Aug. 19-25*, pages 1575–1581, 2017.

[3] De Finetti, B.: *La logique des probabilités*. Dans *Congrès Int. de Philosophie Scientifique, IV. Induction et probabilité*, pages 31–39, Paris,, 1936. Hermann.

[4] Dubois, D. et H. Prade: *Conditional objects as nonmonotonic consequence relationships*. IEEE Trans. on Syst., Man and Cyber., 24 :1724–1740, 1994.

[5] Klein, S.: *Analogy and mysticism and the structure of culture (and Comments & Reply)*. Current Anthropology, 24 (2) :151–180, 1983.

[6] Kraus, S., D. Lehmann et M. Magidor: *Nonmonotonic reasoning, preferential models and cumulative logics*. Artificial Intelligence, 44 :167–207, 1990.

[7] Lepage, Y. et C.-L. Goh: *Towards automatic acquisition of linguistic features*. Dans Jokinen, K. et E. Bick (rédacteurs) : *Proc. 17th Nordic Conf. of Computational Linguistics, NODALIDA'09, Odense, Denmark, May 14-16*, pages 118–125. Northern European Association for Language Technology (NEALT), 2009.

[8] Miclet, L. et H. Prade: *Handling analogical proportions in classical logic and fuzzy logics settings*. Dans *Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09),Verona*, pages 638–650. Springer, LNCS 5590, 2009.

[9] Prade, H. et G. Richard: *Homogeneous logical proportions : Their uniqueness and their role in similarity-based prediction*. Dans Brewka, G., T. Eiter et S. A. McIlraith (rédacteurs) : *Proc. 13th Int. Conf., Principles of Knowledge Representation and Reasoning (KR'12), Rome, June 10-14*, pages 402–412. AAAI Press, 2012.

[10] Prade, H. et G. Richard: *From analogical proportion to logical proportions*. Logica Universalis, 7(4) :441–505, 2013.

[11] Prade, H. et G. Richard: *Homogenous and heterogeneous logical proportions*. IfCoLog J. of Logics and their Applications, 1(1) :1–51, 2014.

Session 4 : Explicabilité

Actes JIAF-JFPDA 2023

Des explications transitives questionnables au service de l'élicitation de préférences additives

Manuel Amoussou[1] **Khaled Belahcene**[1] **Nicolas Maudet**[2] **Vincent Mousseau**[1] **Wassila Ouerdane**[1]

[1]MICS, CentraleSupélec, Université Paris-Saclay, France
[2]LIP6, Sorbonne Université, CNRS, France

{manuel.amoussou, khaled.belahcene}@centralesupelec.fr,
{vincent.mousseau, wassila.ouerdane}@centralesupelec.fr,
nicolas.maudet@lip6.fr

Résumé

On considère un modèle additif de décision multicritère dans lequel les critères sont évalués sur une échelle binaires. Le décideur fournit une information préférentielle ($\mathbb{PI}$) sur la base de laquelle une relation de préférence nécessaire est établie. Nous proposons d'expliquer une paire de cette préférence nécessaire par une chaîne transitive composée de swaps nécessaires, d'éléments de la $\mathbb{PI}$, et de swaps non-nécessaires. Une telle explication est appelée explication transitive questionnable. Nous proposons une méthode de calcul de telles explications et montrons comment de telles explications peuvent s'insérer naturellement dans une procédure d'élicitation de préférences.

Abstract

We consider an additive multi-criteria decision model in which the criteria are evaluated on a binary scale. The decider provides preferential information ($\mathbb{PI}$) on the basis of which a necessary preference relation is established. We propose to explain a pair of this necessary preference by a transitive chain composed of necessary swaps, elements of the $\mathbb{PI}$, and non-necessary swaps. Such an explanation is called questionable transitive explanation. We propose a method for the computation of such explanations and show how such explanations can fit naturally into a preference elicitation procedure.

1 Introduction

La question de l'explication en Aide MultiCritère à la Décision a fait l'objet, ces deux dernières décennies, de travaux scientifiques ([1], [2], [6], [7]) destinés à éclairer un décideur (le plus souvent peu connaisseur des modèles théoriques de décision) sur les éléments d'une recommandation produite par un système automatique piloté ou non par un analyste ; ces travaux visent à plus de transparence et ainsi à accroitre la confiance des utilisateurs de tels systèmes. La plupart de ces travaux se sont donc employés à produire des explications suffisamment simples pour être comprises d'un décideur humain tout en veillant à les garder le plus possible fidèles aux modèles de décision sous-jacents.

Dans cet article, nous nous intéressons à une seconde fonction de l'explication : celle de l'élicitation des préférences. En effet, on attend d'une explication intelligible proposée à un décideur qu'elle suscite chez ce dernier une réaction (acceptation, réfutation, ou clarification). La réaction du décideur devient ainsi, une occasion de collecte d'informations sur ses préférences, lesquelles peuvent être utilisées par l'analyste pour enrichir le modèle et fournir une recommandation plus adaptée.

L'article est organisé de la façon suivante : nous considérons des problèmes de décision dans lesquels les alternatives sont décrites sur un ensemble d'échelles binaires de critères et nous faisons l'hypothèse de la représentabilité des préférences du décideur par un modèle additif (voir Section 2). La Section 3 propose un algorithme de calcul d'explication des comparaisons par paire déduites du modèle basé sur la programmation mathématique. Nous proposons ensuite en Section 4 une ébauche de protocole interactif dans lequel l'explication pourrait remplir pleinement sa fonction d'élicitation ; ce que nous illustrons par un exemple. L'article s'achève par une conclusion et l'évocation de perspectives futures (Section 5).

2 Préliminaires

Dans cette section, nous précisons le contexte de décision considéré (Sous-section 2.1) ainsi que l'ensemble des comparaisons par paire déduites sujettes à explication (Sous-section 2.2) et détaillons notre proposition d'explication transitive questionnable (Sous-Section 2.3).

2.1 Contexte de décision

Le cadre que nous considérons est celui de l'Aide Multi-Critère à la Décision (AMCD) où sont en présence deux acteurs : un analyste et un décideur. Le décideur requiert « l'aide » de l'analyste pour opérer le choix d'un sous-ensemble $\mathbf{A}^*$ d'alternatives parmi un ensemble (plus grand) $\mathbf{A}$ d'alternatives décrites sur m critères binaires c'est-à-dire ayant exactement deux niveaux d'évaluation ("fort" et "faible"). On notera $\mathbf{X}$ le produit cartésien de m échelles binaires ordonnées $X_{\mathtt{i}} = \{\mathtt{0}, 1\}$ où $\mathtt{0}$ et 1 représentent respectivement les niveaux "faible" et "fort" de chaque critère :

$$\mathbf{X} = \prod_{\mathtt{i} \in [m]} X_{\mathtt{i}}$$

avec $[m]$ l'ensemble des critères. On a $\mathbf{A} \subset \mathbf{X}$.

Les préférences du décideur sont collectées sous la forme d'un ensemble noté $\mathbb{PI}$ (Preference Information) de comparaisons par paire d'alternatives (x, y) qui traduisent sa préférence de l'alternative x sur l'alternative y. Ces alternatives, sur lesquelles le décideur sait exprimer des préférences, forment l'ensemble $\mathbf{A}^R$ des alternatives dites de *référence* ; les ensembles $\mathbf{A}^R$ et $\mathbf{A}$ étant le plus souvent disjoints.

Nous faisons l'hypothèse que les préférences du décideur sont représentables par un modèle additif qui peut être décrit comme suit :

Définition 1 (Modèle additif avec critères binaires) *Une relation de préférence $\succsim$ sur $\mathbf{X}$ est représentable par un modèle additif si et seulement si il existe une fonction de score $\omega : \langle \omega_{\mathtt{i}} \rangle_{\mathtt{i} \in [m]}$ avec $\omega_{\mathtt{i}} : X_{\mathtt{i}} \to \mathbb{R}^+$ telle que :*

$$x \succsim y \iff \omega(x) = \sum_{\mathtt{i} \in [m]} \omega_{\mathtt{i}}(x_{\mathtt{i}}) \geq \sum_{\mathtt{j} \in [m]} \omega_{\mathtt{j}}(y_{\mathtt{j}}) = \omega(y)$$

où $x_{\mathtt{i}}$ (resp. $y_{\mathtt{j}}$) symbolise l'évaluation ($\mathtt{0}$ ou 1) de l'alternative x (resp. y) sur le $\mathtt{i}-$ème (resp. $\mathtt{j}-$ième) critère.

Sans perte de généralité, nous considérerons dans la suite que $\omega_{\mathtt{i}}(\mathtt{0}) = 0$ pour tout $\mathtt{i} \in [m]$; ce qui simplifiera entre autres l'écriture de la fonction de score ω.

Nous faisons également l'hypothèse que l'ensemble $\mathbb{PI}$ des préférences du décideur (collectées sous la forme de comparaisons par paire d'alternatives de référence) est *consistant* c'est-à-dire qu'il se « fond » dans au moins une relation de préférence $\succsim$ sur $\mathbf{X}$ représentable par un modèle additif : $\mathbb{PI} \subset \succsim$. L'ensemble des comparaisons composant $\mathbb{PI}$ peut donc être restitué par une ou plusieurs fonctions de score ω qui seront dites *compatibles* avec $\mathbb{PI}$.

Définition 2 (Fonction de score compatible avec $\mathbb{E}$) *Soit $\mathbb{E}$ un ensemble de comparaisons par paire (x, y). La fonction de score ω est dite compatible avec $\mathbb{E}$ si :*

$$\omega(x) \geq \omega(y) \quad \forall\, (x, y) \in \mathbb{E} \tag{1}$$

Lorsqu'il n'existe aucune fonction de score compatible avec $\mathbb{E}$ alors $\mathbb{E}$ est dit inconsistant. Dans le cas contraire, $\mathbb{E}$ est dit consistant.

L'existence d'une fonction de score compatible avec $\mathbb{E}$ peut être testée à l'aide d'un programme linéaire (sans fonction objectif) dont les contraintes sont définies par les inéquations (1).

Dans la suite, nous désignerons par $\mathbb{A}$ (resp. $\mathbb{X}$) l'ensemble des comparaisons par paire $\mathbf{A} \times \mathbf{A}$ (resp. $\mathbf{X} \times \mathbf{X}$).

Exemple 1 *Le présent exemple décrit une situation de décision dans laquelle un décideur sollicite l'aide d'un analyste pour le choix de quatre alternatives parmi les huit que compte l'ensemble $\mathbf{A}$ (voir Table 1). Chacune des alternatives est décrite sur $m = 6$ critères binaires symbolisés par des lettres ($[m] = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$).*

Table 1 – Description de l'ensemble $\mathbf{A}$.

	a	b	c	d	e	f
x^3	0	0	0	0	1	1
x^{13}	0	0	1	1	0	1
x^{21}	0	1	0	1	0	1
x^{24}	0	1	1	0	0	0
x^{34}	1	0	0	0	1	0
x^{37}	1	0	0	1	0	1
x^{40}	1	0	1	0	0	0
x^{49}	1	1	0	0	0	1

La Table 2 décrit l'ensemble des alternatives de référence (que le décideur « connaît bien ») sur lequel il fournit un ordre de préférence complet représenté par $\mathbb{PI}$:

Table 2 – Description de l'ensemble $\mathbf{A}^R$.

	a	b	c	d	e	f
r^7	0	0	0	1	1	1
r^{12}	0	0	1	1	0	0
r^{19}	0	1	0	0	1	1
r^{52}	1	1	0	1	0	0

$$\mathbb{PI} = \{(r^{52}, r^{19}),\ (r^{19}, r^7),\ (r^7, r^{12})\}$$

La fonction de score ω suivante est compatible avec $\mathbb{PI}$: $\omega = \{\mathbf{a} : 23, \mathbf{b} : 49, \mathbf{c} : 40, \mathbf{d} : 19, \mathbf{e} : 6, \mathbf{f} : 35\}$.

En effet, on a bien :

$\omega(r^{52}) = 23 + 49 + 19 = 91 \geq 90 = 49 + 6 + 35 = \omega(r^{19})$

$\omega(r^{19}) = 90 \geq 60 = \omega(r^7)$ *et* $\omega(r^7) = 60 \geq 59 = \omega(r^{12})$

2.2 Relation nécessaire et recommandations

Nous nous proposons ici d'analyser l'impact de l'information préférentielle $\mathbb{PI}$ collectée auprès du décideur sur l'ensemble des comparaisons par paire d'alternatives (d'intérêt) $\mathbf{A}$ d'une part et sur l'ensemble des recommandations qui pourraient être faites au décideur d'autre part.

2.2.1 Relation nécessaire

En désignant par $\Omega_{\mathbb{PI}}$ l'ensemble des fonctions de score compatibles avec $\mathbb{PI}$ (Définition 2), il est évident que l'on a :

$$\omega(x) - \omega(y) \geq 0 \quad \forall\, (x,\, y) \in \mathbb{PI} \;\; \forall\, \omega \in \Omega_{\mathbb{PI}}.$$

Le concept de « relation nécessaire étant donné $\mathbb{PI}$ » va au-delà de la seule considération des comparaisons par paire appartenant à $\mathbb{PI}$ et étend la condition $\omega(x) - \omega(y) \geq 0$ à toutes les comparaisons par paire $(x,\, y) \in \mathbb{X}$:

Définition 3 (Relation nécessaire étant donné $\mathbb{PI}$) *Étant donnés un ensemble consistant de comparaisons par paire $\mathbb{PI}$ et deux alternatives $x,\, y \in \mathbf{X}$, on dit que x est nécessairement préférée à y si et seulement si on a :* $\omega(x) - \omega(y) \geq 0$ *pour toute fonction de score ω compatible avec $\mathbb{PI}$ ($\omega \in \Omega_{\mathbb{PI}}$).*

On notera $\mathbb{N}_{\mathbb{PI}}$ l'ensemble des comparaisons par paire $(x,\, y) \in \mathbb{X}$ telles que x est nécessairement préférée à y.

$\mathbb{N}_{\mathbb{PI}}$ est une relation binaire réflexive et transitive[3].

L'ensemble $\mathbb{P}$ des comparaisons par paire $(x,\, y)$ tel que : $y_i = 1 \Rightarrow x_i = 1$ *traduisant la dominance de Pareto est (trivialement) inclus dans $\mathbb{N}_{\mathbb{PI}}$.*

La Proposition 1 suggère une méthode de calcul des éléments de la relation nécessaire basée sur la résolution d'un programme linéaire. D'autres méthodes sont proposées dans la littérature, en particulier [3] qui est assez proche de la Proposition 1 et [2] basée sur l'utilisation des *cofacteurs*.

Proposition 1 *Étant donné un ensemble consistant de comparaisons par paire $\mathbb{PI}$, la comparaison par paire $(x,\, y) \in \mathbb{X}$ appartient à $\mathbb{N}_{\mathbb{PI}}$ si l'ensemble $\mathbb{PI} \cup \{(y,\, x)\}$ est inconsistant (voir Définition 2).*

Comme $\mathbf{A}$ est l'ensemble des alternatives sur lequel va porter la recommandation de l'analyste, l'ensemble $\mathbb{N}_{\mathbb{PI}} \cap \mathbf{A}$ que nous noterons par la suite $\mathbb{N}_{\mathbb{PI}}^{\mathbf{A}}$ revêt une importance particulière : il s'agit de comparaisons par paire déduites évidentes dont on peut dire que le décideur est « convaincu » (étant donnée l'hypothèse de représentabilité de ses préférences par un modèle additif) et par rapport auxquelles il est potentiellement à même d'exprimer les raisons de ces préférences. Les expliquer de façon « simple et intelligible » pourrait permettre à l'analyste de vérifier l'alignement du raisonnement véhiculé par les explications produites avec les convictions du décideur et donc de capturer le cas échéant de nouvelles informations préférentielles qui serviraient à réduire l'incertitude autour de l'ensemble $\mathbf{A}^*$ des alternatives à recommander.

2.2.2 Recommandations admissibles

Nous avons vu dans le paragraphe précédent, que l'information préférentielle $\mathbb{PI}$ pourrait permettre de conclure qu'une alternative $x \in \mathbf{A}$ est nécessairement préférée à une autre alternative $y \in \mathbf{A} : (x,\, y) \in \mathbb{N}_{\mathbb{PI}}^{\mathbf{A}}$. Cette déduction pourrait également se traduire en la réduction du nombre des ensembles $\mathbf{A}_{cand}$ « candidats » à recommander au décideur. En effet, la déduction $(x,\, y) \in \mathbb{N}_{\mathbb{PI}}^{\mathbf{A}}$ pourrait par exemple disqualifier une ou plusieurs recommandations candidates $\mathbf{A}_{cand}$ telle que $y \in \mathbf{A}_{cand}$.

Plus généralement, en considérant pour chaque recommandation candidate $\mathbf{A}_{cand}$ l'ensemble caractéristique de ses comparaisons par paire $\{(x,\, y),\; x \in \mathbf{A}_{cand}$ et $y \notin \mathbf{A}_{cand}\}$, il est possible de déterminer si $\mathbf{A}_{cand}$ peut être recommandé ou non.

Proposition 2 (Recommandation admissible étant donné $\mathbb{PI}$) *Un sous-ensemble de candidats $\mathbf{A}_{cand} \subset \mathbf{A}$ tel que $|\mathbf{A}_{cand}| = |\mathbf{A}^*|$ est admissible (ou recommandable) si l'ensemble des comparaisons par paire $\mathbb{PI} \cup \{(x,\, y),\; x \in \mathbf{A}_{cand}$ et $y \notin \mathbf{A}_{cand}\}$ est consistant (voir Définition 2).*

Exemple 2 (Suite de l'Exemple 1) *Étant donné $\mathbb{PI}$, les comparaisons par paires de $\mathbb{A}$ déduites sont les suivantes :*

$$\mathbb{N}_{\mathbb{PI}}^{\mathbf{A}} = \{(x^{49},\, x^{37}),\; (x^{37},\, x^3),\; (x^{49},\, x^3)\}$$

et le nombre de recommandations admissibles (de 4 alternatives parmi les 8) est de 20.

L'objet de notre contribution est de montrer comment est-ce que l'explication des éléments de la relation nécessaire pourrait permettre la collecte d'information préférentielle supplémentaire avec comme corollaire la diminution du nombre de recommandations admissibles.

2.3 Explication de la relation nécessaire

Dans [8] on peut lire : « Expliquer un évènement, c'est fournir des informations sur ses causes [1] ». Dans le domaine

1. Traduction de *"To explain an event is to provide some information about its causal history"[8].*

de l'Aide MultiCritère à la Décision (AMCD), un certain nombre de travaux ont développé, à travers des propositions diverses, une logique similaire dans le cas où l'« évènement » à expliquer est un élément de la relation nécessaire.

Ici, nous nous appuierons sur deux propositions de la littérature ([5] et [2]) pour définir une nouvelle structure d'explication qui combine les approches qui y sont développées.

2.3.1 Explication sous forme de *preferential reduct* [5]

Ce que les auteurs entendent par "*preferential reduct*", c'est tout simplement l'ensemble minimal des éléments de la $\mathbb{PI}$ qui justifie l'appartenance de la comparaison par paire déduite à l'ensemble $\mathbb{N}_{\mathbb{PI}}$. Cette façon d'expliquer pointe donc directement les comparaisons par paire fournies par le décideur qui sont la cause de la déduction faite. On remarquera qu'elle est complète dans le sens où l'ensemble explicatif d'éléments de la $\mathbb{PI}$ ne sera jamais vide. Elle ne révèle cependant pas les mécanismes (applications des propriétés du modèle additif) qui sont à l'oeuvre dans cette déduction ; ce qui, dans le cadre de la délivrance d'une explication à un décideur humain, peut s'avérer insuffisant.

Exemple 3 (Suite des Exemples 1 et 2) *Le sous-ensemble* $\{(r^{52}, r^{19}), (r^{19}, r^{7})\}$ *de* $\mathbb{PI}$ *est le preferential reduct de la déduction* $(x^{49}, x^{3}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}}$.

Les mécanismes qui sont à l'oeuvre dans cette déduction sont :

(i) l'application de la propriété de transitivité qui permet d'obtenir (r^{52}, r^{7}) *à partir de* (r^{52}, r^{19}) *et* (r^{19}, r^{7}).

(ii) l'application de la propriété de cancellation d'ordre 1 qui autorise un raisonnement « toute chose égale par ailleurs » :
$(r^{52}, r^{7}) \equiv (110101, 000111)$ *qui est équivalent à* $(x^{49}, x^{3}) \equiv (110001, 000011)$.
Les critères colorés étant ceux communs aux deux alternatives et le critère **d** *en bleu étant celui sur lequel s'est appliqué le changement conjoint d'évaluation.*

2.3.2 Explication à l'aide de *preference swaps* [2]

En guise de résumé très sommaire de la contribution de cet article ([2]), nous pouvons dire que l'explication de la comparaison par paire $(x, y) \in \mathbb{N}_{\mathbb{PI}}$ y est conçue comme une chaîne transitive reliant x à y et transitant par des alternatives réelles ($\in \mathbf{A} \cup \mathbf{A}^{R}$) ou fictives ($\notin \mathbf{A} \cup \mathbf{A}^{R}$) telle que les comparaisons par paire consécutives le long de cette chaîne appartiennent (aussi) à $\mathbb{N}_{\mathbb{PI}}$ et composées de (2) alternatives qui ne divergent que d'au plus deux critères (d'où le terme *swaps* dans *preference swaps*). On utilisera les crochets pour représenter le swap $[\mathtt{i}, \mathtt{j}]$ que l'on distinguera plus aisément de l'écriture d'une éventuelle comparaison par paire $(\mathtt{i}, \mathtt{j})$ qui utilise les parentèses.

Dans ce type d'explication, on note la matérialisation du souci de révéler les mécanismes sous-jacents à la déduction (en particulier la propriété de transitivité) doublé du désir de ramener chacune des comparaisons par paire d'alternatives consécutives (grâce à la propriété de cancellation d'orde 1) soit à l'expression d'une préférence de type dominance de Pareto soit à une confrontation entre exactement 2 critères. Ce faisant, on comprend aisément que cette façon d'expliquer offre un maximum de garantie d'intelligibilité pour un décideur humain car la "complexité" des atomes qui composent la chaîne explicative est réduite à son minimum [2].

Sans surprise, l'inconvénient majeur de ce type d'explication est son incomplétude : certaines comparaisons par paire déduites n'admettent pas d'explication comme on peut le constater dans l'exemple suivant.

Exemple 4 (Suite des Exemples 1 et 2) *La comparaison par paire* $(x^{49}, x^{3}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}}$ *n'est pas explicable à l'aide de preference swaps. Cette comparaison, par application de la propriété de cancellation d'ordre 1 est équivalente à la confrontation* $(\{\mathbf{a}, \mathbf{b}\}, \{\mathbf{e}\})$ *de sous-ensembles de critères. Pour que* (x^{49}, x^{3}) *soit explicable à l'aide de preference swaps, il aurait fallu avoir soit* $(x^{\{\mathbf{a}\}}, x^{\{\mathbf{e}\}}) \in \mathbb{N}_{\mathbb{PI}}$ *soit* $(x^{\{\mathbf{b}\}}, x^{\{\mathbf{e}\}}) \in \mathbb{N}_{\mathbb{PI}}$ *où l'alternative* $x^{\{\mathtt{i}\}}$ *vaut* $\mathbf{0}$ *sur tous les critères sauf sur le critère* $\mathtt{i}$ *; or on n'a ni* $(x^{\{\mathbf{a}\}}, x^{\{\mathbf{e}\}}) \in \mathbb{N}_{\mathbb{PI}}$ *ni* $(x^{\{\mathbf{b}\}}, x^{\{\mathbf{e}\}}) \in \mathbb{N}_{\mathbb{PI}}$.

Par contre, la comparaison par paire déduite : $(x^{49}, x^{37}) \equiv (110001, 100101) \equiv (x^{\{\mathbf{b}\}}, x^{\{\mathbf{d}\}})$ *en tant que preference swap est (trivialement) explicable.*

2.3.3 Explication transitive questionable

Comme indiqué en conclusion de la sous-section 2.2 et en introduction de la sous-section 2.3, notre proposition d'explication a pour ambition de combiner les avantages des approches d'explication résumées aux paragraphes 2.3.1 et 2.3.2 et de permettre la capture d'information préférentielle supplémentaire susceptible de réduire le nombre de recommandations admissibles.

En effet, les explications à construire (voir Section 3) seront :

— transitives et composées de *preference swap(s)* et d'élément(s) de la $\mathbb{PI}$

— et questionnables c'est-à-dire avec au moins un *swap* $[\mathtt{i}, \mathtt{j}]$ dont la projection (voir Définition 5) dans $\mathbb{X}$ n'appartenant pas à $\mathbb{N}_{\mathbb{PI}}$.

Une explication transitive questionnable conserve donc les garanties d'intelligibilité pour un décideur humain du fait de sa structure et du caractère atomique (les *swaps*) des éléments qui la composent. À défaut de constituer une preuve de la déduction de la comparaison à expliquer étant donné $\mathbb{PI}$, elle fournit une **autre** preuve potentielle de cette

2. Le lecteur intéressé pourra consulter [4] qui détaille une méthode « rationnelle » destinée à faciliter la prise de décision et basée sur les *even-swaps*.

déduction dont la validation ou la contestation des swaps contribue à acquérir de l'information préférentielle supplémentaire.

Exemple 5 (Suite des Exemples 1 et 2)

$$[x^{49}]\underbrace{110001 \gtrsim 110100}_{[f,d]}\overbrace{[r^{52}] \gtrsim [r^{19}]}^{\in \mathbb{PI}}\underbrace{010011 \gtrsim 100011}_{[b,a]}$$
$$:= \underbrace{100011 \gtrsim 100101}_{[e,d]}[x^{37}]$$

La séquence ci-dessus est un exemple d'explication transitive questionnable de la déduction (x^{49}, x^{37}). *Elle est composée d'une séquence de comparaisons par paires qui « lient »* x^{49} *à* x^{37}. *Sont mis en évidence les comparaisons par paire qui constituent des swaps et les éléments empruntés à la* $\mathbb{PI}$. *Le swap* $[e, d]$ *de projection dans* $\mathbb{X}$, *la comparaison par paire* $(100011, 100101)$ *est un « chaînon » dont la validation ou la contestation permettra d'enrichir la* $\mathbb{PI}$ *et potentiellement de réduire le nombre de recommandations admissibles.*

3 Calcul d'explications transitives questionnables

Dans cette section, nous allons montrer comment est-ce que le problème du calcul d'une explication transitive questionnable peut être modélisé sous la forme d'un programme mathématique mixte (Mixed Integer Program en anglais). Nous tenterons dans un premier temps une formalisation du problème en le ramenant à la recherche d'un chemin dans un graphe à définir (Sous-section 3.2). Nous donnerons ensuite les détails du programme mathématique dont la résolution fournit, lorsqu'elle existe, l'explication (la plus courte) de la déduction $(x, y) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}}$ (Sous-section 3.3).

Mais avant, nous introduisons un certain nombre de définitions de concepts empruntés à [2] et qui nous seront utiles par la suite.

3.1 Quelques définitions

Étant donnée une comparaison par paire $(u, v) \in \mathbb{X}$, nous rappelons ici les définitions des ensembles de critères $\texttt{pro}(u, v)$, $\texttt{con}(u, v)$ et $\texttt{neutr}(u, v)$ (Définition 4) qui nous permettront de mieux appréhender la notion de swap : expression de la préférence d'un critère sur un autre [3] qui peut se concevoir à travers différentes comparaisons par paire de $\mathbb{X}$ (Définition 5). Ces définitions seront complétées par l'introduction de la notion de *décomposition à base de swap(s)* (Définition 6) et de celle de longueur de la comparaison par paire $(u, v) \in \mathbb{X}$ en termes de swap(s) (Définition 7).

Définition 4 (Ensembles pro, con et neutr) *Soient u et v deux alternatives de* **X** *décrites sur l'ensemble* $[m]$ *de critères.*

Les ensembles $\texttt{pro}(u, v)$, $\texttt{con}(u, v)$ *et* $\texttt{neutr}(u, v)$ *de la comparairon par paire* (u, v) *sont définis comme suit :*

$$\texttt{pro}(u, v) = \{\texttt{i} \in [m],\ u_{\texttt{i}} = 1 \text{ et } v_{\texttt{i}} = 0\}$$
$$\texttt{con}(u, v) = \{\texttt{j} \in [m],\ u_{\texttt{j}} = 0 \text{ et } v_{\texttt{j}} = 1\}$$
$$\texttt{neutr}(u, v) = \{\texttt{i} \in [m],\ u_{\texttt{i}} = v_{\texttt{i}}\}$$

On remarquera que les trois ensembles $\texttt{pro}(u, v)$, $\texttt{con}(u, v)$ *et* $\texttt{neutr}(u, v)$ *partitionnent l'ensemble des critères* $[m]$ *et représentent respectivement l'ensemble de critères en faveur* (pro), *en défaveur* (con) *et neutre(s)* (neutr) *de la préférence traduite par la comparaison par paire* (u, v).

Définition 5 (Projection d'un swap dans $\mathbb{X}$) *Lorsque* $\texttt{pro}(u, v) = \{\texttt{i}\}$ *et* $\texttt{con}(u, v) = \{\texttt{j}\}$, *on dit que* (u, v) *est une projection du swap* $[\texttt{i}, \texttt{j}]$ *dans* $\mathbb{X}$.

Il existe exactement 2^{m-2} *projections différentes du swap* $[\texttt{i}, \texttt{j}]$ *dans* $\mathbb{X}$.

Définition 6 (Décomposition à base de swap(s)) *Soit* $(u, v) \in \mathbb{X}$ *une comparaison par paire. Une décomposition à base de swap(s) de* (u, v) *est une application injective* ψ *de* $\texttt{con}(u, v)$ *vers* $\texttt{pro}(u, v)$. *L'ensemble des swaps :* $\{[\psi^{-1}(\texttt{j}), \texttt{j}],\ \texttt{j} \in \texttt{con}(u, v)\}$ *caractérisent la décomposition lorsqu'elle existe. En effet, les comparaisons par paire* (u, v) *telles que :* $|\texttt{con}(u, v)| > |\texttt{pro}(u, v)|$ *n'admettent (par définition) pas de décomposition à base de swap(s).*

Eu égard à la définition précédente, toutes les *décompositions à base de swap(s)* de (u, v) sont caractérisées chacune par des ensembles de swaps $\{[\psi^{-1}(\texttt{j}), \texttt{j}],\ \texttt{j} \in \texttt{con}(u, v)\}$ qui sont tous de taille égale à $|\texttt{con}(u, v)|$. Ce constat permet donc d'introduire la définition suivante :

Définition 7 (Longueur de (u, v) en termes de swap(s)) *Soit* $(u, v) \in \mathbb{X}$. *On définit comme suit la longueur* $\mathrm{L}(u, v)$ *en termes de swap(s) de la comparaison par paire* (u, v) *:*

$$\mathrm{L}(u, v) = \begin{cases} +\infty & \text{si } (u, v) \text{ n'admet pas de décomposition à base de swaps} \\ |\texttt{con}(u, v)| + \mathrm{I}_{(u,v)} & \text{sinon} \end{cases}$$

avec $\mathrm{I}_{(u,v)} = 1$ *si* $|\texttt{pro}(u, v)| > |\texttt{con}(u, v)|$ *et* 0 *sinon.*

Exemple 6 (Suite des Exemples 1 et 5) *Considérons les 3 comparaisons par paires suivantes :* (x^{40}, x^{21}), (x^{49}, x^{3}) *et* (r^{19}, x^{37}). *La Table 3 en fournit les ensembles* pro *et* con, *une décomposition à base de swap(s) (lorsqu'elle existe) et sa longueur en termes de swap(s) :*

3. Plus rigoureusement, le swap $[\texttt{i}, \texttt{j}]$ traduit en termes de préférence la supériorité de l'intensité de la différence entre les niveaux "fort" et "faible" du critère i sur celle du critère j.

TABLE 3 – Décomposition à base de swap(s)

(u, v)	con(u, v)	pro(u, v)	ψ	L(u, v)
(x^{40}, x^{21})	{**b**, **d**, **f**}	{**a**, **c**}	n'existe pas	-
(x^{49}, x^{3})	{**e**}	{**a**, **b**}	ψ_1(**e**) = **a**	2
(r^{19}, x^{37})	{**a**, **d**}	{**b**, **e**}	ψ_2(**a**) = **b**	
			et ψ_2(**d**) = **e**	2

L'ensemble (singleton) {[**a**, **e**]} *caractérise* ψ_1 *tandis que l'ensemble* {[**b**, **a**], [**e**, **d**]} *caractérisent* ψ_2. *Remarquons, dans le cas de* ψ_2, *que ce sont justement les swaps* [**b**, **a**] *et* [**e**, **d**] *de projections respectives dans* $\mathbb{X}$ (**0**10**0**11, **1**00**0**11) *et* (**1**00**0**11, **1**00**1**0**1**) *qui composent la (sous-)chaîne reliant* r^{19} *à* x^{37} *dans l'explication transitive questionnable de* (x^{49}, x^{37}).

3.2 Formalisation du problème

Étant donnés un ensemble (consistant) $\mathbb{PI} = \{(u^1, v^1), \ldots, (u^k, v^k)\}$ de k comparaisons par paires et deux alternatives x, y telles que $(x, y) \in \mathbb{N}_{\mathbb{PI}}$, on définit le graphe orienté pondéré $\mathcal{G}^{\mathbb{PI}}_{(x,y)} = (V, E)$ appelé *graphe d'explication de* (x, y) *étant donné* $\mathbb{PI}$ comme suit :

- $V = \{x, y\} \cup \{u^1, \ldots, u^k\} \cup \{v^1, \ldots, v^k\}$
  Les sommets V de $\mathcal{G}^{\mathbb{PI}}_{(x,y)}$ représentent donc l'ensemble des alternatives impliquées dans $\mathbb{PI}$ complétées de x et y.
- $E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$ avec
  - $E_1 = \{(x, u^1), \ldots, (x, u^k)\}$
  - $E_2 = \{(v^1, y), \ldots, (v^k, y)\}$
  - $E_3 = \{(v^t, u^{t'}), \forall t, t' \in \{1, \ldots, k\}$ avec $t \neq t'$
  - $E_4 = \{(u^1, v^1), \ldots, (u^k, v^k)\} = \mathbb{PI}$
  - $E_5 = \{(x, y)\}$

  Les arcs E_4 représentent exactement les comparaisons par paire de la $\mathbb{PI}$. Les arcs E_1 symbolisent les liens directs entre le sommet-alternative x et les sommets entrants de E_4 tandis que les arcs E_2 représentent les liens directs entre les sommets sortants de E_4 ($\mathbb{PI}$) et le sommet-alternative y. Quant aux arcs E_3, ils représentent les connections directes entre toute paire d'arcs distincts de E_4. Enfin, l'ensemble E_5 contient uniquement l'arc direct (x, y).
- La fonction g de pondération des arcs est définie comme suit :

$$g((u, v)) = \begin{cases} 1 & \text{si } (u, v) \in E_4\ (\mathbb{PI}) \\ \text{L}(u, v) & \text{sinon} \end{cases} \quad (2)$$

Une schématisation graphique d'un graphe d'explication est illustrée à la Figure 1. Sur ce graphique, on peut remarquer que les arcs de chaque catégorie $(E_1, \ldots, E_5)$ ont une couleur caractéristique. La $\mathbb{PI}$ est composée de deux comparaisons par paires qui suffisent à induire $(x, y) \in \mathbb{N}_{\mathbb{PI}}$. Ces comparaisons par paire sont représentées par des arcs en trait plein pour indiquer qu'elles apparaîtraient telles quelles dans une explication transitive questionnable. Les autres arcs par contre sont représentés en pointillés car les comparaisons par paire correspondantes devront faire l'objet d'une décomposition à base de swap(s) s'ils étaient empruntés dans la chaîne explicative de la déduction (x, y).

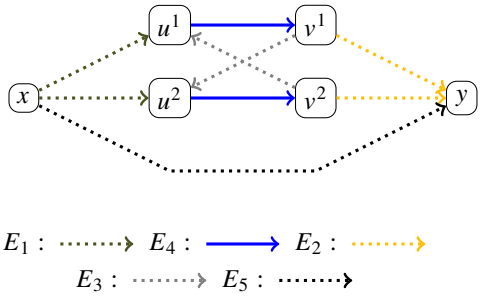


FIGURE 1 – Graphe d'explication $\mathcal{G}^{\mathbb{PI}}_{(x,y)}$

De façon formelle, le problème de l'existence d'une explication transitive questionnable peut être décrit comme suit :

Entrées :
- Un ensemble de comparaisons par paire $\mathbb{PI}$
- Une comparaison par paire $(x, y) \in \mathbb{N}_{\mathbb{PI}}$
- Un entier naturel L_{max}

Question :

Existe-t-il dans le graphe d'explication $\mathcal{G}^{\mathbb{PI}}_{(x,y)}$ un chemin $C = [z^0, \ldots, z^{t-1}, z^t, \ldots, z^l]$ de longueur l tel que :
- $l \leq L_{max}$
- reliant x à y c'est-à-dire : $z^0 = x$ et $z^l = y$
- il existe $t \in [\![1; l]\!]$ tel que $(z^{t-1}, z^t) \notin \mathbb{PI}$ et (z^{t-1}, z^t) admet une décomposition ψ à base de swap(s) dont au moins un des swaps caractéristiques a pour projection dans $\mathbb{X}$ (voir Définition 5) une comparaison par paire qui n'appartient pas à la relation nécessaire (voir Définition 3).

Nous avons fait le choix dans la sous-section 3.3 de modéliser le problème décrit ci-dessus dans sa version optimisation : celle conduisant à retenir l'explication la plus courte. En effet, l'explication est destinée à un décideur (le plus souvent humain) dont il convient de tenir compte des limites de la capacité à traiter plusieurs informations à la fois ([7]).

3.3 Modélisation

La modélisation du problème du calcul de la plus courte explication transitive questionnable va s'appuyer sur l'ensemble noté $\widetilde{S}_{\mathbb{PI}}$ de swaps dont les projections dans $\mathbb{X}$ (Définition 5) n'appartiennent pas à la relation nécessaire (Définition 3) afin de garantir le caractère questionnable des explications à produire.

3.3.1 Les variables

On distinguera trois catégories de variables :

1. Variables continues $\mathsf{w}_\mathtt{i}$

   Il s'agit des m variables $\mathsf{w}_\mathtt{i}$ continues (strictement) positives dont les valeurs correspondent aux évaluations des niveaux "fort" des critères binaires. On rappelle (Définition 1) que nous avons fait l'hypothèse d'une évaluation nulle des niveaux "faible" des m critères.

2. Variables binaires $\mathsf{arc}_{(u,\,v)}$

   Pour chaque arc $(u,\,v)$ du graphe d'explication $\mathcal{G}^{\mathbb{PI}}_{(x,\,y)}$, on définit une variable binaire $\mathsf{arc}_{(u,\,v)}$ valant 1 si et seulement si $(u,\,v)$ appartient au chemin solution du problème.

3. Variables binaires $\mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)}$

   Pour chaque arc $(u,\,v) \notin E_4$, on définit $|\mathtt{pro}(u,\,v)| \times |\mathtt{con}(u,\,v)|$ variables binaires qui permettront de déterminer (lorsqu'elle existe) la décomposition à base de swap(s) (Définition 6) de la comparaison par paire $(u,\,v)$ utilisée dans l'explication transitive de la déduction $(x,\,y)$. $\mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)}$ vaut 1 si et seulement si le swap $[\mathtt{i},\,\mathtt{j}]$ appartient aux swaps caractéristiques de ladite décomposition.

3.3.2 La fonction objectif

Étant donnée la fonction de pondération des arcs (Équation 2), la fonction objectif s'écrit (linéairement) comme suit :

$$\sum_{(u,\,v)\in E} arc_{(u,\,v)} \times g((u,\,v))$$

où E représente l'ensemble des arcs du graphe $\mathcal{G}^{\mathbb{PI}}_{(x,\,y)}$.

3.3.3 Les contraintes

On distinguera six grandes catégories de contraintes :

1. Normalisation de la fonction de score et (stricte) positivité de ses composantes

$$\sum_{\mathtt{i}\in[m]} \mathsf{w}_\mathtt{i} = 1 \tag{3}$$

$$\mathsf{w}_\mathtt{i} \geq \epsilon \quad \forall\, \mathtt{i} \in [m] \tag{4}$$

   avec ϵ une réel positif arbritrairement petit.

2. Prise en compte de la $\mathbb{PI}$

$$\sum_{\mathtt{i}\,\in\,\mathtt{pro}(u,\,v)} \mathsf{w}_\mathtt{i} \geq \sum_{\mathtt{j}\,\in\,\mathtt{con}(u,\,v)} \mathsf{w}_\mathtt{j} \quad \forall\,(u,\,v) \in \mathbb{PI} \tag{5}$$

3. Contraintes de « chemin »

$$\sum_{(x,\,u)\in E} \mathsf{arc}_{(x,\,u)} = 1 \tag{6}$$

$$\sum_{(u,\,z)\in E} \mathsf{arc}_{(u,\,z)} = \sum_{(z,\,v)\in E} \mathsf{arc}_{(z,\,v)} \quad \forall\, z \in V\backslash\{x,\,y\} \tag{7}$$

   avec E (resp. V) l'ensemble des arcs (resp. sommets) du graphe $\mathcal{G}^{\mathbb{PI}}_{(x,\,y)}$.

4. Contraintes liant les variables $\mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)}$ et $\mathsf{arc}_{(u,\,v)}$

   Pour chaque arc $(u,\,v) \notin \mathbb{PI}$,

$$\sum_{i\,\in\,\mathtt{pro}(u,\,v)} \mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)} = \mathsf{arc}_{(u,\,v)} \quad \forall\, \mathtt{j} \in \mathtt{con}(u,\,v) \tag{8}$$

$$\sum_{i\,\in\,\mathtt{con}(u,\,v)} \mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)} \leq \mathsf{arc}_{(u,\,v)} \quad \forall\, \mathtt{i} \in \mathtt{pro}(u,\,v) \tag{9}$$

5. « Au moins un swap de $\widetilde{S}_{\mathbb{PI}}$ utilisé »

$$\sum_{(u,\,v)\,\in\,E\backslash E_4} \sum_{[i,\,j]\,\in\,\widetilde{S}_{\mathbb{PI}}} \mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)} \geq 1 \tag{10}$$

   avec E l'ensemble des arcs du graphe $\mathcal{G}^{\mathbb{PI}}_{(x,\,y)}$ et E_4 ceux représentant les éléments de la $\mathbb{PI}$ (Sous-section 3.2).

6. Contraintes liant les variables $\mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)}$ à $\mathsf{w}_\mathtt{i}$ et $\mathsf{w}_\mathtt{j}$

$$\mathsf{w}_\mathtt{i} - \mathsf{w}_\mathtt{j} \geq (1+\epsilon) \times \mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)} - 1 \tag{11}$$

   avec ϵ une réel positif arbritrairement petit.

3.3.4 Récupération de la solution

Le programme mathématique décrit ci-dessus peut admettre ou non une solution. Si son exécution échoue, cela signifie qu'il n'existe pas d'explication transitive questionnable de la déduction $(x,\,y) \in \mathbb{N}_{\mathbb{PI}}$. Par contre, si elle réussit, la valeur optimale fournit la longueur de l'explication. Les valeurs des variables $\mathsf{arc}_{(u,\,v)}$ indiquent les arcs empruntés et celles des variables $\mathsf{s}^{[\mathtt{i},\,\mathtt{j}]}_{(u,\,v)}$ renseignent sur leur décomposition à base de swap(s).

4 Mobilisation au sein d'un dispositif interactif

L'objet de cette section est de décrire une ébauche de protocole interactif à travers lequel les explications transitives questionnables rempliront une fonction d'élicitation et d'en

fournir une instanciation illustrée. Nous rappelons que les préférences du décideur sont additives. À cette hypothèse, nous ajoutons le fait que l'information préférentielle qu'il fournit est « sûre » c'est-à-dire que :

(i) les comparaisons par paires composant $\mathbb{PI}$ ne seront donc pas remises en cause lors du processus interactif

(ii) le décideur ne se trompe pas dans l'appréciation des swaps qui lui sont fournis.

Sur la base de ces hypothèses, on remarquera que les seules informations préférentielles additionnelles capturées sont des comparaisons par paire résultant de la validation ✔ ou de la contestation ✖ des swaps utilisés dans les explications transitives questionnables. Aussi, remarquerons-nous que le processus « converge » : le nombre d'éléments de la relation nécessaire sur $\mathbf{A}$ ($|\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}}|$) croît tandis que le nombre de recommandations admissibles (Paragraphe 2.2.2) décroît. L'augmentation du nombre d'éléments de la relation nécessaire sur $\mathbf{A}$ offre de nouvelles occasions d'explication qui permettront de capturer de l'information préférentielle additionnelle orchestrant ainsi une sorte d'« effet boule de neige » qui pourrait aboutir à l'identification exacte de l'ensemble $\mathbf{A}^*$ à recommander (comme c'est le cas dans l'illustration de la Sous-section 4.2).

4.1 Ébauche de protocole interactif

Dans ce qui suit, nous identifions différents instants t consécutifs à la « collecte des réactions du décideur » sur la $t-$ième explication transitive questionnable qui lui est soumise. Ainsi, l'instant $t = 0$ correspond au tout début de l'interaction et $\mathbb{PI}^t$ à l'ensemble des comparaisons par paire à l'instant t. On note : $\mathbb{S}^t$ l'ensemble des comparaisons par paire projections dans $\mathbb{X}$ des swaps (validés ou contestés) de la $t-$ième explication transitive questionnable.

Squelette d'un protocole interactif

Entrées. $\mathbf{A}$, $\mathbf{A}^R$ et $\mathbb{PI}^0$.

Objectif. Éliciter les préférences du décideur à l'aide d'explications transitives questionnables.

Le protocole.

$t = 0$

Tant que $|\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^t}| \neq 0$ **et** `ConditionArret` non vérifiée

  `Choisir` $(x, y) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^t}$

  Tenter d'expliquer (x, y) (Section 3)

  Si l'explication existe :

    Collecter $\mathbb{S}^t$ (Les réactions du décideur)

    Mises à jour :

    ☛ $\mathbb{PI}^{t+1} = \mathbb{PI}^t \cup \mathbb{S}^t$

    ☛ $t \leftarrow t + 1$

Dans le squelette décrit ci-dessus, on voit bien (**Tant que** $|\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^t}| \neq 0$) qu'un facteur limitant de la collecte d'information préférentielle additionnelle est l'absence de déductions de comparaisons par paire d'alternatives de l'ensemble $\mathbf{A}$. Lorsque cette condition est vérifiée, le choix (Procédure `Choisir`) de la paire (x, y) à expliquer peut être aléatoire, conforme à une stratégie bien définie ou encore explicitement désignée par le décideur. Quant à la condition d'arrêt `ConditionArret`, elle pourrait traduire une contrainte sur la valeur de t (avoir t faible pour « pas trop fatiguer » le décideur) ou une condition sur l'évolution du nombre de recommandations admissibles (seuil ou convergence).

4.2 Illustration

Nous proposons ici, une instanciation du protocole détaillé précédemment. Les données du problème sont données par les Tables 4 et 5. On remarquera que les ensembles d'alternatives d'intérêt $\mathbf{A}$ et de référence $\mathbf{A}^R$ ne sont pas disjoints ($\mathbf{A} \cap \mathbf{A}^R = \{x^{50}, x^7\}$). Le type de recommandation considéré est un choix ($|\mathbf{A}^*| = 1$) et l'on dispose à l'étape $t = 0$ de l'information préférentielle ci-dessous :

$$\mathbb{PI}^0 = \{(r^{43}, r^{24}), (r^{24}, x^{50}), (x^{50}, x^7), (x^7, r^{17})\}$$

L'ensemble $\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^0}$ des déductions par calcul de la relation nécessaire sur l'ensemble des comparaisons par paires $\mathbb{A}$ se présente comme suit :

$$\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^0} = \{(x^{56}, x^{50}), (x^{56}, x^{37}), (x^{56}, x^7), (x^{26}, x^{50}), (x^{26}, x^7), (x^{25}, x^{50}), (x^{25}, x^7), (x^{50}, x^7)\}$$

L'ensemble des (5) recommandations admissibles étant donné $\mathbb{PI}^0$ est :

$$\{x^{56}\}, \{x^{26}\}, \{x^{25}\}, \{x^{42}\}, \{x^{11}\}$$

En effet, la présence des comparaisons par paire (x^{56}, x^{50}), (x^{56}, x^{37}), (x^{56}, x^7) (mises en évidence par la couleur violet dans $\mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^0}$) suffit à montrer que les alternatives x^{50}, x^{37}, et x^7 ne sauraient être recommandées.

Table 4 – Description de l'ensemble $\mathbf{A}$.

	a	b	c	d	e	f
x^{56}	1	1	1	0	0	0
x^{26}	0	1	1	0	1	0
x^{25}	0	1	1	0	0	1
x^{42}	1	0	1	0	1	0
x^{11}	0	0	1	0	1	1
x^{50}	1	1	0	0	1	0
x^{37}	1	0	0	1	0	1
x^7	0	0	0	1	1	1

Le déroulé complet de l'illustration s'étend sur 4 étapes (décrites de façon uniforme) : l'explication transitive questionnable de la paire déduite $(x, y) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^t}$ est fournie par la mise en évidence d'une chaîne dans un sous-graphe du *graphe d'explication* $\mathcal{G}^{\mathbb{PI}^t}_{(x,y)}$ et la donnée des swaps explicatifs. Les réactions du décideur par rapport à ces derniers

TABLE 5 – Description de l'ensemble $\mathbf{A}^R$.

	a	b	c	d	e	f
r^{43}	1	0	1	0	1	1
r^{24}	0	1	1	0	0	0
x^{50}	1	1	0	0	1	0
x^{7}	0	0	0	1	1	1
r^{17}	0	1	0	0	0	1

sont symbolisées par ✔ pour exprimer une validation et ✖ pour signifier une contestation. La déduction d'une paire supplémentaire suffisante pour réduire le nombre de recommandations admissibles est indiquée le cas échéant. La mise à jour de la $\mathbb{PI}$ ne s'effectue qu'avec les swaps de l'ensemble $\widetilde{S}_{\mathbb{PI}^t}$ (swaps dont les projections dans $\mathbb{X}$ n'appartiennent pas à la relation nécessaire (voir Sous-section 3.3)) : réaliser la mise à jour avec des swaps n'appartenant pas à $\widetilde{S}_{\mathbb{PI}^t}$ ne rajoute que de l'information redondante.

Étape $t = 1$: Explication de (x^{50}, x^{7}) de longueur 2

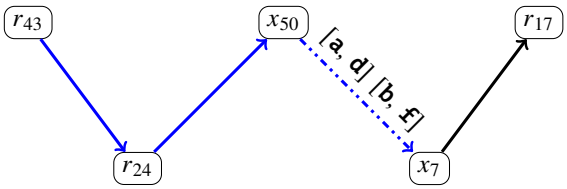


FIGURE 2 – Étape $t = 1$

Réactions du décideur : [a, d] : ✔ [b, f] : ✔
Mise à jour : $\mathbb{PI}^1 = \mathbb{PI}^0 \cup \{[\mathbf{a}, \mathbf{d}], [\mathbf{b}, \mathbf{f}]\}$
On a : $(x^{26}, x^{11}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^1}$, ce qui suffit à disqualifier la recommandation admissible $\{x^{11}\}$ à l'Étape $t = 0$.

L'ensemble des (4) recommandations admissibles est donc désormais :

$$\{x^{56}\}, \{x^{26}\}, \{x^{25}\}, \{x^{42}\}$$

Étape $t = 2$: Explication de (x^{56}, x^{7}) de longueur 3

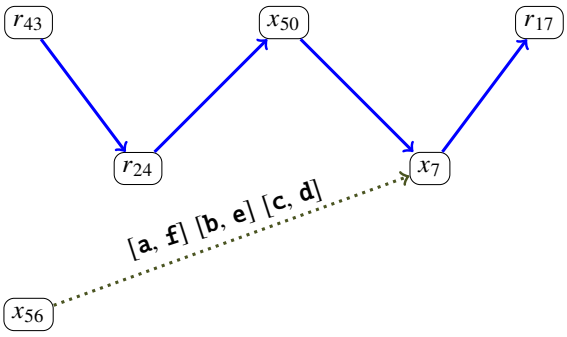


FIGURE 3 – Étape $t = 2$

Réactions du décideur : [a, f] : ✔ [b, e] : ✔ [c, d] : ✔
Mise à jour : $\mathbb{PI}^2 = \mathbb{PI}^1 \cup \{[\mathbf{a}, \mathbf{f}], [\mathbf{b}, \mathbf{e}]\}$ ($[\mathbf{c}, \mathbf{d}] \notin \widetilde{S}_{\mathbb{PI}^1}$)
On a : $(x^{56}, x^{25}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^2}$ et $(x^{56}, x^{42}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^2}$ et l'ensemble des (2) recommandations admissibles devient :

$$\{x^{56}\}, \{x^{26}\}$$

Étape $t = 3$: Explication de (x^{42}, x^{7}) de longueur 3

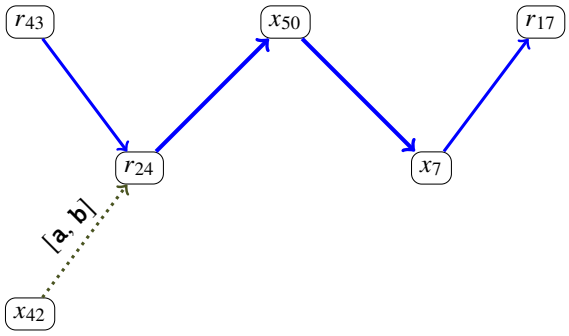


FIGURE 4 – Étape $t = 3$

Réactions du décideur : [a, b] : ✖
Mise à jour : $\mathbb{PI}^3 = \mathbb{PI}^2 \cup \{[\mathbf{b}, \mathbf{a}]\}$)
On n'a ni : $(x^{56}, x^{26}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^3}$ ni $(x^{26}, x^{56}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^3}$ et donc l'ensemble des recommandations admissibles reste :

$$\{x^{56}\}, \{x^{26}\}$$

Étape $t = 4$: Explication de (x^{56}, x^{37}) de longueur 3

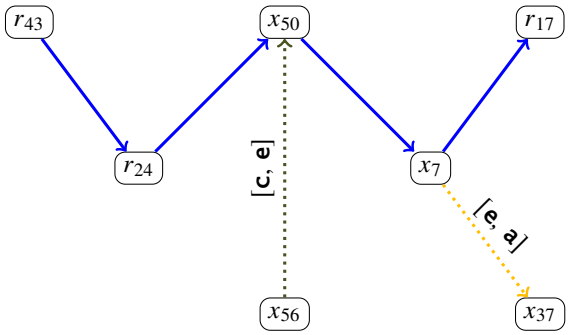


FIGURE 5 – Étape $t = 4$

Réactions du décideur : [c, e] : ✔ [e, a] : ✖
Mise à jour : $\mathbb{PI}^4 = \mathbb{PI}^3 \cup \{[\mathbf{c}, \mathbf{e}], [\mathbf{a}, \mathbf{e}]\}$)
On a : $(x^{56}, x^{26}) \in \mathbb{N}^{\mathbf{A}}_{\mathbb{PI}^4}$ et donc l'alternative à recommander est :

$$\{x^{56}\}$$

Pour convaincre le lecteur de la représentabilité par un modèle additif des préférences exprimées dans cette illustration, nous fournissons la fonction de score suivante :

$$\omega = \{\mathbf{a} : 29, \mathbf{b} : 32, \mathbf{c} : 66, \mathbf{d} : 24, \mathbf{e} : 20, \mathbf{f} : 18\}$$

5 Conclusion et Perspectives

Dans cet article, nous avons essayé de montrer que l'explication peut, en Aide MultiCritère à la Décision (AMCD), remplir une fonction d'élicitation de préférences. En faisant l'hypothèse de préférences représentables par un modèle additif sur domaine de critères binaire, nous avons proposé un programme mathématique qui calcule la plus courte explication transitive questionnable d'un élément $(x,\ y)$ de la relation nécessaire. Pour rappel, la relation nécessaire sur un ensemble $\mathbf{X}$ d'alternatives est l'ensemble des comparaisons par paire $(x,\ y)$ soutenues par l'ensemble des modèles compatibles avec l'information préférentielle $\mathbb{PI}$ fournie par le décideur.

La présente contribution complète la proposition [1] qui décrit une palette de schémas d'arguments utilisés pour justifier la comparaison par paire $(x,\ y)$ soutenue par un modèle précis ω. En effet, elle décrit un nouvel opérateur qui combine suivant des règles précises une ensemble donné de connaissances (ici des préférences) pour en construire de nouvelles (définition du schéma d'arguments) ; ce nouvel opérateur s'inspirant des schémas transitif et de couverture. Notre contribution se distingue cependant desdites propositions de [1] d'une part, en ce qu'elle s'inscrit dans un cadre où les préférences du décideur sont exprimées sous la forme de l'ensemble des comparaisons par paire $\mathbb{PI}$ qui peuvent être soutenues par plusieurs modèles précis différents ω et d'autre part en ce ce que l'explication produite, du fait de son caractère questionnable, contribuera à « aider » l'analyste à déterminer de façon précise l'alternative à recommander.

Les explications transitives questionnables ont donc vocation à être utilisées dans un protocole interactif (entre un décideur et un analyste) devant aboutir à la formulation d'une recommandation. Les grandes lignes d'un tel protocole ont été esquissées en Sous-section 4.1 et une illustration en a été donnée (Sous-section 4.2). Dans cette dernière, on a pu remarquer que la collecte d'information préférentielle à l'aide des explications transitives questionnables a permis de déterminer avec précision la recommandation à faire au décideur. Il convient donc, dans des travaux futurs, d'essayer de quantifier cet apport en réalisant des simulations numériques portant sur une plus large variété de problèmes de décision (nombre de critères supérieur à 6, calibrage de la quantité d'information préférentielle déduite . . .). Les résultats de telles expérimentations seront édifiantes et permettront une instanciation plus précise de protocoles d'interaction (stratégie de choix de la paire à expliquer ou de l'explication à proposer lorsqu'il en existe plusieurs . . .) afin que cet apport soit maximal.

Références

[1] Amoussou, M., Kh. Belahcène, N. Maudet, V. Mousseau et W. Ouerdane: *Step-wise Explanations for the Additive Model*. Working Paper (hal-03964933), 2021.

[2] Belahcene, K., C. Labreuche, N. Maudet, V. Mousseau et W. Ouerdane: *Explaining robust additive utility models by sequences of preference swaps*. Theory and Decision, 82(2) :151–183, 2017.

[3] Greco, S., V. Mousseau et R. Słowiński: *Ordinal regression revisited : Multiple criteria ranking using a set of additive value functions*. EJOR, 191(2) :416–436, 2008.

[4] Hammond, J., R. Keeney et H. Raiffa: *Even Swaps : A Rational Method for Making Trade-offs*. Harvard business review, 76 :137–8, 143, 1998.

[5] Kadzinski, Milosz, Salvatore Corrente, Salvatore Greco et Roman Słowiński: *Preferential reducts and constructs in robust multiple criteria ranking and sorting*. Operations Research-Spektrum, 36 :1021–1053, octobre 2014.

[6] Labreuche, Christophe: *A general framework for explaining the results of a multi-attribute preference model*. Artificial Intelligence, 175(7) :1410–1448, 2011, ISSN 0004-3702. https://www.sciencedirect.com/science/article/pii/S0004370210001979, Representing, Processing, and Learning Preferences : Theoretical and Practical Challenges.

[7] Miller, George A.: *The Magical Number Seven, Plus or Minus Two : Some Limits on Our Capacity for Processing Information*. The Psychological Review, 63(2) :81–97, March 1956. http://www.musanim.com/miller1956/.

[8] Miller, Tim: *Explanation in Artificial Intelligence : Insights from the Social Sciences*. ArXiv, abs/1706.07269, 2019.

Actes JIAF-JFPDA 2023

Classes of Explanations for the Verification Problem in Abstract Argumentation

Sylvie Doutre[1] **Théo Duchatelle**[2] **Marie-Christine Lagasquie-Schiex**[2]

[1] IRIT, Université Toulouse Capitole, France
[2] IRIT, Université Paul Sabatier, France
Sylvie.Doutre@irit.fr
Theo.Duchatelle@irit.fr
Marie-Christine.Lagasquie@irit.fr

Résumé

Le problème de vérification en argumentation abstraite consiste à déterminer si un ensemble est acceptable sous une sémantique donnée dans un graphe d'argumentation donné. Cet article s'attache à expliquer la réponse retournée. Des explications visuelles en termes de sous-graphes du cadre d'argumentation initial sont définies. Ces explications sont regroupées en classes, ce qui permet de sélectionner l'explication qui convient le mieux dans un contexte donné parmi l'ensemble des possibilités offertes. Des résultats montrent comment utiliser les aspects visuels de ces explications pour soutenir l'acceptabilité d'un ensemble d'arguments sous une sémantique. Les aspects computationnels d'explications spécifiques sont également étudiés.

Abstract

The Verification Problem in abstract argumentation consists in checking whether a set is acceptable under a given semantics in a given argumentation graph. Explaining why the answer is so is the challenge tackled by this paper. Visual explanations in the form of subgraphs of the initial argumentation framework are defined. These explanations are grouped into classes, allowing one to select the explanation that suits them best among the several offered possibilities. Results are provided on how to use the visual aspects of these explanations to support the acceptability of a set of arguments under a semantics. Computational aspects of specific explanations are also investigated.

1 Introduction

Abstract Argumentation is increasingly studied as a formal tool to provide explanations in the context of eXplainable Artificial Intelligence (XAI). The term argumentative XAI has emerged, with a number of application domains, ranging from machine learning, to decision, medicine or security (see [19] for an overview). [7] presents the current approaches of argumentative XAI and their open challenges, and underlines that explanations for the argumentative process itself are necessary too.

The basic argumentation process relies on an abstract structure which takes the form of a directed graph, whose nodes are arguments and edges represent attacks between arguments [10]. Characterising the acceptability of arguments can take the form of extension-based semantics : they define sets (extensions) of arguments which are collectively acceptable according to the semantics. The main questions which have been addressed so far in this context concern the global acceptability status of an argument or of a set of arguments, that is, why, under a given semantics, they belong to at least one extension (credulous acceptance) or to every extension (skeptical acceptance). The most common explanation approach consists in identifying set(s) of arguments which act as explanation(s), as in [12, 4, 5, 18, 13, 1]. However, since the argumentative process of Abstract Argumentation already provides ways for selecting arguments, explaining this process by more selection of arguments (although different ones) may not be fully helpful. Moreover, this set approach does not highlight the attacks which are involved in the explanations.

Another question regarding the argumentation process concerns the *Verification Problem* Ver, defined as follows : given an Argumentation Framework $\mathcal{A}$, a set of arguments S and an extension-based semantics σ, "Is S an extension under σ in $\mathcal{A}$?". The answer to this problem is "yes" or "no". In order to explain why the answer is so, the *eXplanation Verification Problem* XVer can be defined using the question Q_σ : "Why is S (not) an extension under σ in

$\mathcal{A}$?".

[2] is one of the only approaches which has addressed this problem and which has provided answers for some acceptability semantics of [10] in the form of relevant subgraphs, as in [17, 15, 16]. Such a visual approach is particularly of interest for human agents, graphs having been shown to be helpful for humans to comply with argumentation reasoning principles [20]. This graph-based approach not only highlights arguments, but also attacks. In [2], properties that these answers satisfy have been established, depending on whether the answer to the corresponding verification problem is "yes" or "no". This methodology follows the line of [6] in that an explanation for a set S satisfying a semantic σ is a (set of) subgraph(s) G of $\mathcal{A}$ such that G satisfies a given graph property C. Another interesting point in [2] is that the considered semantics are based on a modular definition, which allows the explanations to be decomposed.

A limitation of [2] is however that, for each semantic principle, a *single* explanation subgraph is defined. It could be more realistic to consider classes (sets) of explanations. Indeed such classes would be particularly meaningful and useful when several agents, human or artificial, are involved around the explanation of a same problem, in that they offer a variety of answers, which all follow a same schema, but which may differ on their exact content. Any agent can choose or can be presented an explanation that suits them best, and any agent can understand an explanation given by another agent, different from theirs. Classes of explanations adapt to a wide set of agents.

As in [2], the approach that will be presented in this paper goes further, by considering the possibility that the answer to the Verification Problem is not known before an explanation be asked and given. In this case, the explanation graph and its interpretation offer at a same time the answer to the problem and a justification to this answer.

Only few related works can be found concerning this notion of classes of explanation. Such classes have already been proposed in [1] for the problem of credulous acceptance of an argument, where the authors consider explanation schemes made of several elements, one of them being fixed, the other ones varying from one explanation to another. Another related work is [4] in which the authors define a parametric computation of explanations. As such, it is more the computation processes that are grouped in classes, rather than the explanations (i.e. results of the processes) themselves.

Thus, our aim in the current paper is to define classes of explanations following a generic methodology, applied to classical semantics (conflict-free, admissible, stable, complete), by building up on the approach of [2]. Additional properties (emptyness, uniqueness, maximality, minimality, computation) of explanations on these new classes will be defined and investigated.

Sec. 2 recalls background notions relative to abstract argumentation, graph theory, and presents the explanation approach defined in [2]. Classes of explanations are defined in Sec. 3, Sec. 4 studies their properties; Sec. 5 shows how to compute their maximal and minimal explanations and illustrates the whole approach on an example. Sec. 6 concludes and presents some future works. Proofs of all the results can be found in [8]

2 Background notions

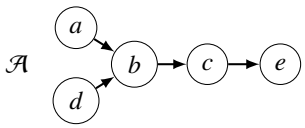
2.1 Argumentation and Graph Theory

We begin by recalling basic notions on Abstract Argumentation.

Definition 1 ([10]) *A* Dung's argumentation framework *(AF) is an ordered pair* (A, R) *such that* $R \subseteq A \times A$.

Each element $a \in A$ is called an *argument* and aRb means that *a attacks b*. For $S \subseteq A$, we say that S attacks $a \in A$ iff bRa for some $b \in S$. Any argumentation framework can be represented as a directed graph (the nodes are the arguments and the edges correspond to the attack relation).

Example 1 *Let consider* $\mathcal{A} = (A = \{a, b, c, d, e\}, R = \{(a, b), (d, b), (b, c), (c, e)\})$. $\mathcal{A}$ *is depicted by the following figure :*



The main asset of Dung's approach is the definition of semantics using some basic properties in order to define sets of acceptable arguments, as follows.

Definition 2 ([10]) *Let* $\mathcal{A} = (A, R)$. *An argument* $a \in A$ *is* acceptable *wrt* $S \subseteq A$ *iff for all* $b \in A$, *if* bRa *then* $\exists c \in S$ *st* cRb.

Definition 3 ([10]) *Given* $\mathcal{A} = (A, R)$, *a subset S of A is :*
- *a* conflict-free set *iff there are no a and b in S such that a attacks b,*
- *an* admissible set *iff S is conflict-free and for any* $a \in S$, *a is acceptable wrt S,*
- *a* complete extension *iff S is admissible and for any* $a \in A$, *if a is acceptable wrt S then* $a \in S$,
- *a* stable extension *iff S is conflict-free and S attacks any* $a \in A \setminus S$.

Example 2 *Let consider again* $\mathcal{A}$ *given in Ex. 1. Here there is a unique complete and stable extension :* $\{a, d, c\}$ *whereas there are 6 admissible sets :* $\{\}$, $\{a\}$, $\{d\}$, $\{a, c\}$, $\{d, c\}$, $\{a, d, c\}$.

The Verification Problem for the four semantics given in Def. 3 can be solved in polynomial time, as indicated by [11].

Example 3 *Considering $\mathcal{A}$ given in Ex. 1, an instance of the Verification problem could be : "Is $\{\}$ a stable extension ?" ; in this case the answer will "no". Another instance would be : "Is $\{a, d, c\}$ a complete extension ?" ; in this case the answer will "yes".*

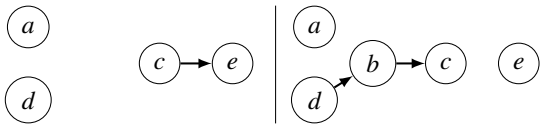
Since an AF can be represented using directed graphs, we also need to recall some basic notions of Graph Theory.

Definition 4 *Let $G = (V, E)$ and $G' = (V', E')$ be two graphs.*

- *G' is a* subgraph *of G iff $V' \subseteq V$ and $E' \subseteq E$.* [1]
- *G' is a* strict subgraph *of G iff it is a subgraph of G and either $V' \subset V$ or $E' \subset E$.* [2]
- *G' is an* induced subgraph of G by V' *if G' is a subgraph of G and for all $a, b \in V', (a, b) \in E'$ iff $(a, b) \in E$. G' is denoted as $G[V']_V$.*
- *G' is a* spanning subgraph of G by E' *if G' is a subgraph of G and $V' = V$. G' is denoted as $G[E']_E$.*

A subgraph G' of G is included in G. In an induced subgraph G' of G by a set of *vertices* S, some vertices of G can be missing but all the edges concerning the kept vertices are present. In a spanning subgraph G' of G by a set of *edges* S, all the vertices of G are present but some edges of G can be missing.

Example 4 *Let consider $\mathcal{A}$ given in Ex. 1. An example of an induced (resp. spanning) subgraph of $\mathcal{A}$ is given in the left (resp. right) following figure :*



Induced and spanning subgraphs are examples of ways to compute a graph from another single graph. Another operation producing a new graph from other ones is the union that represents the aggregation of the information contained in the two graphs :

Definition 5 (Graph union) *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The* union *of G_1 and G_2 is defined by $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.*

Let us consider also a particular kind of graphs, bipartite graphs, whose set of vertices can be split in two disjoint sets and in which every arc connects a vertex of one part to a vertex of the other part :

1. G is then a *supergraph* of G'
2. G is then a *strict supergraph* of G'

Definition 6 (Bipartite Graph) *Let $G = (V, E)$ be a graph. G is* bipartite *(with* parts *T and U) iff there exist $T, U \subseteq V$ such that $T \cup U = V$ and $T \cap U = \emptyset$ (T and U are a partition of V) and for every $(a, b) \in E$, either $a \in T$ and $b \in U$, or $a \in U$ and $b \in T$. G will be denoted with (T, U, E) and U is the* complement part of *T (and vice-versa).*

Some important functions can be defined over graphs.

Definition 7 (Successor and Predecessor functions) *Let $G = (V, E)$ be a graph. The* successor *function of G is the function $E^+ : V \mapsto 2^V$ such that $E^+(v) = \{u \mid (v, u) \in E\}$ and the* predecessor *function of G is the function $E^- : V \mapsto 2^V$ such that $E^-(v) = \{u \mid (u, v) \in E\}$. Let S be a set of vertices, $E^+(S) = \bigcup_{v \in S} E^+(v)$ and $E^-(S) = \bigcup_{v \in S} E^-(v)$.*

Let $n \geq 0$. The n-step successor *(resp.* predecessor*) function of G is $E^{+n}(v) = \overbrace{E^+ \circ \cdots \circ E^+}^{n \text{ times}}(v)$ (resp. $E^{-n}(v) = \overbrace{E^- \circ \cdots \circ E^-}^{n \text{ times}}(v)$). By convention, we have $E^{+0}(v) = E^{-0}(v) = \{v\}$.* [3]

Considering an argumentation framework, the successor (resp. predecessor) function represents the arguments that are attacked by (resp. are the attackers of) some argument(s). An AF being usually denoted by (A, R), the successor and predecessor functions are thus denoted R^+ and R^- in this context.

We then recall some notions on vertices having a particular status in a graph.

Definition 8 (Source, Sink, Isolated vertex) *Let $G = (V, E)$ be a graph and v be a vertex of G. v is said to be a* source *iff $E^-(v) = \emptyset$ and it is said to be a* sink *iff $E^+(v) = \emptyset$. v is said to be* isolated *iff it is both a source and a sink.*

Thus, *sources* (resp. *sinks*) are vertices that may only be origins (resp. endpoints) of arcs. *Isolated* vertices are those that are connected to no other vertices.

Example 5 *Let consider $\mathcal{A}$ given in Ex. 1. Argument a is a 3-step predecessor of e, whereas c is a predecessor of e (and obviously e is a 3-step successor of a, whereas e is a successor of c). Moreover, a and d are the sources of $\mathcal{A}$ and e is the sink of $\mathcal{A}$.*

2.2 Explanations in Argumentation

We recall the main definitions of what explanations are in [2] but only for those answering the questions about semantics results in abstract argumentation. These questions

3. Note that $E^{+1}(v) = E^+(v)$ and $E^{-1}(v) = E^-(v)$

are defined as follows : let σ represent a semantics among conflict-freeness, admissibility, completeness and stability, and given an argumentation framework $\mathcal{A} = (A, R)$ and some set $S \subseteq A$,

Q_σ : `Why is S (not) an extension under` σ `in` $\mathcal{A}$`?`

In order to answer these questions, and hence to provide explanations, [2] uses the decomposition of semantics into principles. The idea is to identify some properties that can be used to provide a modular characterization of semantics. We refer the reader to [9] for further details. Given a set S, the following principles are considered :

Conflict-freeness (CF) :	No internal conflicts in S
Defence (Def) :	$\forall x \in S$, x is acceptable wrt S
Reinstatement (Re) :	$\forall x$ acceptable wrt S, $x \in S$
Complement Attack (CA) :	S attacks all arguments not in S

Note that the reinstatement principle has been split into two sub-principles. Indeed, to decide whether a set S of arguments contains all the arguments acceptable wrt S, one must consider on the one hand the arguments that are unattacked and thus acceptable by lack of attackers (sub-principle denoted by Re_1), and on the other hand the arguments for which S defeats all the attackers (sub-principle denoted by Re_2).

The following has been proven in [9].

Proposition 1 *Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. S is :*

Conflict-free	*iff S respects*	*{CF}*
Admissible	*iff S respects*	*{CF, Def}*
Complete	*iff S respects*	*{CF, Def, Re_1, Re_2}*
Stable	*iff S respects*	*{CF, CA}*

With this result, a straightforward answer arises for Q_σ : a set S is an extension under semantics σ because it respects all the principles listed for σ in Prop. 1. This moves the burden of explanation from semantics to principles. So, in order to answer Q_σ, we are going to answer intermediate questions on principles. Let $\pi \in \{CF, Def, Re_1, Re_2, CA\}$ represent a principle. Given an argumentation framework $\mathcal{A} = (A, R)$ and some set $S \subseteq A$, the questions we will define answers for are :

Q_π : `Why does (not) S respect principle` π`?`

[2] defines visual answers to these questions. These answers take the form of a graph. This allows for the answers to be drawn, as well as to study their visual (i.e. structural) properties. More precisely, as argumentation frameworks are graphs themselves, the answers given are subgraphs of an argumentation framework.

Definition 9 ([2]) *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. $G_\pi(S)$ is defined as :*

$$
\begin{aligned}
G_{CF}(S) &= \mathcal{A}[S]_V \\
G_{Def}(S) &= (\mathcal{A}[S \cup R^{-1}(S)]_V) \\
&\quad [\{(a,b) \in R \mid (a \in R^{-1}(S) \text{ and } b \in S) \\
&\qquad \text{or } (a \in S \text{ and } b \in R^{-1}(S))\}]_E \\
G_{Re_1}(S) &= \mathcal{A}[\{a \in A | R^-(a) = \emptyset\}]_V \\
G_{Re_2}(S) &= (\mathcal{A}[S \cup R^2(S) \cup R^{-1}(R^2(S))]_V) \\
&\quad [\{(a,b) \in R \mid (a \in R^{-1}(R^2(S)), b \in R^2(S)) \\
&\qquad \text{or } (a \in S, b \in R^{-1}(R^2(S)))\}]_E \\
G_{CA}(S) &= \mathcal{A}[\{(a,b) \in R \mid a \in S \text{ and } b \notin S\}]_E
\end{aligned}
$$

Moreover the interpretation of these subgraphs can be done using a "checking procedure" in order to explicitly identify if the given subset satisfies or not the concerned principle :

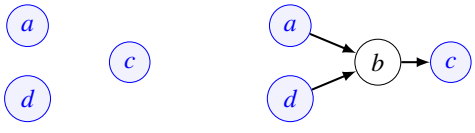
Definition 10 ([2]) *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. Let G be a subgraph of $\mathcal{A}$. The checking procedure $C_\pi(G)$ is defined as :*

$$
\begin{aligned}
C_{CF}(G) &= \textit{no attacks in } G \\
C_{Def}(G) &= \textit{no source vertices in } R^{-1}(S) \textit{ in } G \\
C_{Re_1}(G) &= \textit{all vertices in } G \textit{ are in } S \\
C_{Re_2}(G) &= \textit{all vertices in } R^2(S) \setminus S \textit{ are endpoint of an arc whose origin is a source vertex in } G \\
C'_{Re_2}(G) &= \textit{all vertices in } R^2(S) \setminus S \textit{ are endpoint of an arc whose origin is a source vertex or is in } R^2(S), \textit{ in } G \\
C_{CA}(G) &= \textit{no isolated vertices in the complement part of } S \textit{ in } G
\end{aligned}
$$

For each principle π, [2] has proven that the subgraph G_π associated with the corresponding checking procedure C_π provides an explanation that answers question Q_π. [4] More precisely, if a set S respects a principle π, then G_π verifies C_π, otherwise it does not. When the principles are combined into a semantics σ, the answer to Q_σ is the corresponding set of subgraphs along with their corresponding checking procedures.

Example 6 *Let consider $\mathcal{A}$ given in Ex. 1 and $S = \{a, d, c\}$. The question we are interested in is : "Why is S an extension under admissibility in $\mathcal{A}$?". This question comes down to wondering : "Why S satisfies conflict-freeness CF and defense Def ?". So, an explanation of why S is admissible is a set which contains the explanation for CF and the explanation for Def.*

The $G_{CF}(S)$ and $G_{Def}(S)$ explanations are given in the following figure :



4. This result is slightly more complex in the case of reinstatement. See [2] and Sec. 3.3.

There is no attack in G_{CF}, hence C_{CF} is satisfied. And so we can conclude that S is conflict-free.

Concerning G_{Def}, note that neither e nor (c, e) belong to this explanation since they have no impact on the defence of S. Then applying C_{Def} on G_{Def}, we can see that each attacker of S (here only b) is not a source vertex; so S also satisfies the defence principle.

This allows this form of explanation to be used for two purposes as indicated in the introduction : when the answer to the corresponding verification problem is known, that is, when we know that a set is (resp. is not) acceptable under a given semantics or principle, G_π on which C_π is (resp. is not) verified, offers a visual explanation of the situation, answering XVer. When the answer to the verification problem is not known, G_π and the verification of whether C_π holds or not offers at the same time an answer to Ver and an explanation of this answer.

3 Classes of explanations

In this paper, we are interested in refining the notion of explanation proposed in [2] and recalled in Sec. 2.2. Indeed, considering Ex. 6 leads to the following remark : for explaining the respect of the defence principle it seems useless to consider the two defenders of c in G_{Def} (only one is enough for proving that c is defended). So, in order to propose a more flexible notion of explanation, another approach based on the notion of *classes of explanations* is presented in this section. Of course the definition of these classes allows to recover the explanations described in [2] but also it results in the *possibility of producing several explanations for the same question*.

Hence, for each principle π, we define our explanations so that they contain at least enough information to be able to decide whether or not S respects π. We then prove that our explanations can be used in conjunction with the checking procedures recalled in Def. 10.

3.1 Explanation about Conflict-freeness

To decide whether a set S of arguments is conflict-free, one must know whether or not there are attacks among its arguments. Thus, we firstly require our explanation to contain only arguments of S, and secondly to contain only attacks between these arguments. However, with only these two constraints, it may happen that no attacks are displayed on the explanation when there are some in the original framework, leading at best to an impossibility to decide or at worst, an incorrect decision. Hence, we add a third constraint, which is that if conflicts exist between arguments of S, then at least one must be present in the explanation.

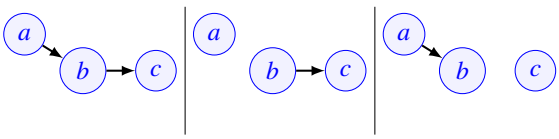
Definition 11 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{(a, b) \in R \mid a, b \in S\}$. The subgraph (A', R') of $\mathcal{A}$ is an* explanation to Q_{CF} *iff*
- $A' = S$
- $R' \subseteq X$
- *If $X \neq \emptyset$, then $R' \neq \emptyset$*

Note that the subgraph G_{CF} recalled in Def. 9 obviously belongs to the class of explanations for conflict-freeness. Moreover, in [2], a result concerning the structural property of explanations for conflict-freeness has been given : a set of arguments is conflict-free iff there is no attack in the subgraph corresponding to its explanation (checking procedure C_{CF} recalled in Def. 10). This result can be extended to all the subgraphs captured by our class of explanations.

Theorem 1 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{CF}. S is conflict-free iff $C_{CF}(A', R')$ is satisfied by S.*

This provides a way of deciding whether a set is conflict-free based on an explanation to Q_{CF}. Note that this also provides a way of deciding whether a set is *not* conflict-free, hence the possibility of handling the negative version of Q_{CF}. The same goes for all the other equivalence results concerning the other principles.

Example 7 *Let consider $\mathcal{A}$ given in Ex. 1 and $S = \{a, b, c\}$. There are 3 explanations for Q_{CF}, each of them proving that S is not conflict-free :*



3.2 Explanation about Defence

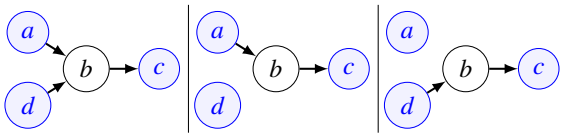
To decide whether a set S of arguments contains only arguments that are acceptable wrt S, one must know whether or not this set defeats all its attackers. Thus, we firstly require our explanation to contain only arguments of S and its attackers, and secondly to contain only attacks from S to its attackers and vice versa. To make sure the attackers are spotted as such, we further require that all the attacks of the second type are contained in the explanation. However, with only these two constraints, it may happen that no attacks targeting a specific attacker are displayed on the explanation when there are some in the original framework. As we wish the explanation to show how S defends itself, this situation is certainly undesirable. Hence, we add a third constraint, which is that if an attacker is attacked by S, then at least one attack from S to this attacker must be present in the explanation.

Definition 12 *Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. Consider $X = \{(b, a) \in R \mid b \in R^{-1}(S), a \in S\}$ and $Y = \{(a, b) \in R \mid a \in S, b \in R^{-1}(S)\}$. The subgraph (A', R') of $\mathcal{A}$ is an* explanation to Q_{Def} *iff*
- $A' = S \cup R^{-1}(S)$
- $X \subseteq R' \subseteq X \cup Y$
- $\forall b \in R^{-1}(S)$, *if* $b \in R^{+1}(S)$, *then* $\exists (a, b) \in R'$ *with* $a \in S$

Note that the subgraph G_{Def} recalled in Def. 9 obviously belongs to the class of explanations for defence. Moreover it has been shown in [2] that a conflict-free set of arguments defends all its arguments iff there is no source vertex among its attackers in $G_{Def}(S)$ (checking procedure C_{Def} recalled in Def. 10). This result can be extended to all the subgraphs captured by our class of explanations.

Theorem 2 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ be a conflict-free set of arguments and (A', R') be an explanation to Q_{Def}. $S \subseteq F_{\mathcal{A}}(S)$ iff $C_{Def}(A', R')$ is satisfied by S.*

Example 8 *Let consider $\mathcal{A}$ given in Ex. 1 and $S = \{a, c, d\}$. There are 3 explanations for proving that S satisfies the defence principle :*



Additionally, the next result extends a similar result given in [2] providing more insight on the behavior of an explanation for defence : when computed using a conflict-free set, the explanation for defence takes the form of a bipartite graph.

Proposition 2 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{Def}. If S is conflict-free, (A', R') is a bipartite graph and S can always be one of its parts.*

The two previous results can thus be used to decide whether a set of arguments effectively defends all its arguments or if it is not conflict-free.

3.3 Explanation about Reinstatement

The first part of the reinstatement principle concerns unattacked arguments. All these arguments are acceptable wrt S and should thus belong to S. Thus, we firstly require our explanation to contain only unattacked arguments, and secondly to contain no attacks (which results from the only arguments displayed being unattacked). However, with only these two constraints, it may happen that an unattacked argument not belonging to S is not displayed on the explanation. Hence, we add a third constraint, which is that if there exists unattacked arguments that are not in S, then at least one must be present in the explanation.

Definition 13 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{a \in A \mid R^{-1}(a) = \emptyset\}$. The subgraph (A', R') of $\mathcal{A}$ is an* explanation to Q_{Re_1} *iff*
- $S \cap X \subseteq A' \subseteq X$
- $R' = \emptyset$
- *If* $(A \setminus S) \cap X \neq \emptyset$, *then* $\exists a \in (A \setminus S) \cap X$ *with* $a \in A'$

The second part concerns arguments for which S defeats the attackers. These arguments must belong to S if S defeats all of their attackers. Thus, we firstly require our explanation to contain the arguments of S, the arguments that S defends (two steps of the attack relation from S), and the attackers of these arguments. Secondly, we require it contains only the attacks from S to the attackers and from the attackers to the arguments S defends. In addition, we require that all the attacks of the second type are displayed on the explanation, so that none is missed. However, with only these two constraints, it may happen that no attacks targeting a specific attacker are displayed on the explanation when there are some in the original framework. Hence, we add a third constraint, which is that if an attacker is attacked by S, then at least one attack from S to this attacker must be present in the explanation.

Definition 14 *Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. Consider $X = \{(b, c) \in R \mid b \in R^{-1}(R^{+2}(S)), c \in R^{+2}(S)\}$ and $Y = \{(a, b) \in R \mid a \in S, b \in R^{-1}(R^{+2}(S))\}$. The subgraph (A', R') of $\mathcal{A}$ is an* explanation to Q_{Re_2} *iff*
- $A' = S \cup R^{+2}(S) \cup R^{-1}(R^{+2}(S))$
- $X \subseteq R' \subseteq X \cup Y$
- *For every* $b \in R^{-1}(R^{+2}(S))$, *if* $b \in R^{+1}(S)$, *then* $\exists (a, b) \in R'$ *with* $a \in S$

Note that the subgraph G_{Re_1} (resp. G_{Re_2}) recalled in Def. 9 obviously belongs to the class of explanations for the first (resp. second) part of the principle of reinstatement. Moreover in the case of reinstatement, two results have been proven in [2] and can be extended to all the subgraphs captured by our class of explanations.

The first one shows how to conclude that a set contains all the arguments that it effectively defends from both parts of the explanation on reinstatement.

Theorem 3 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2}. If $C_{Re_1}(A', R')$ and $C_{Re_2}(A'', R'')$ are satisfied by S then $F_{\mathcal{A}}(S) \subseteq S$.*

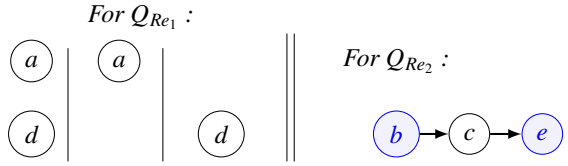
The second results shows the behavior of both parts of the explanation on reinstatement if computed on a set that contains all the arguments it effectively defends.

Theorem 4 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2}. If $F_{\mathcal{A}}(S) \subseteq S$ then $C_{Re_1}(A', R')$ and $C'_{Re_2}(A'', R'')$ are satisfied by S.*

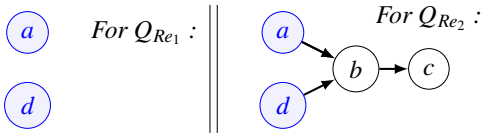
From Th. 3 and 4 follows the next corollary, which shows an equivalence result :

Corollary 1 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ such that $R^2(S)$ is conflict-free, (A', R') be an explanation to Q_{Re_1} and (A'', R'') be an explanation to Q_{Re_2}. $F_{\mathcal{A}}(S) \subseteq S$ iff $C_{Re_1}(A', R')$ and $C_{Re_2}(A'', R'')$ are satisfied by S.*

Example 9 *Let consider $\mathcal{A}$ given in Ex. 1 and $S = \{b, e\}$. There are 3 explanations for proving that S does not satisfy the first reinstatement principle (some unattacked arguments are not in S ; here it is the case for a and d) and one for proving that S satisfies the second reinstatement principle (the arguments defended by S are in S) :*



Let consider now $S = \{a, d\}$. There are one explanation for proving that S satisfies the first reinstatement principle (any unattacked argument is in S) and another one for proving that S does not satisfy the second reinstatement principle (some arguments defended by S are not in S ; here it is the case of c) :



3.4 Explanation about Complement Attack

To decide whether a set S of arguments attacks its complement, one must know whether or not all the arguments not in S are attacked by S. Thus, we firstly require our explanation to contain all the arguments of the original framework (S and its complement), and secondly to contain only attacks from S to arguments not in S. However, with only these two constraints, it may happen that no attacks targeting a specific argument outside of S are displayed on the explanation when there are some in the original framework. Hence, we add a third constraint, which is that if an argument not in S is attacked by S, then at least one attack from S to this argument must be present in the explanation.

Definition 15 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $X = \{(a, b) \in R \mid a \in S, b \notin S\}$. The subgraph (A', R') of $\mathcal{A}$ is an* explanation to Q_{CA} *iff*

- $A' = A$
- $R' \subseteq X$
- $\forall b \in A \setminus S$, *if* $b \in R^{+1}(S)$, *then* $\exists (a, b) \in R'$ *with* $a \in S$

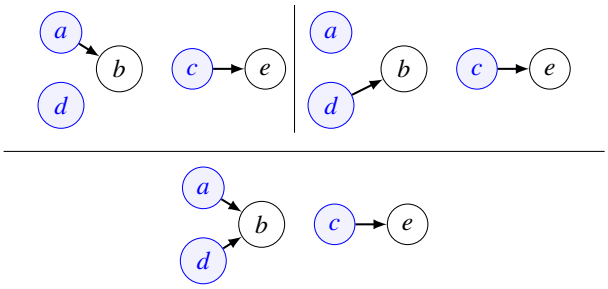
Note that the subgraph G_{CA} recalled in Def. 9 obviously belongs to the class of explanations for the principle of complement attack. Moreover concerning this principle, it was proven in [2] that a set of arguments attacks its complement iff there are no isolated vertices in $G_{CA}(S)$ and the explanation subgraph is always a bipartite graph with the arguments of S being the only possible origins for attacks. We extend these results to our class of explanations for complement attack.

Theorem 5 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and (A', R') be an explanation to Q_{CA}.*

$A \setminus S \subseteq R^{+1}(S)$ iff $C_{CA}(A', R')$ is satisfied by S.

(A', R') is a bipartite graph, S can always be one of its parts and all vertices in S are sources in it.

Example 10 *Let consider $\mathcal{A}$ given in Ex. 1 and $S = \{a, b, c\}$. There are three explanations to Q_{CA} proving that S satisfies the principle of complement attack :*



4 Properties of Explanations

We now turn to the definition of explanation properties and to a formal study of our classes of explanations according to them. This will allow to highlight some particular kinds of explanations, as well as to better understand their behavior. The properties that we will consider are : minimality, maximality, emptyness and uniqueness.

4.1 Some specific explanations

In this section, we identify some specific properties that could be respected by our explanations.

Minimality, Maximality A minimal (resp. maximal) explanation is an explanation which contains the least (resp. all the) possible amount of information. In a sense, a minimal explanation only provides what is required to explain whereas a maximal explanation in fact provides everything that might be relevant to explain, even if it might be redundant.

Definition 16 *Let $\mathcal{A} = (A, R)$ and $S \subseteq A$. The subgraph (A', R') of $\mathcal{A}$ is a* minimal *(resp.* maximal*)* explanation *that answers Q_π iff there is no subgraph (A'', R'') of $\mathcal{A}$ which is also an explanation that answers Q_π such that (A'', R'')*

is a strict subgraph of (A', R') *(resp.* (A', R') *is a strict subgraph of* (A'', R'')*).*

Example 7 (cont'd) *In this example, the maximal explanation is the first one and the two other ones are minimal.*

Emptyness The notion of an empty explanation is one that should be avoided when providing explanations, in the sense that it somewhat represents the incapacity of the system to answer the question that has been asked.

Definition 17 *Let* $\mathcal{A} = (A, R)$ *and* $S \subseteq A$*. The subgraph* (A', R') *is an* empty explanation that answers Q_π *iff* $(A', R') = (\emptyset, \emptyset)$.

Uniqueness We consider an explanation to be unique when there is only one of its kind. Although we defined classes of explanations in an attempt to represent all the different points of view that could emerge as to how to answer a question, in some situations, there can only be one way to answer that question.

Definition 18 *Let* $\mathcal{A} = (A, R)$ *be a graph. The subgraph* (A', R') *is a* unique explanation that answers Q_π *iff there is no subgraph* (A'', R'') *with* $(A'', R'') \neq (A', R')$ *which is also an explanation that answers* Q_π*.*

Example 9 (cont'd) *In this example, the explanations for the second reinstatement principle are unique (for* $S = \{b, e\}$ *or* $S = \{a, d\}$*) whereas the explanation for the first reinstatement principle is unique for* $S = \{a, d\}$ *but not for* $S = \{b, e\}$*.*

Minimality and uniqueness are seen as explanation principles in [13]. However, these two notions are defined differently in [13], relatively to another concept of explanation based on sets of arguments, not on subgraphs, as we do.

4.2 Properties of specific explanations

Here, we provide the results of our formal study on our explanations using the aforementioned properties. We begin with empty explanations. The results show that, although empty explanations can occur, they only do so in very specific situations.

The following theorem establishes a characterisation of empty explanations, which generalises a similar result given in [2]. Moreover if this empty explanation occurs, it is the only possible one.

Theorem 6 *Let* $\mathcal{A} = (A, R)$ *and* $S \subseteq A$*.* $(\emptyset, \emptyset)$ *is an explanation that answers*

1. Q_π *with* $\pi \in \{CF, Def, Re_2\}$ *iff* $S = \emptyset$*.*
2. Q_{Re_1} *iff* $\{a \in A \mid R^{-1}(a) = \emptyset\} = \emptyset$*.*
3. Q_{CA} *iff* $\mathcal{A} = (\emptyset, \emptyset)$*.*

If $(\emptyset, \emptyset)$ *is an explanation to* Q_π *with* $\pi \in \{CF, Def, Re_1, Re_2, CA\}$*, then it is unique.*

Now, we turn to our study of maximal explanations. The next theorem states for each principle that there is only one possible maximal explanation.

Theorem 7 *Let* $\mathcal{A} = (A, R)$ *and* $S \subseteq A$*. If* (A', R') *is a maximal explanation that answers* Q_π *with* $\pi \in \{CF, Def, Re_1, Re_2, CA\}$*, then it is the unique maximal explanation that answers* Q_π*.*

In the worst case, the number of explanations can be exponential in the size of some specific sets of elements, depending on the type of explanation (for instance the set of the attacks between the arguments belonging to the extension S in the case of explanations for the conflict-free principle). Thus considering only minimal explanations is a first step towards a computationally efficient method.

Nevertheless, as it turns out, there can be multiple minimal explanations in general for each principle. The next theorem studies the relation between minimal and maximal explanations and shows that the maximal explanation is exactly the union of all the minimal explanations.

Theorem 8 *Let* $\mathcal{A} = (A, R)$ *and* $S \subseteq A$*. Consider* $\pi \in \{CF, Def, Re_1, Re_2, CA\}$ *and let* (A', R') *be the maximal explanation that answers* Q_π *and* M *be the set of all minimal explanations that answers* Q_π*. Then,* $(A', R') = \bigcup_{G \in M} G$*.*

This result opens the way to algorithmic solutions since, for a given principle, a maximal explanation covers all the possible explanations (the minimal ones but also all the intermediate explanations).

5 Computation of Explanations

This section investigates how to compute the maximal and minimal explanations of a class.

Maximal Explanations It turns out that maximal explanations exactly correspond to the explanations defined in [2] (recalled in Def. 9) :

Proposition 3 *Let* $\mathcal{A} = (A, R)$*,* $S \subseteq A$ *and* $\pi \in \{CF, Def, Re_1, Re_2, CA\}$*.* $G_\pi(S)$ *is the maximal explanation that answers* Q_π*.*

This result entails that maximal explanations can be computed using only the graph operators of induced and spanning subgraphs, thus ensuring an efficient computation.

Note that Prop. 3 aggregated with Th. 7 allows to recover a unicity result given in [2].

From Maximal to Minimal Explanations In order to compute the minimal explanations for each principle π, we start from the maximal explanation :

Given $\mathcal{A} = (A, R)$ and $S \subseteq A$, $(A', R') \leftarrow G_\pi(S)$

Then, we gradually remove elements until obtaining a minimal explanation. This leads to five algorithms Alg_π (one for each principle π) that are built following the same schema. They also use the same condition for stopping the removal : "it remains at most one element to remove". The only differences between these algorithms concern the "nature" of the removed elements : [5]

For *CF*, removal of attacks between elements of S :

While $|R'| > 1$
$(x, y) \leftarrow choose(R')$; $R' \leftarrow R' \setminus \{(x, y)\}$

For *Def*, for each attacker of S that is not in S, removal of attacks that target it :

For $y \in R^{-1}(S) \setminus S$
While $|R'^{-1}(y)| > 1$
$x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

For Re_1, removal of unattacked arguments not in S :

While $|A' \setminus S| > 1$
$x \leftarrow choose(A' \setminus S)$; $A' \leftarrow A' \setminus \{x\}$

For Re_2, for each argument that is an attacker of the arguments S defends and that is not defended by S, removal of attacks that target it :

For $y \in R^{-1}(R^{+2}(S)) \setminus R^{+2}(S)$
While $|R'^{-1}(y)| > 1$
$x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

For *CA*, for each argument that is not in S, removal of attacks that target it :

For $y \in A \setminus S$
While $|R'^{-1}(y)| > 1$
$x \leftarrow choose(R'^{-1}(y))$; $R' \leftarrow R' \setminus \{(x, y)\}$

Our algorithms are sound and complete for the computation of minimal explanations as shown by the following proposition.

Proposition 4 *Let $\mathcal{A} = (A, R)$, $S \subseteq A$ and $\pi \in \{CF, Def, Re_1, Re_2, CA\}$. Algorithm Alg_π using $\mathcal{A}$ and S as inputs is sound and complete for the computation of a minimal explanation that answers Q_π.*

The computation of minimal explanations thus relies on the computation of maximal explanations, and the removal of some arcs (or arguments) in them. The computation of maximal explanations is already known to be polynomial (see [2]). Moreover the complexity of the removal operation in the worst case is linear in the number of removed elements and this number is either quadratic in the number of vertices in the graph when these elements are attacks (so for any principle except the one for the first part of reinstatement), or linear in the number of vertices in the graph when these elements are vertices (for the first part of reinstatement). From these considerations, our algorithms can be considered as computationally efficient.

Note also that a slight adaptation of these algorithms could produce random intermediate explanations (so neither minimal, nor maximal). This could be done by randomly stopping the removal process after a parametric number of steps. It is also the way to create more specific explanations responding to certain constraints given by users (for instance, explanations containing only x elements of a given type among the $y \geq x$ existing ones).

5. Note that these elements are generally attacks except in the case of the principle Re_1.

6 Conclusion and Future Work

This paper has defined *classes of explanations* for principles and semantics for the explanation Verification Problem XVer in Abstract Argumentation. These classes of explanations have been studied according to general properties such as maximality, minimality, emptyness and uniqueness. They extend and generalize the single explanations of [2], allowing more flexibility in the choice of explanations that could be presented to potential users. Moreover we have established that the explanations of [2] correspond to the maximal explanations of the defined classes, thus providing a way to compute them using graph operators. A procedure to compute minimal explanations from the maximal ones has also been provided and proven sound and complete for each class of explanations.

These results make an implementation of the proposed approach ready to be done. From this implementation, like in any XAI approach, as underlined by [7], an empirical evaluation should be conducted to assess to which extent these visual explanations actually are helpful for human agents to understand the answer to the Verification Problem. This is a first important future work, clearly related with the explainability social process described in [14].

Moreover, this evaluation could also provide a first study about what is a "best explanation" and how to select it. It is therefore also related to a second important future work : how take into account the issue of the "realizability", or personalization of an explanation. Indeed, one may have in mind parts of an explanation (some arguments, some attacks), but not a correct and complete explanation ; determining whether there exists such an explanation, and providing it, would ensure a personalized answer. In order to do so, a deeper investigation of the inner structure of the classes of explanation, and more specifically of the links they could have with lattices, may be of help.

This contribution and its research avenues will be of help in any application which uses computational abstract argumentation [19, 7].

In addition, the approach may be extended in several directions :

— to some semantics that use additional principles like maximality/minimality for set inclusion, for instance, the preferred or grounded semantics ; in this case, some new visual criteria must be identified *in order to be able to explain why* a given set is or is not a preferred or a grounded extension ; note that the visualization difficulty is not related to the complexity of the underlying problem (since the Ver problem for the grounded semantics is a polynomial problem whereas it is an exponential one for the preferred semantics) ;

— to contrastive questions : single explanations to such questions have been proposed in [2] ; their generalisation to classes of explanations may be studied using the work presented here since, very often, a contrastive question can be viewed as the conjunction of some specific single questions.

Moreover, extending XVer to additional semantics and additional questions can be considered as an attempt to produce a generic approach for the computation of explanations, on the model of the approach of [3].

Finally, more notions of Graph Theory may be investigated in order to provide other kinds of visual explanations. In particular, the notion of graph isomorphism seems of great interest, especially to provide ways of reasoning by association (explaining a result via a structurally identical argumentation framework that one already accepted).

Références

[1] Baumann, Ringo et Markus Ulbricht: *Choices and their Consequences - Explaining Acceptable Sets in Abstract Argumentation Frameworks*. Dans *Proc. of KR*, pages 110–119, Online event, 2021. IJCAI Organization.

[2] Besnard, Philippe, Sylvie Doutre, Théo Duchatelle et Marie Christine Lagasquie-Schiex: *Explaining Semantics and Extension Membership in Abstract Argumentation*. Intelligent Systems with Applications, 16 :200118, 2022.

[3] Besnard, Philippe, Sylvie Doutre, Théo Duchatelle et Marie Christine Lagasquie-Schiex: *Generic logical encoding for argumentation*. Journal of Logic and Computation, 2022, ISSN 0955-792X. https://doi.org/10.1093/logcom/exac039.

[4] Borg, AnneMarie et Floris Bex: *A Basic Framework for Explanations in Argumentation*. IEEE Intelligent Systems, 36(2) :25–35, 2021.

[5] Borg, AnneMarie et Floris Bex: *Necessary and Sufficient Explanations for Argumentation-Based Conclusions*. Dans *Proc. of ECSQARU*, tome 12897 de *LNCS*, pages 45–58, Prague, Czech Republic, 2021. Springer.

[6] Cocarascu, Oana, Kristijonas Čyras, Antonio Rago et Francesca Toni: *Explaining with argumentation frameworks mined from data*. Dans *Proc. of DEXAHAI*, Southampton, United Kingdom, 2018.

[7] Čyras, Kristijonas, Antonio Rago, Emanuele Albini, Pietro Baroni et Francesca Toni: *Argumentative XAI : A Survey*. Dans *Proc. of IJCAI*, pages 4392–4399, Online Event / Montreal, Canada, 2021. IJCAI Organization.

[8] Doutre, Sylvie, Théo Duchatelle et Marie Christine Lagasquie-Schiex: *Classes of Explanations for the Verification Problem in Abstract Argumentation*. Research Report IRIT/RR–2022–09–FR, IRIT : Institut de Recherche en Informatique de Toulouse, France, 2022.

[9] Doutre, Sylvie et Jean Guy Mailly: *Quantifying the Difference Between Argumentation Semantics*. Dans *Proc. of COMMA*, tome 287, pages 255–262, Potsdam, Germany, 2016. IOS Press.

[10] Dung, Phan Minh: *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*. Artificial Intelligence, 77(2) :321–357, 1995.

[11] Dvorák, Wolfgang et Paul E Dunne: *Computational problems in formal argumentation and their complexity*. Handbook of formal argumentation, 4 :631–688, 2018.

[12] Fan, Xiuyi et Francesca Toni: *On Computing Explanations in Argumentation*. Dans *Proc. of AAAI*, pages 1496–1502, Austin, Texas, USA, 2015. AAAI Press.

[13] Liao, Beishui et Leendert van der Torre: *Explanation Semantics for Abstract Argumentation*. Dans *Proc. of COMMA*, tome 326, pages 271–282, Perugia, Italy, 2020. IOS Press.

[14] Miller, Tim: *Explanation in artificial intelligence : Insights from the social sciences*. Artificial Intelligence, 267 :1–38, 2019.

[15] Niskanen, Andreas et Matti Järvisalo: *Smallest Explanations and Diagnoses of Rejection in Abstract Argumentation*. Dans *Proc. of KR*, pages 667–671, Rhodes, Greece, 2020. IJCAI Organization.

[16] Racharak, Teeradaj et Satoshi Tojo: *On Explanation of Propositional Logic-based Argumentation System*. Dans *Proc. of ICAART*, tome 2, pages 323–332, Online Streaming, 2021. SCITEPRESS.

[17] Saribatur, Zeynep Gozen, Johannes Peter Wallner et Stefan Woltran: *Explaining Non-Acceptability in Abstract Argumentation*. Dans *Proc. of ECAI*, tome 325, pages 881–888, Santiago de Compostela, Spain, 2020. IOS Press.

[18] Ulbricht, Markus et Johannes Peter Wallner: *Strong Explanations in Abstract Argumentation*. Dans *Proc. of AAAI*, pages 6496–6504, Online event, 2021. AAAI Press.

[19] Vassiliades, Alexandros, Nick Bassiliades et Theodore Patkos: *Argumentation and explainable artificial intelligence : a survey*. The Knowledge Engineering Review, 36, 2021.

[20] Vesic, Srdjan, Bruno Yun et Predrag Teovanovic: *Graphical Representation Enhances Human Compliance with Principles for Graded Argumentation Semantics*. Dans *Proc. of AAMAS*, pages 1319–1327, Auckland, New Zealand, 2022. IFAAMAS.

Actes JIAF-JFPDA 2023

Temporalité et causalité en argumentation abstraite

Yann Munro* **Camilo Sarmiento***
Isabelle Bloch **Gauvain Bourgne** **Marie-Jeanne Lesot**
Sorbonne Université, CNRS, LIP6, Paris, France
{prenom.nom}@lip6.fr

Résumé

Dans le cadre de l'argumentation abstraite, nous présentons les bénéfices de prendre en compte la temporalité, c'est-à-dire l'ordre d'énonciation des arguments, ainsi que la causalité. Nous proposons une réécriture des graphes d'argumentation abstraits acycliques dans un langage d'action permettant de modéliser l'évolution du monde et d'établir des relations causales entre l'énonciation des arguments et leurs conséquences directes comme indirectes. Une implémentation en *Answer Set Programming* est également proposée ainsi que des perspectives pour aller vers des explications.

Abstract

In the context of abstract argumentation, we present the benefits of considering temporality, i.e. the order in which arguments are enunciated, as well as causality. We propose a formal method to rewrite the concepts of acyclic abstract argumentation frameworks into an action language, that allows us to model the evolution of the world, and to establish causal relationships between the enunciation of arguments and their consequences, whether direct or indirect. An Answer Set Programming implementation is also proposed, as well as perspectives towards explanations.

1 Introduction

Un système d'argumentation abstrait (AAF) offre un cadre propice pour représenter et raisonner sur des informations contradictoires par l'intermédiaire d'arguments. Ce cadre permet de trouver des ensembles d'arguments pouvant être acceptés et fournit des explications sur les raisons pour lesquelles ces ensembles ont été acceptés ou non. Les AAF proposent donc des outils appropriés pour modéliser et raisonner sur des débats. Cependant, il s'agit d'un cadre statique qui n'inclut pas de notion de temporalité qui semble cruciale pour modéliser des dialogues. Pour résoudre ce problème, plusieurs types d'approche ont été proposées. Une première catégorie modifie le graphe d'argumentation en ajoutant ou supprimant des attaques et des arguments à l'aide d'opérateurs spécifiques [3], et revient à considérer un AAF à chaque pas de temps. Une autre propose de transformer un système d'argumentation vers un formalisme logique pour ensuite utiliser des opérateurs de révision ou de changement de croyances afin de mettre à jour le système d'argumentation [15]. Nous proposons d'utiliser un autre formalisme logique, les langages d'action, afin de pouvoir modéliser la dynamique d'un dialogue.

En effet, les langages d'action, comme celui proposé dans [16], ont été naturellement conçus pour inclure cette notion dans le modèle. Ce dernier vise à déterminer l'évolution du monde étant donné un ensemble d'actions choisies délibérément par des agents et dont l'occurrence peut entraîner une réaction en chaîne d'évènements dit exogènes. Nous avons choisi le langage d'action de [16] pour trois raisons principales. Tout d'abord, il permet de gérer la concurrence d'évènements. C'est également le cas des langages comme *C* [7] ou PDDL+ [6], mais leur sémantique est adaptée respectivement aux actions non déterministes ou aux actions duratives, ce qui augmente la complexité et n'est pas utile dans notre cadre. Ensuite, ce langage comporte une définition de la notion de causalité effective. Enfin, une traduction complète et correcte en ASP est proposée dans [17].

Cet article est organisé comme suit. La section 2 présente brièvement le formalisme des AAF de Dung [4]. La section 3 fournit une description du langage d'action choisi et une définition de la notion de causalité effective. Dans la section 4, nous détaillons les principales contributions de cet article : une réécriture des AAF acycliques dans le langage d'action, l'implémentation associée et quelques propriétés de cette transformation. Elles concernent principalement la correction et la complétude de notre transformation, ainsi que la pertinence de l'inclusion de la temporalité. La section 5 est une discussion autour d'un exemple sur les

* Ces auteurs ont contribué de façon égale.

apports de notre transformation pour obtenir des informations enrichies sous forme de représentation graphique et de relations causales.

2 Système d'argumentation abstrait

Cette section rappelle les principes de base des AAF [4].

Un système abstrait d'argumentation est un couple (A, R) où A est un ensemble fini d'arguments et R est une relation binaire sur $A \times A$. On appelle R la relation d'attaque et on dit qu'un argument $a \in A$ attaque $b \in A$ si $(a, b) \in R$, ce qui s'écrit $R(a, b)$. Comme R est une relation binaire à support fini, on peut naturellement représenter un système abstrait d'argumentation sous la forme d'un graphe.

Exemple 1 *Pour illustrer ces notions, on introduit ici un scénario argumentatif modélisant l'interaction entre un médecin demandeur, D, et un radiologue, R, à propos d'un examen d'un bébé de n mois pour la pathologie X.*
D : Peux-tu me faire un scanner pour ce bébé ? (a)
R : Il vaut mieux éviter les radiations ionisantes pour les jeunes bébés. (b)
R : Je peux te proposer une IRM dans deux jours. (c)
D : On peut voir X sur une IRM ? (d)
R : Oui bien sûr ! Si tu veux une confirmation, regarde le guide des bonnes pratiques en radiologie. (e)
D : Mais puisqu'il s'agit d'un bébé, il risque de bouger et donc on pourrait manquer l'information que l'on cherche car l'image ne sera pas très nette. (f)
R : Ne t'inquiète pas, j'ai l'habitude de faire ce genre d'examen pour des bébés. (g)
D : Est-ce que cela ne coûte pas beaucoup plus cher à l'hôpital de faire une IRM ? (h) Il faut aussi que je voie avec la famille du patient car ça pourrait leur revenir plus cher (i).
R : Aucun problème dans ces cas là. Ce coût élevé englobe l'expérience acquise par mon équipe, de sorte qu'à l'avenir, elle puisse réaliser ce type d'examen délicat sans moi. (j)
D : Je viens de discuter avec la famille, aucun problème avec l'IRM elle est couverte pour ça. (k)
D : Cependant, la famille n'est pas rassurée de devoir attendre deux jours, peux-tu faire l'IRM dans la journée ? (l)
R : Non je n'ai vraiment plus de place. Mon prochain créneau est dans deux jours comme je te l'ai dit. (m)

À la suite de cet échange, la décision est donc arrêtée sur une IRM programmée dans deux jours. Mais plus tard dans la journée, le médecin reçoit un appel de la famille pour prévenir que le bébé ne va vraiment pas bien et insister sur l'urgence de l'examen. Le médecin recontacte donc le radiologue pour ajouter un dernier argument :
D : C'est vraiment urgent pour le bébé, il faut une place aujourd'hui ! (n)

A partir de ce dialogue, on peut extraire manuellement des arguments et les relations entre eux afin d'obtenir un AAF représenté en figure 1 avec les arguments suivants : {$\boldsymbol{a}$ *: Scanner,* b *: Radiations ionisantes,* $\boldsymbol{c}$ *: IRM dans deux jours,* d *: X non visible par IRM,* e *: X visible par IRM,* f *: Conditions difficiles,* g *: Expérience,* h *: Coût élevé pour l'hôpital,* i *: Coût élevé pour le patient,* j *: Pas problématique pour l'hôpital,* k *: Famille couverte pour une IRM,* $\boldsymbol{l}$ *: IRM aujourd'hui,* m *: Pas de disponibilité aujourd'hui,* n *: C'est une urgence !*}. *Les arguments* a, c, l *sont appelés les variables de décision, leur acceptation étant le critère déclencheur d'une décision : scanner, IRM dans deux jours, ou IRM aujourd'hui.*

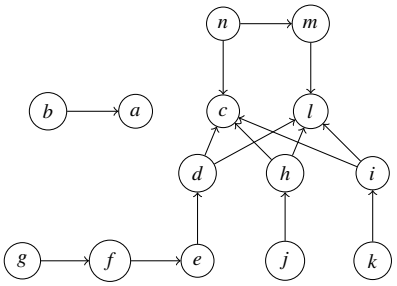


FIGURE 1 – Graphe d'argumentation associé à l'exemple 1.

*Le système d'argumentation obtenu est un graphe que l'on peut associer à ce dialogue. Ce processus d'extraction peut également être effectué automatiquement, en utilisant des méthodes dites d'*argument mining *[9].*

Remarque – Il s'agit d'une représentation statique du dialogue dont toute notion de temporalité a été effacée. Ainsi, si les arguments avaient été énoncés dans un ordre différent, cela ne changerait pas pour autant le graphe. Cela a de l'importance quand on s'intéresse aux notions de causalité, cf section 5.2.

Une fois le graphe d'argumentation construit, il est possible de raisonner sur ce graphe afin de déterminer les ensembles d'arguments qui peuvent être acceptés. Pour cela, on rappelle quelques définitions supplémentaires :

• On note Att_a **l'ensemble des attaquants directs** de a pour la relation R : $Att_a = \{b \in A \mid R(b, a)\}$.

• Un ensemble S est **sans conflit** s'il n'y pas d'arguments $(a, b) \in S^2$ qui s'attaquent l'un l'autre.

• Un argument $a \in A$ est **acceptable** par un ensemble S si S attaque tous les attaquants de a.

• Un ensemble S sans conflit est dit **admissible** si tous ses éléments sont acceptables par S.

On peut également définir des sémantiques à base d'extension. Ce sont des propriétés qui doivent être respectées par un ensemble d'arguments afin qu'il soit accepté. Dans le cas des graphes acycliques, toutes ces sémantiques coïncident et ne forment qu'une unique extension, admissible, et ne seront donc pas évoquées ici [14].

Exemple 1 *(suite) – Le modèle obtenu pour modéliser le dialogue entre le radiologue et le médecin est acy-*

clique. Pour déterminer l'ensemble des arguments acceptables, il suffit de partir des arguments non attaqués, ici $\{b, g, j, k, n\}$*. Ces derniers sont par défaut acceptés. Ensuite, un argument attaqué par au moins un argument accepté ne peut être accepté. En appliquant ce principe, on obtient que l'argument l est accepté à l'inverse de a et c. La décision finale est donc de réaliser une IRM en urgence dans la journée.*

3 Langage d'action et causalité

Cette section présente d'abord la notion de langage d'action telle que définie dans [16]. Elle introduit ensuite brièvement ce qui y est défini comme causalité effective.

3.1 Sémantique et syntaxe

Le langage d'action introduit dans [16] a été conçu dans l'optique de déterminer l'évolution du monde étant donné un ensemble d'actions choisies délibérément par des agents. L'occurrence de ces actions pouvant entraîner une réaction en chaîne d'évènements dits exogènes, il est nécessaire pour avoir une connaissance complète de l'évolution du monde de s'intéresser aussi bien à l'évolution des états du monde qu'à l'occurrence des évènements. Le formalisme utilisé s'appuie sur une décomposition du monde en deux ensembles : $\mathbb{F}$ contient les variables décrivant l'état dans lequel se trouve le monde, plus précisément il s'agit de fluents instanciés représentant des propriétés du monde pouvant varier dans le temps ; $\mathbb{E}$ contient des variables décrivant des transitions dont l'occurrence modifie les fluents.

Un *littéral de fluent* est soit un fluent $f \in \mathbb{F}$, ou sa négation $\neg f$. L'ensemble des littéraux de fluents dans $\mathbb{F}$ est noté $Lit_{\mathbb{F}}$, défini par $Lit_{\mathbb{F}} = \mathbb{F} \cup \{\neg f \mid f \in \mathbb{F}\}$. Le *complément* d'un littéral de fluent l est défini comme $\overline{l} = \neg f$ si $l = f$, ou $\overline{l} = f$ si $l = \neg f$.

Définition 1 (État S) *L'ensemble* $L \subseteq Lit_{\mathbb{F}}$ *est un* état *si :*
- *il est cohérent :* $\forall l \in L, \overline{l} \notin L$ *;*
- *il est complet :* $\forall f \in \mathbb{F}, f \in L$ *ou* $\neg f \in L$*.*

Un état est donc un ensemble $L \subseteq Lit_{\mathbb{F}}$ donnant la valeur de chaque fluent décrivant le monde. Le temps est modélisé de façon linéaire de sorte à obtenir un état $S(t)$ pour chaque pas de temps t de l'ensemble $\mathbb{T} = \{-1, 0, \ldots, N\}$, avec $S(0)$ l'*état initial*. Il s'agit d'une formalisation bornée dans le passé d'un problème réel qui lui n'est pas borné. Pour avoir la formalisation la plus fidèle possible, tous les états précédant $t = 0$ sont recueillis dans un état $S(-1) = \mathbb{F} \backslash S(0)$.

Un évènement $e \in \mathbb{E}$ est une formule atomique caractérisée par trois composantes : des *préconditions* indiquant les conditions devant être satisfaites par l'état S pour que l'évènement puisse se déclencher ; des *conditions de déclenchement* donnant toutes les conditions devant être satisfaites au temps t pour que l'évènement puisse se déclencher, conditions dont la singularité par rapport aux préconditions est détaillée ci-dessous ; des *effets* précisant les changements de l'état du monde attendus si l'évènement se produit. Il faut en effet noter qu'un évènement peut avoir moins d'effets que ceux formalisés lorsqu'il se produit dans certains contextes.

Les préconditions et les effets sont respectivement représentés par des formules des langages $\mathcal{P} ::= l | \psi_1 \wedge \psi_2 | \psi_1 \vee \psi_2$ et $\mathcal{E} ::= l | \varphi_1 \wedge \varphi_2$. Les fonctions associant à chaque évènement préconditions, conditions de déclenchement et effets sont respectivement notées pre, tri et eff, et sont définies comme : $pre : \mathbb{E} \to \mathcal{P}$, $tri : \mathbb{E} \to \mathcal{P}$, et $eff : \mathbb{E} \to \mathcal{E}$. Deux ensembles disjoints $\mathbb{A}$ et $\mathbb{U}$ forment une partition de $\mathbb{E}$ avec : $\mathbb{A}$ contient les actions réalisées par des agents et donc soumises à leur volition ; $\mathbb{U}$ contient les évènements exogènes se déclenchant aussitôt que leurs préconditions pre sont satisfaites, sans qu'un agent n'ait besoin de les réaliser. Pour les évènements exogènes, il n'y a pas de différence entre pre et tri. À l'opposé, les conditions de déclenchement des actions ne se limitent pas aux préconditions, il faut en plus la volonté de réaliser l'action de la part de l'agent, ou une sorte de manipulation d'un agent tiers qui s'y substituerait.

L'ensemble contenant tous les évènements se produisant au pas de temps t est noté $E(t)$. Le fait de gérer la concurrence d'évènements (plus d'un évènement peut avoir lieu à chaque pas de temps) est l'un des avantages principaux de ce langage d'action.

Le langage d'action décrit peut être résumé comme étant un système de transition classique, où $E(t)$ génère la transition entre les états $S(t)$ et $S(t+1)$. De ce fait, les états s'enchaînent au fur et à mesure que les évènements se produisent, simulant ainsi l'évolution du monde.

Afin d'être en mesure d'obtenir des relations causales en accord avec la conception communément admise dans la communauté de philosophes qui s'intéressent à la causalité, et cela malgré le fait d'avoir une formalisation bornée dans le passé, il est nécessaire que les évènements s'étant produits avant $t = 0$ soient représentés. Pour chaque littéral $l \in S(0)$ on introduit un évènement $ini_l \in \mathbb{E}$ tel que $eff(ini_l) = l$. On note alors $E(-1) = \{ini_l, l \in S(0)\}$ qui vérifie $eff(E(-1)) = S(0)$.

Pour résoudre des conflits potentiels ou établir des priorités entre les événements, un ordre partiel strict $\succ_{\mathbb{E}}$ est introduit, qui garantit la priorité de déclenchement d'un événement par rapport à un autre.

Définition 2 (Contexte κ) *Le* contexte *noté* κ *est l'octuple* $(\mathbb{E}, \mathbb{F}, pre, tri, eff, S(0), \succ_{\mathbb{E}}, \mathbb{T})$*, où* $\mathbb{E}$*,* $\mathbb{F}$*,* pre*,* tri*,* eff*,* $S(0)$*,* $\succ_{\mathbb{E}}$*, et* $\mathbb{T}$ *ont été définis précédemment.*

Définition 3 (Exécution valide) *Une* exécution *est une séquence* $E(-1), S(0), E(0), \ldots, E(N), S(N+1)$*. Elle est* va-

lide *étant donné un contexte κ si elle vérifie* $\forall t \in \mathbb{T}$ *:*

1. $S(t) \subseteq Lit_{\mathbb{F}}$ *est un état au sens de la définition 1.*
2. $E(t) \subseteq \mathbb{E}$ *vérifie :*
   - 2.a $\forall e \in E(t),\ S(t) \models pre(e)$;
   - 2.b $\nexists (e, e') \in E(t)^2,\ e \succ_{\mathbb{E}} e'$;
   - 2.c $\forall e \in \mathbb{E}$ *tel que* $S(t) \models tri(e)$, $e \in E(t)$ *ou* $\exists e' \in E(t),\ e' \succ_{\mathbb{E}} e$;
3. $S(t+1) = \left\{ l \in S(t), \forall e \in E(t), \bar{l} \notin \mathit{eff}(e) \right\} \cup \{ l \in Lit_{\mathbb{F}}, \exists e \in E(t), l \in \mathit{eff}(e) \}$.

Pour un contexte κ donné, il existe potentiellement plus d'une exécution valide. En effet, aucune spécification du moment où les actions sont réalisées n'est inclus dans le contexte. Leurs préconditions peuvent être satisfaites, et donc des exécutions peuvent être valides, mais leurs conditions de déclenchement ne le peuvent pas. L'ajout en entrée d'un ensemble d'actions couplées à un temps $\sigma \subseteq \mathbb{A} \times \mathbb{T}$ qui modélise la volition des agents, appelé *scénario*, permet d'obtenir une unique exécution valide. D'une telle exécution il est possible d'extraire deux types de traces :

Définition 4 (Traces $\tau^e_{\sigma,\kappa}$ et $\tau^s_{\sigma,\kappa}$) *Étant donné un scénario σ et un contexte κ, la* trace d'évènements $\tau^e_{\sigma,\kappa}$ *de σ, κ est la séquence d'évènements $E(-1), E(0), \ldots, E(N)$ contenue dans une des exécutions valides étant donné κ, telle que :* $\forall t, \forall e \in E(t),\ e \in \mathbb{A} \Leftrightarrow (e,t) \in \sigma$.
La trace d'états $\tau^s_{\sigma,\kappa}$ *est la séquence d'états $S(0), S(1), \ldots, S(N+1)$ correspondant à $\tau^e_{\sigma,\kappa}$.*

3.2 Causalité effective

La définition de causalité effective proposée dans [16] est une formalisation adaptée aux langages d'action du « NESS test ». Celui-ci stipule [18] : *«A particular condition was a cause of a specific consequence if and only if it was a necessary element of a set of antecedent actual conditions that was sufficient for the occurrence of the consequence.»*

Une relation causale est un lien entre une cause à un effet. Le fait que les langages d'action représentent le monde comme une succession d'états produits par des occurrences d'évènements introduit des états entre les évènements. De ce fait, en plus de la relation de causalité effective qui relie deux occurrences d'évènements entre elles, comme communément accepté par les philosophes, il est nécessaire de définir des relations causales où les causes sont des occurrences d'évènements, et les effets sont la véracité de formules du langage $\mathcal{P}$ à un temps donné. Le NESS test est utilisé pour définir ces relations intermédiaires. Pour pouvoir fournir la définition de causalité effective adaptée aux langages d'action, trois relations causales sont introduites dans [16] (pour les détails voir [17]). (i) Les NESS-causes directes donnent des informations essentielles en se basant sur les effets que l'occurrence d'un évènement a réellement eus, qui à nouveau ne sont pas nécessairement les mêmes que ceux attendus. Comme mentionné précédemment, il s'agit donc d'une relation entre l'occurrence d'un évènement et la véracité de formules du langage $\mathcal{P}$. Malgré leur aspect indispensable, ces relations ne sont pas toujours les plus intéressantes. En effet, l'ensemble des NESS-causes directes d'une formule peut contenir un certain nombre d'évènements exogènes qui ne sont pas nécessairement pertinents. Il est donc essentiel d'établir une chaîne causale en remontant le temps de sorte à retrouver l'ensemble d'actions qui sont derrière la véracité de la formule de $\mathcal{P}$. (ii) Les NESS-causes permettent de retrouver cette chaîne causale. En notant $\psi \in \mathcal{P}$ la formule qui nous intéresse à l'instant t_ψ et C l'ensemble des NESS-causes directes de (ψ, t_ψ), la NESS-cause s'intéresse à ce qui a causé $(tri(C), t)$, où $t < t_\psi$ nécessairement. Il faut noter que par définition les NESS-causes directes sont un type particulier de NESS-causes. Enfin, l'occurrence d'un premier évènement est considéré comme une cause effective (iii) de l'occurrence d'un second d'évènement si l'occurrence du premier est une NESS-cause des conditions de déclenchement du deuxième. De cela nous pouvons déduire que, si l'occurrence (e', t_2) est une NESS-cause directe de (ψ, t_3) et que l'occurrence (e, t_1) est une cause effective de (e', t_2) avec $t_1 < t_2 < t_3$, alors l'occurrence (e, t_1) est une NESS-cause de (ψ, t_3). Ces trois relations causales sont illustrées à l'aide de l'exemple en section 5.2.

4 Passage des AAF au langage d'action

Dans cette section, nous présentons la contribution principale de cet article, à savoir une réécriture d'un graphe d'argumentation abstrait acyclique dans le langage d'action présenté dans la section précédente. Pour cela, la section 4.1 présente la définition du contexte argumentatif κ, la section 4.2 fournit les définitions modifiées de la sémantique associée au langage d'action, la section 4.3 décrit brièvement l'implémentation ASP. Enfin, la section 4.4 présente les propriétés de la transformation proposée.

Contrairement aux AAF, nous proposons de prendre en compte l'ordre d'énonciation des arguments. Au lieu d'avoir seulement un couple (A, R), l'entrée est un couple (Δ, R), où Δ est un dialogue, c'est-à-dire une séquence d'énoncés en langage naturel :

Définition 5 (Dialogue Δ) *Un* dialogue*, Δ, est défini comme $\Delta = \{(a, o) \mid a \in A, o \in \mathbb{N}\}$, où chaque argument a est associé à son ordre d'énonciation o.*

4.1 Instanciation du contexte κ

Pour pouvoir passer d'un graphe d'argumentation au langage d'action décrit en section 3, il faut d'abord définir les fluents $\mathbb{F}$ c'est-à-dire les variables nécessaires pour décrire

le monde, ici le graphe d'argumentation. Deux éléments doivent être pris en compte : les arguments et les relations d'attaque. Pour décrire un argument x, nous introduisons deux fluents : $p_x \in \mathbb{F}$ qui indique si l'argument est présent ou non dans le graphe et $a_x \in \mathbb{F}$ qui indique l'acceptabilité de l'argument. Pour R, nous utilisons le fluent $cA_{y,x} \in \mathbb{F}$ exprimant le fait que y peut attaquer x. Comme nous ne traitons que des AAF acycliques, $\nexists (x_1, \ldots, x_n) \in A$ tel que $\left(cA_{x_1,x_2}, \ldots, cA_{x_{n-1},x_n}, cA_{x_n,x_1}\right) \in \mathbb{F}$. Nous appelons cette propriété l'acyclicité des fluents cA.

En ce qui concerne les évènements $\mathbb{E}$, dans le cas de l'argumentation abstraite la seule action volontaire possible est d'énoncer un argument x, notée $enunciate_x \in \mathbb{A}$. Pour cela, il faut que l'argument en question n'ait pas déjà été prononcé. Dans ce cas, x devient présent et acceptable par défaut. Ce choix est justifié car on évaluera son acceptabilité à l'état suivant avant qu'il n'ait un impact sur le reste du graphe. Formellement :

$$pre(enunciate_x) \equiv \neg p_x$$
$$eff(enunciate_x) \equiv p_x \wedge a_x$$

Remarque – Aucun des évènements décrits par la suite n'a pour effet de rendre un argument non présent. Cela implique qu'il n'est pas possible de ré-énoncer un argument déjà énoncé. Cette hypothèse n'est pas en contradiction avec le cadre de l'argumentation classique. En effet, un argument répété se manifesterait par un argument identique mais de nom différent dans le graphe ce qu'il est évidemment possible aussi avec notre transformation. Cependant, le langage d'action que nous utilisons offrant des outils pour tenir compte de la temporalité, une meilleure approche pourrait exister mais nécessiterait une étude approfondie. Malgré tout, cet article étant une première étape, il vise à poser des bases solides au prix de quelques hypothèses simplificatrices.

Contrairement au cadre de l'argumentation abstraite, nous prenons ici en compte l'ordre d'énonciation des arguments. Cela implique de mettre à jour l'acceptabilité de tous les autres arguments présents après l'énonciation d'un nouvel argument et avant l'énonciation du suivant. Cela définit des états que nous appelons *états argumentatifs*.

Définition 6 (État argumentatif) *Un état $S(t)$ est dit argumentatif si :*
i) $\forall x, y, \left[S(t) \models a_x \wedge p_y \wedge cA_{y,x} \Rightarrow S(t) \models \neg a_y\right]$;
ii) $\forall x, \left[S(t) \models p_x \wedge \left(\bigwedge_y \neg a_y \vee \neg cA_{y,x}\right) \Rightarrow S(t) \models a_x\right]$.

Après l'énonciation d'un argument, nous souhaitons que des mises à jour soient déclenchées automatiquement. Nous les représentons par deux évènements exogènes : $makesUnacc_{y,x} \in \mathbb{U}$ et $makesAcc_x \in \mathbb{U}$. Pour rappel, un argument n'est acceptable que s'il est non attaqué ou attaqué uniquement par des arguments non acceptables. De fait, il suffit que l'un des attaquants soit acceptable pour rendre l'argument attaqué non acceptable. Cela implique donc au moins deux cas à envisager :

Mise à jour de l'acceptabilité : Supposons que l'argument y venant d'être énoncé peut attaquer l'argument x et que x et y sont acceptables. Alors, x étant attaqué par un argument acceptable y, x devient non acceptable. Formellement, l'évènement exogène $makesUnacc_{y,x}$ peut s'écrire :

$$tri(makesUnacc_{y,x}) \equiv a_x \wedge a_y \wedge cA_{y,x}$$
$$eff(makesUnacc_{y,x}) \equiv \neg a_x$$

Cette écriture permet également de traiter les cas où un nouvel argument z rend un attaquant y de x à nouveau acceptable. Dans ce cas, x devient non acceptable.

Mise à jour de la non-acceptabilité : Supposons que l'argument x est non acceptable et qu'un argument z vient d'être prononcé. Celui-ci n'a pas de lien direct avec l'argument x mais a pu impacter l'acceptabilité de certains attaquants de x. On vérifie donc si tous les arguments pouvant attaquer x sont acceptables ou non. Si aucun d'entre eux n'est effectivement acceptable, alors x le redevient. Dans le langage d'action, cela se traduit par l'évènement exogène $makesAcc_x$ tel que :

$$tri(makesAcc_x) \equiv p_x \wedge \neg a_x \wedge \left(\bigwedge_y \neg cA_{y,x} \vee \neg a_y\right)$$
$$eff(makesAcc_x) \equiv a_x$$

Enfin, lorsqu'un argument x est énoncé, il faut vérifier qu'il n'est pas rendu non acceptable par un argument y déjà présent avant qu'il ne rende non acceptables d'autres arguments, z par exemple. Cela se traduit par la règle de priorité ci-dessous :

$$makesUnacc_{y,x} \succ_{\mathbb{E}} makesUnacc_{x,z}$$

Notons qu'ajouter un argument dans le graphe ne peut impacter les autres arguments de manière directe qu'en les rendant non acceptables. Pour cette raison, il n'est pas nécessaire d'établir une règle de priorité de la forme $makesUnacc_{y,x} \succ_{\mathbb{E}} makesAcc_z$ car cette situation est déjà couverte par la règle précédente.

Remarque – Dans la transformation proposée, on ne fait pas de distinction entre la notion d'attaque potentielle et celle d'attaque réelle car la différence entre ces dernières disparaît dans les équations. En effet, considérons un fluent $attack_{y,x} \in \mathbb{F}$ traduisant le fait que l'argument y attaque effectivement l'argument x. Définissons l'évènement exogène $isAttacking_{y,x} \in \mathbb{U}$ comme :

$$tri(isAttacking_{y,x}) \equiv p_x \wedge p_y \wedge cA_{y,x}$$
$$eff(isAttacking_{y,x}) \equiv attack_{y,x}$$

D'après cette définition, y attaque x si les deux sont présents et y peut attaquer x. Cependant, pour que cette attaque soit prise en compte, il faut toujours que l'attaquant y soit acceptable. On obtient des conditions de la forme $a_y \wedge attack_{y,x}$, c'est-à-dire $a_y \wedge p_y \wedge cA_{y,x}$. Or un argument ne peut pas être acceptable sans être présent donc $a_y \wedge p_y \equiv a_y$. Ainsi, avec ce nouveau fluent on aurait : $tri(makesUnacc_{y,x}) = a_y \wedge a_x \wedge attack_{y,x} = a_y \wedge a_x \wedge cA_{y,x}$. On retrouve la même précondition que sans l'introduction de $isAttacking$ et de $attack$. De fait, nous avons choisi de n'utiliser que $canAttack$.

4.2 Modification de la sémantique

Après avoir défini le contexte κ pour le cadre argumentatif, il faut modifier les définitions associées à la sémantique du langage d'action. Cela va permettre en particulier d'obtenir des traces représentatives de la réalité. Pour cela, les arguments sont énoncés à partir d'états argumentatifs étape par étape dans l'ordre de l'interaction. Comme nous ne considérons que des graphes acycliques, il existe toujours un tel état après l'ajout d'un nouvel argument et il est donc toujours possible de continuer l'interaction.

La définition actuelle du scénario σ n'est pas adaptée à ce cas. En effet, elle demande la connaissance préalable du nombre d'étapes nécessaires pour revenir à un état argumentatif, de sorte à prévoir l'état exact dans lequel l'argument suivant pourra être énoncé. Nous proposons pour résoudre ce problème d'introduire un ensemble d'actions ordonnées $\varsigma \subseteq \mathbb{A} \times \mathbb{N}$, appelé *séquence*. L'unicité de l'exécution valide n'est plus obtenue grâce au scénario σ, mais à la séquence ς. Il faut donc modifier les définitions 3 et 4.

Définition 7 (Configuration argumentative χ) *La* configuration argumentative*, notée χ, est le couple (ς, κ) avec ς une séquence et κ un contexte.*

La définition 8 suivante modifie la définition 3 : la condition 2.d est ajoutée et dans la condition 2.c $\forall e \in \mathbb{E}$ est remplacé par $\forall e \in \mathbb{U}$. Ces modifications expriment le fait qu'une action de la séquence ς ne peut être déclenchée que si aucun évènement exogène ne se déclenche au même pas de temps. Les conditions 1, 2.a, 2.b, et 3 restent identiques, nous ne modifions donc rien vis-à-vis du déclenchement des évènements exogènes.

Définition 8 (Exécution valide dans le cadre des AAF) *Une exécution est une séquence :* $E(-1), S(0), E(0), \ldots, E(N), S(N+1)$. *Elle est valide étant donné κ si, en plus des conditions 1, 2.a, 2.b et 3 de la définition 3, elle vérifie $\forall t \in \mathbb{T}$:*

2 $E(t) \subseteq \mathbb{E}$ vérifie :

2.c $\forall e \in \mathbb{U}$ tel que $S(t) \models tri(e)$, $e \in E(t)$ ou $\exists e' \in E(t),\ e' \succ_{\mathbb{E}} e$;

2.d Si $\exists e \in E(t) \cap \mathbb{A}$, alors $\forall e' \in \mathbb{U}$, $S(t) \not\models tri(e')$;

2.e $E(t) \neq \emptyset$.

Dans la définition 4, les traces ont été définies comme des extraits d'une exécution valide compte tenu de κ et de conditions supplémentaires liées à σ. Au lieu de définir directement des traces, la définition 9 suivante correspond à une exécution valide étant donnée $\chi = (\varsigma, \kappa)$. Les traces sont simplement des extraits de ces exécutions valides.

Définition 9 (Exécution valide étant donné χ) *Soit une configuration argumentative $\chi = (\varsigma, \kappa)$. Une exécution valide étant donné κ est* valide étant donné χ *si :*

1. $\forall t \in \mathbb{T}$, $E(t) \subset (\{a, \exists o \in \mathbb{N}, (a, o) \in \varsigma\} \cup \mathbb{U})$;
2. $\forall ((e, o), (e', o')) \in \varsigma^2$ *tel que* $o < o'$, $\exists t, t'$ *tel que* $e \in E(t)$ *et* $e' \in E(t')$ *et* $t < t'$;
3. $\forall ((e, o), (e', o')) \in \varsigma^2$ *tel que* $o = o'$, $\exists t$ *tel que* $(e, e') \in E(t)^2$.

Soit une exécution valide étant donné χ. Sa *trace d'évènements* τ^e_χ est sa séquence d'évènements $E(-1), E(0), \ldots, E(N)$, et sa *trace d'états* τ^s_χ sa séquence d'états $S(0), S(1), \ldots, S(N+1)$.

4.3 Implémentation en ASP

Nous proposons une implémentation en ASP, sur la base de celle décrite dans [17]. Les programmes ASP $\pi_{con}(\kappa)$ et $\pi_{seq}(\varsigma)$ sont obtenus par la traduction respectivement du contexte κ et de la séquence ς, $\pi_{\mathbb{A}}$ est obtenu par la traduction de la sémantique du langage d'action introduite dans la section 3.1 et modifiée dans la section 4.2, et $\pi_{\mathbb{C}}$ est obtenu par la traduction des définitions des relations causales introduites par [16]. Une traduction complète et correcte est proposée dans [17]. Le programme complet $\Pi(\chi) = \pi_{sce}(\varsigma) \cup \pi_{con}(\kappa) \cup \pi_{\mathbb{A}} \cup \pi_{\mathbb{C}}$ est disponible [1].

4.4 Quelques propriétés formelles

Cette section donne les propriétés formelles de la transformation proposée. Tout d'abord, nous établissons que la notion de temporalité est bien prise en compte par la transformation. Ensuite, nous établissons sa correction et sa complétude. Enfin, nous introduisons une proposition qui ouvre la voie à la discussion de la section 5. Toutes les preuves sont omises pour cause de place, elles sont détaillées dans [13].

Le premier résultat montre que, bien que les exécutions valides étant donné κ ne soient pas uniques, les exécutions valides étant donné χ le sont, ainsi que les traces correspondantes τ^e_χ et τ^s_χ.

Proposition 1 *Soit une configuration argumentative $\chi = (\varsigma, \kappa)$, les traces τ^e_χ et τ^s_χ sont uniques.*

1. https://gitlab.lip6.fr/sarmiento/kr_2023.git

Dorénavant, lorsqu'il sera question d'évènements et d'états, il s'agira de ceux étant donné les traces uniques τ^e_χ et τ^s_χ. Ainsi, l'ensemble des événements qui se sont effectivement produits à l'instant t est $E^\chi(t) = \tau^e_\chi(t)$. De même, l'état réel à l'instant t est $S^\chi(t) = \tau^s_\chi(t)$.

Nous établissons à présent l'aspect complet et correct de notre transformation. Pour cela, nous introduisons d'abord la notion de graphe associé.

Définition 10 *Soit $S^\chi(t)$ un état. On appelle graphe associé le graphe $AF' = (A', R')$, tel que $A' = \{x \mid S^\chi(t) \models p_x\}$ et $R' = \{(y, x) \mid S^\chi(t) \models cA_{y,x}\}$.*

Un état argumentatif est considéré comme un état où rien ne se passe tant qu'une action volontaire n'est pas effectuée. Nous montrons maintenant qu'il est toujours possible d'atteindre un tel état à partir d'un état argumentatif dans lequel un argument $x \in A$ est énoncé.

Proposition 2 *Soit $S^\chi(t)$ un état argumentatif et $x \in A$ un argument. Si $enunciate_x \in E^\chi(t)$, alors $\exists t' \in \mathbb{T}$, $t < t'$ tel que $S^\chi(t')$ est un état argumentatif.*

Enfin, la proposition suivante permet de prouver qu'un argument acceptable dans l'état argumentatif est acceptable dans le graphe associé et vice-versa.

Proposition 3 *Soit $S^\chi(t)$ un état argumentatif et $AF = (A, R)$ son graphe associé. Alors, pour tout x, $x \in A$ est acceptable par A si et seulement si $S^\chi(t) \models a_x$.*

Nous avons établi qu'il existe une équivalence entre un état argumentatif et son graphe associé. Maintenant, à partir d'un dialogue et de la relation d'attaque, les traces sont générées ainsi qu'un AAF. Nous établissons alors l'existence d'un état dont le graphe associé est égal à l'AAF initial. Un tel état est appelé *état argumentatif final* et est défini comme un état argumentatif $S^\chi(t)$ tel que $\forall x \in A$, $\exists t' \in \mathbb{T}$ tel que $t' < t$ et $enunciate_x \in E^\chi(t')$.

Théorème 1 (Correction et complétude) *Soit un dialogue Δ et R un ensemble de relations d'attaque. Étant donné une configuration argumentative χ, le graphe argumentatif associé AF' à un état argumentatif final $S^\chi(t)$, et $AF = (A, R)$ le graphe d'argumentation construit à partir de (Δ, R), on a $AF' = AF$.*

Les résultats précédents permettent de montrer la cohérence de l'état final du langage d'action avec l'argumentation. En particulier, le théorème 1 est essentiel car il établit la correction et la complétude de notre approche avec l'argumentation, et permet ainsi d'assurer qu'aucune information n'est perdue.

À l'inverse, intégrer la temporalité permet d'ajouter des informations supplémentaires grâce aux états intermédiaires, comme illustré dans la section suivante, et aux relations causales qui peuvent en être déduites. Ainsi, on a le résultat suivant :

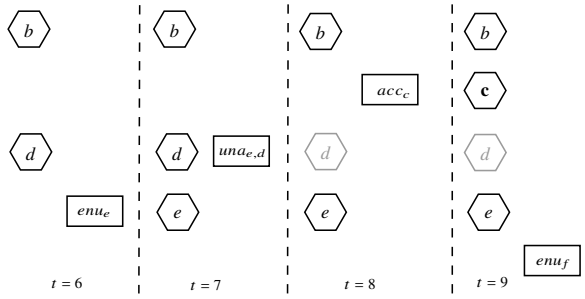


FIGURE 2 – Représentation possible d'un extrait des traces d'évènements $\tau^e_{\varsigma,\kappa}$ et d'états $\tau^s_{\varsigma,\kappa}$. Les fluents sont représentés par des hexagones et les évènements par des rectangles.

Proposition 4 *Les relations causales sont dépendantes de la séquence ς.*

Cette proposition montre que les relations causales sont dépendantes de l'ordre d'énonciation des arguments. Ainsi, même si, comme dans le cadre classique de l'argumentation, l'acceptabilité d'un argument dans l'état argumentatif final n'en dépend pas (cf théorème 1), il est tout de même essentiel de tenir compte de la temporalité lorsque l'on s'intéresse à des notions proches de la causalité notamment pour l'explicabilité. Ce résultat peut être visualisé à partir de l'exemple 2 et illustre l'enrichissement qu'apporte le passage à un langage d'action. L'utilisation de ces relations causales est discutée dans la section suivante.

5 Application à l'exemple et discussion

Dans cette section, nous appliquons le programme $\Pi(\chi)$ à l'exemple 1. Les traces d'évènements $\tau^e_{\varsigma,\kappa}$ et d'états $\tau^s_{\varsigma,\kappa}$, ainsi que les relations causales permettent de construire les figures 2, 3 présentées ci-dessous.

5.1 Représentation graphique et explication

En argumentation, Cyras et al. [19] proposent une classification des méthodes permettant de générer des explications. Parmi elles, une catégorie se concentre sur l'extraction de sous-graphes argumentatifs pour justifier l'acceptation ou le rejet d'un argument pour une certaine sémantique, produisant une représentation graphique du processus d'acceptation ou de rejet d'un argument.

Grâce à la transformation décrite dans la section précédente, nous proposons également des représentations graphiques du processus argumentatif. En effet, les traces d'évènements et d'états permettent d'obtenir une narration de l'interaction représentable graphiquement. Une première forme que peut prendre cette visualisation est présentée pour l'exemple 1 avec le schéma simplifié en figure 2.

Exemple 1 *(suite) – Comme nous nous concentrons principalement sur l'acceptabilité des arguments, nous avons*

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h,i*	*j*	*k*	*l*	*m*	*n*
a	•	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦
b		•	•	•	•	•	•	•	•	•	•	•	•
c			•	◦	•	◦	•	◦	◦	•	•	•	◦
d				•	◦	•	◦	◦	◦	◦	◦	◦	◦
e					•	◦	•	•	•	•	•	•	•
f						•	◦	◦	◦	◦	◦	◦	◦
g							•	•	•	•	•	•	•
h								•	◦	◦	◦	◦	◦
i								•	•	◦	◦	◦	◦
j									•	•	•	•	•
k										•	•	•	•
l											•	◦	•
m												•	◦
n													•

TABLE 1 – Représentation graphique de l'interaction.

décidé de représenter uniquement les fluents a_x de $\tau^s_{\varsigma,\kappa}$. Pour rester concis, nous n'utilisons que le nom des arguments pour représenter leur acceptabilité. Qui plus est, pour des raisons de lisibilité, nous ne faisons pas apparaître ce fluent lorsque c'est sa négation qui est vraie dans l'état. Nous faisons une exception lorsque c'est l'occurrence d'un évènement représenté qui a pour effet la négation du fluent. Dans ce cas, la négation est représentée par une nuance plus claire. Les évènements $enunciate_x$, $makesUnacc_{y,x}$, et $makesAcc_x$ sont représentés respectivement par les noms plus courts enu_x, $una_{y,x}$, et acc_x.

Le premier état représenté correspond à $S(6)$, état argumentatif au sens de la définition 6 permettant l'énonciation de l'argument suivant. Tous les arguments précédant e ayant déjà été énoncés, l'action $enunciate_e$ peut être faite. L'occurrence de cet évènement est la transition vers l'état suivant $S(7)$ où, comme le présente la figure 2, l'argument e est acceptable. Contrairement à $S(6)$, $S(7)$ n'est pas un état argumentatif. En effet, la condition (i) de la définition 6 n'est pas respectée car $(a_d \wedge cA_{e,d}) \in S(7)$ et $a_e \in S(7)$. L'état n'étant pas argumentatif, l'argument qui suit ne peut pas être énoncé. Toutefois, les conditions de déclenchement de $makesUnacc_{e,d}$ étant satisfaites, cet évènement exogène est déclenché ce qui entraîne une nouvelle transition d'état. L'argument d n'étant plus acceptable dans $S(8)$, la condition (i) est à nouveau satisfaite. Cela ne suffit pas pour rendre l'état argumentatif. En effet, la condition (ii) n'est pas satisfaite par $S(8)$ empêchant l'argument suivant d'être énoncé. $makesAcc_c$ est déclenché conduisant à l'état suivant $S(9)$. Ici, comme le représente la figure 2, l'argument c est acceptable. Ce nouvel état étant argumentatif, l'argument suivant, f, peut être énoncé. Le dialogue se poursuit ainsi pas à pas et se termine à l'état $S(31)$.

Pour des raisons de place, cette figure ne représente qu'une partie de l'interaction.

Une deuxième forme possible, plus compacte, est proposée dans le tableau 1 : la première colonne représente les arguments du graphe et la première ligne l'ordre des actions

ς_1	c	d	e	f	g	h,i	j	k	l	m	n
c	•	◦	•	◦	•	◦	◦	•	•	•	◦
ς_2	c	l	m	n	d	e	f	g	h,i	j	k
c	•	•	•	◦	◦	◦	◦	◦	◦	◦	◦

TABLE 2 – Impact de l'ordre d'énonciation des arguments (lignes 1, 3) sur l'acceptabilité des arguments (lignes 2, 4).

réalisées. Par souci de lisibilité, l'action $enunciate_x$ est résumée en x. Enfin, • signifie que l'argument est acceptable tandis que ◦ signifie qu'il ne l'est pas. Si un argument n'a pas encore été énoncé, la notion d'acceptabilité n'a pas de sens, ce qui a été représenté par les cases grisées. Contrairement à la représentation précédente où l'on faisait apparaître les étapes de mises à jour, cette deuxième forme a l'avantage d'être plus compacte et donc de permettre de mieux visualiser l'échange dans sa globalité. Ainsi, en regardant la ligne *c*, il est possible de suivre l'évolution de l'acceptabilité de cet argument en fonction des arguments qui ont été énoncés. Par exemple, l'énonciation de *e* (colonne *e*) fait passer *c* de ◦ (cf colonne précédente) à •, c'est-à-dire de non acceptable à acceptable. Cette forme de représentation permet également de voir rapidement l'impact direct et indirect de l'ordre d'énonciation des arguments sur l'évolution du scénario complet. Cela est illustré avec l'exemple 2.

Exemple 2 *Modifions un peu le scénario de l'exemple 1. Le dialogue débute de la même façon avec l'énonciation des arguments a, b, c. À ce moment là, le médecin demande ensuite directement s'il n'est pas possible de faire l'IRM aujourd'hui même (l). Le radiologue répond qu'il ne peut que dans deux jours au plus tôt (m). Le médecin précise alors qu'il s'agit d'une urgence (n). Ensuite, le reste du dialogue se déroule dans l'ordre présenté dans la troisième ligne du tableau 2.*

Dans celui-ci, nous avons représenté uniquement l'évolution de l'acceptabilité de c, à partir de son énonciation, variable de décision avec a et l. Même si l'état final du graphe d'argumentation n'est pas modifié, on observe l'impact très important que peut avoir l'ordre des actions dans les étapes intermédiaires qui mènent à ce dernier. Ainsi, dans ce nouveau scénario, la variable de décision c est refusée dès la 6^e action, c'est-à-dire l'énonciation de n, et cela sans modification jusqu'à la fin. Ces nuances ne sont pas représentées par les approches statiques.

Cependant, ni les méthodes par extraction de sous-graphe, ni les représentations que nous venons de présenter ne s'intéressent ou ne mettent en valeur la chaîne causale qui a conduit à l'action ou à la décision, propriété importante pour une explication d'après Miller [11].

5.2 Causalité et explication

Dans [19], une autre catégorie de méthodes se rapproche de cette notion d'explication causale en recherchant quels arguments doivent être retirés d'un graphe d'argumentation pour rendre acceptable un argument qui ne l'était pas [5]. En matière de causalité, cela correspond à une recherche de *but-for* cause de la non acceptabilité d'un argument. Cependant, ce test ne permet pas de résoudre les cas où l'occurrence de l'un de deux évènements aurait été suffisante pour causer un effet en l'absence de l'autre, appelés surdétermination [10]. Pour cela, il faut utiliser d'autres méthodes comme celle présentée dans la section 3.2, ou encore les équations structurelles de Halpern et Pearl [8]. Bien que ces deux méthodes permettent de traiter les cas de surdétermination, elles ne le font pas de la même façon et ne s'accordent pas sur un même résultat. D'un point de vue philosophique, la définition de causalité sous-jacente au NESS test appartient à la famille des approches par régularité [1], alors que les définitions de Halpern et Pearl appartiennent à la famille des approches contrefactuelles [10]. Résoudre le débat sur quelle approche est la plus adéquate est en dehors du cadre de cet article. Discutons maintenant comment nous nous démarquons de la transformation des graphes d'argumentation abstraits acycliques proposée dans [12], permettant d'exploiter la dernière définition de causalité proposée par Halpern afin de générer des explications causales en argumentation.

Le premier point de différenciation est la façon dont nous représentons le monde. En effet, comme montré dans la section 5.1, l'utilisation d'un langage d'action nous permet de prendre en compte la dynamique du dialogue. Cela n'est pas sans importance dans la compréhension de celui-ci, étant donné que la temporalité est fondamentale dans la façon dont on se représente le monde. Le deuxième point de différenciation est lié à la causalité. D'un point de vue purement mathématique, la définition de causalité de Halpern peut être qualifiée de « *Contrastive actual weak sufficiency* » d'après Beckers [2], alors que celle utilisée ici est « *Minimal actual strong sufficiency* » d'après cette même typologie. En quelques mots, alors que la première accorde beaucoup d'importance au fait qu'une cause doit être nécessaire à l'effet, d'où l'aspect contrastif, la deuxième place la suffisance au premier plan et asservit la nécessité à cette dernière. Plus de détails sur les implications de ces différences sont discutés dans [2, 18]. D'un point de vue pratique, l'avantage de l'approche causale utilisée ici est qu'elle ne nécessite pas de raisonnement contrefactuel ni d'interventionnisme, des mécanismes coûteux d'un point de vue computationel et critiqués du fait qu'ils introduisent de la subjectivité dans l'enquête causale [16, 18]. Le fait d'être dans un cadre où l'analyse causale se fait a posteriori, et donc en pleine connaissance du déroulement des évènements, enlève un avantage aux méthodes contrefactuelles qui sont très utiles lorsque l'on raisonne a priori et que l'on

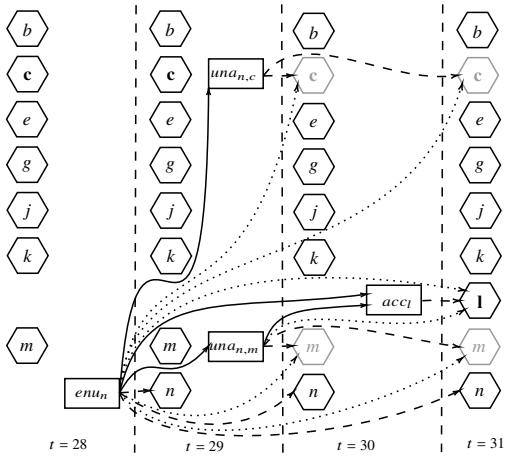


FIGURE 3 – Représentation possible d'un extrait de la trace d'évènements $\tau^e_{\varsigma,\kappa}$, de la trace d'états $\tau^s_{\varsigma,\kappa}$ et des relations causales. NESS-causes directes (– –), NESS-causes (· · · · ·), et causes effectives (——).

souhaite explorer les autres déroulements possibles.

En ajoutant les relations causales trouvées par le programme à la représentation des traces d'évènements et d'états, il est possible de générer une représentation visuelle plus riche de l'interaction qui s'est produite. La figure 3 en est une illustration pour les quatre derniers états de trace de l'exemple 1. Il s'agit de la partie du dialogue correspondant à l'énonciation de l'argument *n* et aux mécanismes de mise à jour qui en découlent.

Exemple 1 *(suite) – La représentation des traces d'évènements et d'états étant la même que pour la figure 2, ici nous commentons uniquement les relations causales que l'on trouve dans la figure 3. Rappelons que les arguments a, c et l sont les variables de décision. L'argument a devenant non acceptable très tôt dans le dialogue et le restant tout le long, nous avons choisi de ne pas le représenter dans les figures.*

L'argument n autour duquel toute la figure est articulée est celui qui vient clore le débat. Cet argument, énoncé par le médecin demandeur, porte sur le caractère urgent de l'examen. Sur la figure 3 nous pouvons voir que l'énonciation de cet argument dans l'état $S(28)$ *est une NESS-cause directe de l'acceptabilité de l'argument dans les états suivants, relation que nous notons* $(enunciate_n, 28)$ *NESS-cause directe* $(a_n, 29{-}31)$*. De même, nous avons* $(makesUnacc_{n,c}, 29)$ *NESS-cause directe* $(\neg a_c, 30-31)$*,* $(makesUnacc_{n,m}, 29)$ *NESS-cause directe* $(\neg a_m, 30-31)$*, et* $(makesAcc_l, 30)$ *NESS-cause directe* $(a_l, 31)$*. Comme le montrent ces exemples, cette première relation est la brique de base de la causalité, elle s'intéresse aux relations causales étant donné les effets réels de l'occurrence d'un évènement. Toutefois, cette relation n'est pas suffisante. Si nous souhaitons savoir pour-*

quoi l'argument l est acceptable à la fin du dialogue, et donc pourquoi la décision prise est de faire une IRM le jour même, dire simplement que c'est à cause de l'occurrence $(makesAcc_l, 30)$ *n'est pas satisfaisant.*

Cherchons alors à en savoir un peu plus sur les raisons pour lesquelles l'occurrence $(makesAcc_l, 30)$ *a eu lieu. Pour cela, il faut s'intéresser aux NESS causes et causes effectives afin de construire la chaîne causale qui a mené à* $(makesAcc_l, 30)$*. Puis par transitivité, nous obtenons que* $(makesUnacc_{n,m}, 29)$ *est une cause du fait que* $makesAcc_l$ *se soit déclenché, et donc des effets que ce déclenchement a pu avoir. En remontant encore plus loin en recherchant les causes pour lesquelles l'occurrence* $(makesUnacc_{n,m}, 29)$ *a eu lieu, nous trouvons* $(enunciate_n, 28)$ *cause effective* $(makesUnacc_{n,m}, 29)$ *et donc* $(enunciate_n, 28)$ *NESS-cause* $(\neg a_m, 30 - 31)$*. Par transitivité, nous pouvons déduire* $(enunciate_n, 28)$ *NESS-cause* $(a_l, 31)$*. Cette nouvelle relation nous permet de dire que le médecin demandeur précisant qu'il s'agit d'une urgence est une des causes de la décision finale, réponse qui paraît déjà plus satisfaisante et pouvant faire partie d'une explication. Ce même raisonnement peut-être appliqué pour trouver les causes de* $(\neg a_c, 31)$*, l'autre variable décisionnelle.*

Discussion – L'exemple précédent montre que les chaînes causales sont composées d'un nombre important de relations même si le nombre d'états étudiés est restreint. Dans le contexte de l'intelligence artificielle explicable (XAI), Miller explique dans [11] que dans le cadre d'une explication, la chaîne causale est très importante. Pour autant, il ajoute qu'une explication doit également être courte. De fait, la question des relations à mettre en avant reste à résoudre si nous voulons utiliser cette méthode pour générer des explications.

Miller précise également [11] qu'une explication est contrastive. De fait, lors du processus de recherche d'explications, il est important de pouvoir raisonner sur des scénarios contrefactuels afin de fournir des explications effectivement contrastives. Or, comme nous l'avons décrit précédemment, l'approche causale adoptée n'utilise pas ce genre de raisonnement.

Pour ces raisons, dans notre approche nous ne proposons pour le moment qu'une représentation visuelle du mécanisme conduisant à la décision. Cette dernière a l'avantage de permettre de représenter toute la chaîne causale grâce à des schémas comme celui présenté en figure 3. Ils peuvent évidemment être un support à une explication mais n'en constituent pas une indépendamment.

6 Conclusion

Nous avons proposé dans cet article une formalisation des systèmes abstraits d'argumentation acycliques dans le langage d'action présenté dans [16]. Cette transformation permet tout d'abord d'augmenter l'expressivité de ces modèles grâce à l'intégration de la temporalité, ce qui permet d'examiner l'effet de l'ordre d'énonciation des arguments. De plus, elle nous permet aussi d'exploiter la notion de causalité associée au langage d'action, offrant la possibilité de donner des informations supplémentaires sur l'acceptation ou le rejet d'un argument ainsi que des justifications sur ce dernier. Pour cela, nous avons proposé deux types de représentations graphiques du processus d'argumentation formant un support visuel et ouvrant la voie à de nouvelles formes d'explications en argumentation.

Les perspectives de ce travail visent à développer de telles explications, en appliquant les principes développés dans le contexte de l'intelligence artificielle explicable, par exemple détaillés dans [11] : les chaînes causales sont établies comme essentielles pour les explications, mais elles doivent également être courtes. La question des relations à privilégier reste donc ouverte, ainsi que la manière dont elles peuvent être utilisées pour définir des explications contrastives, nécessitant de pouvoir raisonner sur des scénarios contrefactuels.

Remerciements : Les auteurs remercient la Professeure Catherine Adamsbaum, radio-pédiatre, pour les discussions sur les exemples. Ce travail a été en partie financé par la chaire d'I. Bloch en intelligence artificielle (Sorbonne Université et SCAI).

Références

[1] Andreas, H. et M. Guenther: *Regularity and Inferential Theories of Causation*. Dans *The Stanford Encyclopedia of Philosophy*. Stanford University, 2021.

[2] Beckers, S.: *Causal Sufficiency and Actual Causation*. J. Philos. Log., 50(6) :1341–1374, 2021.

[3] Doutre, Sylvie, Faustine Maffre et Peter McBurney: *A dynamic logic framework for abstract argumentation : adding and removing arguments*. Dans *Advances in Artificial Intelligence : From Theory to Practice : 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017*. Springer, 2017.

[4] Dung, P. M.: *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*. Artificial Intelligence, 77 :321–357, 1995.

[5] Fan, X. et F. Toni: *On explanations for non-acceptable arguments*. Dans *Theory and Applications of Formal Argumentation : 3rd Int. Workshop*, pages 112–127. Springer, 2015.

[6] Fox, M. et D. Long: *Modelling Mixed Discrete-Continuous Domains for Planning*. Journal of Artificial Intelligence Research, 27 :235–297, 2006.

[7] Giunchiglia, E. et V. Lifschitz: *An Action Language Based on Causal Explanation : Preliminary Report*. Dans *AAAI*, pages 623–630, 1998.

[8] Halpern, J. Y. et J. Pearl: *Causes and explanations : A structural-model approach. Part I : Causes*. The British J. Philosophy of Science, 2005.

[9] Lippi, M. et P. Torroni: *Argumentation mining : State of the art and emerging trends*. ACM Trans. on Internet Technology, 16(2) :1–25, 2016.

[10] Menzies, P. et H. Beebee: *Counterfactual Theories of Causation*. Dans *The Stanford Encyclopedia of Philosophy*. Stanford University, 2020.

[11] Miller, T.: *Explanation in Artificial Intelligence : Insights from the Social Sciences*. Artificial Intelligence, 267 :1–38, 2019.

[12] Munro, Y., I. Bloch, M. Chetouani, M.-J. Lesot et C. Pelachaud: *Argumentation and Causal Models in Human-Machine Interaction : A Round Trip*. Dans *8th Int. Workshop on AI and Cognition*, 2022.

[13] Munro, Y., C. Sarmiento, I. Bloch, G. Bourgne et M.-J. Lesot: *Temporality and Causality in Abstract Argumentation*. CoRR, abs/2303.09197, 2023.

[14] Rahwan, I. et G. R. Simari: *Argumentation in artificial intelligence*, tome 47. Springer, 2009.

[15] Saint-Cyr, Florence Dupin de, Pierre Bisquert, Claudette Cayrol et Marie Christine Lagasquie-Schiex: *Argumentation update in YALLA (yet another logic language for argumentation)*. International Journal of Approximate Reasoning, 75 :57–92, 2016.

[16] Sarmiento, C., G. Bourgne, K. Inoue et J. G. Ganascia: *Action Languages Based Actual Causality in Decision Making Contexts*. Dans *PRIMA*, pages 243–259, 2022.

[17] Sarmiento, Camilo, Gauvain Bourgne, Katsumi Inoue, Daniele Cavalli et Jean Gabriel Ganascia: *Action Languages Based Actual Causality for Computational Ethics : a Sound and Complete Implementation in ASP*. CoRR, abs/2205.02919, 2023.

[18] Wright, R. W.: *Causation in Tort Law*. California Law Review, 73(6) :1735–1828, 1985.

[19] Čyras, K., A. Rago, E. Albini, P. Baroni et F. Toni: *Argumentative XAI : A Survey*. Dans *IJCAI-21*, pages 4392–4399, 2021.

Actes JIAF-JFPDA 2023

Les implicants premiers, un outil polyvalent pour l'explication de classification robuste

Hénoïk Willot[1] **Sébastien Destercke**[1] **Khaled Belahcene**[2]

[1] Heudiasyc, University of Technology of Compiegne, France
[2] MICS, CentraleSupélec, Université Paris-Saclay, France
henoik.willot@hds.utc.fr
sebastien.destercke@hds.utc.fr
khaled.belahcene@centralesupelec.fr

Résumé

Dans cet article, nous étudions comment les résultats d'un classifieur robuste peuvent être expliqués à l'aide d'implicants premiers, en nous concentrant sur l'explication de dominances par paires. Par robustes, nous sous-entendons des modèles prudents pouvant s'abstenir de classer ou de comparer deux classes lorsqu'ils manquent d'informations. Cela peut se faire en utilisant des ensembles (convexes) de probabilités. Par implicant premier, nous parlons d'un ensemble minimal d'attributs que nous devons figer afin d'obtenir une certaine conclusion (soit une dominance, soit une non-dominance entre deux classes). Après avoir présenté les concepts généraux, nous les appliquerons au cas bien connu du classifieur Bayesien naïf.

Abstract

In this paper, we investigate how robust classification results can be explained by the notion of prime implicants, focusing on explaining pairwise dominance relations. By robust, we mean that we consider imprecise models that may abstain to classify or to compare two classes when information is insufficient. This will be reflected by considering (convex) sets of probabilities. By prime implicants, we understand a minimal number of attributes that we need to know or specify before reaching a specified conclusion (either of dominance or non-dominance between two classes). After presenting the general concepts, we derive them in the case of the well-known naive credal classifier.

1 Introduction

Two key aspects of trustworthy AI are the ability to provide robust and safe inferences or predictions, and to be able to provide explanations as of why those have been made.

Regarding explainability, the notion of prime implicants corresponds to providing minimal sufficient condition to make a given statement, e.g., the attributes that need to be instantiated to make a classification. They have been successfully proposed as components of explanations for large classes of models such as graphical ones [17], with very efficient procedure existing for specific structures such as the Naive one [15]. In contrast with other methods such as SHAP [5] that tries to compute the average influence of attributes, prime implicants have the advantage to be well-grounded in logic, and to provide certifiable explanation (in the sense that the identified attributes are logical, sufficient reasons).

However, explainable AI tools have been mostly applied to precise models, at least in the machine learning domain (this is less true, e.g., in preference modelling [3]). Yet, in applications involving sensitive issues or in which the decision maker wants to identify ambiguous cases, it may be preferable to use models that will return sets of classes when information is insufficient rather than always returning a point-valued prediction. Several frameworks such as conformal prediction [2], indeterminate classifiers [10] or imprecise probabilistic models [8] have been proposed to handle this issue. While some explanation methods for such models have been recently proposed [18, 21], none of them explicitly adopts a logical standpoint regarding explanations, meaning that the present work is complementary to those.

Imprecise probabilistic models in particular have the interest that they are direct extensions and generalisations of probabilistic classifiers, hence one can directly try to transport well-grounded explanation principles existing for precise probabilistic classifier to this setting. This is what we

intend to do in this paper for prime implicant explanations.

We will start by introducing how the idea of prime implicants can be adapted to classifiers considering sets of probabilities as their uncertainty models. Section 2 will be a short reminder of the robust classification setting, and will introduce our notations. In Section 3, we will present the idea of prime implicant, as well as how it can answer various explanatory needs. As the formulated problems are likely to be computationally challenging for generic models, we focus in Section 4 on the naive credal classifier, that generalise the naive Bayes classifier. We show that for such a model, computing and enumerating prime implicants can be done in polynomial time, thanks to its independence assumption and decompositional properties. We also provide an experiment in Section 5 illustrating our approach.

2 Preliminaries on robust classification

In this section, we lay down our basic notations and provide necessary reminders about imprecise probabilities.

We consider a usual discrete multi-class problem, where we must predict a variable Y taking values in $\mathcal{Y} = \{y_1, \ldots, y_m\}$ using n input variables $X_1, \ldots, X_n$ that respectively takes values in $\mathcal{X}_i = \{\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{k_i}\}$. We note $\mathcal{X} = \times_{i=1}^n \mathcal{X}_i$ and $\mathbf{x} \in \mathcal{X}$ a vector in this space. When considering a subset $E \subseteq \{1, \ldots, n\}$ of dimensions, we will denote by $\mathcal{X}_E = \times^{i \in E} \mathcal{X}_i$ the corresponding domain, and by $\mathbf{x}_E$ the values of a vector on this sub-domain. We will also denote by $-E := \{1, \ldots, n\} \setminus E$ all dimensions not in E, with $\mathcal{X}_{-E}, \mathbf{x}_{-E}$ following the same conventions as $\mathcal{X}_E, \mathbf{x}_E$. We will also denote by $(\mathbf{x}_E, \mathbf{y}_{-E})$ the concatenation of two vectors whose values are given for different elements. Notation $(\mathbf{x}_E, \cdot)$ means that all features in $-E$ can take any value. If $E = \{1, \ldots, N\}$, then we will simply ignore the subscript.

In the rest of the paper, we will often refer to partially ordered sets, their corresponding relations and sets of sufficient elements that allows to asset them. Those sufficient elements will here be composed of a vector of specified feature values and of a set of probabilities. We will denote by $y \geq_{p,(\mathbf{x})} y'$ the fact that considering the model p and the vector $(\mathbf{x})$ is sufficient to state (or implies) $y \geq y'$.

In the case of precise classifiers, we have $y \geq_{p,(\mathbf{x})} y'$ when the condition [1]

$$\frac{p(y|\mathbf{x})}{p(y'|\mathbf{x})} \geq 1 \tag{1}$$

is met, or in other words when $p(y|\mathbf{x}) \geq p(y'|\mathbf{x})$. However, probabilistic classifiers can be deceptively precise, for instance when only a small number of data are available to estimate them, or when data become imprecise.

1. Using dominance expressed this way will be useful in the sequel. We will also restrict ourselves to 0/1 loss functions here.

This is why, in this paper, we consider generalised probabilistic settings, and more specifically imprecise probability theory, where one considers that the probability p belongs to some subset $\mathcal{P}$, often assumed to be convex (this will be the case here). One then needs to extend the relation $\geq_p$ to such a case, and a common and robust way to do so is to require $\geq_p$ to be true for all elements $p \in \mathcal{P}$. In this case, y is said to robustly dominate y' upon observing a vector $\mathbf{x}$, written $y \succ_{\mathcal{P},(\mathbf{x})} y'$, when the condition

$$\inf_{p \in \mathcal{P}} \frac{p(y|\mathbf{x})}{p(y'|\mathbf{x})} \geq 1 \tag{2}$$

is met, or in other words when $p(y|\mathbf{x}) \geq p(y'|\mathbf{x})$ for all $p \in \mathcal{P}$. Going from the precise to imprecise probabilities can introduce incomparabilities between classes, written $y \succ\!\prec_{\mathcal{P},(\mathbf{x})} y'$ when both

$$\inf_{p \in \mathcal{P}} \frac{p(y|\mathbf{x})}{p(y'|\mathbf{x})} < 1 \text{ and } \inf_{p' \in \mathcal{P}} \frac{p'(y'|\mathbf{x})}{p'(y|\mathbf{x})} < 1. \tag{3}$$

3 Explaining robust classification through prime implicants

Explaining the conclusion or deduction of an algorithm, and in particular of a learning algorithm, has become (again) an important issue [6]. A notion that can play a key role in explanation mechanisms is the one of prime implicants, i.e., which elements are sufficient for drawing a given conclusion. In this section, we detail how prime implicants can be used to answer the needs of different explanatory mechanisms, within the setting of robust, imprecise probabilistic classifiers.

3.1 Prime implicants as validatory explanation

When observing a vector $\mathbf{x}^o$ and making a prediction about whether y dominates y', finding a prime implicant confirming that y dominates y' corresponds to finding the values of $\mathbf{x}^o$ that are sufficient to state that y dominates y', and that are minimal with this property.

With this idea in mind, we will say that a subset $E \subseteq \{1, \ldots, n\} := [\![1, n]\!]$ of attributes (where E are the indices of the considered attributes) is a *validatory implicant* of $y \succ_{\mathcal{P},(\mathbf{x}_E^o, \cdot)} y'$ iff

$$\inf_{\mathbf{x}_{-E}^v \in \mathcal{X}_{-E}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}_E^o, \mathbf{x}_{-E}^v))}{p(y'|(\mathbf{x}_E^o, \mathbf{x}_{-E}^v))} \geq 1, \tag{4}$$

that is if dominance holds for any values of attributes whose indices are outside E, and any probability $p \in \mathcal{P}$. This means that knowing $\mathbf{x}_E^o$ alone is sufficient to deduce $y \succ y'$. A set E is a *prime implicant* iff we satisfy (4) and for any $i \in E$, we have

$$\inf_{\mathbf{x}_{-E\cup\{i\}}^v \in \mathcal{X}_{-E\cup\{i\}}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}_{E\setminus\{i\}}^o, \mathbf{x}_{-E\cup\{i\}}^v))}{p(y'|(\mathbf{x}_{E\setminus\{i\}}^o, \mathbf{x}_{-E\cup\{i\}}^v))} \leq 1, \tag{5}$$

that is if removing any attribute from E makes our deduction invalid, so that E is a minimal sufficient condition for $y \succ_{\mathcal{P},(\mathbf{x}^o_E,\cdot)} y'$ to hold for any completion of $-E$. In the sequel, it will prove useful to consider the function that associates to each possible subset the value of the ratio between the obtained posterior probabilities. This function ϕ^v is defined by :

$$\phi^v(y, y', \mathbf{x}^o, E) := \inf_{\mathbf{x}^v_{-E} \in \mathcal{X}_{-E}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^o_E, \mathbf{x}^v_{-E}))}{p(y'|(\mathbf{x}^o_E, \mathbf{x}^v_{-E}))}. \quad (6)$$

To ease the use of this function, we will omit the observed vector $\mathbf{x}^o$ when context is clear and we will write "y, y'" as a subscript meaning that class y is the numerator and y' the denominator, i.e., $\phi^v_{y,y'}(E) := \phi^v(y, y', \mathbf{x}^o, E)$

Note that when the set $\mathcal{P}$ reduces to a singleton, that is when we consider precise classifiers instead of robust ones, then our notion of prime implicant reduces to previously proposed ones [15], and our approach is therefore a formal generalisation of those.

Monotony with respect to imprecision one can note that the notion of validatory prime implicant is monotonic with respect to imprecision, in the following sense

Proposition 1. *Consider two credal sets $\mathcal{P}' \subseteq \mathcal{P}$, then*

$$y \succ_{\mathcal{P},(\mathbf{x}^o_E,\cdot)} y' \implies y \succ_{\mathcal{P}',(\mathbf{x}^o_E,\cdot)} y'$$

Démonstration. Immediate, since if Equation (5) is true for E and $\mathcal{P}$, it must be true for E and $\mathcal{P}'$, as the infimum is taken over a smaller domain. □

This means that a validatory implicant will remain so if we consider a more precise model (obtained, e.g., by observing additional data). However, if a subset E was prime for $\mathcal{P}$, it does not need to be so for $\mathcal{P}'$, meaning that the size of validatory prime implicants should decrease as imprecision decreases. This is somehow natural, as a more informative models should need less measurements to provide a conclusion.

3.2 Prime implicants as contrastive explanations

Another quite common way to audit or explain a statement *"Why X is P ?"* is by answering the implicit question *"Why X is P and not Q ?"* [16, 12]. This can classically be answered by finding a counter-factual, i.e., a modification of the example with sufficient changes so as to change our conclusion. Replying to this question in a minimal way can be seen as the task of finding a minimal set of attributes or features for which a modification could change our decision. We will call $E \subseteq [\![1, n]\!]$ a *contrastive prime implicant* if modifying the attributes within E is a minimal sufficient condition to change our decision, that is, if

$$\inf_{\mathbf{x}^c_E \in \mathcal{X}_E} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))}{p(y'|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))} < 1, \quad (7)$$

and if for any $i \notin E$, we have

$$\inf_{\mathbf{x}^c_{E\setminus\{i\}} \in \mathcal{X}_{E\setminus\{i\}}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^c_{E\setminus\{i\}}, \mathbf{x}^o_{-E\cup\{i\}}))}{p(y'|(\mathbf{x}^c_{E\setminus\{i\}}, \mathbf{x}^o_{-E\cup\{i\}}))} \geq 1, \quad (8)$$

that is there is at least one modification of feature values in E that lead to a different decision, and any change done within a subset of it would not change the decision. Denoting $\mathbf{x}^c_E$ the argument of Equation (7), E is a contrastive implicant if $y \not\succ_{\mathcal{P},(\mathbf{x}^c_E,\mathbf{x}^o_{-E})} y'$. We also consider the function $\phi^c_{y,y'}$ that associates to each possible subset the value

$$\phi^c_{y,y'}(E) := \inf_{\mathbf{x}^c_E \in \mathcal{X}_E} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))}{p(y'|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))} \quad (9)$$

One of the interesting aspects of considering imprecise models is that contrastive explanations do not necessary lead to reversing the initial preference (which is the case for precise models). Indeed, modifying the conclusion $y \succ_{\mathcal{P},\mathbf{x}^o} y'$ by considering the modified vector $(\mathbf{x}^c_E, \mathbf{x}^o_{-E})$ can lead to two quite different situations, resulting either in $y' \succ_{\mathcal{P},(\mathbf{x}^c_E,\mathbf{x}^o_{-E})} y$ (reversing of preference) or $y \succ\!\prec_{\mathcal{P},(\mathbf{x}^c_E,\mathbf{x}^o_{-E})} y'$ (weakening of preference) and we will define two notions of contrastive explanations.

Given that E is a contrastive prime implicant, we say that it is also a *reversing prime implicant* if in addition we have [2]

$$\inf_{p \in \mathcal{P}} \frac{p(y'|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))}{p(y|(\mathbf{x}^c_E, \mathbf{x}^o_{-E}))} \geq 1, \quad (10)$$

as this contrastive prime implicant change the initial statement or conclusion into its reverse. Otherwise, if it does satisfy Equations (7) and (8), but not (10), we say that E is a *weakening prime implicant*, as it changes a preference between two classes into incomparability. The vector $(\mathbf{x}^c_E, \mathbf{x}^o_{-E})$ also provides us with a contrastive example for which the decision would change.

Monotony with respect to imprecision as with validatory implicants, the notion of contrastive implicants is monotonic with respect to imprecision, but in the other direction.

Proposition 2. *Consider two credal sets $\mathcal{P} \subseteq \mathcal{P}'$, then*

$$y \not\succ_{\mathcal{P},(\mathbf{x}^c_E,\mathbf{x}^o_{-E})} y' \implies y \not\succ_{\mathcal{P}',(\mathbf{x}^c_E,\mathbf{x}^o_{-E})} y'$$

Démonstration. Immediate, since if Equation (7) is true for E and $\mathcal{P}$, it must be true for E and $\mathcal{P}'$, as the infinimum is taken over a larger domain, and is of lower value. □

This means that a contrastive implicant and the associated example $(\mathbf{x}^c_E, \mathbf{x}^o_{-E})$ will remain so if we consider a more imprecise model. However, if a subset E was prime for $\mathcal{P}$, it does not need to be so for $\mathcal{P}'$, meaning that the size of

2. Recall that $\mathbf{x}^c_E$ is the argument of Equation (7).

contrastive prime implicants should decrease as imprecision increases. Again, this is somehow intuitive, as a dominance obtained for a more imprecise model should be easier to modify than the same dominance obtained from a more precise model. It should also be noted that the argument $\mathbf{x}^c_E$ obtained for $\mathcal{P}$ in Equation (7) may actually change when considering $\mathcal{P}'$.

Remark 3. *Equation* (7) *corresponds to finding some minimal changes that could modify our conclusion, and can therefore be viewed as a tool to analyse the robustness of this decision. As such, it can be useful to analyse the model and its robustness. However, answering the question "what should I change to be sure to reverse the dominance" would necessitate another notion where the satisfaction of Equation* (10) *is enforced, that we will not consider here, leaving it for future work.*

3.3 Prime implicants as explanation of doubt

For precise models, the statement *"Why X is P ?"* that we have to explain is typically a precise assignment or a dominance relation between two classes. In the case of robust classification, the question *"Why X is neither P nor Q ?"*, and for what reasons cannot I classify X precisely, also makes sense.

In this case, we say that $E \subseteq [\![1, n]\!]$ is an *implicant of doubt* if

$$\sup_{\mathbf{x}^d_{-E} \in \mathcal{X}_{-E}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))}{p(y'|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))} < 1, \tag{11a}$$

and

$$\sup_{\mathbf{x}^d_{-E} \in \mathcal{X}_{-E}} \inf_{p \in \mathcal{P}} \frac{p(y'|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))}{p(y|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))} < 1, \tag{11b}$$

that is any change performed outside of E (in particular the changes for the most favourable values for y in Equation (11a) and the most favourable for y' in Equation (11b)) will not modify the fact that the two classes are incomparable given our model and knowledge of it. It is further more minimal, *i.e. prime*, if for any $i \notin E$, we have either

$$\sup_{\mathbf{x}^d_{-E\cup\{i\}} \in \mathcal{X}_{-E\cup\{i\}}} \inf_{p \in \mathcal{P}} \frac{p(y'|(\mathbf{x}^o_{E\setminus\{i\}}, \mathbf{x}^d_{-E\cup\{i\}}))}{p(y|(\mathbf{x}^o_{E\setminus\{i\}}, \mathbf{x}^d_{-E\cup\{i\}}))} \geq 1, \tag{12a}$$

or

$$\sup_{\mathbf{x}^d_{-E\cup\{i\}} \in \mathcal{X}_{-E\cup\{i\}}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^o_{E\setminus\{i\}}, \mathbf{x}^d_{-E\cup\{i\}}))}{p(y'|(\mathbf{x}^o_{E\setminus\{i\}}, \mathbf{x}^d_{-E\cup\{i\}}))} \geq 1, \tag{12b}$$

We also consider the function $\phi^d_{y,y'}$ that associates to each possible subset the value

$$\phi^d_{y,y'}(E) := \sup_{\mathbf{x}^d_{-E} \in \mathcal{X}_{-E}} \inf_{p \in \mathcal{P}} \frac{p(y|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))}{p(y'|(\mathbf{x}^o_E, \mathbf{x}^d_{-E}))} \tag{13}$$

$\phi^d_{y,y'}$ corresponds to Equation (11a) and $\phi^d_{y',y}$ to Equation (11b) and both will be used in the computations as we will see in Section 4 for the naive credal classifier. In general, the vectors $\mathbf{x}^d_{-E}$ for which the bounds of (11a)-(11b) are obtained will be different.

Monotony with respect to imprecision as before, we can easily show that implicants of doubt are somehow monotonic with imprecision, in the following sense

Proposition 4. *Consider two credal sets $\mathcal{P} \subseteq \mathcal{P}'$, then*

$$y \succ\!\prec_{\mathcal{P},(\mathbf{x}^o_E,\cdot)} y' \implies y \succ\!\prec_{\mathcal{P}',(\mathbf{x}^o_E,\cdot)} y'$$

Démonstration. Immediate, since if Equations (11a)-(11b) are true for E and $\mathcal{P}$, it must be true for E and $\mathcal{P}'$, as the infinimum is taken over a larger domain, and is of lower value. □

It should be remarked that while those implicants are also of validatory nature, in the sense that they confirm our conclusion, their monotonicity is not in the same direction as the validatory implicants of dominance relations. This is however not surprising : as imprecision increases, it becomes easier to obtain that two classes are incomparable, hence prime implicants should decrease in size as credal sets become more imprecise.

Remark 5. *We could also have considered contrastive implicants of doubt that would transform the incomparability into a dominance relation, as done for example in [21]. Such implicants would be of the same kind as the ones mentioned in Remark 3, as they would answer the question "what should I change to be sure to have a dominance relation".*

Remark 6. *In contrast with the previous implicants trying to either verify or contradict a dominance relation between two classes, it may be that $E = \emptyset$ is the only prime implicant of $y \succ\!\prec y'$, in which case doubt is simply due to inherent imprecision of our information (think, for instance, about the case of total ignorance).*

3.4 Short discussion about the three types of Prime implicants

We defined three functions $\phi^c_{y,y'}, \phi^d_{y,y'}, \phi^v_{y,y'}$ which are inclusion-monotonic : for $\phi^d_{y,y'}$, $\phi^v_{y,y'}$ and $E \subseteq F$, we have $\phi^{\cdot}_{y,y'}(E) \leq \phi^{\cdot}_{y,y'}(F)$, and for $\phi^c_{y,y'}$ and $E \subseteq F$, we have $\phi^c_{y,y'}(E) \geq \phi^c_{y,y'}(F)$.

This means that they can be seen as value functions associated to E, and that finding a prime implicant amounts to the task of finding a minimal "bundle of items" [3] E such that $\phi^v_{y,y'}(E) \geq 1$, $\phi^c_{y,y'}(E) < 1$ or $\phi^d_{y,y'}(E) < 1$, therefore allowing us to map the finding of robust prime implicants to an item selection problem or to knapsack

3. Each index of an attribute being associated to an item.

problems where we have to fill the sack until it reaches a certain value. Unfortunately, in general, the log-functions [4] of each problem will not be additive, as we will not have $\log \dot{\phi}_{y,y'}(E \cup \{i\}) = \log \dot{\phi}_{y,y'}(E) + \log \dot{\phi}_{y,y'}(\{i\})$. We will nevertheless show in Section 4 that it is the case for the Naive credal classifier.

While providing a minimal subset of features such that a preference/dominance is preserved or changed can be considered to some extent as satisfactory for the user [19, 7] (as long as those features have a meaning for the user), the same cannot really be said about non-dominance or incomparability. In such a case, the user will probably not be satisfied by the mere fact that features values in E are sufficient to claim incomparability, and will request to know why this incomparability happens.

In a machine learning setting, it makes sense to differentiate between incomparability due to ambiguity, where a small change in our knowledge representation $\mathcal{P}$ would lead to a decision, from incomparability due to lack of knowledge, where it would require significantly more knowledge to obtain a decision. These two types of uncertainty sources are often referred to as epistemic and aleatoric uncertainties, and those can be quantified [13].

It seems reasonable that the complementary explanation to incomparability should differ according to the dominating source of uncertainty or indecision. In particular :

— if the indecision is mainly due to aleatoric uncertainties, it is clear that collecting more data is unlikely to solve the issue, and that it would be important to identify those features that generate the ambiguity. In this case, it would seem preferable to provide a contrastive explanation (in the sense of Section 3.2) rather than recommending the collection of further data, so as to answer the question : "which features generate my ambiguity ?".

— if the indecision is mainly due to epistemic uncertainties, a possible way to answer this question is to know how many further data points would we need to collect (and which ones) in order to reach a conclusion rather than producing none. The question we would answer would then be "what data should I collect to gain knowledge ?"

It is clear to us that providing formal reasons as to why an incomparability is observed, and proposing tools in this direction is a worthwile undertaking, and that our proposal could be useful to the analyst as a way to audit the model (why is my model doubting, and what could I do about it ?). It is less clear that the notion proposed in this paper is instrumental to the end-user. Indeed, once $\mathbf{x}^o$ is known, letting the end-user know that we could have known earlier (i.e., with less measurement) that we could not reach a decision is not very helpful. However, our approach can also be considered to detect from partial observations, and before measuring all features, that incomparability will ensue whatever happens, therefore sending an early signal that with this model and this degree of cautiousness, taking more measurements is fruitless.

A definite goal we have in mind for future work is to go beyond the definition of prime implicants of doubts, and investigate problems such as active learning or feature acquisition in which they could offer an operational advantage.

[4] As we will deal later with joint probabilities and independence, using log transform will allow us to turn products into sums.

4 The case of the Naive credal classifier

We now study the specific case of the Naive credal classifier [20], and show that in this case, computing prime implicants become easy, as such a computation can be brought back to selecting items with an additive value functions, or equivalently to simple knapsack problems.

4.1 Generic case

The basic idea of the Naive credal classifier (the same as its precise counterpart) is to assume that attributes are independent of each other given the class. This modelling assumption means that

$$p(y|\mathbf{x}) = \frac{\prod_{i=1}^{n} p_i(\mathbf{x}_i|y) \times p_Y(y)}{p(\mathbf{x})}$$

once we apply the Naive assumption and Bayes rule. This means in particular that

$$\frac{p(y|\mathbf{x})}{p(y'|\mathbf{x})} = \frac{p_Y(y)}{p_Y(y')} \prod_{i=1}^{n} \frac{p_i(\mathbf{x}_i|y)}{p_i(\mathbf{x}_i|y')}$$

with every $p_i(\cdot|y)$ being independent of $p_i(\cdot|y')$, and every $p_i(\cdot|y)$, $p_j(\cdot|y)$ independent for $i \neq j$. When switching to credal models, one considers sets of conditional distributions $\mathcal{P}_{X_i}(\cdot|y)$ and a set $\mathcal{P}_Y$ of priors rather than precise probabilities. We will abuse the notation $\mathcal{P}_{X_i}$ by $\mathcal{P}_i$ and p_{X_i} by p_i for the sake of conciseness.

In our Equations (4), (7) and (11a, 11b) we have two optimisation problems, one in $\mathcal{X}_E$ (or $\mathcal{X}_{-E}$) and one in $\mathcal{P}$. Thanks to the independence assumptions, the two problems can be solved independently. Let us now see how the common part of the three Equations, the problem in $\mathcal{P}$, transform in this case. We have

$$\inf_{p \in \mathcal{P}} \frac{p(y|\mathbf{x})}{p(y'|\mathbf{x})} = \inf_{p_Y \in \mathcal{P}_Y} \frac{p_Y(y)}{p_Y(y')} \prod_{i=1}^{n} \inf_{p_i \in \mathcal{P}_i} \frac{p_i(\mathbf{x}_i|y)}{p_i(\mathbf{x}_i|y')} \quad (14)$$

Once again, thanks to our Independence assumption, each term of the equation can be taken independently of the others (different variables) and inside each feature the numerator is independent of the denominator (different conditioning element). Moreover, as our probability sets $\mathcal{P}_i$ are

convex, finding the minimum and maximum value is usually easy. Finally, we get that Equation (14) becomes

$$\inf_{p_Y \in \mathcal{P}_\mathcal{Y}} \frac{p_Y(y)}{p_Y(y')} \prod_{i=1}^{n} \frac{\underline{p}_i(\mathbf{x}_i|y)}{\overline{p}_i(\mathbf{x}_i|y')} \tag{15}$$

where $\underline{p}(\mathbf{x}) = \inf_{p \in \mathcal{P}} p(\mathbf{x})$ and $\overline{p}(\mathbf{x}) = \sup_{p \in \mathcal{P}} p(\mathbf{x})$. Let's note $\underline{p}^{y,y'} = \inf_{p_Y \in \mathcal{P}_\mathcal{Y}} \frac{p_Y(y)}{p_Y(y')}$. We can now rewrite our functions as :

$$\phi^v_{y,y'}(E) = \underline{p}^{y,y'} \prod_{i \in E} \frac{\underline{p}_i(\mathbf{x}^o_i|y)}{\overline{p}_i(\mathbf{x}^o_i|y')} \prod_{i \in -E} \inf_{\mathbf{x}^v_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^v_i|y)}{\overline{p}_i(\mathbf{x}^v_i|y')} \tag{16a}$$

$$\phi^c_{y,y'}(E) = \underline{p}^{y,y'} \prod_{i \in E} \inf_{\mathbf{x}^c_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^c_i|y)}{\overline{p}_i(\mathbf{x}^c_i|y')} \prod_{i \in -E} \frac{\underline{p}_i(\mathbf{x}^o_i|y)}{\overline{p}_i(\mathbf{x}^o_i|y')} \tag{16b}$$

$$\phi^d_{y,y'}(E) = \underline{p}^{y,y'} \prod_{i \in E} \frac{\underline{p}_i(\mathbf{x}^o_i|y)}{\overline{p}_i(\mathbf{x}^o_i|y')} \prod_{i \in -E} \sup_{\mathbf{x}^d_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^d_i|y)}{\overline{p}_i(\mathbf{x}^d_i|y')} \tag{16c}$$

As we see in Equations (16a), (16b) and (16c), in the case of the NCC the optimisation on $\mathcal{X}_E$ (or $\mathcal{X}_{-E}$) is independent of the computation of $\phi_{y,y'}(E)$. It follows that the results are unique and can be computed before choosing the items in E. We can represent them by unique "worst opponent" vectors, depending only on classes y and y' (the former in the numerator and the later at the denominator) :

$$\mathbf{x}^{v:y,y'} = \times_{i=1}^n \arg \inf_{\mathbf{x}^v_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^v_i|y)}{\overline{p}_i(\mathbf{x}^v_i|y')}$$

$$\mathbf{x}^{c:y,y'} = \times_{i=1}^n \arg \inf_{\mathbf{x}^c_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^c_i|y)}{\overline{p}_i(\mathbf{x}^c_i|y')}$$

$$\mathbf{x}^{d:y,y'} = \times_{i=1}^n \arg \sup_{\mathbf{x}^d_i \in \mathcal{X}_i} \frac{\underline{p}_i(\mathbf{x}^d_i|y)}{\overline{p}_i(\mathbf{x}^d_i|y')}$$

When we solve the problem of selecting E, in the case of validatory and contrastive prime implicants, we will only use the "worst opponent" vectors with "y, y'", whereas we also need the converse "y', y" for prime implicants of doubt. We will then refer to $\mathbf{x}^v$ and $\mathbf{x}^c$ instead of $\mathbf{x}^{v:y,y'}$ and $\mathbf{x}^{c:y,y'}$. We can also note that $\mathbf{x}^v$ and $\mathbf{x}^c$ are equal in the case of NCC, the two problems of finding validatory and contrastive prime implicants only differing by the fact that in the validatory case, elements of E are fixed, while they are modified in the contrastive case. It should be noted that this uniqueness of "worst opponent" is not true for more generic models, in the sense that the arguments Equations (4), (7) and (11a, 11b) in $\mathcal{X}_E$ will typically depend on E.

Coming back to the NCC, we can also see that each of our function is additive on their log form. Indeed, for instance for $\phi^v_{y,y'}$ we have :

$$\begin{aligned}\log \phi^v_{y,y'}(E \cup \{i\}) - \log \phi^v_{y,y'}(E) = \\ (\log\ \underline{p}_i(\mathbf{x}^o_i|y) - \log\ \overline{p}_i(\mathbf{x}^o_i|y')) \\ - (\log\ \underline{p}_i(\mathbf{x}^v_i|y) - \log\ \overline{p}_i(\mathbf{x}^v_i|y'))\end{aligned} \tag{17}$$

As this value is independent of any feature (inside or outside E) different from i. We can therefore define contribution functions G^v, G^c and $G^d_{y,y'}$ mapping each feature i to the contribution of adding i to E :

$$\begin{aligned}G^v(i) = (\log\ \underline{p}_i(\mathbf{x}^o_i|y) - \log\ \overline{p}_i(\mathbf{x}^o_i|y')) \\ - (\log\ \underline{p}_i(\mathbf{x}^v_i|y) - \log\ \overline{p}_i(\mathbf{x}^v_i|y'))\end{aligned} \tag{18a}$$

$$\begin{aligned}G^c(i) = (\log\ \underline{p}_i(\mathbf{x}^c_i|y) - \log\ \overline{p}_i(\mathbf{x}^c_i|y')) \\ - (\log\ \underline{p}_i(\mathbf{x}^o_i|y) - \log\ \overline{p}_i(\mathbf{x}^o_i|y'))\end{aligned} \tag{18b}$$

$$\begin{aligned}G^d_{y,y'}(i) = (\log\ \underline{p}_i(\mathbf{x}^o_i|y) - \log\ \overline{p}_i(\mathbf{x}^o_i|y')) \\ - (\log\ \underline{p}_i(\mathbf{x}_i^{d:y,y'}|y) - \log\ \overline{p}_i(\mathbf{x}_i^{d:y,y'}|y'))\end{aligned} \tag{18c}$$

We see that, by definition, values of functions G^v's are at least zero because we replace the worst opponent value with a better value (the observed one) [5] and is at most 0 for G^c and $G^d_{y,y'}$. It follows that

$$\log \phi^v_{y,y'}(E) = \log \phi^v_{y,y'}(\emptyset) + \sum_{i \in E} G^v(i) \tag{19a}$$

$$\log \phi^c_{y,y'}(E) = \log \phi^c_{y,y'}(\emptyset) + \sum_{i \in E} G^c(i) \tag{19b}$$

$$\log \phi^d_{y,y'}(E) = \log \phi^d_{y,y'}(\emptyset) + \sum_{i \in E} G^d_{y,y'}(i) \tag{19c}$$

We will note the log-contributions of the empty set by C^v, C^c and $C^d_{y,y'}$.

Using these additive rewriting, we will now investigate how to compute our three types of prime implicants (validatory, contrastive and doubt), and the associated complexity.

4.2 Validatory Prime implicants

From Equation (19a), we have that $\log \phi^v_{y,y'}(E) = C^v + \sum_{i \in E} G^v(i)$ and our goal (4) is to find subsets $E \subseteq [\![1, n]\!]$ such that $\log \phi^v_{y,y'}(E) \geq 0$.

It follows that we want to optimise E so that the sum of positive contributions is greater than C^v. Finding a smallest prime implicant is then computationally easy, as it amounts to order the $G^v(i)'s$ in decreasing order, and add them until $\sum_{i \in E} G^v(i) \geq -C^v$. The whole procedure is summarised in Algorithm 1.

The complexity of Algorithm 1 is linear over the ordered contributions, in number of attributes. Computing

5. Indeed, $\log\ \underline{p}_i(\mathbf{x}^v_i|y) - \log\ \overline{p}_i(\mathbf{x}^v_i|y') < \log\ \underline{p}_i(\mathbf{x}^o_i|y) - \log\ \overline{p}_i(\mathbf{x}^o_i|y')$ by definition.

Input: C^v ; G^v
Output: $Xpl = (E, \mathbf{x}^o_E)$: explanation in terms of attribute
Order G^v in decreasing order, with σ the associated permutation
$i \leftarrow 1$
while $\phi^v_{y,y'}(E) + C^v < 0$ **do**
  $i \leftarrow i + 1$
  $E \leftarrow E \cup \{\sigma^{-1}(i)\}$
  $\phi^v_{y,y'}(E) \leftarrow \phi^v_{y,y'}(E) + G^v(\sigma(i))$
end
$Xpl \leftarrow (E, \mathbf{x}^o_E)$
return (Xpl)
Algorithm 1: Compute first available prime implicants explanation

the contributions remains easy as the only complexity is to compute the "worst case" vector $\mathbf{x}^v$, whose components $\mathbf{x}^v_i$ requires $|X_i| = k_i$ evaluations on each dimensions. As sets $\mathcal{P}$ are typically polytopes defined by linear constraints, finding the values $\underline{p}$ and $\overline{p}$ amounts to solving linear programs, something that can be done in polynomial time. For some specific cases such as probability intervals [9] (induced, e.g., by the classical Imprecise Dirichlet Model [4]), this can even be done in linear time. Therefore, the overall method is polynomial, with a linear pre-treatment over the sum of k_i's, followed by a sorting algorithm, after which Algorithm 1 is linear over the number of attributes.

4.3 Contrastive Prime implicants

The case of the contrastive prime implicants is straightforward once we solved the validatory prime implicants. Indeed, as suggested by the similarity between the definitions of $\phi^v_{y,y'}$ and $\phi^c_{y,y'}$ in Equations (16a) and (16b) and the definitions of $\mathbf{x}^v$ and $\mathbf{x}^c$, we almost compute the same thing, the difference being that the role of E for $\phi^v_{y,y'}$ is fulfilled by $-E$ for $\phi^c_{y,y'}$. We obtain that $C^c > 0$ whereas we had $C^v < 0$, as C^c is obtained when we observe the full vector $\mathbf{x}^o$, and that $G^c(i) \leq 0$. To use Algorithm 1, we only need to change the while condition to $\phi^c_{y,y'}(E) + C^c > 0$, and the vector G^v to be ordered in ascending order.

That such strong duality relations hold in general is unlikely, even if validatory and contrastive explanations are known to be linked in general [14].

4.4 Prime implicants of doubt

From the definition of prime implicant of doubt in Equations (11a) and (11b) we have to investigate simultaneously two problems, one in favour of y against y' and one in favour of y' against y. To do so we have two functions $\phi^d_{y,y'}$ and $\phi^d_{y',y}$, which in the case of the NCC are additive :

$$\log \phi^d_{y,y'}(E) = C^d_{y,y'} + \sum_{i \in E} G^d_{y,y'}(i),$$
$$\log \phi^d_{y',y}(E) = C^d_{y',y} + \sum_{i \in E} G^d_{y',y}(i).$$

$C^d_{y,y'}$ and $C^d_{y',y}$ are obtained when we assume observing the "worst case opponents" $\mathbf{x}^{d:y,y'}$ and $\mathbf{x}^{d:y',y}$, the two vectors the most in favour of y against y' and of y' against y. In practice, we then want to find which features of $\mathbf{x}^o$ are sufficient to observe so that both dominance relationships (if they hold for some vectors $\mathbf{x}^{d:y,y'}$, $\mathbf{x}^{d:y',y}$, which may not be the case as hinted by Remark 6) are broken, *i.e.*, both $y \not\succ_{\mathcal{P},(\mathbf{x}^o_E, \mathbf{x}^{d:y,y'}_{-E})} y'$ and $y' \not\succ_{\mathcal{P},(\mathbf{x}^o_E, \mathbf{x}^{d:y',y}_{-E})} y$. This problem can be represented as a 2-dimensional Knapsack where the objects are the features and the two Knapsacks corresponds to the dominance of y over y' and the converse. We obtain the following formulation

$$\begin{aligned} &\min \sum_{i=1}^{n} x_i \\ &\text{subject to} \\ &\sum_{i=1}^{n} x_i * G^d_{y,y'}(i) \leq -C^d_{y,y'}, \\ &\sum_{i=1}^{n} x_i * G^d_{y',y}(i) \leq -C^d_{y',y}, \\ &\forall i \in \{1, \ldots, n\}\ x_i \in \{0, 1\}. \end{aligned} \tag{20}$$

This problem can be solved by using an efficient MILP solver. The indexes with a non zero associated x_i are the components of E, *i.e.* are our prime implicants.

4.5 NCC with the Imprecise Dirichlet Model

In this section we will present the Imprecise Dirichlet Model [4], which is a classical model of representation of domain of probabilities, and study how the prime implicants will behave in this case.

The main idea of the IDM is to build a cautious interval around a precise probability distribution. Let's note the number of observation of an event X by n_X, same notations for a conditional event $X|Y$ by $n_{X|Y}$ and N the total number of observation. We obtain that the probability of witnessing X is $\frac{n_X}{N}$. We introduce the meta-parameter s of the IDM which can be interpreted as a number of "unwitnessed" observations. As these could be or not witnessed for X the probability of X belongs to the interval $\left[\frac{n_X}{N+s}, \frac{n_X+s}{N+s}\right]$.

We can easily see that, in the case of the IDM, the s hyper-parameter allows us to go from fully precise ($s = 0$) to fully imprecise ($s = \infty$), meaning that the monotonicity properties we mentioned so far can easily be checked by modifying its value.

5 First Experiments

This Section will present an illustrative case based on the data from the Zoo dataset from UCI repository [11] using the NCC alongside the IDM. To avoid probabilities of 0 we will regularize them by mixing them with a uniform distribution (using a coefficient $\epsilon = 0.05$ to weight this uniform). The experiment will be separated in two parts. The first one will focus on trying to answer with a quantitative study the questions "How do the different implicants behave, in size, absence, number, based on the kind of implicants we search, and on whether we justify a relation consistent with the ground truth ?" and "Is the size of pairwise explanations dependent of the imprecision and the number of predicted classes ?". Second one will illustrate how a discussion with a user could occur based on this data for the different types of explanations.

The Zoo dataset is a classification dataset containing 101 samples of animals with 16 input features and the class. The classes are numbers from 1 to 7 corresponding to Mammal, Bird, Reptile, Fish, Amphibian, Bug and Invertebrate. We used 14 features for classification ('feathers' : $\{fe, \neg fe\}$, 'eggs' : $\{e, \neg e\}$, 'airborne' : $\{ai, \neg ai\}$, 'aquatic' : $\{aq, \neg aq\}$, 'predator' : $\{p, \neg p\}$, 'toothed' : $\{to, \neg to\}$, 'backbone' : $\{b, \neg b\}$, 'breathes with lungs' : $\{l, \neg l\}$, 'venomous' : $\{v, \neg v\}$, 'fins' : $\{fi, \neg fi\}$, 'legs' : $\{0, 2, 4, 5, 6, 8\}$, 'tail' : $\{ta, \neg ta\}$, 'domestic' : $\{d, \neg d\}$, 'at least catsize' : $\{c, \neg c\}$), all binary except for the number of legs. To have sufficient classification errors, we removed 2 features ('hair' and 'milk') from the original features.

5.1 Quantitative study

We performed this study using a 4-Fold cross-validation with a stratified data separation, due to the samples by class being very unbalanced, e.g. 41 for Mammals and 4 for Amphibians.

Are Validatory and Contrastive explanations size dependent of miss-classification ? The idea behind this question is to verify the shape of explanations when the observation is well classified against when it is miss-classified. If y is the true class, we could expect explanation for an observed dominance $y \succ_{\mathcal{P},\mathbf{x}^o} y'$ that are "true" to differ from observed dominance $y' \succ_{\mathcal{P},\mathbf{x}^o} y$ that are false. In Figure 1, we plot the size of such explanations.

First note that the monotonicity in terms of imprecision are well observed : as s increases, the size of validatory and contrastive explanations respectively increases and decreases.

Then, we can see that there is no significant difference between the distributions when the prediction is correct or not, except maybe for a bigger variability in the case of wrong prediction. While further experiments would be needed to confirm this, it seems that the length of explanation is not related to whether they explain a correct or incorrect prediction, suggesting that one would have to check their plausibility.

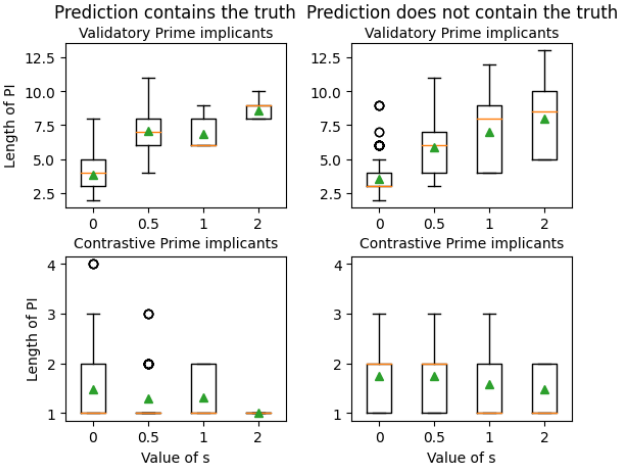


FIGURE 1 – Explanation length according to prediction truth. Green triangles are mean values.

5.1.1 Are Doubt explanation dependent of the number of undominated items ?

We will now focus on prime implicants of doubt explanations. As said in Section 3.4, incomparabilities may arise from lack of knowledge or from ambiguity about the observed element. A question is then to know whether this affects the length of our explanation.

As a proxy, we plotted in Figure 2 the length of implicants explaining incomparabilities against the number of undominated classes, with the idea that this is a reasonable proxy of ambiguity versus lack of knowledge (the more the number of undominated, the more incomparabilities are likely to be due to lack of knowledge). Again, while the Figure 2 does show the expected monotonicity, it seems that the size of pairwise explanation is not especially affected by the final number of classes in the prediction. As we used a proxy, this independence would however have to be confirmed by more precise assessment of whether our incomparability is mainly due to epistemic or aleatoric uncertainty.

5.2 Illustrative explanations

Let us now present some results we can get from the experiments. We will focus on values $s \in \{0.5, 1, 2\}$ and on 3 animals : Giraffe, Seal and Tortoise. We will denote NCCs the corresponding classifier.

Giraffe as a non-ambiguous problem. Described by

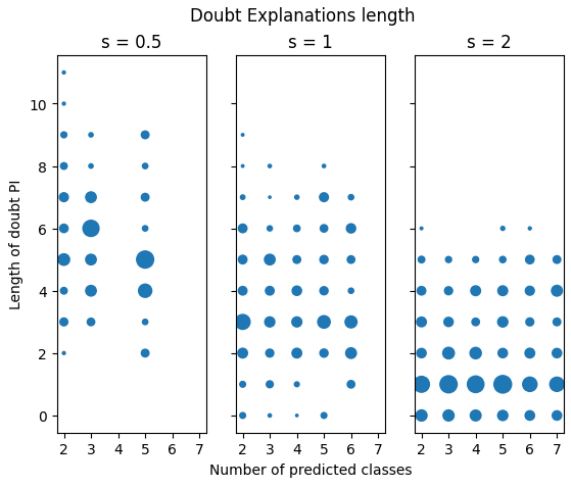


FIGURE 2 – Length of pairwise doubt prime implicants by size of prediction and values of IDM (ball size is normalized with respect to the number of examples having the same number of undominated classes)

$(\neg fe, \neg e, \neg ai, \neg aq, \neg p, to, b, l, \neg v, \neg fi, 4, ta, \neg d, c)$, the Giraffe is a prototypical example of Mammals, as all NCC0.5, NCC1 and NCC2 classifies it as a Mammal only. To illustrate the validatory prime implicants explanations, we will take a look into the preference "Mammal $\succ_{\mathcal{P},Giraffe}$ Bird". For NCC0.5, a sufficient reason to classify the Giraffe as Mammal and not Bird is that it has **no feathers** ($\neg fe$), does **not** produce **eggs** ($\neg e$) and is **toothed** (to). With the increasing cautiousness of NCC1 we need to add the fact that the Giraffe has **4 legs** to the explanation and for NCC2 we then add that it is **not airborne**. All advanced reasons correspond to attributes of Mammals and not of Birds.

A contrastive explanation showing how "robust" our classification is for NCC0.5 and NCC1 that we change the values of the features **feathers**, **eggs** and **toothed**. So, an animal like the giraffe, but with feathers, laying eggs and no teeth could be either a mammal or a bird.

Seal as an ambiguous animal. Described by $(\neg fe, \neg e, \neg ai, aq, p, to, b, l, \neg v, fi, 0, \neg ta, \neg d, c)$, it is classified as a Mammal for NCC0.5, but NCC1 and NCC2 are more cautious by predicting the set $\{Mammal, Reptile, Fish, Amphibian\}$. Let us now investigate the comparison between Mammal (the true class) with Fishes.

For NCC0.5 a sufficient reason to classify as a Mammal is that the Seal **has lungs**, does **not** produce **eggs**, is (at least) **catsized**, is **not venomous**, has **no feathers** and has a **backbone**. Note that this time explanations contain element that support Mammal but can nevertheless be met in fishes as well (e.g., has no feathers). The decision is also less robust, as contrastive explanation shows that flipping one of the features ['lungs', 'eggs', 'catsized', 'venomous'] is enough to make Mammal and Fish incomparable.

When going to NCC1, Mammal $\succ\!\prec_{\mathcal{P},Seal}$ Fish can be explained by the fact that the Seal does **not** produce **eggs**, is **aquatic**, **breathes**, has **fins**, has **no legs**. Interestingly, we can see that the explanation shows that the seal is somehow ambiguous, having some typical features of fishes (aq, fi) as well as of Mammals $(\neg e, l)$.

Tortoise as a mistaken animal. Described by $(\neg fe, e, \neg ai, \neg aq, \neg p, \neg to, b, l, \neg v, \neg fi, 4, ta, \neg d, c)$, it is wrongly labelled by NCC0.5 and NCC1 as a Mammal rather than a Reptile. NCC2 is much less precise and predicts that a Tortoise can be every class except for Fish.

If we investigate the reasons why NCC0.5 believes the Tortoise is a Mammal we obtain the validatory prime implicant **not venomous**, has **4 legs**, is **catsized**, **breathes**, is **not** a **predator**, has a **backbone**, is **not airborne** (for NCC1 we add that it has **no feathers**, is **not aquatic** and has **no fins**.)

The explanation is reasonable but quite long, and does not use the fact that a Tortoise lay eggs (the Platypus being one of the mammal, it is possible for mammals to lay eggs). Also, the Reptile class is poorly represented (4 examples) and most by "serpent like" animals with **no legs**, pretty **venomous**, small (so not **catsized**) and **predators**.

Finally, when increasing to NCC2, we obtain that the doubt between Reptile and Amphibian is not caused by any feature (empty prime implicant of doubt). This clearly shows that Reptile and Amphibian are indistinguishable "by default" and are underrepresented, as the Amphibian and reptile classes have respectively 3 and 4 learning observations which is too little compared to s=2.

6 Conclusion and future works

Considering explanations for imprecise classifier opens up many questions, for instance in relation with the possibility of observing incomparability, or of increasing/decreasing the imprecision of a model. In this paper, we focused on prime implicants, extending notions proposed so far in the precise setting. We introduced three notions of prime implicants in the case of pairwise comparison, answering the questions *"Why X is P ?"*, *"Why X is P and not Q ?"* and *"Why is X neither P nor Q ?"*. When applying them to the Naïve credal classifier, we obtain that the computations are computationally easy (at least for validatory and contrastive explanations).

In the future, we would like to focus on various questions not investigated here, such as for which robust models (e.g., including some dependence statements) do computations remain tractable ? What happens with interaction between attributes ? When trying to explain the complete

partial order, should we use pairwise or holistic (i.e., prime implicants explaining the non-dominated classes at once) explanations ? How do we select what dominance to explain in such a case ? There are also several other explanation mechanisms we could consider [1]. We also want to investigate the case with prediction costs. Indeed, we can associate costs to making a prediction y' when the truth is y. During this paper we used the usual 0/1 cost for all types of miss-classification.

Finally, we also feel that we have only skimmed the surface of the role of incomparability explanations, in the sense that the operational role and advantage of such explanations still remain to be explored.

Références

[1] Audemard, Gilles, Frédéric Koriche et Pierre Marquis: *On tractable XAI queries based on compiled representations*. Dans *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, tome 17, pages 838–849, 2020.

[2] Balasubramanian, Vineeth, Shen Shyang Ho et Vladimir Vovk: *Conformal prediction for reliable machine learning : theory, adaptations and applications*. Newnes, 2014.

[3] Belahcene, Khaled, Christophe Labreuche, Nicolas Maudet, Vincent Mousseau et Wassila Ouerdane: *Explaining robust additive utility models by sequences of preference swaps*. Theory and Decision, 82(2) :151–183, 2017.

[4] Bernard, Jean Marc: *An introduction to the imprecise Dirichlet model for multinomial data*. International Journal of Approximate Reasoning, 39(2-3) :123–150, 2005.

[5] Broeck, Guy Van den, Anton Lykov, Maximilian Schleich et Dan Suciu: *On the tractability of SHAP explanations*. Dans *Proceedings of the 35th AAAI*, 2021.

[6] Burkart, Nadia et Marco F Huber: *A survey on the explainability of supervised machine learning*. Journal of Artificial Intelligence Research, 70 :245–317, 2021.

[7] Chin-Parker, Seth et Julie Cantelon: *Contrastive Constraints Guide Explanation-Based Category Learning*. Cognitive Science, 41(6) :1645–1655, 2017.

[8] Corani, Giorgio, Alessandro Antonucci et Marco Zaffalon: *Bayesian networks with imprecise probabilities : Theory and application to classification*. Dans *Data Mining : Foundations and Intelligent Paradigms*, pages 49–93. Springer, 2012.

[9] De Campos, Luis M, Juan F Huete et Serafin Moral: *Probability intervals : a tool for uncertain reasoning*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2(02) :167–196, 1994.

[10] Del Coz, Juan José, Jorge Díez et Antonio Bahamonde: *Learning Nondeterministic Classifiers*. Journal of Machine Learning Research, 10(10), 2009.

[11] Dua, Dheeru et Casey Graff: *UCI Machine Learning Repository*, 2017.

[12] Guidotti, Riccardo: *Counterfactual explanations and how to find them : literature review and benchmarking*. Data Mining and Knowledge Discovery, pages 1–55, 2022.

[13] Hüllermeier, Eyke, Sébastien Destercke et Mohammad Hossein Shaker: *Quantification of credal uncertainty in machine learning : A critical analysis and empirical comparison*. Dans *Uncertainty in Artificial Intelligence*, pages 548–557. PMLR, 2022.

[14] Ignatiev, Alexey, Nina Narodytska, Nicholas Asher et Joao Marques-Silva: *On Relating'Why ?'and'Why Not ?'Explanations*. arXiv preprint arXiv :2012.11067, 2020.

[15] Marques-Silva, João, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev et Nina Narodytska: *Explaining Naive Bayes and Other Linear Classifiers with Polynomial Time and Delay*. Dans *NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[16] Miller, Tim: *Explanation in artificial intelligence : Insights from the social sciences*. Artif. Intell., 267 :1–38, 2019.

[17] Shih, Andy, Arthur Choi et Adnan Darwiche: *A symbolic approach to explaining Bayesian network classifiers*. Dans *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5103–5111, 2018.

[18] Utkin, Lev V, Andrei V Konstantinov et Kirill A Vishniakov: *An Imprecise SHAP as a Tool for Explaining the Class Probability Distributions under Limited Training Data*. arXiv preprint arXiv :2106.09111, 2021.

[19] Williams, Joseph Jay et Tania Lombrozo: *The role of explanation in discovery and generalization : evidence from category learning*. Cognitive science, 34 5 :776–806, 2010.

[20] Zaffalon, Marco: *The naive credal classifier*. Journal of Statistical Planning and Inference, 105(1) :5–21, 2002.

[21] Zhang, Haifei, Benjamin Quost et Marie Hélène Masson: *Explaining Cautious Random Forests via Counterfactuals*. Dans *International Conference on Soft Methods in Probability and Statistics*, pages 390–397. Springer, 2023.

Session 5 : Choix social et éthique

Actes JIAF-JFPDA 2023

Revisiter l'équité pour le partage de loyer avec budgets

Stéphane Airiau[1] **Hugo Gilbert**[1] **Umberto Grandi**[2]
Jérôme Lang[1] **Anaëlle Wilczynski**[3]

[1] Université Paris-Dauphine, Université PSL, CNRS, LAMSADE, Paris, France
[2] IRIT, Université Toulouse Capitole, Toulouse, France
[3] MICS, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France

stephane.airiau@dauphine.fr hugo.gilbert@lamsade.dauphine.fr
umberto.grandi@irit.fr jerome.lang@lamsade.dauphine.fr
anaelle.wilczynski@centralesupelec.fr

Résumé

Le partage des loyers consiste à calculer simultanément une affectation des chambres aux agents et un paiement, à partir des valuations individuelles de chaque chambre par chaque agent. Lorsque les agents ont une certaine contrainte budgétaire à respecter, une solution sans envie n'existe pas nécessairement. Nous proposons deux manières de contourner ce problème. Premièrement, nous relâchons le critère d'absence d'envie pour tenir compte des disparités budgétaires. Deuxièmement, nous autorisons les allocations fractionnaires, dans lesquelles les agents peuvent changer de chambres pendant la durée de la location.

Abstract

Rent division consists in simultaneously computing an allocation of rooms to agents and a payment, starting from an individual valuation of each room by each agent. When agents have budget limits, it is known that envy-free solutions do not necessarily exist. We propose two solutions to overcome this problem. In the first one, we relax envy-freeness to account for budget disparities. In the second one, we allow fractional allocations, in which agents may change rooms during the duration of the lease.

1 Introduction

Un ensemble de n agents souhaite faire une colocation dans un appartement de n chambres. Les chambres étant différentes, les agents ont des préférences particulières, modélisées par des valuations représentant le montant maximum qu'ils seraient capables de payer pour une chambre. Le problème de partage de loyer standard revient à se demander comment affecter les chambres aux agents et diviser le loyer entre eux. Il existe une solution individuellement rationnelle (aucun agent ne paie plus pour sa chambre que sa valuation pour elle), sans envie (aucun agent ne préfère la situation d'un autre agent) et qui maximise le bien-être social [1].

Cependant, ce problème standard peut ne pas être réaliste dans la mesure où les agents ont en général un budget à respecter pour le paiement de leur chambre. Dans ce contexte, la recherche de solutions individuellement rationnelles et sans envie est loin d'être triviale [2], et peut même s'avérer vaine, comme illustré dans l'exemple suivant :

Considérons une colocation à deux agents 1 et 2 et deux chambres r_1 et r_2 pour un loyer total de 1000. Les deux agents attribuent les mêmes valuations aux chambres, 800 pour r_1 et 400 pour r_2, mais ont des budgets différents, 600 pour l'agent 1 et 500 pour l'agent 2. Si l'on affecte la chambre r_2 à l'agent 1, alors la rationalité individuelle implique que 1 doit payer au plus 400, obligeant 2 à payer 600, ce qui excède son budget. Donc r_1 doit être affectée à 1 et r_2 à 2. Par rationalité individuelle, l'agent 2 ne peut pas payer plus que 400 donc, l'agent 1 doit payer sa limite de budget, 600, pour atteindre un loyer de 1000. L'agent 2, qui est dans une chambre qu'il évalue à 400 pendant qu'il paie exactement ce prix, a une utilité de 0, alors qu'il obtiendrait une utilité de 800-600=200 s'il était dans la situation de l'agent 1. L'agent 2 envie donc l'agent 1.

Cet exemple montre que l'on ne peut pas toujours simultanément satisfaire la rationalité individuelle, les contraintes de budget et l'absence d'envie.

Cependant, deux solutions pourraient être envisagées :

1. Allouer, de manière à respecter les contraintes de budget et de rationalité individuelle, la chambre r_1 à l'agent 1

avec un paiement de 600 et la chambre r_2 à l'agent 2 avec un paiement de 400. On peut argumenter que l'envie de 2 envers 1 n'est pas justifiée puisque 2 n'est pas capable de payer le même paiement que 1. Cette solution peut donc être considérée comme sans envie *vis-à-vis du budget (B-EF)*.

2. Allouer r_1 à 1 et r_2 à 2 *pour la première moitié de l'année* et échanger les chambres pour la seconde moitié, en demandant un paiement de 500 à chacun. Cette solution *fractionnaire* est sans envie et rationnelle individuellement.

Nous explorons dans cet article ces deux manières pour élargir l'ensemble des solutions équitables : l'absence d'envie vis-à-vis du budget et les allocations fractionnaires.

2 Le modèle

Un ensemble R de n chambres doivent être affectées à un ensemble A de n agents. Chaque agent i a des valuations $v_{ij} \in \mathbb{R}^+$ sur chaque chambre $r_j \in R$, et L est le montant total du loyer. Chaque agent i dispose d'un budget $b_i \in \mathbb{R}^+$.

Une solution au problème de partage de loyer consiste en une allocation $\sigma : A \to R$ et un vecteur de paiements $p : A \to \mathbb{R}$ tel que $\sum_i p_i = L$. Une solution est *abordable* si $p_i \leq b_i$ pour tout $i \in A$.

Les agents sont supposés avoir des utilités quasi-linéaires, et une solution (σ, p) est dite *sans envie* si aucun agent ne peut améliorer son utilité en échangeant la chambre et le paiement qui lui sont affectés avec ceux d'un autre agent : (σ, p) est sans envie si $v_{i\sigma(i)} - p_i \geq v_{i\sigma(j)} - p_j$ pour tous agents i, j. Une solution (σ, p) est *individuellement rationnelle (IR)* si pour tout agent i, il est vrai que $v_{i\sigma(i)} - p_i \geq 0$.

3 Absence d'envie vis-à-vis du budget

Lorsque les paiements individuels sont contraints par un budget, la notion d'envie peut naturellement être restreinte aux chambres qui sont abordables pour un agent donné, permettant un relâchement de l'absence d'envie.

Définition 1 *Une solution (σ, p) est* sans envie vis-à-vis du budget *(B-EF) si pour tout agent i il est vrai que $v_{i\sigma(i)} - p_i \geq v_{i\sigma(j)} - p_j$ pour tout agent j t.q. $p_j \leq b_i$.*

Contrairement au cadre sans budget, une allocation B-EF peut ne pas exister et ne pas maximiser le bien-être social. En revanche, nous montrons qu'une solution B-EF ainsi que son allocation sans paiement sont Pareto-optimales. De plus, une solution à la fois B-EF et IR peut être construite, si elle existe, à l'aide d'un programme linéaire en nombre entiers. Nous donnons également deux algorithmes efficaces pour des restrictions particulières : l'un en temps pseudo-polynomial lorsque l'allocation est fixée et l'autre en temps polynomial lorsque les paiements sont fixés.

4 Solutions fractionnaires

Une autre voie pour trouver des allocations sans envie en présence de budgets est d'autoriser les agents à passer une partie de leur temps dans des chambres différentes.

Définition 2 *Une* solution fractionnaire *à un problème de partage de loyer est une matrice bistochastique X de taille $n \times n$, avec x_{ij} la part du temps que l'agent i passe dans la chambre r_j, et un vecteur de paiement $p : A \to \mathbb{R}$.*

Une solution fractionnaire (X, p) est *individuellement rationnelle (IR)* si pour tout agent i, il est vrai que $\sum_{r_j \in R} x_{ij} v_{ij} - p_i \geq 0$. De plus, une solution fractionnaire (X, p) est *sans envie* s'il est vrai pour tous agents i et i' que : $\sum_{r_j \in R} x_{ij} v_{ij} - p_i \geq \sum_{r_j \in R} x_{i'j} v_{ij} - p_{i'}$.

Une solution fractionnaire sans envie n'existe pas toujours. Néanmoins, une solution fractionnaire, à la fois IR et EF, peut être construite en temps polynomial lorsqu'une telle allocation existe, en utilisant un programme linéaire.

Une solution fractionnaire peut donner lieu à de multiples implémentations, en fonction de l'ordre dans lequel les agents prennent leur chambres. Par le théorème de Birkhoff-von Neumann, pour toute solution fractionnaire X, il existe $\lambda_1, \ldots, \lambda_k \in (0, 1]$, avec $\sum_t \lambda_t = 1$, et $\sigma_1, \ldots, \sigma_k$ des allocations déterministes, tel que X peut être décomposée de la manière suivante : pour tout agent $i \in A$ et $r_j \in R$, $\sum_{\{t | \sigma_t(i) = j\}} \lambda_t = x_{ij}$. Une implémentation précise l'ordre dans lequel les allocations déterministes d'une décomposition sont effectuées. Nous montrons que trouver une implémentation d'une allocation fractionnaire qui minimise le nombre total de changements de chambres est NP-difficile. De plus, même lorsque la décomposition est donnée, trouver une implémentation qui minimise le nombre total de changements de chambres est NP-difficile.

5 Discussion

Avec quelques simulations sur des données synthétiques, nous remarquons qu'en pratique B-EF et les solutions fractionnaires permettent d'accroître significativement la proportion d'instances admettant une solution équitable.

Identifier la complexité du problème de décision concernant l'existence de solutions B-EF en général apparaît comme un problème ouvert intéressant.

Références

[1] Gal, Ya'akov, Moshe Mash, Ariel D Procaccia et Yair Zick: *Which is the fairest (rent division) of them all ?* Journal of the ACM (JACM), 64(6) :1–22, 2017.

[2] Procaccia, Ariel, Rodrigo Velez et Dingli Yu: *Fair rent division on a budget*. Dans *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1177–1184, 2018.

Actes JIAF-JFPDA 2023

L'abus de comparaisons est mauvais pour la santé *

Emma Caizergues[1,2] **François Durand**[1] **Fabien Mathieu**[3]

[1] Nokia Bell Labs, Massy, France
[2] Université Paris-Dauphine, Paris, France
[3] Swapcard, Paris, France
emma.caizergues@nokia.com
francois.durand@nokia.com
fabien@swapcard.com

Résumé

Un *algorithme interruptible* (*anytime algorithm* en anglais) est un algorithme capable de renvoyer une estimation du résultat à chaque étape de son exécution. Dans cet article, nous nous intéressons au problème du *tri interruptible*. Nous considérons que chaque comparaison est une étape d'exécution de l'algorithme, et nous mesurons la proximité entre l'estimation et la liste triée à l'aide de la distance tau de Kendall. Nous présentons *Corsort*, une famille d'algorithmes de tris interruptibles reposant sur des estimateurs. Par simulation, nous montrons qu'un Corsort bien configuré a un temps de terminaison quasi-optimal, et fournit de meilleures estimations intermédiaires que les meilleurs tris dont nous avons connaissance.

Abstract

An *anytime algorithm* (*algorithme interruptible* in French) is an algorithm that is able to give an estimation of the result after each step of execution. In this article, we study the problem of *anytime sorting*. We consider that each comparison is a step of execution, and we measure the proximity between the estimation and the sorted list with the Kendall tau distance. We present *Corsort*, a family of anytime sorting algorithms using estimators. By simulation, we show that a well-configured Corsort has a quasi-optimal termination time, and gives better estimations than the other algorithms of our benchmark.

1 Introduction

Agathe, viticultrice dans le Haut-Rhin, possède n crus de Riesling qu'elle aimerait classer afin de mieux satisfaire les demandes de ses acheteurs et acheteuses. Chaque cru a une qualité distincte dont Agathe peut juger car elle est une experte : quand deux vins lui sont proposés, elle est capable d'identifier le meilleur, sans jamais faire d'erreur. Cependant, comme la perception de la qualité d'un vin est aussi complexe qu'éphémère, Agathe ne peut déguster que deux vins à la fois et ne peut pas comparer des vins d'une dégustation à l'autre. Autrement dit, Agathe ne peut pas insérer chaque nouveau vin qu'elle goûte dans une liste triée des vins déjà dégustés, mais doit comparer les crus deux à deux.

Pour éviter un excès de dégustations, Agathe souhaite effectuer un minimum de comparaisons. A minima, il faut qu'elle n'effectue pas deux fois la même comparaison. Également, elle est consciente que son palais peut être saturé à tout moment, rendant toute nouvelle dégustation impossible. Si elle doit s'arrêter avant d'avoir fini de trier tous les crus, elle souhaite avoir une bonne estimation du classement des vins. Elle cherche donc un algorithme de tri qui minimise le nombre de comparaisons tout en donnant une bonne estimation du résultat après chaque comparaison.

Contributions

Afin d'aider Agathe dans son classement, nous apportons les contributions suivantes :

- Nous formalisons le problème de la recherche d'un bon *tri interruptible par comparaisons*, dont la performance est mesurée par la distance tau de Kendall entre résultat provisoire et liste parfaitement triée ;
- Nous revisitons les algorithmes de tri classiques du point de vue *interruptible* ;
- Nous proposons des heuristiques simples pour estimer, à partir de l'ordre partiel correspondant à l'étape d'exécution en cours, le résultat final du tri (encore inconnu) ;

*Ce travail a été effectué au LINCS (https://www.lincs.fr/).

— Nous introduisons *Corsort*, une famille de tris interruptibles dont la logique repose sur des estimateurs ;
— Nous publions un paquet Python dédié à l'analyse de performance des tris interruptible par comparaisons ;
— Par simulation, nous montrons qu'un Corsort bien configuré a un temps de terminaison quasi-optimal, et fournit de meilleures estimations intermédiaires que les meilleurs tris dont nous avons connaissance.

Travaux connexes

Le classement de crus de Riesling s'inscrit dans le problème plus général du classement d'une liste par intervention humaine. Dans ce contexte, on considère généralement qu'attribuer une utilité à chaque élément de la liste n'est pas une technique fiable et qu'il vaut mieux procéder en comparant des paires [9]. Par exemple, le site web `https://www.pubmeeple.com/ranking-engine` propose une interface pour réaliser des classements par comparaisons successives. La comparaison humaine est beaucoup plus coûteuse que toute opération de base réalisée par ordinateur, et justifie que l'on sépare la complexité œnologique (en nombre de comparaisons) de la complexité globale [5, 17]. Des problèmes similaires apparaissent en dehors de toute intervention humaine dès que le coût de comparaison entre deux objets est prohibitifs. C'est par exemple le cas si les objets sont constitués de données massives et que chaque comparaison nécessite un transfert de données [15].

Si Agathe était sûre d'arriver au bout de la dégustation, on retrouverait un problème classique : trier en minimisant le nombre de comparaisons [6]. En plus d'algorithmes célèbres tels que le tri rapide, le tri fusion et le tri par tas (parmi bien d'autres), on peut citer l'algorithme de Ford-Johnson [8], extrêmement proche de la borne théorique en nombre de comparaisons, et même optimal pour certaines valeurs [16].

Une problème proche est celui *tri sous information partielle* : étant donné un ordre total qu'on ne connaît que partiellement (au sens où un ordre partiel compatible est connu), comment retrouver l'ordre total en un minimum de comparaisons [5] ? La plupart des variantes de problèmes de tri se déclinent *sous information partielle*, par exemple le classement des k meilleures valeurs [7]. Le problème du tri sous information partielle est à l'origine du principe de fonctionnement des tris *Corsort* proposés dans cet article.

La possible interruption de la dégustation nous place dans le domaine des *algorithmes interruptibles* (*anytime algorithms*), qui maintiennent à tout instant une estimation du résultat [18]. Étonnamment, les tris ont été peu étudiés dans cette littérature : les études ne concernent que le tri par sélection, le tri de Shell ou le tri rapide, sans introduire d'algorithme plus adapté, et les mesures d'écart au résultat final ne sont pas des distances [10, 12].

Parmi les problèmes connexes, les *algorithmes progressifs* peuvent aussi être interrompus à tout moment, mais l'accent est mis sur des bornes prouvables de performance en pire cas plutôt que sur l'efficacité moyenne empirique [1]. Les *algorithmes par contrat* opèrent également un compromis entre temps et précision, mais supposent que le temps disponible est connu à l'avance [18]. À l'inverse, le *tri approximatif* fixe un objectif en terme d'erreur maximale, et essaie de borner le nombre d'opérations nécessaire pour l'atteindre. L'algorithme ASort, sur lequel nous reviendrons dans la suite de cet article, donne des garanties de cette nature [9].

Le reste de l'article est organisé comme suit : la section 2 formalise les notions de tri interruptible et d'estimateur de classement ; la section 3 présente notre principale contribution, les tris *Corsort* ; la section 4 évalue la qualité des différentes solutions considérées au moyen de simulations ; la section 5 conclut.

2 Tris interruptibles

Formellement, nous voulons trier une liste $X = (X[1], \ldots, X[n])$, où $n > 0$, en effectuant des comparaisons du type : *est-ce que* $X[i] < X[j]$ *?* Un *tri interruptible* (*anytime sorting algorithm*) est un algorithme capable, à chaque étape k de son exécution, de renvoyer une estimation X_k du résultat. Dans notre modèle, chaque comparaison constitue une étape de l'algorithme [1]. On mesure la qualité de X_k par la distance tau de Kendall [14] entre X_k et la liste triée, c'est-à-dire par le nombre de paires d'éléments que l'estimation classe dans le mauvais sens : $\tau_k = |\{(i, j) : i < j, X_k[i] > X_k[j]\}|$. Idéalement, nous cherchons un tri interruptible dont le *profil de performance* $k \to \tau_k$, qui représente l'évolution de l'erreur commise au fur et à mesure de l'exécution de l'algorithme, est constamment plus bas que celui des autres algorithmes testés.

2.1 Tris classiques

Certains algorithmes classiques peuvent être vus comme interruptibles car ils maintiennent une liste courante qui converge vers la liste triée et peut servir d'estimation X_k. C'est par exemple le cas du tri rapide et du tri fusion, que nous avons tous deux implantés d'une manière favorable à l'esprit de l'algorithme initial : par exemple, pour le tri rapide, la position du pivot est mise à jour dans la liste après chaque comparaison.

Modifier l'ordre des comparaisons effectuées peut améliorer les estimations intermédiaires X_k. Pour le tri fusion, il est naturel de parcourir l'arbre de récursion en profondeur (DFS), c'est-à-dire en triant récursivement des parties gauche, puis droite de chaque sous-liste considérée. Cependant, on peut aussi le parcourir en largeur (BFS), en

1. Par convention, si l'algorithme termine en moins de k comparaisons, alors X_k est le résultat final, c'est-à-dire la liste triée.

DFS	BFS
(ab)	(ab)
(cd)	(cd)
$(abcd)$	(ef)
(ef)	(gh)
(gh)	$(abcd)$
$(efgh)$	$(efgh)$
$(abcdefgh)$	$(abcdefgh)$

TABLE 1 – Déroulement d'un tri fusion pour une liste de taille 8 ($abcdefg$), avec parcours récursif *classique* (DFS) ou *en largeur* (BFS). Chaque ligne représente une sous-liste que l'algorithme doit trier.

triant des sous-listes de taille croissante. La table 1 illustre la différence sur un exemple simple [2]. Les mêmes comparaisons sont faites dans les deux cas mais dans un ordre différent. L'ordre du parcours en largeur, en maintenant une répartition des comparaisons équilibrée sur l'ensemble des éléments de la liste, semble a priori plus favorable pour les estimations intermédiaires X_k (et nous le vérifierons expérimentalement).

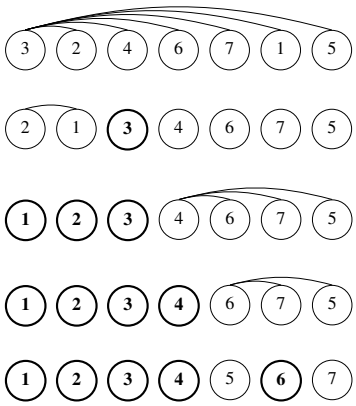


FIGURE 1 – Exécution du tri rapide sur la liste $X = (3246715)$. Chaque étape représente l'application complète d'un pivot. Les arêtes représentent les comparaisons effectuées. Un nœud apparaît en gras s'il a déjà été utilisé comme pivot : il partitionne alors la liste en éléments plus petits à gauche et plus grands à droite. On a omis les étapes sans aucune comparaison.

Pour le tri rapide, notre implémentation améliorée est équivalente à l'algorithme ASort [9], en utilisant la sélection rapide [11] comme sous-algorithme d'identification de la médiane. Le principe d'ASort est d'identifier la médiane de la liste et de séparer les éléments plus petits et les éléments plus grands, qu'on va ensuite trier récursivement de

2. Par concision, nous omettons les virgules pour noter les listes prises en exemples dans les tables et les figures. Par exemple, la liste (a, b, c, d) est notée $(abcd)$.

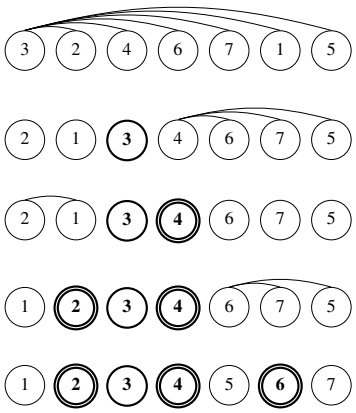


FIGURE 2 – Exécution d'ASort sur la liste $X = (3246715)$, avec la sélection rapide comme sous-algorithme de médiane. Chaque étape représente l'application complète d'un pivot par la sélection rapide. Un nœud apparaît en gras s'il a déjà été utilisé comme pivot, et doublement entouré si c'est une des médianes déjà trouvées. On a omis les étapes sans aucune comparaison. Lors de la troisième étape, ayant trouvé la médiane 4, on doit trouver la médiane de la sous-liste de gauche, (213) ; mais il est inutile d'effectuer toute comparaison avec l'élément 3 car celui-ci, précédemment utilisé comme pivot, est déjà à sa place définitive.

la même façon. Reste à décider comment on identifie la médiane. L'algorithme de sélection rapide, similaire au tri rapide, permet de la trouver par l'application de pivots successifs. Des exemples d'exécution du tri rapide et d'ASort sont donnés dans les figures 1 et 2. On constate que les comparaisons effectuées sont les mêmes, quoique dans un ordre différent, et il est facile de se convaincre que c'est toujours le cas.

D'autres algorithmes classiques permettent d'obtenir une estimation X_k par une transformation simple et naturelle de l'état courant. C'est le cas du tri par tas, qui conserve en mémoire le tas associé à la liste partiellement triée. Pour obtenir une estimation juste à l'égard de l'algorithme, il suffit donc de parcourir le tas à l'envers, puis les éléments déjà triés à l'endroit.

Enfin, pour certains algorithmes comme Ford-Johnson, il est difficile d'associer une estimation simple respectant « l'esprit » de l'algorithme. En effet, l'état courant dans l'algorithme de Ford-Johnson n'a pas une structure naturellement proche d'une liste : il n'existe aucune transformation simple qui puisse donner une estimation raisonnable. Comme il n'est pas toujours trivial de trouver une transposition naturelle de *l'idée* d'un algorithme en estimateurs intermédiaires, nous montrons dans la section suivante comment produire des X_k pour n'importe quel algorithme de tri.

2.2 Estimateurs

Pour rendre n'importe quel tri interruptible, nous proposons d'utiliser un estimateur qui ignore l'algorithme de tri utilisé et repose uniquement sur l'historique des comparaisons effectuées.

Notons $C_k = \{X[i_1] < X[j_1], \ldots, X[i_k] < X[j_k]\}$ le résultat de k comparaisons. C_k définit par clôture transitive un ordre partiel $\preceq_k$ sur les éléments de la liste [3]. Un *estimateur* est une fonction qui associe à tout ordre partiel un ordre total compatible.

Il existe un estimateur optimal qu'on peut résumer en deux étapes. D'abord, on considère l'ensemble des ordres totaux compatibles avec $\preceq_k$, appelés ses *extensions linéaires*. Ensuite, on trouve l'ordre qui minimise l'espérance de la distance tau de Kendall avec une extension linéaire aléatoire uniforme : autrement dit, on applique la *méthode de Kemeny* [13] au *profil de vote* constitué des extensions linéaires. Malheureusement, énumérer les extensions linéaires est un problème #P-complet [3], et appliquer la méthode de Kemeny à un profil de vote est aussi un problème NP-difficile [2]. Le coût de cette approche semble donc absolument prohibitif.

En pratique, pour construire un estimateur, nous allons dorénavant utiliser une fonction de *score* qui associe à chaque élément une valeur qui reflète sa position estimée dans la liste. L'estimation renvoyée est la liste associée au tri des scores. Rappelons que les complexités globale et œnologique sont distinctes : il est bien moins coûteux de trier n scores que de comparer deux crus. En cas d'égalité des scores, on renvoie les éléments dans leur ordre d'origine. Par un léger abus de langage, nous appelons aussi estimateur la fonction de score utilisée.

Une première idée est d'associer à chaque élément x un score correspondant à sa position moyenne $p_k(x)$ dans les extensions linéaires de $\preceq_k$. C'est bien un estimateur : si un élément est plus grand qu'un autre dans l'ordre partiel $\preceq_k$, alors il le sera dans toutes ses extensions linéaires, donc aussi selon la position moyenne p_k. Autrement dit, p_k renvoie toujours un ordre total cohérent avec l'ordre partiel. En outre, renvoyer une extension linéaire « moyenne » assure une distance raisonnable à la liste triée. Malheureusement, le coût de l'estimateur p_k reste prohibitif en complexité globale, puisque c'est également un problème #P-complet [3].

Nous proposons donc d'introduire des fonctions de score heuristiques plus simples à calculer pour produire des estimations X_k raisonnables. Pour cela, si x est un élément de la liste, on note $d_k(x) = |\{y \in X : y \preceq_k x\}|$ et $a_k(x) = |\{y \in X : x \preceq_k y\}|$ respectivement le nombre de descendants et d'ancêtres connus de x (lequel est inclus dans les deux ensembles par convention). Une manière simple de calculer d_k et a_k est de partir de C_k et de construire sa clôture transitive, pour un coût en $O(n^2)$.

Nous avons considéré les fonctions de scores suivantes :

— Δ_k, définie par $\Delta_k(x) = d_k(x) - a_k(x)$. Δ_k attribue à chaque x un score qui reflète la moyenne entre sa plus basse et sa plus haute positions possibles.

— ρ_k, définie par $\rho_k(x) = d_k(x)/(d_k(x) + a_k(x))$. Cela revient à positionner x comme si ses descendants et ses ancêtres avaient en moyenne des positions régulièrement espacées au sein de l'ensemble de la liste.

S'il n'y a pas d'ambiguïté, nous omettrons par la suite l'indice k. Tout comme p, Δ et ρ sont des estimateurs : il est facile de vérifier que si un élément est plus grand qu'un autre dans l'ordre partiel, alors il aura aussi un plus grand score [4].

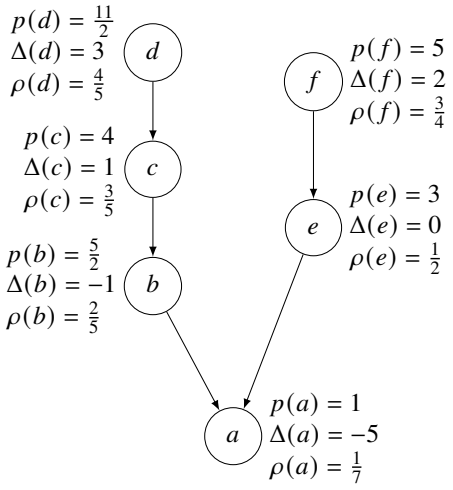


FIGURE 3 – Exemple de sélection d'extension linéaire. Ici, les estimateurs p, Δ et ρ renvoient la même estimation $(abecfd)$, en accord avec la méthode de Kemeny.

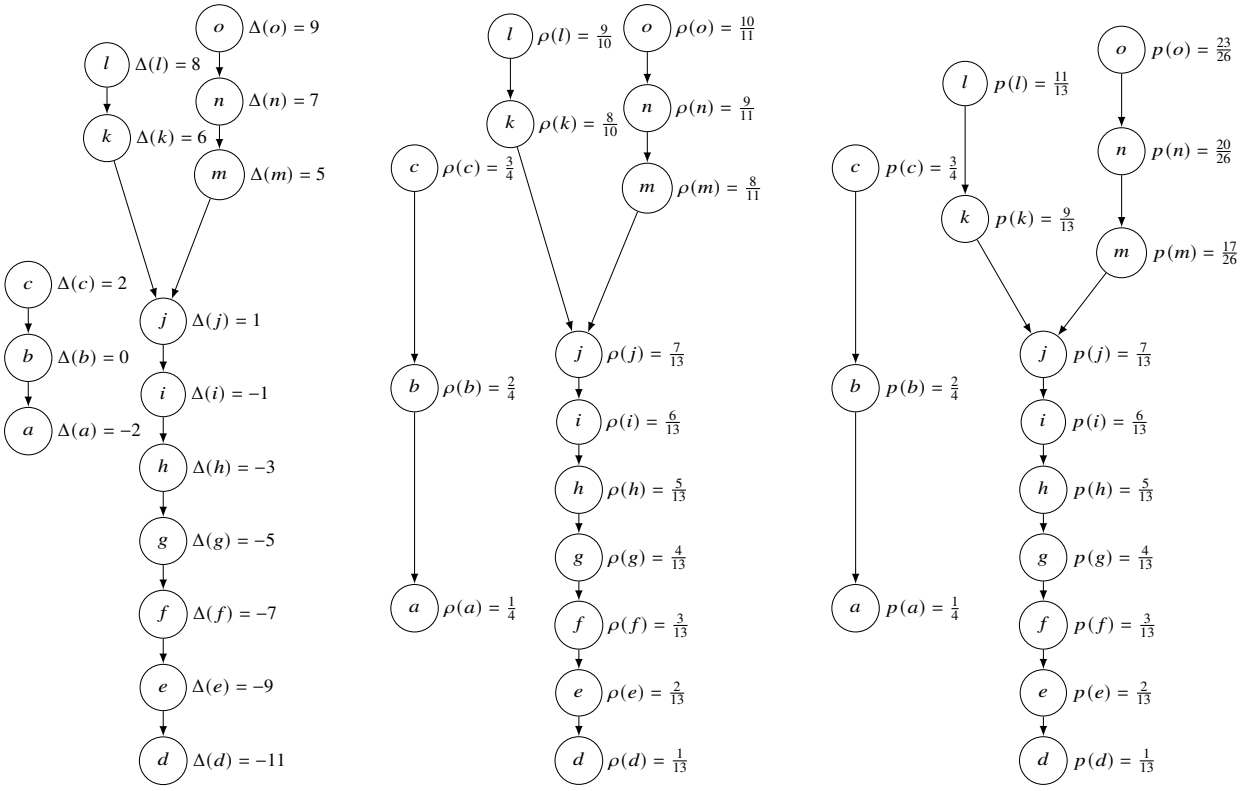
La figure 3 montre sur un exemple simple le résultat des différents estimateurs. Dans ce cas, les trois estimateurs Δ, ρ et p sont optimaux, au sens où ils renvoient le même ordre total que la méthode de Kemeny.

La figure 4 exhibe un cas plus complexe où les quatre estimateurs donnent des résultats distincts. Ici, ρ est meilleur que Δ, notamment car il positionne mieux les éléments de la petite composante connexe (nœuds a à c) par rapport à ceux de la grande (nœuds d à o). Il reste cependant surpassé par l'indicateur p, lui-même moins performant que l'optimum donné par la méthode de Kemeny.

En Section 3, lors de l'optimisation de notre tri Corsort, nous verrons que de manière générale, ρ semble être un

3. D'un point de vue purement formel, quitte à étiqueter chaque élément x par son indice i dans la liste initiale et à noter le résultat obtenu (x, i), on peut supposer que tous les éléments sont distincts. Ainsi, lors du tri de la liste $(17, 42, 42)$, il est possible de se trouver dans une situation où on sait déjà que $(17, 1) \preceq_k (42, 2)$ mais où on ne sait pas encore comparer $(17, 1)$ et $(42, 3)$.

4. En particulier, quand on connaît le résultat de toutes les comparaisons, classer les éléments selon l'estimateur Δ ou ρ renvoie bien la liste triée.



(a) Scores Δ. (b) Scores ρ. (c) Scores p.

Estimateur	Estimation renvoyée	$\overline{\tau}$
Δ	$(defghaibjcmknlo)$	10,0
ρ	$(defaghibjmcknlo)$	8,91
p	$(defaghibjmkcnlo)$	8,66
Kemeny (optimal)	$(deafghibjmknclo)$	8,61

FIGURE 4 – Exemple de sélection d'extension linéaire par attribution de score. L'estimateur p (à droite) est normalisé par $n+1 = 16$ afin de faciliter la comparaison avec ρ (au centre). Le tableau regroupe les estimations renvoyées et l'espérance $\overline{\tau}$ de l'erreur τ. La méthode de Kemeny (optimale) n'attribue pas de score aux éléments et n'a pas de figure associée, mais est également donnée pour comparaison. La position verticale de chaque nœud est proportionnelle à la valeur de l'estimateur considéré (Δ, ρ ou p).

meilleur estimateur que Δ. Mais nous verrons également que Δ présente un autre type d'intérêt.

3 Tris orientés comparaisons (Corsort)

Nous appelons tri *Corsort* (*Comparison-ORiented Sort*) un tri qui, à chaque étape de son exécution, sélectionne la comparaison suivante en fonction de l'ordre partiel courant $\preceq_k$. On suppose qu'on choisit toujours des paires non comparables selon $\preceq_k$. Cela assure de ne jamais effectuer deux fois la même comparaison, donc de terminer en au plus $n(n-1)/2$ étapes, et plus généralement de ne jamais effectuer des comparaisons déductibles par transitivité.

Le cœur d'un corsort, c'est-à-dire le choix de la prochaine comparaison, vise deux objectifs. D'une part, il doit assurer une terminaison rapide, c'est-à-dire minimiser le nombre total de comparaisons. C'est un problème de *tri sous information partielle*, qui revient à choisir une comparaison dont les deux issues sont aussi équiprobables que possible [5]. À cette fin, nous proposons d'utiliser un estimateur basé sur une notion de score et de comparer des éléments dont les scores sont proches. En effet, cela indique une grande incertitude quant au résultat de la comparaison, ce qui rapproche de l'équiprobabilité souhaitée. D'autre part, pour améliorer

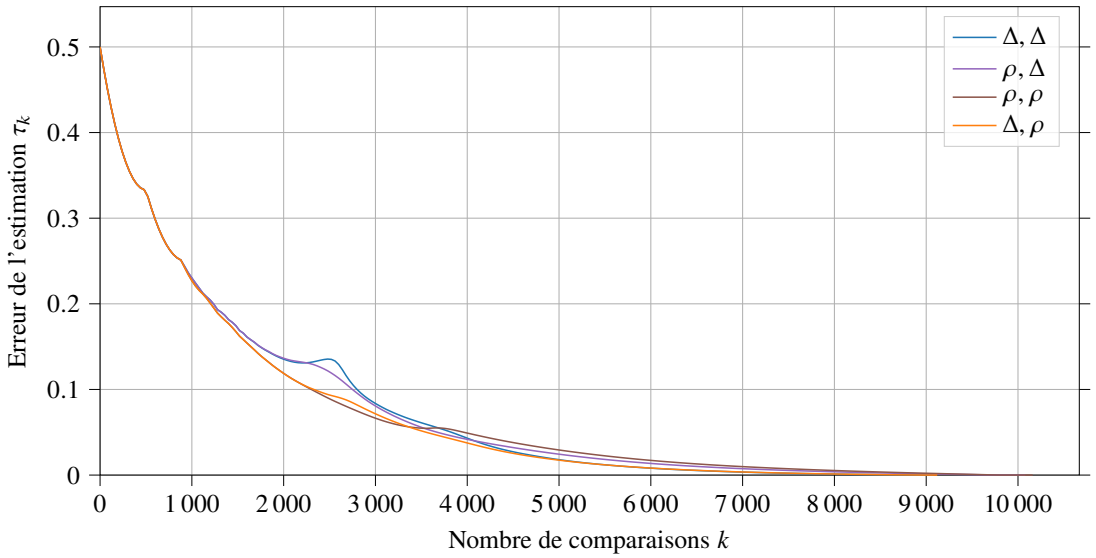


FIGURE 5 – Profils de performances de tris Corsort pour $n = 1\,000$. Chaque courbe est obtenue en triant 10 000 listes aléatoires. Pour chaque valeur de k, on calcule l'erreur τ_k médiane, normalisée par $n(n-1)/2$. En légende, chaque Corsort est défini par le critère principal à minimiser pour déterminer la prochaine comparaison, puis l'estimateur utilisé pour renvoyer X_k.

Algorithme 1 : Corsort sélectionné

$X \leftarrow$ une liste de n éléments munie d'un ordre total
$k \leftarrow 0$
$\preceq_k \leftarrow$ l'ordre partiel vide sur X
tant que $\preceq_k$ *est incomplet et pas d'interruption* **faire**
  $i, j \leftarrow \arg\min_{i,j \text{ non comparables dans } \preceq_k} \left(|\Delta_k(i) - \Delta_k(j)|, \max(I_k(i), I_k(j))\right)$
  $\preceq_{k+1} \leftarrow$ clôture transitive de $\preceq_k$ augmenté de la comparaison i, j
  $k \leftarrow k + 1$
retourner X *trié selon l'estimateur* ρ_k

la qualité des estimations intermédiaires X_k, il faut acquérir de l'information sur les éléments pour lesquels on en a peu, car ces derniers créent une incertitude qui va se répercuter sur τ_k. Pour représenter la quantité d'information acquise sur un sommet x, on introduit donc $I_k(x) = a_k(x) + d_k(x)$, et on souhaite comparer des éléments pour lesquels I_k est faible. Empiriquement, nous avons constaté que le premier objectif devait être prioritaire sur le second et avons opté pour une sélection lexicographique. En résumé :

— Le choix de la prochaine comparaison se fait en cherchant à minimiser l'écart des scores selon un estimateur (Δ ou ρ dans nos expériences).
— En cas d'égalité, on cherche à comparer une paire (x, y) pour laquelle $I_k(x)$ et $I_k(y)$ sont faibles. À cette fin, nous cherchons une paire qui minimise $\max(I_k(x), I_k(y))$.
— Les valeurs de X_k sont données par un estimateur, Δ ou ρ, qui n'est pas nécessairement le même que pour choisir la prochaine comparaison.

Afin de déterminer la meilleure combinaison possible, nous avons développé un paquet Python pour créer des tris interruptibles et mesurer leurs performances [4]. Ce paquet permet, entre autres choses, d'étudier le déroulement d'un algorithme de tri. Nous avons ensuite testé les différentes variantes de Corsort par simulation. La figure 5 montre les profils de performance médians obtenus pour $n = 1\,000$. On observe que tous les Corsorts utilisant l'estimateur Δ présentent une « bosse » entre $k = 2\,000$ et $k = 3\,000$, en particulier quand la prochaine comparaison est aussi déterminée par Δ. Cela nous amène à choisir ρ pour les estimations X_k (nous avons vérifié que ρ est aussi globalement plus performant pour les algorithmes de tris classiques). À l'inverse, Δ est presque toujours plus intéressant que ρ pour choisir la prochaine comparaison, en particulier dans la deuxième moitié de l'exécution du tri [5].

5. Non montré par souci de lisibilité de la figure 5 : pour le critère secondaire de choix de la prochaine comparaison, à la place de $\max(I_k(x), I_k(y))$, on peut utiliser $I_k(x) + I_k(y)$. Notre choix d'uti-

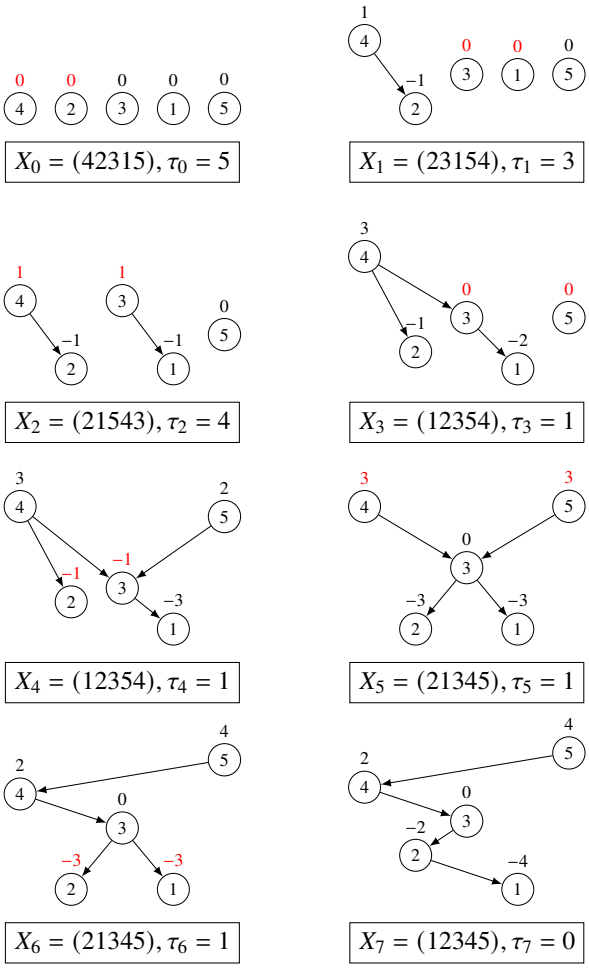


FIGURE 6 – Exécution du Corsort sélectionné sur la liste $X = (42315)$. Chaque étape k montre l'ordre partiel courant après k comparaisons, Δ_k (indiqué au-dessus de chaque élément), ρ_k (représenté par sa hauteur), l'estimation X_k renvoyée et l'erreur τ_k. À chaque étape, la prochaine comparaison se fera entre les deux sommets dont les valeurs de Δ_k sont en rouge.

Nous avons donc choisi le Corsort décrit par l'algorithme 1 : parmi les paires encore incomparables, choisir la paire (x, y) qui minimise lexicographiquement $\big(|\Delta_k(x) - \Delta_k(y)|, \max(I_k(x), I_k(y))\big)$; et utiliser ρ comme estimateur. À titre d'exemple, la figure 6 montre l'exécution du Corsort sélectionné sur la liste $X = (42315)$. En l'occurrence, on retrouve la quasi-monotonie du profil de performance déjà constaté de manière plus générale dans la figure 5.

liser le maximum améliore la performance mais de manière marginale.

4 Évaluation

À présent, nous souhaitons comparer notre tri Corsort avec les algorithmes classiques présentés en section 2.1. Notre méthodologie d'évaluation est la suivante : pour une valeur de n donnée, nous tirons 10 000 listes aléatoires et calculons le nombre de comparaisons nécessaires pour un tri complet, ainsi que les profils de performance $k \to \tau_k$. Pour donner un aperçu de la distribution des résultats, nous traçons pour chaque algorithme la médiane (courbe foncée), les quantiles de 25% à 75% (zone claire), et les quantiles de 2,5% à 97,5%, qui représentent un intervalle de confiance à 95% (zone très claire).

La figure 7 montre le temps de terminaison (en nombre de comparaisons) pour des valeurs de n allant de 8 à 1024 et les tris suivants : par tas, rapide, Corsort, fusion et Ford-Johnson. L'axe des ordonnées montre l'écart relatif par rapport à la borne inférieure théorique $n \log_2(n) - n/\ln(2) + \log_2(2\pi n)/2$ [6] : plus une courbe est proche de 0, plus elle est proche de l'optimum.

Nous observons que le tri par tas effectue presque deux fois plus de comparaisons que nécessaire. Le tri rapide est meilleur (moins de 30% de surcoût) mais avec une grande variance : pour $n = 1024$, l'intervalle de confiance à 95% va de 17% à 46%. Cette grande variance est due au choix du pivot, dont la valeur influe grandement sur la rapidité de l'algorithme. À chaque étape, plus le pivot est proche de la médiane (de la sous-liste courante considérée), plus l'algorithme sera rapide. Les trois tris restants ont un surcoût encore plus faible (5% pour Corsort, 2% pour le tri fusion, 0,3% pour Ford-Johnson) et une variance négligeable. Nous concluons que Corsort est un bon candidat puisqu'il n'est battu que par des algorithmes dont la terminaison est asymptotiquement optimale, i.e. équivalente à $n \log_2(n)$ [8].

La figure 8 montre les profils de performance des quatre meilleurs tris de la figure 7 : tri rapide, Corsort, tri fusion, et algorithme de Ford-Johnson. Ce dernier, tout comme Corsort, utilise l'estimateur ρ. Pour les tris rapide et fusion, deux variantes sont utilisées : la version de base avec son estimation naturelle (l'état courant de la liste), et une version améliorée (respectivement fusion-BFS et ASort) munie de l'estimateur ρ.

Premièrement, on voit que le tri rapide et Asort ont une très grande variance. Ceci s'explique encore par le système de pivot, qui influe sur la structure des comparaisons effectuées, et donc sur la qualité de l'estimation. Les variances des tris fusion, fusion-BFS muni de ρ et Ford-Johnson muni de ρ sont assez faibles, tout comme Corsort, dont la variance est quasiment négligeable. Ces tris sont donc plus robustes que le tri rapide et Asort.

Ensuite, on remarque que l'utilisation de l'estimateur ρ améliore le profil de performance [6]. Pour le tri fusion, cela

6. Non montré par souci de lisibilité de la figure : Δ améliore également le profil des tris classiques, mais est généralement moins performant que ρ.

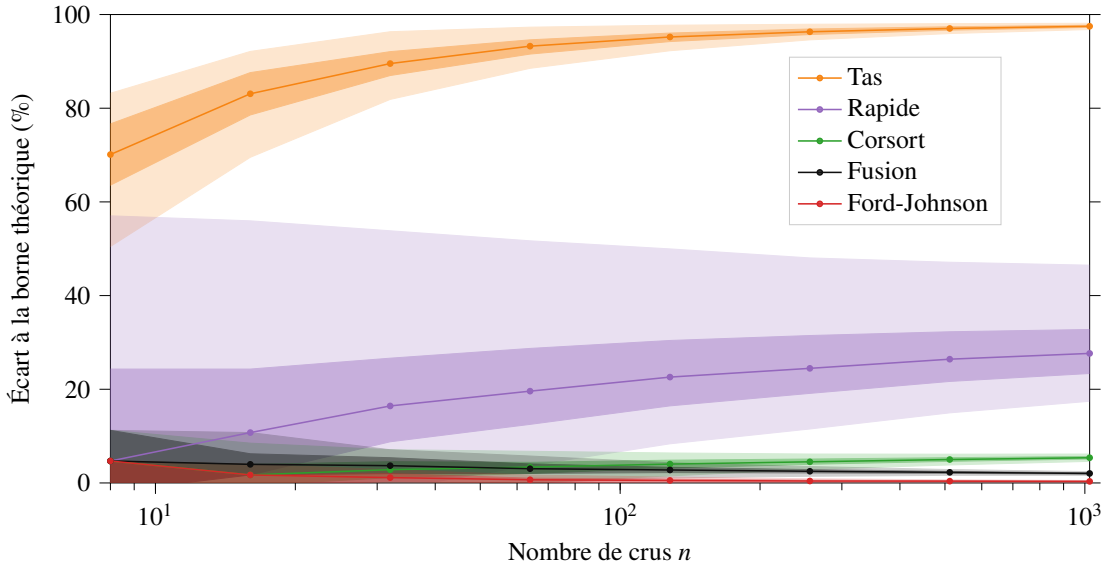


FIGURE 7 – Nombre de comparaisons pour terminer l'algorithme, exprimé en écart par rapport à la borne théorique. Chaque point est obtenu en triant 10 000 listes aléatoires.

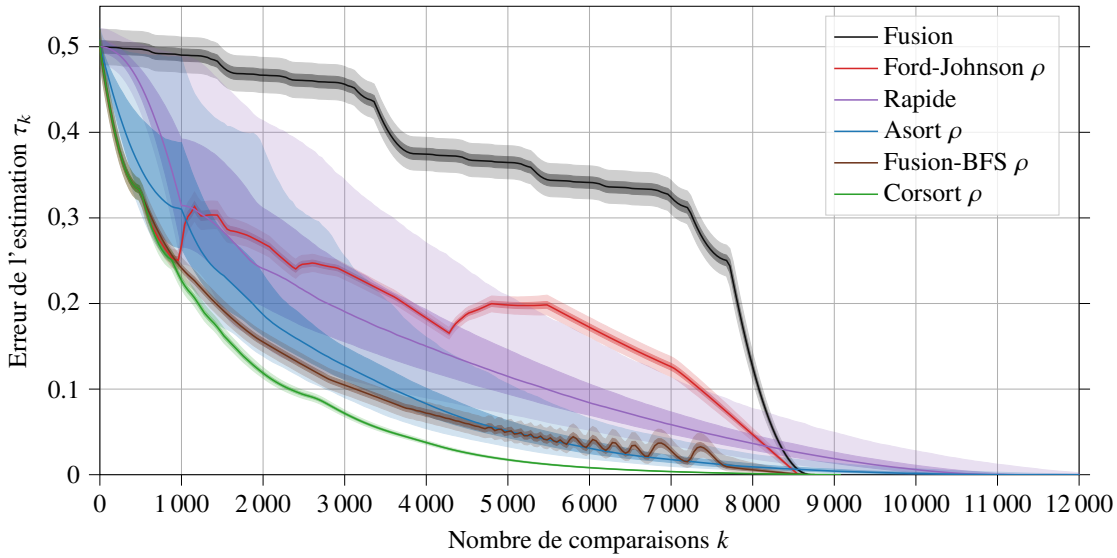


FIGURE 8 – Profils de performance de tris interruptibles pour $n = 1000$. Chaque courbe est obtenue en triant 10 000 listes aléatoires. L'erreur τ est normalisée par $n(n-1)/2$.

se fait au prix d'un comportement non-monotone en fin de parcours [7]. Cependant, grâce à une implémentation BFS et à notre estimateur, le profil de performance du tri fusion amélioré reste presque constamment en dessous du tri rapide, que celui-ci soit en version naïve ou améliorée (Asort). En plus d'améliorer les estimations, l'estimateur ρ réduit aussi la variance (qui reste tout de même conséquente pour Asort).

Enfin, il ressort la supériorité du profil de Corsort : il est presque tout le temps monotone, avec une très faible variance, et il est constamment sous les autres à part en terminaison (il termine un peu plus tard que fusion ou Ford-

7. La non-monotonie du tri fusion-BFS muni de ρ se produit au moment de la fusion des plus grandes sous-listes, ce qui donne lieu à des ordres partiels similaires à l'exemple de la figure 4, avec des composantes connexes en forme de chaîne et d'autre en forme de Y. Les deux dernières « bosses » de la figure 8 correspondent aux fusions des quarts de liste ; les quatre précédentes, aux huitièmes de listes. La dernière fusion n'occasionne pas de non-monotonie car l'estimateur ρ est performant sur une configuration en Y si celle-ci n'est pas accompagnée d'une chaîne à côté.

Johnson). Corsort est donc un excellent tri interruptible que nous recommandons à Agathe pour trier ses crus de Riesling. Enfin, on peut constater la relative bonne performance du tri fusion-BFS muni de ρ, qui peut être un choix intéressant si l'on désire un tri interruptible qui soit également rapide, avec une terminaison, hors estimateur, en $O(n\log(n))$ opérations totales (pas seulement en nombre de comparaisons).

5 Conclusion et travaux futurs

Nous avons étudié des tris interruptibles générant un minimum de comparaisons. Nous avons proposé une méthode pour rendre tout tri interruptible, avec interruption possible après chaque comparaison. Nous avons introduit Corsort, une famille de tris à base d'estimateurs. Par simulation, nous avons montré qu'un tri Corsort bien configuré a un temps de terminaison (en nombre de comparaisons) quasi-optimal et possède un profil de performance meilleur que les meilleurs tris dont nous avons connaissance. Nous avons aussi montré que munir les tris classiques de nos estimateurs améliore leurs profils de performance.

La relative nouveauté de notre approche laisse place à plusieurs pistes de réflexion pour des travaux futurs.

D'abord, il sera intéressant de réaliser des nouvelles simulations avec d'autres fonctions de score pour essayer d'améliorer la terminaison de notre tri Corsort et/ou son profil de performance. Il sera aussi souhaitable de trouver des fonctions de score qui améliorent encore les profils de performance des tris classiques, en particulier qui tendent à rendre monotones l'algorithme de Ford-Johnson et le tri fusion.

À terme, on voudra essayer de répondre à plusieurs questions encore ouvertes. Le comportement en pire cas du tri Corsort choisi est-il en $O(n\log(n))$? Quelle est la borne théorique du profil de performance ? Peut-on trouver d'autres estimateurs raisonnables pour qu'un tri Corsort s'en approche ?

Références

[1] Alewijnse, S. P. A., T. M. Bagautdinov, M. de Berg, Q. W. Bouts, A. P. ten Brink, K. Buchin et M. A. Westenberg: *Progressive Geometric Algorithms*. Dans *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 50–59, Kyoto Japan, juin 2014. ACM, ISBN 978-1-4503-2594-3. https://dl.acm.org/doi/10.1145/2582112.2582156.

[2] Bartholdi, John, Craig A Tovey et Michael A Trick: *Voting schemes for which it can be difficult to tell who won the election*. Social Choice and welfare, 6 :157–165, 1989.

[3] Brightwell, Graham et Peter Winkler: *Counting linear extensions is #P-complete*. Dans *Proceedings of the twenty-third annual ACM symposium on Theory of computing - STOC '91*, pages 175–181, New Orleans, Louisiana, United States, 1991. ACM Press, ISBN 978-0-89791-397-3. http://portal.acm.org/citation.cfm?doid=103418.103441.

[4] Caizergues, Emma, François Durand et Fabien Mathieu: *Corsort : Comparaison ORiented Sort*. https://emczg.github.io/corsort/, 2023.

[5] Cardinal, Jean, Samuel Fiorini, Gwenaël Joret, Raphaël M. Jungers et J. Ian Munro: *Sorting under partial information (without the ellipsoid algorithm)*. Dans *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 359–368, Cambridge Massachusetts USA, juin 2010. ACM, ISBN 978-1-4503-0050-6. https://dl.acm.org/doi/10.1145/1806689.1806740.

[6] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest et Clifford Stein: *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009, ISBN 9780262533058.

[7] Dushkin, Eyal et Tova Milo: *Top-k Sorting Under Partial Order Information*. Dans *Proceedings of the 2018 International Conference on Management of Data*, pages 1007–1019, Houston TX USA, mai 2018. ACM, ISBN 978-1-4503-4703-7. https://dl.acm.org/doi/10.1145/3183713.3199672.

[8] Ford, Lester R. et Selmer M. Johnson: *A Tournament Problem*. The American Mathematical Monthly, 66(5) :387–389, mai 1959, ISSN 0002-9890, 1930-0972. https://www.tandfonline.com/doi/full/10.1080/00029890.1959.11989306.

[9] Giesen, Joachim, Eva Schuberth et Miloš Stojaković: *Approximate Sorting*. Fundam. Inf., 90(1–2) :67–72, jan 2009, ISSN 0169-2968.

[10] Grass, Joshua et Shlomo Zilberstein: *Anytime Algorithm Development Tools*. SIGART Bulletin Special Issue on Anytime Algorithms and Deliberation Scheduling, 7(2) :20–27, 1996. http://rbr.cs.umass.edu/shlomo/papers/GZsigart96.html.

[11] Hoare, C. A. R.: *Algorithm 65 : find*. Communications of the ACM, 4(7) :321–322, 1961.

[12] Horvitz, Eric: *Reasoning Under Varying and Uncertain Resource Constraints*. Morgan Kaufmann Publishers, janvier 1988. https://www.microsoft.com/en-us/research/publication/reasoning-under-varying-and-uncertain-resource-

[13] Kemeny, John G: *Mathematics without numbers*. Daedalus, 88(4) :577–591, 1959.

[14] Kendall, M. G.: *A New Measure of Rank Correlation*. Biometrika, 30(1/2) :81, juin

1938, ISSN 00063444. https://www.jstor.org/stable/2332226?origin=crossref.

[15] Mesrikhani, Amir et Mohammad Farshi: *Progressive sorting in the external memory model*. The CSI Journal on Computer Science and Engineering, 15(2), 2018.

[16] Peczarski, Marcin: *New Results in Minimum-Comparison Sorting*. Algorithmica, 40(2) :133–145, octobre 2004, ISSN 0178-4617, 1432-0541. http://link.springer.com/10.1007/s00453-004-1100-7.

[17] Peters, Dominik et Ariel D Procaccia: *Preference Elicitation as Average-Case Sorting*. Dans *Proceedings of the AAAI Conference on Artificial Intelligence*, tome 35, pages 5647–5655, 2021.

[18] Zilberstein, Shlomo: *Using Anytime Algorithms in Intelligent Systems*. AI Magazine, 17(3) :73, mars 1996. https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1232, Section : Articles.

Actes JIAF-JFPDA 2023

Différentiation des modalités du Bien : au-delà de l'optimalité de Pareto

Guillaume Gervois Gauvain Bourgne Marie-Jeanne Lesot

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
{prenom.nom}@lip6.fr

Résumé

L'éthique computationnelle étudie les restrictions et les préférences éthiques à intégrer aux algorithmes de prise de décision. Une approche pour faire face à une critique commune envers l'approche utilitariste de l'éthique computationnelle consiste à introduire des modalités différenciées du Bien, où les modalités sont définies comme les valeurs philosophiques qui correspondent aux différentes composantes du Bien. La différenciation permet alors qu'aucune modalité ne puisse en compenser une autre en définissant des classes distinctes de modalités. L'optimalité de Pareto modélise un cas extrême de différenciation, où chaque modalité constitue sa propre classe. Cet article propose une nouvelle approche, ordinale, pour traiter les modalités différenciées : la différenciation est modélisée par un ordre partiel strict sur les modalités, qui exprime quelles modalités prévalent sur les autres. L'article propose une axiomatisation de la supériorité pour prendre en compte ces comparaisons de modalités dans la détermination des actions éthiques : il discute de la manière d'induire une relation de préférence éthique entre les actions possibles, basée sur l'ordre partiel entre les modalités. En outre, il étudie les propriétés de cette relation induite, établissant qu'elle est asymétrique et transitive, prouvant ainsi qu'elle constitue une relation d'ordre.

Abstract

Computational ethics studies the ethical restrictions and preferences to be embedded into decision-making algorithms. One approach to deal with a common criticism towards the utilitarian approach to computational ethics consists in introducing differentiated modalities of the Good, where modalities are defined as philosophical values that correspond to different components of the Good. Differentiation then does not allow that any modality can compensate for any other one, distinct classes of modalities are defined. Pareto optimality models an extreme case of differentiation, where each modality constitutes its own class. This paper proposes a new, ordinal, approach to deal with differentiated modalities: differentiation is modelled by a strict partial order on the modalities, that expresses which modalities supersede others. The paper proposes an axiomatisation of superiority, to take into account these declared modality comparisons in the determination of ethical actions: it discusses how to derive an ethical preference relation between the possible actions, based on the partial order between the modalities. In addition, it studies the properties of this induced relation, establishing it is asymmetric and transitive, thus proving it constitutes a sound order relation.

1 Introduction

Les outils de prise de décision automatique sont de plus en plus répandus et utilisés. Face à cette popularité, on observe une demande croissante pour de nouveaux outils respectant les lois et les principes éthiques, c'est-à-dire vérifiant les contraintes de *conformité éthique*. Le domaine en pleine expansion de l'éthique computationnelle [1, 12] cherche à répondre à ces demandes. De nombreux principes éthiques proposés par des philosophes peuvent aider les informaticiens à aborder la question de la conformité éthique des algorithmes. L'utilitarisme, promu par Bentham et Mill à la fin du 18ème siècle, est l'une des théories morales les plus célèbres, mais aussi l'un des principes éthiques les plus implémentés [3, 10]. D'un point de vue computationnel, le principe utilitariste est séduisant car il est facilement représentable : il quantifie le Bien par des valeurs numériques, nommées utilités, qui peuvent ensuite être additionnées. Cependant, ce principe fait l'objet de débats philosophiques, notamment parce qu'il considère les différentes *modalités du Bien* comme étant toutes *indifférenciées*. Le terme *modalité* fait référence, ici et dans cet article, aux différentes valeurs philosophiques qui permettent de définir le Bien.

Prenons l'exemple d'une médecin dans un hôpital pour illustrer le fait que l'utilitarisme suppose que les modalités sont indifférenciées. Elle a le choix entre soigner un patient, ce que l'on note dans la suite *treat_patient* et qui aura pour effet de sauver une vie, et distribuer des chocolats à un

grand nombre de patients, noté *distribute_chocolat* et qui aura simplement pour effet de leur faire plaisir. Cet exemple confronte deux modalités : la vie humaine et le plaisir de manger du chocolat, notées respectivement *human_life* et *choco_pleasure*. Si l'on considère un nombre suffisamment important de patients, la somme des utilités attribuées au plaisir de manger du chocolat dépassera l'utilité attribuée au fait de sauver une vie, quelle que soit la valeur de cette dernière. L'utilitarisme conclut donc que la médecin doit distribuer du chocolat plutôt que de soigner le patient. Un tel cas montre que toute modalité peut être compensée par une autre : l'utilitarisme ne permet pas de modéliser la nature conflictuelle des modalités.

Les principales critiques de cette hypothèse d'indifférence font appel à une différenciation des modalités [8]. On peut par exemple considérer que le statut de médecin oblige à se préoccuper de la vie des patients plutôt que du plaisir de manger du chocolat, on peut aussi considérer que la vie humaine est plus importante que le plaisir de manger du chocolat. Cette dernière option introduit une notion de supériorité entre les modalités en accordant à certaines d'entre elles un statut particulier [5] : les modalités supérieures doivent être considérées en premier lorsqu'une décision doit être prise.

Suivant ces remarques, cet article propose une nouvelle approche, ordinale, pour traiter de la différentiation des modalités dans un système de conformité éthique : à notre connaissance, il propose une première tentative de relier cette préoccupation philosophique aux préférences ordinales. Plus précisément, il considère que la notion de supériorité est exprimée par un ordre partiel strict sur les modalités et il propose une *axiomatisation* de la supériorité, formalisant la prise en compte ces comparaisons de modalités afin d'en déduire des préférences ordinales entre les actions.

Le principe proposé peut être vu comme un principe de décision multicritère, où chaque modalité constitue un critère, allant au-delà du principe d'optimalité de Pareto : ce dernier, d'abord appliqué à des problèmes de prise de décision et ensuite à des problèmes éthiques [10], peut être considéré comme un cas extrême de différenciation des modalités. En effet, les modalités ne sont comparées qu'à elles-mêmes, et non les unes aux autres. Dans l'exemple médical précédent, aucune action n'est considérée comme dominant éthiquement l'autre : pour le principe de Pareto, les modalités sont incomparables entre elles. L'approche de supériorité que nous proposons et généralise le principe de Pareto en ajoutant la comparaison de supériorité des modalités.

L'article est structuré comme suit. La section 2 propose une formalisation du problème de conformité éthique afin de représenter les principes utilitariste et de Pareto, ainsi que la notion de comparaisons de modalité. La section 3 présente l'axiomatisation proposée de la supériorité qui prend en compte ces comparaisons pour déterminer une relation de préférence éthique entre les actions possibles. La section 4 étudie les propriétés de la relation induite proposée, établissant qu'elle constitue une relation d'ordre, prouvant qu'elle est asymétrique et transitive. La section 5 discute les hypothèses faites sur les relations de comparaison de modalité, au-delà du cas asymétrique et transitif. La section 6 conclut l'article et discute de certaines directions pour des travaux futurs.

2 Formalisation de la conformité éthique

Cette section décrit le formalisme considéré pour représenter un problème de conformité éthique, en présentant d'abord le cadre ordinal considéré et les notations utilisées tout au long de l'article. Elle introduit ensuite la représentation de la différenciation des modalités par un ordre partiel strict et montre enfin comment les principes utilitariste et de Pareto classiques sont exprimés dans ce cadre.

2.1 Formalisation ordinale de la conformité éthique

Un problème éthique consiste à sélectionner, parmi un ensemble $\mathcal{A}$ d'actions possibles (par exemple les options de soigner un patient ou distribuer du chocolat), l'ensemble $\mathcal{A}_p$ des *actions permissibles*, définies comme celles qu'il est éthiquement acceptable de réaliser selon un principe éthique donné. Dans l'article, les lettres a, a', o et o' seront utilisées pour représenter les éléments de $\mathcal{A}$.

Parmi les principes éthiques proposés par les philosophes et ceux qui ont été implémentés en éthique computationnelle, on retrouve l'utilitarisme de l'acte [13]. C'est une version courante de l'utilitarisme que l'on peut décomposer en trois étapes. Premièrement, les conséquences des actions sont éthiquement quantifiées par une *valeur d'utilité*. Dans la deuxième étape, ces valeurs d'utilité sont agrégées pour chaque action afin d'obtenir un nombre représentant l'utilité globale produite par l'action. Dans la dernière étape, les actions permissibles sont définies comme étant celles qui maximisent l'utilité.

Ces étapes peuvent être formalisées comme suit. Chaque action est représentée par un vecteur composé des valeurs d'utilité. Chaque valeur du vecteur correspond à une *modalité*, c'est-à-dire à l'une des valeurs philosophiques qui permettent de définir le Bien (par exemple la vie humaine ou le plaisir du chocolat). On note $\mathcal{M}$ l'ensemble fini des modalités et on considère que $\mathcal{A} \subset \mathbb{R}^{|\mathcal{M}|}$: plus la valeur du vecteur est élevée, plus l'action est intéressante du point de vue éthique selon cette modalité. Si l'action possède une valeur non nulle pour une modalité, on dit que l'action *porte* la modalité. Cette caractérisation des actions se situe dans le cadre usuel de la prise de décision multicritère [7], où les valeurs du vecteur peuvent être interprétées comme les

performances de l'action pour chacun des critères que sont les modalités.

Décrivons l'exemple présenté dans l'introduction avec ce formalisme.

Exemple 1. *Notons a l'action treat_patient et a' l'action distribute_chocolat. Considérons que sauver le patient a une valeur de* 10 *pour la modalité human_life, ainsi* $a_{human\_life} = 10$. *Ne procurant pas de plaisir aux patients, on a* $a_{choco\_pleasure} = 0$. *De même* $a'_{human\_life} = 0$. *Considérons que la distribution de chocolat procure* 1 *d'utilité et qu'il y a onze patients, soit* $a'_{choco\_pleasure} = 11$. *En utilisant la notation* $a = (a_{human\_life}, a_{choco\_pleasure})$, *cet exemple définit* $a = (10, 0)$ *et* $a' = (0, 11)$.

Cette quantification du Bien des conséquences est discutable : elle masque les relations causales en attribuant une seule valeur par modalité pour toutes les conséquences. On peut le voir directement avec l'action de distribuer du chocolat dans l'exemple 1. L'action telle qu'elle a été décrite cause un petit plaisir pour chacun des patients séparément. Elle possède donc un grand nombre de conséquences qui sont toutes valuées positivement pour la modalité *choco_pleasure*. Le formalisme proposé considère donc qu'une étape antérieure d'agrégation a déjà eu lieu, comme une somme pour l'utilitarisme de l'acte, afin de déterminer l'unique valeur de l'action *distribute_chocolat* pour la modalité *choco_pleasure*. En choisissant une autre fonction d'agrégation, il est possible de proposer d'autres solutions au problème du médecin. Ces solutions sont masquées par ce formalisme. Cependant, cette discussion dépasse le cadre souhaité dans cet article : la caractérisation choisie suffit à montrer l'intérêt d'une prise en compte différenciée des modalités.

Comme nous l'avons rappelé plus haut, l'utilitarisme de l'acte ordonne les actions en fonction de leurs utilités et définit comme permises celles qui ont les utilités les plus élevées. Pour formaliser cette vision ordinale, nous introduisons une relation de comparaison $\succsim_e$ sur $\mathcal{A} \times \mathcal{A}$ pour dénoter ces préférences éthiques. Ainsi $o \succsim_e o'$ signifie que l'action o est éthiquement préférée ou équivalente à l'action o'. La question est de savoir comment définir cette relation sur les actions, dont $\mathcal{A}_p$ est dérivé.

2.2 Différenciation ordinale des modalités

Comme discuté dans l'introduction, nous proposons de formaliser la différenciation des modalités comme un ordre partiel strict sur les modalités, que nous désignons par $\succ_m$, c'est-à-dire une relation asymétrique et transitive : $x \succ_m y$ signifie que la modalité x prévaut sur la modalité y. La modalité x est dite *dominante* et la modalité y *dominée*. L'ordre partiel strict peut être vu comme un ensemble de paires : $\succ_m \subset \mathcal{M}^2$. Chaque paire de modalités (x, y) est appelée une *comparaison*.

La difficulté de la définition de la supériorité consiste alors à prendre en compte ces comparaisons de modalités dans la détermination des actions admissibles : en ajoutant à la caractérisation des actions l'ordre partiel strict $\succ_m$ sur les modalités, il s'agit d'obtenir des informations sur la relation de comparaison $\succsim_e$, qui permettra ensuite d'obtenir l'ensemble $\mathcal{A}_p$.

Notre formalisation de la supériorité entre modalités a pour objectif de modéliser le fait qu'aucun plaisir issu de la consommation de chocolat, aussi grand qu'il soit, ne peut jamais égaler ou dépasser le fait de sauver une vie. Autrement dit de modéliser la préférence pour les actions qui portent une modalité dominante plutôt que n'importe laquelle des actions ne portant que des modalités dominées. De ce fait, la formalisation ne fournit que des relations de préférence stricte entre les actions, et pas de relations d'équivalence. Nous nous intéressons donc particulièrement à la partie asymétrique de $\succsim_e$ qui est notée $\succ_e$, où $o \succ_e o'$ signifie que o est strictement préférée à o'.

2.3 Formalisation des principes éthiques classiques

Dans cet article, la relation de préférence est définie à l'aide de propriétés de la forme suivante :

$$\forall o, o' \in \mathcal{A}, [\textit{conditions sur } o,\ o' \textit{et les modalités}] \Rightarrow o \succ_e o' \quad (1)$$

2.3.1 Utilitarisme de l'acte

Le principe de l'utilitarisme de l'acte rappelé dans la section précédente peut être exprimé comme suit :

$$\forall o, o' \in \mathcal{A}, \left[\sum_{x \in \mathcal{M}} o_x > \sum_{x \in \mathcal{M}} o'_x \right] \Rightarrow o \succ_e o' \quad (2)$$

On obtient pour l'exemple 1 : $\sum_{x \in \mathcal{M}} a_x = 10$ et $\sum_{x \in \mathcal{M}} a'_x = 11$. Selon l'équation 2, l'utilitarisme de l'acte conclut $a' \succ_e a$.

2.3.2 Optimalité de Pareto

Le principe de Pareto classique utilisé dans le cadre de la décision multicritère peut être écrit dans sa version stricte comme suit :

$$\forall o, o' \in \mathcal{A}, \left[\exists x \in \mathcal{M},\ (o_x > o'_x) \wedge (\forall y \in \mathcal{M} \backslash \{x\},\ o_y \geq o'_y) \right] \Rightarrow o \succ_e o' \quad (3)$$

Pour l'exemple 1, on observe $a_{human\_life} > a'_{human\_life}$ et $a'_{choco\_pleasure} > a_{choco\_pleasure}$. Les deux modalités ne favorisant pas la même action, l'optimalité de Pareto ne fournit aucune préférence.

2.3.3 Discussion

Les deux principes précédents assurent la transitivité et l'asymétrie de $\succ_e$. En terme de traitement des modalités, le principe utilitariste considère que les modalités sont équivalentes, puisque dans l'équation 2 la somme est une fonction d'agrégation commutative : on peut inverser les modalités sans modifier le résultat. Au contraire, le principe de Pareto considère que les modalités sont incomparables : dans l'équation 3, seules les quantifications d'une même modalité sont comparées entre les actions considérées.

La contribution de cet article, telle que décrite dans les sections suivantes, se concentre sur la définition d'une nouvelle condition plus expressive que ces deux cas extrêmes pour le traitement des comparaisons entre modalités. Elle combine les quantifications par modalités et l'ordre $\succ_m$ entre les modalités afin d'introduire la supériorité entre les modalités.

3 Proposition d'axiomatisation de la supériorité entre les modalités

Cette section décrit la définition proposée d'une relation de préférence éthique entre les actions possibles, basée sur l'ordre partiel entre les modalités, résultant en une axiomatisation de la supériorité, comme une généralisation du principe de Pareto. Elle formalise d'abord la définition du comportement de supériorité souhaité, puis décrit en trois étapes l'axiomatisation proposée, en fonction du nombre de modalités dominantes et dominées.

3.1 Définition de la supériorité

Afin de définir le comportement de supériorité souhaité, nous considérons d'abord le cas où l'ordre partiel strict sur la modalité contient une seule comparaison, notée $x \succ_m z$. La supériorité de la modalité x sur la modalité z est alors définie dans le formalisme par l'équivalence suivante :

$$x \succ_m z \Leftrightarrow \big[\forall o, o' \in \mathcal{A},\ [(o_x > o'_x \wedge \forall y \in \mathcal{M}\backslash\{x,z\},\ o_y \geq o'_y)] \Rightarrow o \succ_e o'\big] \quad (4)$$

Le point important de cette définition est que les quantifications de la modalité dominante x sont suffisantes pour déterminer la préférence entre deux actions, indépendamment des quantifications de la modalité dominée. Il n'y a donc pas de compensation possible entre une modalité dominante et une modalité dominée. Quant aux autres modalités y qui ne sont pas impliquées dans la comparaison, comme pour l'optimalité de Pareto, il est nécessaire qu'elles favorisent la même action que la modalité dominante.

3.2 One Over One : un dominant, un dominé

Dans le cas où l'ensemble de comparaison définit une seule modalité dominante et une seule modalité dominée, la définition de la relation $\succ_e$ induite découle directement de la définition de la supériorité de l'équation 4 :

$$\forall o, o' \in \mathcal{A}, \big[\exists x, z \in \mathcal{M},\ x \succ_m z \wedge (o_x > o'_x) \wedge (\forall y \in \mathcal{M}\backslash\{x,z\},\ o_y \geq o'_y)\big] \Rightarrow o \succ_e o' \quad (5)$$

Reprenons l'exemple 1 en y ajoutant la comparaison $human\_life \succ_m choco\_pleasure$. On sait que $a_{human\_life} > a'_{human\_life}$. Ne disposant que de deux modalités dans cet exemple, la condition sur les y est vérifiée également. Ainsi, l'équation déduit la préférence $treat\_patient \succ_e distribute\_chocolat$.

3.3 One Over Many : un dominant, plusieurs dominés

Dans un problème complexe, on peut être amené à considérer un ensemble de comparaisons. Cette section considère le cas où une seule modalité dominante prévaut sur un ensemble de modalités dominées. Dans ce cas, nous considérons la généralisation suivante de l'équation 5 : quel que soit le nombre de modalités dominées, elles ne peuvent pas contrer la préférence induite par la modalité dominante. Cette généralisation est une supposition forte qui donne à la propriété de supériorité proposée un caractère *absolu* : rien ne peut la contredire.

$$\forall o, o' \in \mathcal{A}, \big[\exists x \in \mathcal{M}, \exists Z \subset \mathcal{M}\backslash\{x\}, (\forall z \in Z,\ x \succ_m z) \wedge (o_x > o'_x)\ \wedge (\forall y \in \mathcal{M}\backslash\{x\} \cup Z,\ o_y \geq o'_y)\big] \Rightarrow o \succ_e o' \quad (6)$$

Cette propriété est équivalente à la reformulation suivante :

$$\forall o, o' \in \mathcal{A}, \big[\exists x \in \mathcal{M}, (o_x > o'_x) \wedge \forall y \in \mathcal{M}\backslash\{x\}, (x \succ_m y \vee o_y \geq o'_y)\big] \Rightarrow o \succ_e o' \quad (7)$$

Cette dernière formule souligne le fait qu'elle peut être considérée comme une généralisation de l'optimalité de Pareto. En effet, si aucune comparaison n'est considérée, alors $x \succ_m y$ est faux pour toutes les modalités et la formule est identique à l'équation 3.

3.4 Many Over Many : cas général

Dans le cas général, pour toute paire d'actions o et o', il faut distinguer trois sous-ensembles de modalités de $\mathcal{M}$:

— L'ensemble X des modalités favorisant une même action, qui doit être non vide afin d'obtenir une préférence stricte en favorisant une action o par rapport à une action o' :

$$X = \{x \in \mathcal{M} \mid o_x > o'_x\}$$

— L'ensemble des modalités dominées, qui représente les modalités dominées par au moins une modalité de l'ensemble X :

$$\{y \in \mathcal{M}\backslash X \mid \exists x \in X,\ x \succ_m y\}$$

— L'ensemble des modalités non dominantes et non dominées, qui doivent être en accord avec les modalités de l'ensemble X :

$$\{y \in \mathcal{M}\backslash X \mid o_y \geq o'_y\}$$

Par rapport au cas précédent, cette généralisation renforce le caractère *absolu* de la supériorité en précisant que la présence d'une seule modalité dominante suffit à considérer une modalité comme étant dominée. Une modalité dominée ne participe activement que si aucune préférence n'est exprimée pour toutes ses modalités dominantes. Dans ce cas, nous proposons la définition suivante :

$$\begin{aligned}&\forall o, o' \in \mathcal{A},\\ &\quad \big[\exists X \subset \mathcal{M},\ X \neq \emptyset,\ (\forall x \in X,\ o_x > o'_x) \wedge\\ &\quad [\forall y \in \mathcal{M}\backslash X,\ (\exists x \in X,\ x \succ_m y) \vee o_y \geq o'_y]\big]\\ &\qquad\qquad \Rightarrow o \succ_e o' \quad (8)\end{aligned}$$

Cette propriété est équivalente à la reformulation suivante :

$$\begin{aligned}&\forall o, o' \in \mathcal{A},\ \big[\exists x \in \mathcal{M}, (o_x > o'_x)\ \wedge [\forall y \in \mathcal{M},\\ &\quad (\exists x' \in \mathcal{M}, x' \succ_m y \wedge o_{x'} > o'_{x'}) \vee o_y \geq o'_y]\big]\\ &\qquad\qquad \Rightarrow o \succ_e o' \quad (9)\end{aligned}$$

Comme pour le cas précédent, il s'agit d'une généralisation de l'optimalité de Pareto. Si aucune comparaison n'est considérée, alors $x' \succ_m y$ est faux pour toutes les modalités et l'équation 9 est identique à l'équation de l'optimalité de Pareto.

3.5 Définition de la préférence minimale induite $\succ_e^m$

Parmi l'ensemble de toutes les préférences $\succ_e$ qui satisfont l'équation 9, la relation de préférence minimale est définie comme celle qui ne contient que les paires induites par l'équation. Ainsi pour définir cette relation, il suffit de remplacer l'implication de l'équation 9 par une équivalence.

Définition 1. *La* préférence éthique minimale*, notée $\succ_e^m$, est la relation de préférence induite uniquement par l'équation 9 :*

$$\begin{aligned}&\forall o, o' \in \mathcal{A},\\ &\exists x \in \mathcal{M}, (o_x > o'_x) \wedge\\ &[\forall y \in \mathcal{M}, (\exists x' \in \mathcal{M}, x' \succ_m y \wedge o_{x'} > o'_{x'}) \vee o_y \geq o'_y]\\ &\Leftrightarrow o \succ_e^m o'\end{aligned}$$

En utilisant cette définition, un ordre $\succ_e$ satisfait l'axiomatisation de la supériorité que nous proposons dans l'équation 9 si et seulement si il est un sur-ensemble de cette relation minimale : $\succ_e^m \subseteq \succ_e$.

La section suivante étudie les propriétés de cette relation de préférence minimale, en établissant qu'elle est asymétrique et transitive, ce qui implique que c'est un ordre partiel strict.

4 Propriétés de la relation $\succ_e^m$ proposée

Cette section établit que la relation $\succ_e^m$ proposée satisfait la propriété requise de définir une relation d'ordre sur les actions :

Théorème 1. *$\succ_e^m$ est un ordre partiel strict.*

Les sections 4.1 et 4.2 prouvent respectivement qu'il est asymétrique et transitif. Les deux preuves utilisent le lemme suivant où $\oplus$ désigne le XOR binaire :

Lemme 1. *Pour tout ensemble non vide $X \subseteq \mathcal{M}$, en notant l'ensemble des modalités maximales* $\max_{\succ_m}(X) = \{x \in X \mid \forall x' \in X,\ \neg(x' \succ_m x)\}$*, on a :*

$$\forall x \in X, (x \in \max_{\succ_m}(X)) \ \oplus\ (\exists x' \in \max_{\succ_m}(X),\ x' \succ_m x)$$

Démonstration. Ce lemme est prouvé par récurrence sur $|X|$.

— Si $|X| = 1$, alors $X = \{x\} = \max_{\succ_m}(X)$.

— Si $|X| = n+1$, avec $n \in \mathbb{N}^*$. On a $X = X' \cup \{x\}$, avec $|X'| = n$. Dans ce cas, on distingue deux possibilités :

  — $x \in \max_{\succ_m}(X)$.

  — $x \notin \max_{\succ_m}(X)$, par définition de $\max_{\succ_m}(X)$, on a $\exists x' \in X,\ x' \succ_m x$. D'après l'asymétrie de $\succ_m$, on peut conclure que $x \neq x'$ d'où $x' \in X'$. Par hypothèse de récurrence sur X' on obtient soit $x' \in \max_{\succ_m}(X')$, et on pose $x'' = x'$, soit $\exists x'' \in \max_{\succ_m}(X'),\ x'' \succ_m x'$. Par transitivité et asymétrie, on a $x'' \succ_m x$ et $\neg(x \succ_m x'')$. Donc on obtient $x'' \in \max_{\succ_m}(X)$ et $x'' \succ_m x$.

□

4.1 Asymétrie de la relation $\succ_e^m$ proposée

Proposition 1. *$\succ_e^m$ est asymétrique :*
elle vérifie $\forall o, o' \in \mathcal{A},\ o \succ_e^m o' \Rightarrow \neg(o' \succ_e^m o)$.

Démonstration. On suppose que $o \succ_e^m o'$ et par absurde que $o' \succ_e^m o$. D'après la définition 1, on obtient :

— $\exists x_0 \in \mathcal{M},\ (o_{x_0} > o'_{x_0})$ (A)

— $\forall y \in \mathcal{M}$,
$o_y \geq o'_y \vee (\exists x' \in \mathcal{M},\ x' \succ_m y \wedge o_{x'} > o'_{x'})$ (B)

— $\exists x_1 \in \mathcal{M},\ (o'_{x_1} > o_{x_1})$ (C)

— $\forall y \in \mathcal{M}$,
$o'_y \geq o_y \vee (\exists x' \in \mathcal{M},\ x' \succ_m y \wedge o'_{x'} > o_{x'})$ (D)

Appelons S l'ensemble des modalités qui ont une préférence pour o plutôt que o' et I l'ensemble des modalités qui ont une préférence pour o' plutôt que o.

$S = \{x \in \mathcal{M} \mid o_x > o'_x\}$ et $I = \{x \in \mathcal{M} \mid o'_x > o_x\}$. D'après (A) et (B), on sait que ces ensembles sont non vides. S étant non vide et en utilisant le Lemme 1, on peut prendre un $z \in \max_{\succ_m}(S)$. Ainsi, $z \in S$ donc $o_z > o'_z$ et donc, avec (D), $\exists x_2 \in \mathcal{M}$, $x_2 \succ_m z \wedge o'_{x_2} > o_{x_2}$. $o'_{x_2} > o_{x_2} \Rightarrow x_2 \in I$ et en utilisant le Lemme 1 sur I :

— si $x_2 \in \max_{\succ_m}(I)$, on note $x_3 = x_2$.
— sinon $\exists x_3 \in \max_{\succ_m}(I)$, $x_3 \succ_m x_2$.

Dans les deux cas on obtient $x_3 \succ_m z$ par transitivité de $\succ_m$. $x_3 \in I$ donc $o'_{x_3} > o_{x_3}$ et avec (B), $\exists x_4 \in \mathcal{M}, x_4 \succ_m x_3 \wedge o_{x_4} > o'_{x_4}$. On déduit $o_{x_4} > o'_{x_4}$ donc $x_4 \in S$. Par transitivité $x_4 \succ_m z$, de plus par définition de $\max_{\succ_m}(S)$, $x_4 \in S$ et $x_4 \succ_m z$ donc $z \notin \max_{\succ_m} S$, ce qui est contradictoire. On conclut donc que $\neg(o \succ_e^m o' \wedge o' \succ_e^m o)$. □

4.2 Transitivité de la relation $\succ_e^m$ proposée

Proposition 2. *$\succ_e^m$ est transitive :*
elle vérifie $\forall o, o' \in \mathcal{A}$,
$$(o \succ_e^m o' \wedge o' \succ_e^m o'') \Rightarrow (o \succ_e^m o'').$$

Démonstration. Considérons o, o', o'' tel que $o \succ_e^m o'$ et $o' \succ_e^m o''$. En utilisant la définition 1, on obtient :

— $\exists x_0 \in \mathcal{M}$, $(o_{x_0} > o'_{x_0})$ (E_1)
— $\forall y \in \mathcal{M}$, $o_y < o'_y \Rightarrow (\exists x' \in \mathcal{M}$, $x' \succ_m y \wedge o_{x'} > o'_{x'})$ (E_2)
— $\exists x_1 \in \mathcal{M}$, $(o'_{x_1} > o''_{x_1})$ (F_1)
— $\forall y \in \mathcal{M}$, $o'_y < o''_y \Rightarrow (\exists x' \in \mathcal{M}$, $x' \succ_m y \wedge o'_{x'} > o''_{x'})$ (F_2)

On doit prouver $o \succ_e^m o''$, soit d'après la définition 1, $P_1 : \exists x \in \mathcal{M}, o_x > o''_x$, et pour tout $y \in \mathcal{M}$,
$P_2(y) : o_y < o''_y \Rightarrow \exists z \in \mathcal{M}. z \succ_e^m y \wedge o_z > o''_z$.

Preuve de P_1. Par E_1, on a x_0 tel que $o_{x_0} > o'_{x_0}$. Si $o'_{x_0} \geq o''_{x_0}$ alors $o_{x_0} > o''_{x_0}$ et P_1 est satisfait. Sinon, $o'_{x_0} < o''_{x_0}$. Selon le lemme 1 et F_2, $S_0 = \{x \in \mathcal{M} | x \succ_m x_0 \wedge o'_x > o''_x\}$ est non vide, ainsi on peut choisir x_2 dans $max_{\succ_m} S_0$. Si $o_{x_2} \geq o'_{x_2}$ alors $o_{x_2} > o''_{x_2}$ et P_1 est satisfait. Sinon, $o_{x_2} < o'_{x_2}$. Par E_2, on obtient une modalité x_3 telle que $x_3 \succ_m x_2$ et $o_{x_3} > o'_{x_3}$ Comme x_2 est maximale pour $\succ_m$ dans S_0, on a $x_3 \notin S_0$ et donc $o'_{x_3} \leq o''_{x_3}$. Si $o'_{x_3} < o''_{x_3}$, utiliser F_2 donnerait une modalité de S_0 supérieure à x_2, ce qui contredirait sa maximalité. Donc $o'_{x_3} = o''_{x_3}$. Avec $o_{x_3} > o'_{x_3}$, cela implique P_1.

Preuve de $\forall y, P_2(y)$. Considérons une modalité $y_0 \in \mathcal{M}$. Si $o_{y_0} \geq o''_{y_0}$, $P_2(y_0)$ est trivialement vérifié. Sinon, on a $o_{y_0} < o''_{y_0}$ (H_1). On a donc deux cas :

(A) Supposons $o_{y_0} < o'_{y_0}$ (H_2). Selon le lemme 1, E_2 et H_2, $S_2 = \{x \in \mathcal{M} | x \succ_m y_0 \wedge o_x > o'_x\}$ est non vide, ainsi on peut choisir x' dans $max_{\succ_m} S_2$.
  (A.1) Supposons $o'_{x'} < o''_{x'}$ (H_3). Par F_2 et H_3, on obtient une modalité z telle que $z \succ_m x' \wedge o'_z > o''_z$. On a $z \succ_m x'$ et $x' \succ_m y_0$, donc, par transitivité de $\succ_m$, $z \succ_m y_0$. Etant donné que x' est maximal pour $\succ_m$, on doit avoir $z \notin S_2$ ce qui donne $o_z \geq o'_z$. Avoir $o_z > o'_z$ n'est pas possible car cela autoriserait à dériver depuis E_2 une modalité qui appartiendrait à S_1 tout en étant supérieure à x', contredisant encore la maximalité de x'. On peut conclure $o_z = o'_z$, et donc $o_z > o''_z$, ce qui prouve $P_2(y_0)$.
  (A.2) Sinon, $o'_{x'} \geq o''_{x'}$. Etant donné un $x' \in S_1$, on obtient $o_{x'} > o''_{x'}$. On a ainsi (prenant x' pour z), $P_2(y_0)$.

(B) Dans l'autre cas, $o_{y_0} \geq o'_{y_0}$ (H_4). On considère ensuite les modalités qui sont supérieures à y_0.
  (B.1) Supposons que $\exists y' \in \mathcal{M}, y' \succ_m y_0 \wedge o_{y'} < o'_{y'}$. Alors, en appliquant le raisonnement du cas A.1 à y', on obtient un $z \in \mathcal{M}$ tel que $z \succ_m y'$ et $o_z > o''_z$. Par transitivité de $\succ_m$, $z \succ_m y_0$, ce qui prouve $P_2(y_0)$.
  (B.2) Sinon, on doit avoir : $\forall y' \in \mathcal{M}, y' \succ_m y_0 \Rightarrow o_{y'} \geq o'_{y'}$ (H_5). Avec H_1 et H_4, on a $o'_{y_0} < o''_{y_0}$. Appliquer F_2 donne une modalité z telle que $z \succ y_0$ et $o'_z > o''_z$. Etant donné H_5 on a $o_z \geq o'_z$ et donc $o_z > o''_z$, ce qui prouve $P_2(y_0)$.

Nous avons ainsi prouvé $P_2(y_0)$ dans tous les cas et pour tout y_0. □

Ceci conclut la preuve du théorème 1. $\succ_e^m$ est un ordre partiel strict.

5 Discussions sur les hypothèses faites sur $\succ_m$

Nous avons supposé que la relation $\succ_m$ est asymétrique et transitive, cela englobe de nombreuses situations, néanmoins nous discutons dans cette section deux cas alternatifs.

5.1 Cas d'une relation $\succ_m$ totale

L'ajout d'autres hypothèses peut donner des informations supplémentaires sur $\succ_e$. Par exemple, si nous supposons que la relation $\succ_m$ est également totale, alors l'axiomatisation devient un ordre lexicographique sur les modalités [6]. Il suffit alors pour départager les actions d'observer s'il existe une préférence pour la modalité au sommet de l'ordre, et ainsi de suite jusqu'à la fin de l'ordre $\succ_m$. Ainsi, pour toute paire d'actions non égales o et o', une préférence sera déduite de l'équation 9. Cette propriété est utile si l'on souhaite une action unique à réaliser. Cependant, le fait d'avoir une seule action permise peut être vu comme une propriété restrictive pour un système de conformité éthique.

5.2 Cas d'une relation $\succ_m$ non transitive

On peut aussi souhaiter que la relation $\succ_m$ ne soit pas transitive. Cependant, cette section montre que c'est une condition nécessaire à notre axiomatisation si l'on souhaite

définir des préférences rationnelles entre différentes actions. En effet, si on autorise une relation $\succ_m$ qui n'est pas transitive, il est alors possible de définir des boucles de supériorité entre modalités. Supposons par exemple que l'on dispose de quatre modalités w, x, y, z telles que $w \succ_m x$, $x \succ_m y$, $y \succ_m z$ et $z \succ_m w$. Dans ce cas, l'axiomatisation proposée dans l'équation 9 ne garantit plus l'asymétrie et la transitivité de la relation minimale induite $\succ_e^m$. Pour l'illustrer, considérons deux actions o et o' telles que $o_w > o'_w$, $o_x < o'_x$, $o_y > o'_y$ et $o_z < o'_z$. Appliquons maintenant l'équation 9 :

— Pour obtenir $o \succ_e^m o'$: on observe $o_w > o'_w$ donc $\exists x \in \mathcal{M}, (o_x > o'_x)$ est vérifiée. De plus on vérifie $\forall y \in \mathcal{M}, o_y \geq o'_y$ pour les modalités w et y. Quant aux modalités x et z, il existe w et y telles que $w \succ_m x \wedge o_w > o'_w$ et $y \succ_m z \wedge o_y > o'_y$. Ainsi x et z vérifient $\forall y \in \mathcal{M}, (\exists x' \in \mathcal{M}, x' \succ_m y \wedge o_{x'} > o'_{x'})$. Toutes les conditions sont vérifiées, on en déduit $o \succ_e o'$.

— Pour obtenir $o' \succ_e^m o$, on applique exactement le même raisonnement mais en partant de la modalité x au lieu de la modalité w.

La relation induite $\succ_e^m$ n'est donc pas asymétrique. On peut alors observer des cycles dans les préférences obtenues entre les différentes actions. Ces cycles dans les préférences posent différents problèmes [9], comme l'argument de la pompe monétaire, qui empêchent de les considérer comme rationnelles. Si l'on souhaite éviter les cycles tout en conservant une relation $\succ_m$ qui n'est pas transitive, il est nécessaire de modifier l'axiomatisation proposée. Néanmoins, de telles modifications sortent du cadre voulu pour cet article étant donné qu'elles nécessitent d'introduire des conditions ne provenant pas directement du concept de supériorité entre modalités.

6 Conclusion et perspectives

Cet article propose une axiomatisation du concept philosophique de supériorité entre les modalités du Bien. Pour ce faire, un formalisme de décision multicritère ordinal adapté à la prise de décision éthique a été défini, basé sur une approche utilitariste. En tant que généralisation du principe d'optimalité de Pareto, l'axiomatisation proposée permet de déduire les préférences à partir de la différenciation des modalités.

Le travail présenté dans cet article ouvre de multiples perspectives. Il constitue une première étape pour relier les préoccupations philosophiques aux préférences ordinales. Les travaux en cours visent à étudier les cadres formels existants qui offrent des propriétés similaires à celles que nous proposons, comme par exemple les hiérarchies de contraintes [2], les CP-nets et TCP-nets [4] ou encore des variantes des méthodes de surclassement avec seuil [11].

Une limite du travail actuel réside dans les simplifications faites sur les relations causales dans le formalisme utilisé, comme discuté dans la section 2. Afin de prendre en compte les questions éthiques qui interviennent sur ces relations causales, nous envisageons d'étendre le formalisme pour pouvoir prendre en compte chaque conséquence des actions séparément.

Comme discuté dans la section 5, l'ensemble minimal de préférences éthiques qui respectent un ordre de supériorité $\succ_m$ entre les modalités n'est pas nécessairement total. En effet, le principe de supériorité qui est axiomatisé n'a pas pour objectif de résoudre toutes les décisions éthiques. Cela soulève donc des questions sur la combinaison de plusieurs principes afin d'obtenir un unique ensemble d'actions permissibles. Ainsi, des travaux en cours ont pour objectif de formaliser une version plus générale du concept de principe éthique et des conditions que le mélange de plusieurs principes doit respecter.

Références

[1] Anderson, Michael et Susan Leigh Anderson: *Machine Ethics*. Cambridge University Press, 2011.

[2] Borning, Alan, Bjorn Freeman-Benson et Molly Wilson: *Constraint hierarchies*. LISP and symbolic computation, 5(3) :223–270, 1992.

[3] Bourgne, Gauvain, Camilo Sarmiento et Jean Gabriel Ganascia: *ACE modular framework for computational ethics : dealing with multiple actions, concurrency and omission*. Dans *1st Workshop on Computational Machine Ethics*, 2021.

[4] Brafman, Ronen I, Carmel Domshlak et Solomon Eyal Shimony: *On graphical modeling of preference and importance*. Journal of Artificial Intelligence Research, 25 :389–424, 2006.

[5] Chang, Ruth: *Incommensurability (and Incomparability)*. John Wiley & Sons, Ltd, 2013.

[6] Fishburn, Peter C.: *Axioms for Lexicographic Preferences*. The Review of Economic Studies, 42(3) :415–419, 1975.

[7] Gonzales, Christophe et Patrice Perny: *Multicriteria Decision Making*, page 519–548. Springer International Publishing, 2020.

[8] Griffin, James: *Are There Incommensurable Values ?* Philosophy & Public Affairs, 7(1) :39–59, 1977.

[9] Hansson, Sven Ove et Till Grüne-Yanoff: *Preferences*. Dans *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2022.

[10] Lindner, Felix, Martin Mose Bentzen et Bernhard Nebel: *The HERA approach to morally competent robots*. Dans *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6991–6997. IEEE, 2017.

[11] Rogers, Martin, Michael Bruen et Lucien Yves Maystre: *The Electre Methodology*, pages 45–85. Springer US, Boston, MA, 2000.

[12] Tolmeijer, Suzanne, Markus Kneer, Cristina Sarasua, Markus Christen et Abraham Bernstein: *Implementations in Machine Ethics : A Survey*. ACM Comput. Surv., 53(6), décembre 2021.

[13] Vallentyne, Peter: *Consequentialism*. Dans *Philosophy publications*. Wiley-Blackwell, 2006.

Actes JIAF-JFPDA 2023

Processus de décision markoviens éthiques

Mihail Stojanovski Nadjet Bourdache Grégory Bonnet Abdel-Illah Mouaddib

Normandie Univ., UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France
{prenom.nom}@unicaen.fr

Résumé

Avec l'introduction d'agents autonomes dans la vie quotidienne, l'intégration de l'éthique dans les processus décisionnels de ces agents devient une question importante. Afin d'avoir un modèle générique fournissant un cadre expressif pour juger un comportement, nous proposons dans cet article les processus de décision markoviens éthiques, (E-MDP pour *Ethical Markov Decision Processes*), qui étendent les MDP classiques avec la représentation explicite des valeurs morales – positives ou négatives – que les décisions des agents peuvent promouvoir ou trahir. Nous proposons un algorithme pour résoudre les E-MDP et nous illustrons notre modèle sur trois cadres éthiques distincts : le cadre de la Théorie du Commandement Divin, le cadre des Devoirs *Prima Facie* et le cadre de l'Éthique de la Vertu, cadres qui raisonnent respectivement sur des états interdits, sur des paires état-action obligatoires et sur des trajectoires exemplaires représentées par des ensembles de transitions. Enfin, nous expérimentons le modèle sur une application de véhicule autonome et évaluons la perte de valeur entre les politiques morales et amorales.

Abstract

With the introduction of autonomous agents in everyday life, the integration of ethics in the decision-making processes of these agents becomes an important issue. In order to have a generic model which provides an expressive framework for judging behavior, we propose in this article the Ethical Markov Decision Processes (E-MDPs), that extend classical MDPs with the explicit representation of moral values – positive or negative – that agents' decisions may promote or demote. We propose an algorithm for solving E-MDPs and illustrate our model on three distinct ethical frameworks: the Divine Command Theory framework, the *Prima Facie* Duties framework, and the Virtue Ethics framework, which respectively focus on prohibited states, mandatory state-action pairs, and exemplary trajectories represented by sets of transitions. Finally, we experiment our model on an autonomous vehicle setting and evaluate the value loss between moral and amoral policies.

1 Introduction

Le développement d'agents autonomes interagissant avec des êtres humains peut poser des problèmes dans certaines situations en l'absence d'une composante éthique. Par exemple, dans le domaine des véhicules autonomes, le choix d'une vitesse de conduite peut être considéré comme plus ou moins éthique selon les circonstances. En effet, il peut être préférable de conduire à vitesse réduite devant une école à l'heure de la fin des cours, plutôt que de respecter simplement la limite de vitesse. Dans ce cas, l'éthique n'est pas une contrainte stricte mais plutôt une recommandation à laquelle l'agent devrait se conformer s'il en a la possibilité.

La littérature propose majoritairement des modèles qualitatifs fondés sur la logique, comme par exemple les cadres d'argumentation valuée [4], la logique modale [11], la logique non monotone [5], ou les architectures BDI [6]. Cependant, dans cet article, nous nous concentrons sur des processus décisionnels quantitatifs qui sont les *processus de décisions markoviens* (MDP). Traiter de l'éthique dans ce type de modèles est relativement nouveau et, à notre connaissance, les approches proposées manquent de généricité. En effet, elles ne permettent souvent de traiter qu'une unique règle parmi un ensemble de règles éthiques, appelées *principes éthiques*, qui indiquent de manière exogène au modèle quelles sont les valeurs morales importantes et quelles sont les décisions autorisées. Le problème est qu'il est nécessaire d'avoir un modèle différent pour chaque principe éthique auquel l'agent doit se conformer, ce qui peut être coûteux en temps et en effort.

Afin d'intégrer plus facilement l'éthique dans le raisonnement des agents autonomes, il est nécessaire de créer un modèle qui soit générique et qui permette d'exprimer avec richesse les comportements éthiques. Nous proposons alors le modèle des *processus de décision markoviens éthiques* (E-MDP) qui étend les MDP avec des valeurs morales explicites et utilise une fonction de récompense multicritère qui représente distinctement la satisfaction (promotion) et violation (trahison) de ces valeurs.

Remarquons qu'une fois une représentation de l'éthique choisie et spécifiée, le problème de décision éthique peut être vu comme un problème d'optimisation multicritère séquentiel. En effet, nous voulons d'abord assurer un comportement où l'agent fait le moins de mal possible, et parmi les comportements obtenus choisir ceux qui font le plus de bien, en accord avec le principe éthique choisi. Dans le cas où l'agent autonome a une tâche à accomplir, le comportement optimal par rapport à la tâche sera choisi parmi les comportements qui minimisent d'abord le mal et maximisent ensuite le bien.

Cet article est structuré comme suit. La section 2 présente un aperçu de l'état de l'art en termes d'intégration de l'éthique dans les MDP ainsi qu'un positionnement philosophique de notre proposition. Nous discutons d'une caractérisation de la prise de décision éthique dans la section 3 et détaillons le modèle E-MDP dans la section 4. Nous introduisons un algorithme pour résoudre les E-MDP en section 5 et illustrons notre modèle avec trois principes éthiques distincts en section 6. Enfin, la section 7 est consacrée à nos résultats expérimentaux.

2 État de l'art

Nous rappelons ici la définition des MDP, ainsi que quelques approches récentes visant à intégrer l'éthique dans ces modèles ; nous présentons ensuite une vision de haut niveau de l'éthique sur laquelle notre proposition est fondée.

2.1 Processus de décision markoviens

Un *processus de décision markovien* [14] est un modèle de prise de décision dans un environnement stochastique entièrement observable. Un MDP est un quadruplet $\langle S, A, \mathcal{T}, R\rangle$ où S est un ensemble fini d'états, A est un ensemble fini d'actions, $\mathcal{T} : S \times A \times S \to [0, 1]$ est une fonction telle que $\mathcal{T}(s, a, s')$ est la probabilité que choisir l'action $a \in A$ dans l'état $s \in S$ produise l'état résultant $s' \in S$, $R : S \times A \times S \to \mathbb{R}$ est une fonction telle que $R(s, a, s')$ est la récompense immédiate que l'agent recevra en arrivant à l'état $s' \in S$ après avoir choisi l'action $a \in A$ dans l'état $s \in S$.

Un solution à un MDP est une fonction, appelée *politique*, qui associe une action (ou un ensemble d'actions) à chaque état du MDP. Il existe plusieurs types de politiques comme des politiques déterministes ou stochastiques [14] ou encore non-déterministes [8]. Les politiques déterministes $\pi : S \to A$ décrivent quelle action doit faire l'agent dans chaque état s tandis que les politiques stochastiques $\pi : S \times A \to [0, 1]$ décrivent pour chaque état s la probabilité de choisir l'action a. Dans cet article, nous nous intéressons en particulier aux politiques non-déterministes. Une telle politique $\pi : S \to 2^{|A|}$ décrit l'ensemble des choix d'action que l'agent peut faire dans chaque état. Une solution optimale d'un MDP est une *politique optimale* π^*, qui maximise la récompense cumulée attendue dans chaque état $s \in S$ en résolvant l'équation de Bellman. Pour la maximisation, l'équation de Bellman prend la forme suivante $\forall s \in S$:

$$V^*(s) = \max_a \sum_{s' \in S} \mathcal{T}(s, a, s')[\mathcal{R}(s, a, s') + \gamma V^*(s')]$$

Les principaux algorithmes pour résoudre un MDP et obtenir des politiques optimales sont *Value Iteration* (VI) [3] et *Policy Iteration* (PI) [10]. Pour nos besoins, nous utilisons une variante de VI qui utilise les Q-valeurs [19].

2.2 Intégrer l'éthique dans les MDP

Plusieurs approches récentes ont été proposées pour intégrer l'éthique dans les MDP. Tout d'abord, certains travaux s'intéressent à l'apprentissage par renforcement, qui consiste à apprendre un comportement par essais et erreurs. Cependant, ces premières approches nécessitent une certaine part d'intervention humaine pour juger de la moralité des actions de l'agent. Par exemple, Abel *et al.* [1] utilisent l'expertise d'un humain pour juger a priori de la moralité de chaque action dans un MDP, tandis que Wu *et al.* [21] utilisent une base de données qui agrège de nombreuses réponses humaines pour un POMDP (MDP partiellement observable). Dans les deux cas, la récompense obtenue par l'agent dépend de la similarité entre la décision humaine et celle de l'agent.

Une autre manière d'intégrer l'éthique dans les MDP est d'utiliser une technique de façonnage de la fonction de récompense – *reward function shaping* – qui consiste à introduire des modifications locales de cette fonction. Pour cela, des récompenses et des pénalités fondées sur la moralité des actions choisies sont ajoutées à la fonction. Par exemple, dans le cas d'un véhicule autonome, De Moura *et al.* [7] utilisent une fonction de récompense fondée sur la proximité des autres usagers de la route, le respect de la loi et la distance au but. En cas de collisions inévitables, la fonction de récompense est modifiée selon des préférences éthiques pour décider de la collision la plus acceptable.

Une autre approche consiste à ajouter une composante éthique au MDP. Par exemple, Svegliato *et al.* [17] définissent des *systèmes autonomes éthiquement conformes* (ECAS pour ethically compliant autonomous systems) qui intègrent un principe moral particulier. Pour cela, une contrainte qui n'est pas directement exprimée dans le MDP est ajoutée au modèle. L'approche est alors fondée sur de l'optimisation sous contrainte, c'est-à-dire la recherche d'une politique qui satisfait cette contrainte. En conséquence, si les contraintes sont incohérentes, une politique éthique peut ne pas exister. Une extension multi-agent de l'ECAS est proposée dans Nashed *et al.* [13]. Ici, chaque agent a un espace d'état particulier et une fonction de valeur

sur celui-ci. Le principe moral est défini pour tenir compte des communautés morales auxquelles l'agent appartient. Dans ces approches, chaque nouveau principe moral nécessite une contrainte spécifique.

Une autre approche, proposée par Rodriguez-Soto *et al.* [16], consiste à utiliser une combinaison de fonctions de récompense. Ici, l'éthique est modélisée par une fonction normative et une fonction évaluative, qui, avec une fonction de récompense classique, guident les agents. La fonction normative pénalise l'agent qui viole une norme (agit d'une manière interdite ou néglige une obligation) tandis que la fonction évaluative le récompense pour des actions autorisées ou obligatoires. La fonction de valeur agrège toutes ces fonctions, et une politique éthique est une politique qui ne viole aucune norme tout en étant louable. Une telle politique peut ne pas toujours exister. Dans ce cas, en raison de l'agrégation, le modèle ne peut pas différencier les politiques en fonction de leurs valeurs. En effet, les politiques ayant de nombreuses normes violées et de nombreuses actions louables peuvent être classées au même rang que celles ayant peu de normes violées et peu d'actions louables.

2.3 Une approche à haut niveau pour l'éthique

Plusieurs travaux en éthique computationnelle [5, 6] distinguent clairement la morale de l'éthique, en se fondant sur la littérature en philosophie. Ici, les options sont évaluées en termes de morale, c'est-à-dire en leur associant le fait de causer du bien ou du mal par rapport à leur conformité aux mœurs, valeurs et usages d'un groupe ou d'une personne [18]. La morale est en effet fondée sur des valeurs morales, qui sont des notions abstraites – positives (ex. courage, sens de la justice) ou négatives (ex. avidité ou cruauté) – caractérisant un large ensemble d'objets : états, actions ou même normes. Prendre une décision qui promeut une valeur morale positive cause du bien, tandis qu'en trahir une correspond généralement à causer du mal. Lorsque deux options sont soutenues par des valeurs morales différentes, chacune apportant un certain regret étant donné que l'exécution des deux est impossible, il s'agit d'un *dilemme moral* [12]. Les sciences sociales ont montré que ces valeurs varient en importance et forment des organisations hiérarchiques appelées systèmes de valeurs [20]. L'éthique est alors à la fois la manière d'éliciter ces valeurs morales (c'est-à-dire de choisir les valeurs qui sont importantes pour le décideur), et de les agréger afin de prendre une décision juste.

Les premières réflexions sur l'éthique distinguent les obligations et les interdictions, des recommandations (positives ou négatives) : ne pas remplir une obligation ou violer une interdiction devrait être sanctionné par une importante pénalité, tandis que faire une action non recommandée ou ne pas faire une action recommandée devrait être sanctionné plus faiblement [2]. D'autres idées – par exemple, la doctrine du double effet [9] – réfutent la dualité entre causer le bien et causer le mal : nous ne pouvons pas facilement justifier de causer le mal en causant le bien car l'un n'est pas la négation de l'autre. Plus encore, la doctrine du faire et du permettre (*Doing and Allowing*) [15] met en évidence la différence entre faire du mal et permettre que du mal se produise. En effet, causer du mal est plus blâmable que de permettre que du mal se produise à l'avenir.

Bien que nous soyons conscients que ces notions de morale et d'éthique peuvent être critiquées ou affinées, nous avons choisi de représenter l'éthique grâce aux notions précédentes. Par conséquent, le modèle que nous devons définir doit : (1) avoir une représentation explicite des valeurs morales qui peuvent être élicitées comme positives ou négatives par le décideur ; (2) considérer explicitement le mal et le bien comme distincts, c'est-à-dire que causer un mal (resp. empêcher un mal) ne signifie pas nécessairement que le bien a été empêché (resp. le bien a été causé) ; (3) faire la distinction entre causer un mal (resp. un bien) et permettre un mal (resp. un bien) ; (4) être capable d'exprimer des obligations, des interdictions et des actions recommandées.

3 Prise de décision éthique

Décider d'un point de vue éthique nécessite un *contexte éthique* qui est propre au décideur. Ce contexte représente les valeurs morales de l'agent qui peuvent être divisées en valeurs positives, négatives et neutres. En effet, la polarité des valeurs peut différer d'un agent à l'autre. Par exemple, l'*obéissance* est une valeur qui peut être positive ou négative selon la position philosophique de l'agent.

Définition 1 *Soit* $\mathcal{V} = \{v_1, \ldots, v_k\}$ *un ensemble de valeurs morales. Un* contexte éthique C *est un tuple* $\{C_1, \ldots, C_k\}$ *où* $C_i \in \{1, -1, 0\}$. *Une évaluation de* 1 *(resp.* -1 *et* 0*) signifie que la valeur est considérée comme positive (resp. négative et neutre) pour l'agent. Soit* $\mathcal{G}_C$ *(resp.* $\mathcal{B}_C$ *et* $\mathcal{N}_C$*) l'ensemble des valeurs positives (resp. négatives et neutres) dans le contexte* C *:* $\mathcal{G}_C = \{v_i \in \mathcal{V} : C_i = 1\}$ *(resp.* $\mathcal{B}_C = \{v_i \in \mathcal{V} : C_i = -1\}$ *et* $\mathcal{N}_C = \{v_i \in \mathcal{V} : C_i = 0\}$*).*

Remarquons que les valeurs ne diffèrent pas en importance entre elles : une valeur positive ne peut pas être considérée comme meilleure qu'une autre (resp. négative, pire). La prise en compte d'une hiérarchie entre les valeurs est laissée à des travaux futurs. À ce stade, une question se pose : comment un agent doit-il prendre une décision en fonction d'un contexte éthique ? D'un point de vue général, une valeur peut être promue ou trahie par une décision. La promotion d'une valeur signifie que le comportement de l'agent est conforme à celle-ci, tandis que la trahison d'une valeur signifie qu'elle est violée par le comportement de l'agent (que la valeur soit positive ou négative). Il semble donc naturel qu'une prise de décision éthique intéressante maximise les valeurs morales positives promues

et minimise les valeurs négatives promues. De plus, elle doit maximiser les valeurs négatives trahies et minimiser les valeurs positives trahies. Comme les valeurs positives ou négatives promues et trahies ne sont pas duales, un tel processus de décision est un processus de décision multi-critère. On souhaite qu'il satisfasse la propriété suivante.

Propriété 1 *Un processus décisionnel éthique maximise la promotion des valeurs positives et la trahison des valeurs négatives, tout en minimisant la trahison des valeurs positives et la promotion des valeurs négatives.*

Cependant, il n'est pas toujours possible de satisfaire la propriété 1. Par conséquent, nous proposons le compromis exprimé par la propriété 2.

Propriété 2 *Un processus décisionnel éthique minimise d'abord le mal en minimisant les valeurs négatives promues et en maximisant celles trahies (le 1^e critère étant plus important que le 2^e), puis maximise le bien en maximisant les valeurs positives promues et en minimisant celles trahies (le 1^e critère étant plus important que le 2^e).*

4 Processus de décision markoviens éthiques

Nous proposons des *processus de décision markoviens éthiques* (E-MDP), qui sont des MDP étendus avec une morale représentée par des valeurs promues ou trahies associées aux transitions, et une éthique, qui dicte comment l'agent optimise sa décision par rapport à la morale.

Définition 2 (MDP éthique) *Un E-MDP est un sextuplet $\langle S, A, \mathcal{T}, C, \mathcal{E}, \mathcal{R}\rangle$ où S, A et $\mathcal{T}$ sont les ensembles classiques d'états, d'actions et la fonction de transition (voir section 2.1), C est un contexte éthique (voir définition 1), $\mathcal{E}$ est une fonction d'évaluation morale (voir définition 3) et $\mathcal{R}$ est une fonction de récompense éthique (voir définition 4).*

Pour exprimer les principes éthiques, nous devons décrire l'alignement du comportement de l'agent avec les valeurs morales. Nous introduisons donc l'évaluation morale des transitions, qui mesure si une transition promeut, trahie ou ne considère pas une valeur donnée du contexte.

Définition 3 (Évaluation morale des transitions) *Chaque transition $(s, a, s') \in S \times A \times S$ où $\mathcal{T}(s, a, s') > 0$, est associée à un tuple $\mathcal{E}(s, a, s') = \langle \epsilon_1, \ldots, \epsilon_k\rangle$ représentant son évaluation morale. Le i-ème élément de $\mathcal{E}(s, a, s')$, noté $\mathcal{E}(s, a, s')_i$, prend une valeur dans $\{1, -1, 0\}$, qui signifie que la valeur morale $v_i \in \mathcal{V}$ est respectivement promue, trahie ou n'est pas considérée.*

4.1 Modélisation de l'éthique

La fonction de récompense des E-MDPs décrit comment les valeurs morales du contexte éthique de l'agent sont alignées avec son comportement. L'agent peut promouvoir ou trahir une valeur, ce qui donne quatre comportements distincts : la promotion d'une valeur positive ou négative – c'est-à-dire *causer du bien ou du mal* – et les comportements trahissant une valeur positive ou négative – c'est-à-dire *empêcher le bien ou le mal*.

Définition 4 (Fonctions de récompense éthique) *La fonction de récompense $\mathcal{R}$ produit un quadruplet où Δ compte le bien causé, ∇ le mal causé, $\overline{\Delta}$ le bien empêché, $\overline{\nabla}$ le mal empêché. Ainsi, $\mathcal{R}(s, a, s') = \langle \Delta, \nabla, \overline{\Delta}, \overline{\nabla}\rangle$ avec $\Delta, \nabla, \overline{\Delta}, \overline{\nabla}$ tels que :*

$$\Delta = \sum_{v_i \in \mathcal{G}_C} x_i \text{ et } \nabla = \sum_{v_i \in \mathcal{B}_C} x_i \text{ où } x_i = \begin{cases} 1 & \text{si } \mathcal{E}(s, a, s')_i = 1, \\ 0 & \text{sinon.} \end{cases}$$

$$\overline{\Delta} = \sum_{v_i \in \mathcal{G}_C} x_i \text{ et } \overline{\nabla} = \sum_{v_i \in \mathcal{B}_C} x_i \text{ où } x_i = \begin{cases} 1 & \text{si } \mathcal{E}(s, a, s')_i = -1, \\ 0 & \text{sinon.} \end{cases}$$

Nous notons $\mathcal{R}_\star(s, a, s')$ avec $\star \in \{\Delta, \nabla, \overline{\Delta}, \overline{\nabla}\}$ la composante $\star$ de la fonction de récompense pour une transition donnée. Suivant la définition 4, une politique d'un E-MDP peut être évaluée selon plusieurs fonctions de valeur éthiques qui donnent la valeur d'une politique dans un état donné selon une composante $\star$ de $\mathcal{R}$.

Définition 5 (Fonction de valeur éthique) *Une fonction de valeur éthique $V^\pi_\star(s)$ pour $\star \in \{\Delta, \nabla, \overline{\Delta}, \overline{\nabla}\}$ est définie comme :*

$$V^\pi_\star(s) = \sum_{a \in A} \pi(a \mid s) \sum_{s' \in S} p(s' \mid s, a)\big(\mathcal{R}_\star(s, a, s') + \big(\gamma V^\pi_\star(s')\big)\big)$$

Étant donné que causer et empêcher le mal et le bien sont distincts et potentiellement conflictuels, la notion d'optimalité devient subjective. De ce fait, nous voulons être en mesure de les exprimer de manière explicite et nous n'agrégeons pas les quatre aspects car nous risquons de perdre des informations nécessaires pour la prise de décision. Au regard des propriétés 1 et 2, une question se pose : « La fin justifie-t-elle les moyens ? En d'autres termes, cherchons-nous à produire le plus grand bien, qui peut résulter d'un mal, ou nous concentrons-nous sur le fait de faire le moins de mal possible, ce qui peut dans certaines circonstances empêcher de faire beaucoup de bien ? » Comme il n'y a pas de hiérarchie entre les valeurs, nous estimons qu'il est plus important pour l'agent d'éviter de causer du mal autant que possible, puis de se focaliser sur le fait de faire autant de bien que possible dans l'espace de décision restant. À cette fin, nous utilisons un ordre lexicographique qui consiste à privilégier les politiques qui causent le moins de mal, puis, à partir de l'ensemble de ces politiques avec un minimum de mal, nous choisissons celles qui font le plus de bien, comme le montrent les définitions suivantes.

Définition 6 *Les politiques optimales π^*_B par rapport aux valeurs négatives sont données par :*

$$\pi^*_B \in \operatorname*{argmin}_\pi V^\pi_\nabla(s) - V^\pi_{\overline{\nabla}}(s) + \epsilon V^\pi_{\overline{\nabla}}(s) \text{ où } \epsilon > 0.$$

Le troisième terme, c'est-à-dire la valeur pondérée du mal empêché, permet de mesurer l'importance de la prévention du mal, qui empêche l'agent d'être absous de causer du mal en le réparant (totalement ou partiellement). Cela incite l'agent à éviter complètement le mal, au lieu de le causer intentionnellement pour le réparer plus tard. En outre, si seule une soustraction entre le mal causé et le mal empêché était prise en compte, les politiques dans lesquelles un mal a été causé et entièrement réparé ne se distingueraient pas de celles dans lesquelles aucun mal n'a été causé. Cela s'applique également aux politiques optimales en ce qui concerne les valeurs positives du contexte.

Définition 7 *Les politiques optimales π^*_G par rapport aux valeurs positives, prises parmi les politiques optimales par rapport aux valeurs négatives, sont données par :*

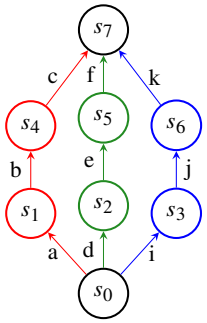
$$\pi^*_G \in \underset{\pi \in \pi^*_B}{\operatorname{argmax}} \; V^\pi_\Delta(s) - V^\pi_{\overline{\Delta}}(s) - \epsilon' V^\pi_{\overline{\Delta}}(s) \text{ où } \epsilon' > 0.$$

Remarquons que la politique optimale par rapport au bien est choisie parmi les politiques qui sont déjà optimales par rapport au mal. Ce choix contraint la notion de politique optimale par rapport au bien, mais permet de nous assurer que l'agent ne considérera pas comme optimal de faire du mal pour obtenir un plus grand bien.

Propriété 3 *La résolution d'un E-MDP en utilisant les définitions 6 et 7 nous permet de calculer une politique qui satisfait la propriété 1 lorsqu'une telle politique existe, ou qui satisfait la propriété 2 sinon.*

Démonstration (Esquisse). Si une politique satisfaisant la propriété 1 existe, alors calculer une politique grâce à la définition 6 puis à la définition 7 retourne cette politique. En effet, si une telle politique existe, notons la π^*, alors elle est telle que pour toute autre politique π, et pour tout $s \in \mathcal{S}, V^{\pi^*}_\star(s) \geq V^\pi_\star(s)$ (resp. $V^{\pi^*}_\star(s) \leq V^\pi_\star(s)$) pour $\star \in \{\Delta, \overline{\nabla}\}$ (resp. $\star \in \{\overline{\Delta}, \nabla\}$). Par conséquent, π^* est optimale pour les critères données par les définitions 6 et 7 et sera donc retournée par tout algorithme optimisant ces derniers. S'il n'existe aucune politique satisfaisant la propriété 1, alors tout algorithme optimisant les critères donnés par les définitions 6 et 7 retournera une politique satisfaisant la propriété 2 par définition. □

Une fois que ces deux critères ont été satisfaits, nous obtenons un ensemble de politiques où l'agent fait le plus de bien possible tout en causant le moins de mal possible. Pour illustrer ces politiques optimales, nous présentons en figure 1 ci-dessous un exemple comparatif de politiques.



T	$\mathcal{E}(T)_n$	$\mathcal{E}(T)_p$
(s_0, a, s_1)	1	1
(s_1, b, s_4)	1	1
(s_4, c, s_7)	0	1
(s_0, d, s_2)	1	1
(s_2, e, s_5)	1	1
(s_5, f, s_7)	−1	0
(s_0, i, s_3)	1	1
(s_3, j, s_6)	0	1
(s_6, k, s_7)	0	0

FIGURE 1 – La figure présente trois politiques : rouge, verte et bleue. La table indique les évaluations morales positives ($\mathcal{E}(T)_p$) et négatives ($\mathcal{E}(T)_n$) pour chaque transition de ces politiques. Comme nous l'avons mentionné, une politique qui satisfait la propriété 1 n'existe pas toujours. Dans cet exemple, nous pouvons facilement voir qu'aucune politique n'optimise tous les critères en même temps. En effet, la politique rouge maximise le bien causé ($\Delta = 3$ contre $\Delta = 2$ pour les politiques bleue et verte), la politique verte maximise la quantité de mal réparé ($\overline{\nabla} = 1$ contre $\overline{\nabla} = 0$ pour les politiques rouge et bleu), tandis que la politique bleu minimise la quantité de mal causé ($\nabla = 1$ contre $\nabla = 2$ pour les politiques rouge et verte). En utilisant les définitions 6 et 7 nous pouvons déduire l'ordre de préférence suivant : la politique bleue est la meilleure politique (c'est celle qui cause le moins de mal), la politique verte est la deuxième meilleure (elle cause autant de mal que la rouge, mais en répare une partie), et la politique rouge est la moins bonne (même si c'est celle qui fait le plus de bien).

4.2 Traitement des tâches amorales

Comme la fonction de récompense éthique ne prend en compte que la moralité, le traitement d'une tâche amorale nécessite un traitement particulier. Une première approche peut consister à considérer que la réalisation de cette tâche promeut une valeur positive spécifique. Ainsi, cette valeur particulière servira d'incitation (dans un sens éthique) pour que l'agent s'efforce également d'atteindre son objectif tout en se comportant de manière éthique. Cependant, comme il n'y a pas de hiérarchie entre les valeurs, si le modèle contient d'autres valeurs morales positives, la tâche amorale pourrait être négligée au profit d'autres valeurs positives. Pour cette raison, nous ajoutons explicitement un autre critère d'optimisation. Nous supposons alors que nous avons une fonction de récompense classique $\mathcal{R}_T$ et, étant donnée une politique optimale par rapport aux valeurs positives, nous nous servons de cette fonction de récompense pour sélectionner la politique la plus optimale au regard de la tâche.

5 Résolution des E-MDP

Pour résoudre les E-MDP, nous adaptons l'algorithme VI pour prendre en compte les spécificités de la fonction de récompense éthique (voir section 4). Notre algorithme fournit une politique non déterministe, c'est-à-dire que plu-

sieurs actions optimales sont associées à chaque état sans choix probabiliste. Remarquons qu'il n'est pas possible de calculer de manière simultanée les Q-valeurs selon chaque critère. En effet, le calcul des Q-valeurs s'appuie sur les actions optimales calculées précédemment. Or, il est possible que des actions non optimales par rapport au mal le soient par rapport au bien. Ainsi, les Q-valeurs propagées d'état en état ne se fonderont pas sur les mêmes actions selon le critère. Pour cette raison, les Q-valeurs et la politique doivent être calculées de manière successive et affinées : une politique par rapport à un critère doit être calculée en ne considérant que les actions optimales pour le critère précédent. Notre approche suit donc les étapes ci-après :

1. Calculer les Q-valeurs en fonction du mal et en extraire une politique optimale, en partant d'une politique avec toutes les actions possibles pour chaque état.

2. Calculer les Q-valeurs en fonction du bien et en extraire une politique optimale, en ne considérant que les actions de la politique optimale par rapport au mal dans chaque état.

3. Si une tâche doit être accomplie (voir section 4.2), calculer les Q-valeurs en fonction de $\mathcal{R}_T$ et en extraire une politique optimale, en ne considérant, dans chaque état, que les actions de la politique optimale par rapport au bien.

Cette procédure assure que la politique optimale finale est celle qui adhère d'abord aux critères éthiques puis, s'il y a une tâche amorale, qu'elle accomplisse la tâche si possible. Pour chaque étape, nous utilisons une variante de l'algorithme VI dont la forme générique est présentée dans l'algorithme 1.

— Le paramètre ★ indique le critère pour lequel les politiques sont calculées : $\nabla\overline{\nabla}$ est le critère par rapport au mal et $\Delta\overline{\Delta}$ le critère par rapport au bien.
— Pour le critère $\nabla\overline{\nabla}$, la politique d'entrée est une politique générique qui considère toutes les actions possibles pour chaque état. Pour le critère $\Delta\overline{\Delta}$, la politique d'entrée est la politique optimale par rapport au mal, qui ne considère que les actions optimales pour le mal. Dans le cas où une tâche est présente, la politique d'entrée ne considère que les actions optimales extraites du critère $\Delta\overline{\Delta}$.
— La fonction f calcule les Q-valeurs soit par rapport au mal, soit par rapport au bien, soit classiquement par rapport à $\mathcal{R}_T$. Dans le cas du mal ou du bien, les formules suivantes sont celles présentées dans les définitions 6 et 7. Formellement :

$$f(Q_{\nabla\overline{\nabla}}(s,a)) = \sum_{s'} T(t)\Big[\big[R_{\nabla}(t) - R_{\overline{\nabla}}(t) + \epsilon R_{\overline{\nabla}}(t)\big] + \gamma \min_{a'} Q_{\nabla\overline{\nabla}}(s',a')\Big],$$

$$f(Q_{\Delta\overline{\Delta}}(s,a)) = \sum_{s'} T(t)\Big[\big[R_{\Delta}(t) - R_{\overline{\Delta}}(t) - \epsilon' R_{\overline{\Delta}}(t)\big] + \gamma \max_{a'} Q_{\Delta\overline{\Delta}}(s',a')\Big].$$

où $t = (s,a,s'), \epsilon > 0, \epsilon' > 0$.
— De même, la fonction *Opt* se concentre sur la valeur minimale ou maximale donnée par Q pour une action $a \in A$ en fonction de ★. Pour le mal, nous voulons les politiques dans lesquelles le moins de mal a été fait. Pour le bien, nous voulons les politiques dans lesquelles le plus de bien a été fait. Formellement :

$$Opt_{\nabla\overline{\nabla}}(Q_{\nabla\overline{\nabla}},\pi,s) = \big\{a \in \pi(s) : Q_{\nabla\overline{\nabla}}(s,a) = \operatorname*{argmin}_{\pi(s)} Q_{\nabla\overline{\nabla}}(s,\pi(s))\big\},$$

$$Opt_{\Delta\overline{\Delta}}(Q_{\Delta\overline{\Delta}},\pi,s) = \big\{a \in \pi(s) : Q_{\Delta\overline{\Delta}}(s,a) = \operatorname*{argmax}_{\pi(s)} Q_{\Delta\overline{\Delta}}(s,\pi(s))\big\}.$$

Algorithme 1 Itération de la valeur lexicographique

Entrée : la politique π et le critère ★
Sortie : la politique optimale $\pi^*_{\bigstar}$

for all $s \in S$ **do**
  for all $a \in \pi(s)$ **do**
    $Q_{\bigstar}(s,a) \leftarrow 0$
  end for
end for
$\Delta \leftarrow 0$
while $\Delta \geq \Delta_o$ **do**
  $\Delta \leftarrow 0$
  for all $s \in S$ **do**
    for all $a \in \pi(s)$ **do**
      $\texttt{temp} \leftarrow Q_{\bigstar}(s,a)$
      $Q_{\bigstar}(s,a) \leftarrow f(Q_{\bigstar}(s,a))$
      $\Delta \leftarrow \max(\Delta, |\texttt{temp} - Q_{\bigstar}(s,a)|)$
    end for
  end for
end while
$\pi^*_{\bigstar}(s) \leftarrow Opt_{\bigstar}(Q^*_{\bigstar},\pi,s) \quad \forall s \in S$
retourner π^*

6 Cadres éthiques dans les E-MDP

Afin de mettre en application les concepts présentés en sections 4 et 5, nous modélisons les cadres éthiques considérés par Svegliato *et al.* [17] : la *théorie du commandement divin* (DCT), les *devoirs prima facie* (PFD) et l'*éthique de*

la vertu (VE). Ces cadres éthiques sont intéressants car ils se focalisent respectivement sur les états, les couples état-action et les transitions. De plus, Svegliato *et al.* ont produit des résultats expérimentaux qui peuvent être comparés à notre modèle. Pour chaque cadre éthique, nous définissons le contexte éthique et les contraintes qu'il implique sur la fonction de l'évaluation morale.

Dans la suite de l'article, nous désignons par T l'ensemble des transitions avec une probabilité non nulle : $T = \{(s, a, s') \in S \times A \times S \mid \mathcal{T}(s, a, s') > 0\}$.

6.1 Théorie du commandement divin

La DCT suppose que certains états sont interdits, et que l'agent cause du mal en y entrant. Nous définissons ces états ci-dessous et les contraintes qu'ils imposent à la fonction de valeur morale.

Définition 8 (États interdits) *Soit $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_k\}$ un ensemble d'ensembles d'états interdits $\mathcal{F}_i \subseteq S$. Chaque $\mathcal{F}_i$ est associé à une valeur négative $v_i \in \mathcal{B}_C$ du contexte.*

Le fait de disposer de plusieurs sous-ensembles d'états interdits nous permet de les associer à des valeurs morales différentes. Par exemple, un état pourrait être interdit pour éviter de mettre en danger des personnes, tandis qu'une autre pourrait être interdit pour protéger la vie privée des personnes. Comme il s'agit de valeurs différentes, chaque état fera partie d'un sous-ensemble différent d'états interdits.

Transiter dans un état appartenant à un sous-ensemble interdit promeut la valeur négative associée, car l'agent cause du mal en faisant cela.

Définition 9 (Entrée dans un état interdit) *Les transitions qui se terminent dans un état interdit promeuvent la valeur associée.*

$$\forall (s, a, s') \in T, \forall \mathcal{F}_i \in \mathcal{F} \text{ t.q. } s' \in \mathcal{F}_i : \mathcal{E}(s, a, s')_i = 1$$

Lorsque l'agent se trouve dans un état interdit, il doit en sortir le plus rapidement possible. Pour inciter l'agent à quitter ces états, transiter d'un état interdit vers un état qui ne l'est pas permet d'éviter le mal.

Définition 10 (Sortie d'un état interdit) *Les transitions qui commencent dans un état interdit et se terminent dans un état qui n'est pas interdit trahissent la valeur associée.*

$$\forall (s, a, s') \in T, \forall \mathcal{F}_i \in \mathcal{F} \text{ t.q. } (s \in \mathcal{F}_i \wedge s' \notin \mathcal{F}_i) : \mathcal{E}(s, a, s')_i = -1$$

Les transitions dans lesquelles ni l'état de départ ni l'état résultant ne sont interdits ont une évaluation morale qui ne considère aucune valeur.

Définition 11 (Autres transitions)

$$\forall (s, a, s') \in T, \forall \mathcal{F}_i \in \mathcal{F} \text{ t.q. } (s \notin \mathcal{F}_i \wedge s' \notin \mathcal{F}_i) : \mathcal{E}(s, a, s')_i = 0$$

La figure 2 illustre les définitions précédentes.

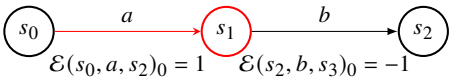


FIGURE 2 – Un exemple de cadre DCT. Soit le contexte éthique $C = \langle -1 \rangle$ et les états interdits $\mathcal{F}_0 = \{s_1\}$. L'état s_1 est interdit et la transition qui y entre promeut le mal. La transition qui en sort et qui arrive en s_2 (qui n'est pas interdit) trahit la valeur négative, et empêche ainsi le mal.

6.2 Devoirs *Prima Facie*

Les PFD supposent l'existence de devoirs fondamentaux que l'agent doit accomplir. Nous définissons ci-dessous ces devoirs et les contraintes qu'ils imposent à la fonction d'évaluation morale.

Définition 12 (Devoir) *Soit $\mathcal{D} = \{\delta_1, \ldots, \delta_k\}$ un ensemble de devoirs. Un devoir δ_i est un ensemble de couples état-action : $\delta_i = \{(s_0, a_0), \ldots, (s_n, a_n)\}$. Chaque devoir est associé à une valeur négative v_i dans C.*

L'agent *accomplit le devoir* en choisissant l'action du devoir dans l'état associé, et *néglige le devoir* si une action différente est choisie.

Accomplir ou négliger un devoir peut être interprété de deux manières : **positivement** où l'agent cause du bien chaque fois qu'un devoir est accompli, et **négativement** où l'agent cause du mal chaque fois qu'un devoir est négligé. Dans ce cas précis, les notions de *causer le bien* et *causer le mal* deviennent duales, et nous pouvons donc soit nous concentrer sur les devoirs accomplis (sur la maximisation du bien), soit nous concentrer sur les devoirs négligés (sur la minimisation du mal). Nous choisissons ici de nous concentrer sur l'interprétation négative. Comme les devoirs sont considérés comme fondamentaux, c'est-à-dire que l'agent est tenu de les accomplir, le fait de revenir à un état où un devoir a été précédemment négligé et de l'accomplir ne sera pas considéré comme empêchant un mal. Par conséquent, seul un nouveau mal peut être évité et, en raison de la nature des devoirs, le mal causé ne peut pas du tout être empêché dans ce cadre.

Définition 13 (Négliger un devoir) *Négliger un devoir δ_i consiste à passer par des transitions (s, a, s') où il existe $(s, a') \in \delta_i$ et où $(s, a) \notin \delta_i$. Dans ce cas, (s, a, s') promeut*

la valeur négative v_i.

$$\forall (s,a,s') \in T \text{ t.q. } \exists\, \delta_i \in \mathcal{D}, (s,a') \in \delta_i \ \wedge\ (s,a) \notin \delta_i : \\ \mathcal{E}(s,a,s')_i = 1.$$

Les autres transitions ne prennent en considération aucune valeur dans leur évaluation.

Définition 14 (Autres transitions) *Toutes les transitions (s,a,s') où ni l'état de départ, ni l'action ne font partie d'un devoir δ_i ont une évaluation morale non applicable pour la valeur négative associée $v_i \in \mathcal{B}_C$:*

$$\forall (s,a,s') \in T, \forall\, \delta_i \in \mathcal{D} \text{ t.q. } (s,a) \notin \delta_i : \mathcal{E}(s,a,s')_i = 0$$

La figure 3 illustre les définitions précédentes.

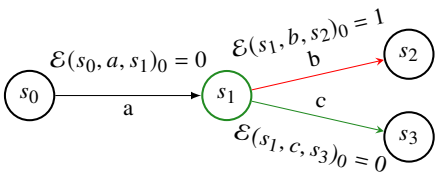


FIGURE 3 – Un exemple de cadre PFD. Soit un contexte $C = \langle -1 \rangle$ et le devoir $\delta = (s_1, c)$. Choisir l'action c dans s_1 accomplit le devoir et ne cause pas de mal. Le choix de b promeut la valeur négative et cause donc un mal.

6.3 L'éthique de la vertu

Le cadre VE suppose qu'il existe un exemplaire moral qui se comporte comme nous voulons que notre agent se comporte. L'agent doit alors se conformer à ce comportement, représenté par des *trajectoires morales* (MT).

Définition 15 (Trajectoire morale) *Une trajectoire morale d'un exemplaire moral est un ensemble $MT = \{(s_0,a_0,s'_0),\ldots,(s_n,a_n,s'_n)\}$ de transitions. Une trajectoire morale est associée à deux valeurs morales v_n et v_p telles que $C_n = -1$ et $C_p = 1$: v_n est une valeur négative et v_p une valeur positive.*

Comme les résultats des actions ne sont pas déterministes, il peut y avoir des sorties de trajectoire non intentionnelles : l'agent essaie de suivre la trajectoire mais n'y parvient pas en raison d'effets stochastiques. Dans les autres cadres éthiques, ceci n'a pas d'importance car ils reposent sur des interdictions, i.e. ne pas atteindre un état ou ne pas négliger un devoir. Dans l'éthique de la vertu, au contraire, ces sorties accidentelles ont leur importance car ce n'est pas tant le résultat de l'action qui compte que l'intention qui était exprimée derrière. Deux valeurs sont alors nécessaires pour prendre cela en compte, l'une pour récompenser le suivi intentionnel ou pénaliser les sorties non intentionnelles, et l'autre pour pénaliser les sorties intentionnelles. Dans la suite, nous distiguons les *trajectoires moralement bonnes* (MGT) et les *trajectoires moralement mauvaises* (MBT), respectivement les trajectoires morales que nous voulons que l'agent suive et les trajectoires que nous voulons que l'agent évite. Les contraintes sur la fonction d'évaluation morale dépendent alors du type de MT.

6.3.1 Trajectoires moralement bonnes

Suivre une trajectoire définie par un exemplaire moral est considéré comme moralement bon. Ainsi, les transitions qui font partie d'une MGT promeuvent la valeur positive associée et causent du bien.

Définition 16 (Maintenir une MGT) *Les transitions qui appartiennent à une MGT promeuvent la valeur positive associée.*

$$\forall (s,a,s') \in MGT : \mathcal{E}(s,a,s')_p = 1.$$

Lorsque l'agent choisit une action qui suit une MGT mais que l'état successeur n'est pas celui attendu, il fait une sortie accidentelle de MGT. Comme l'agent n'avait pas l'intention de quitter la trajectoire, il ne fait qu'empêcher le bien, sans pour autant causer du mal.

Définition 17 (Sortie accidentelle d'une MGT) *Les transitions qui ont le même état de départ et la même action qu'une transition dans une MGT, mais dont l'état successeur est différent, trahissent sa valeur positive.*

$$\forall (s,a,s') \in T \text{ t.q. } \exists (s_g,a_g,s'_g) \in MGT \\ \text{où } s = s_g \wedge a = a_g \wedge s' \neq s'_g \ \wedge\ (s,a,s') \notin MGT : \\ \mathcal{E}(s,a,s')_p = -1.$$

Tandis que dans une MGT, si l'agent choisit une action qui n'en fait pas partie tout en ayant la possibilité de choisir une action correcte, il choisit de sortir intentionnellement de la trajectoire. Par conséquent, non seulement il empêche le bien, mais il cause aussi du mal.

Définition 18 (Sortie intentionnelle d'une MGT) *Les transitions ayant un état de départ qui est un état de départ dans un MGT, mais dont l'action est différente de celle du MGT, promeuvent sa valeur négative et trahissent sa valeur positive.*

$$\forall (s,a,s') \in T \text{ t.q. } \exists (s_g,a_g,s'_g) \in MGT \\ \text{où} \quad s = s_g \wedge a \neq a_g : \\ \mathcal{E}(s,a,s')_n = 1 \quad et \quad \mathcal{E}(s,a,s')_p = -1.$$

Pour toutes les autres transitions, l'évaluation morale ne considère aucune valeur positive ou négative.

Définition 19 (Autres transitions) *Les transitions qui ne satisfont pas les définitions 16, 17, 18 sont évaluées comme suit :*

$$\mathcal{E}(s, a, s')_{n,p} = 0.$$

La figure 4 illustre les définitions précédentes.

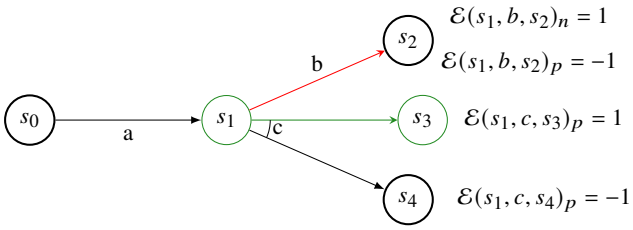


FIGURE 4 – Un exemple de MGT pour un cadre VE. Soit le contexte $C = \langle -1, 1\rangle$ et une MGT contenant (s_1, c, s_3). Si l'agent effectue l'action c, il peut se retrouver en s_3 et cause le bien, ou en s_4 et empêche le bien. Si l'agent choisit b, il sort intentionnellement de la trajectoire, causant du mal.

6.3.2 Trajectoires moralement mauvaises

Un agent se trouvant dans un état qui fait partie d'une MBT ne doit pas y rester. Par conséquent, les transitions qui partent de cet état et qui y retournent causent du mal, même si elle n'appartiennent pas à la MBT.

Définition 20 (Rester dans une MBT) *Les transitions, dont l'état de départ et l'état sucesseur sont les mêmes et qui sont aussi un état de départ d'une transition dans une MBT, promeuvent la valeur négative.*

$$\forall (s, a, s') \in T \text{ t.q. } \exists (s_b, a_b, s'_b) \in MBT$$
$$où\ s = s_b \wedge s = s' : \mathcal{E}(s, a, s')_n = 1.$$

Bien entendu, le fait de suivre une MBT est également considéré comme causant du mal.

Définition 21 (Maintenir une MBT) *Les transitions qui appartiennent à une MBT promeuvent sa valeur négative.*

$$\forall (s, a, s') \in MBT : \mathcal{E}(s, a, s')_n = 1.$$

À cause de la stochasticité, les sorties accidentelles des MBT sont également possibles. Cela signifie que l'agent a choisi de rester dans une MBT mais a échoué. Une telle transition cause certes du bien car l'agent est sorti de la MBT mais cause également du mal car l'intention était d'y rester.

Définition 22 (Sortie accidentelle d'une MBT) *Les transitions ayant le même état de départ et action qu'une transition dans une MBT, mais dont l'état successeur est différent, promeuvent à la fois sa valeur négative et positive.*

$$\forall (s, a, s') \in T \text{ t.q. } \exists (s_b, a_b, s'_b) \in MBT$$
$$où \quad s = s_b \wedge a = a_b \wedge s' \neq s'_b :$$
$$\mathcal{E}(s, a, s')_n = 1 \quad et \quad \mathcal{E}(s, a, s')_p = 1.$$

Dans un MBT, si l'agent choisit d'en sortir alors qu'il a la possibilité de le suivre, il empêche le mal et cause le bien.

Définition 23 (Sortie intentionnelle d'une MBT) *Les transitions ayant un état de départ qui est un état de départ d'une transition d'une MBT, mais dont l'action ne fait pas partie de celle-ci, trahissent la valeur négative et promeuvent la valeur positive.*

$$\forall (s, a, s') \in T \text{ t.q. } \exists (s_b, a_b, s'_b) \in MBT$$
$$où \quad s = s_b \wedge a \neq a_b \wedge \exists a' = a_b :$$
$$\mathcal{E}(s, a, s')_n = -1 \quad et \quad \mathcal{E}(s, a, s')_p = 1.$$

Pour toutes les autres transitions, l'évaluation morale ne considère aucune valeur positive ou négative.

Définition 24 (Autres transitions) *Les transitions qui ne satisfont pas les définitions 20, 21, 22, 23 sont évaluées comme suit :*

$$\mathcal{E}(s, a, s')_{n,p} = 0.$$

La figure 5 illustre les définitions précédentes.

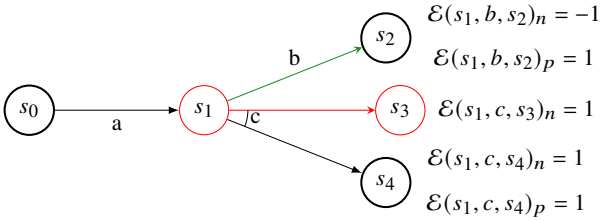


FIGURE 5 – Un exemple de MBT pour un cadre VE. Soit le contexte $C = \langle -1, 1\rangle$ et le MBT contenant (s_1, c, s_3). Si l'agent effectue l'action c, il peut se retrouver en s_3 et ne cause que du mal, ou en s_4 et cause à la fois du mal et du bien, car l'agent a choisi de suivre la mauvaise trajectoire, mais a eu une sortie accidentelle. Si l'agent choisit b, il choisit activement de sortir de la trajectoire, causant ainsi du bien et empêchant le mal.

7 Expériences

Pour évaluer notre modèle, nous avons implémenté le scénario de véhicule autonome proposé par Svegliato *et*

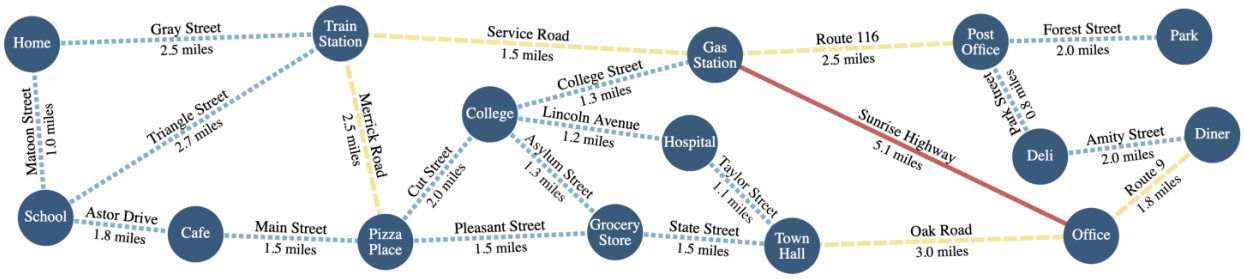


FIGURE 6 – Graphe des états (lieux et routes) utilisés dans nos expériences, tiré de Svegliato *et al.* [17].

al. [17]. Dans celui-ci, un véhicule doit naviguer dans une zone urbaine où différents lieux sont représentés par des états reliés entre eux par d'autres états représentant des routes. La figure 6 représente ces états et la manière dont ils sont reliés entre eux. Les états de route sont caractérisés par des informations sur leur type (encodant les limitations légales de vitesse), la circulation piétonne présente (élevée ou faible) et la vitesse à laquelle le véhicule roule (élevée, normale ou faible). Lorsque l'agent se trouve dans un état de lieu, il peut choisir sur quelle route il va tourner et s'il se trouve sur une route, il peut choisir la vitesse à laquelle il conduira jusqu'au prochain lieu.

Les expériences consistent en trois tâches de navigation entre un état initial et un état but donnés tout en respectant des contraintes éthiques qui peuvent être combinées. Les différentes contraintes éthiques pour chaque cadre sont données ci-après. (**DCT - H**) Tous les états où l'agent roule à vitesse élevée sont interdits. (**DCT - I**) Les états où l'agent roule à vitesse normale ou élevée alors qu'il y a une circulation piétonne élevée sont interdits. (**PFD -** δ_1) Dans tous les états où la circulation piétonne est faible, l'agent a le devoir de conduire à une vitesse normale ou élevée. (**PFD -** δ_2) Dans tous les états où la circulation piétonne est élevée, l'agent a l'obligation de conduire à vitesse faible. (**VE - C**) L'agent ne considère qu'une MGT composées des transitions où la conduite est à vitesse normale lorsqu'il y a une circulation piétonne faible et celles où la conduite est à faible vitesse lorsque la circulation piétonne est élevée. (**VE - P**) L'agent ne considère qu'une MBT composée de toutes les transitions qui atteignent un états de type « Autoroute » ou « École ».

Le table 1 indique le *prix de la moralité*, qui est le pourcentage de perte de valeur entre les politiques éthiques et les politiques optimales en absence de tout cadre éthique. Naturellement, le prix de la moralité est nul lorsqu'il n'y a pas de cadre éthique. De plus, il est important de noter que les valeurs prises par le prix de la moralité n'ont de sens que relativement au modèle de l'application, càd la manière dont la récompense amorale est construite. Par exemple, DCT-I et PFD-δ_1 ont un prix de la moralité nul car ces contraintes éthiques n'interdisent pas les trajectoires optimale amorale. Remarquons que combiner des contraintes,

Éthique	Contraintes	Tâche 1	Tâche 2	Tâche 3
Aucune	-	0%	0%	0%
DCT	H	25.82%	26.25%	34.22%
	I	0%	0%	0%
	H ∪ I	36.12%	36.98%	43.59%
PFD	δ_1	0%	0%	0%
	δ_2	15.47%	15.98%	22.46%
	$\delta_1 \cup \delta_2$	15.47%	15.98%	22.46%
VE	C	36.12%	36.98%	43.59%
	P	11.54%	52.22%	0%
	C ∪ P	64.38%	127.73%	50.28%

TABLE 1 – Prix de la moralité dans l'état de départ.

même certaines ayant individuellement un prix de la moralité nul, peut entraîner un prix de la moralité plus élevé. En effet, une contrainte individuelle peut impacter la politique morale optimale qui aurait été calculée indépendamment avec l'autre contrainte. Remarquons également que les tâches 1, 2 et 3 sont de difficulté croissante pour satisfaire tout à la fois le but amoral et les contraintes éthiques. Il est important de noter que le but des expériences n'est pas nécessairement d'obtenir des résultats quantifiés en termes de performance plus ou moins importante, mais d'illustrer le fait que notre modèle capture les différentes contraintes éthiques considérées par Svegliato *et al* [17]. L'objectif est donc de montrer que le prix de la moralité évolue de manière similaire pour les mêmes tâches et contraintes, mettant ainsi en lumière que le modèle est générique et peut représenter différents principes éthiques.

Nous voyons alors que de manière cohérente le prix de la moralité augmente puisque le modèle privilégie l'éthique sur le but amoral. Lorsque nous comparons ces résultats à ceux de Svegliato *et al.* [17] nous retrouvons les mêmes tendances dans la dégradation du prix de la moralité, à savoir une dégradation avec la difficulté de la tâche et avec un cumul des contraintes. La différence entre notre approche et celle de Svegliato *et al.* est que ces derniers font de l'optimisation sous contraintes, c'est-à-dire qu'ils calculent la politique optimale amorale et la dégrade jusqu'à satisfaire les contraintes éthiques, tandis que nous calculons des

politiques qui adhèrent d'abord aux cadres éthiques, puis sont optimisées sur la base des critères restants (comme les tâches amorales). Une autre différence est que notre approche est générique au sens où l'éthique est définie dans le modèle – dans la récompense elle-même et la fonction de valeur à partir des concepts de valeurs – et non pas comme contrainte exogène qui est une fonction spécifiquement définie avec des éléments qui lui sont propres.

8 Conclusion

Le modèle E-MDP intègre explicitement l'éthique dans les MDP comme des valeurs morales positives et négatives, promues ou trahies. Il utilise une fonction de récompense sous forme de quadruplet pour optimiser d'abord le fait de causer le moins de mal possible, puis de causer le plus de bien possible. Nous avons illustré ce modèle sur trois cadres éthiques, DCT, PFD et VE, qui peuvent être combinés. Enfin, nous avons comparé nos résultats avec ceux de Svegliato *et al.* [17], exhibant des similarités qui nous amènent à penser que notre approche est une manière générique adéquate d'intégrer l'éthique dans les MDP.

Comme perspectives sur la représentation de l'éthique, ce modèle doit être étendu pour des valeurs hiérarchiques mais aussi pour traiter des agents multiples. Intuitivement, chaque agent devrait avoir son propre contexte éthique, et il serait intéressant que des valeurs morales représentent la façon dont les agents prennent en compte les autres. Par exemple, une valeur égalitaire pourrait être promue lorsqu'une décision favorise ou trahit le même nombre de valeurs pour tous les agents.

Comme perspectives algorithmiques, nous pouvons envisager une extension de notre modèle aux observations partielles (PO-MDP). Il serait également intéressant d'étudier l'utilisation d'autres algorithmes de calcul de politique pour trouver les politiques optimales plus efficacement, tel que l'emploi de techniques de recherche heuristique.

Références

[1] Abel, David, James MacGlashan et Michael L. Littman: *Reinforcement Learning as a Framework for Ethical Decision Making*. Dans *AAAI Workshop on AI, Ethics, and Society*, pages 54–61, 2016.

[2] Averroes: *The Decisive Treatise*, 1178.

[3] Bellman, Richard: *A Markovian Decision Process*. Indiana Univ. Math. J., 6 :679–684, 1957.

[4] Bench-Capon, Trevor: *Persuasion in Practical Argumentation Using Value-Based Argumentation Frameworks*. J. Log. Comput., 13(3) :429–448, 2003.

[5] Berreby, Fiona, Gauvain Bourgne et Jean Gabriel Ganascia: *A Declarative Modular Framework for Representing and Applying Ethical Principles*. Dans *16e AAMAS*, page 96–104, 2017.

[6] Cointe, Nicolas, Grégory Bonnet et Olivier Boissier: *Ethical Judgment of Agents' Behaviors in Multi-Agent Systems*. Dans *15e AAMAS*, pages 1106–1114, 2016.

[7] De Moura, Nelson, Raja Chatila, Katherine Evans, Stéphane Chauvier et Ebru Dogan: *Ethical Decision Making for Autonomous Vehicles*. Dans *IEEE Intelligent Vehicles Symposium*, pages 2006–2013, 2020.

[8] Fard, Mahdi Milani et Joelle Pineau: *Non-Deterministic Policies in Markovian Decision Processes*. Journal of Artificial Intelligence Research, 40 :1–24, 2011.

[9] Foot, Philippa: *The Problem of Abortion and the Doctrine of the Double Effect*. Oxf. Rev., 5 :5–15, 1967.

[10] Howard, Ronald A.: *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

[11] Lorini, Emiliano: *On the Logical Foundations of Moral Agency*. Dans *11e DEON*, tome 7393 de *LNCS*, pages 108–122. Springer-Verlag, 2012.

[12] McConnell, Terrance: *Moral Dilemmas*. Dans Zalta, Edward N. (rédacteur) : *The Stanford Encyclopedia of Philosophy*. 2018.

[13] Nashed, Samer, Justin Svegliato et Shlomo Zilberstein: *Ethically Compliant Planning within Moral Communities*. Dans *4e AIES*, pages 188–198, 2021.

[14] Puterman, M.: *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. Wiley & Sons, 1994.

[15] Rickless, Samuel C.: *The Doctrine of Doing and Allowing*. Philos. Rev., 106(4) :555–575, 1997.

[16] Rodriguez-Soto, Manel, Maite Lopez-Sanchez et Juan A. Rodriguez-Aguilar: *A Structural Solution to Sequential Moral Dilemmas*. Dans *19e AAMAS*, pages 1152–1160, 2020.

[17] Svegliato, Justin, Samer Nashed et Shlomo Zilberstein: *Ethically Compliant Sequential Decision Making*. Dans *35e AAAI*, pages 11657–11665, 2021.

[18] Timmons, Mark: *Moral Theory : an Introduction*. Rowman & Littlefield Publishers, 2012.

[19] Watkins, Christopher: *Learning From Delayed Rewards*. Thèse de doctorat, Cambridge Univ., 1989.

[20] Wiener, Yoash: *Forms of Value Systems : A Focus on Organisational Effectiveness and Cultural Change and Maintenance*. Acad. Manage. Rev., 13(4) :534–545, 1988.

[21] Wu, Yueh Hua et Shou De Lin: *A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents*. Dans *32e AAAI*, pages 1687–1694, 2018.

Session 5 : Planification

Actes JIAF-JFPDA 2023

Mixture of Public and Private Distributions in Imperfect Information Games.

Jérôme Arjonilla[1] **Tristan Cazenave**[1] **Abdallah Saffidine**[2]

[1] LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France
[2] University of New South Wales, Sydney, Australia
jerome.arjonilla@lamsade.dauphine.fr
tristan.cazenave@lamsade.dauphine.fr
abdallah.saffidine@gmail.com

Résumé

Dans les jeux à information imparfaite (par exemple, le bridge, le skat, le poker), l'une des considérations fondamentales est de déduire l'information manquante tout en évitant de divulguer des informations privées. Ne pas tenir compte de l'information privée révélée peut conduire à des performances très exploitables. Toutefois, une attention excessive à cet égard conduit à des hésitations qui ne sont plus cohérentes avec nos informations privées. Dans notre travail, nous montrons que, pour améliorer les performances, il faut choisir d'utiliser ou non l'information privée d'un joueur. Nous étendons notre travail en proposant une nouvelle distribution de croyances en fonction de la quantité d'informations privées et publiques souhaitée. Nous démontrons empiriquement une augmentation des performances et montrons que, afin d'améliorer encore les performances, la nouvelle distribution devrait être utilisée en fonction de la position dans le jeu. Nos expériences ont été réalisées sur plusieurs jeux à information imparfaite et dans plusieurs algorithmes basés sur la détermination (PIMC et IS-MCTS).

Abstract

In imperfect information games (*e.g.* Bridge, Skat, Poker), one of the fundamental considerations is to infer the missing information while at the same time avoiding the disclosure of private information. Disregarding the issue of protecting private information can lead to a highly exploitable performance. Yet, excessive attention to it leads to hesitations that are no longer consistent with our private information. In our work, we show that to improve performance, one must choose whether to use a player's private information. We extend our work by proposing a new belief distribution depending on the amount of private and public information desired. We empirically demonstrate an increase in performance and, with the aim of further improving performance, the new distribution should be used according to the position in the game. Our experiments have been done on multiple benchmarks and in multiple determinization-based algorithms (PIMC and IS-MCTS).

1 Introduction

Search in artificial intelligence has been constantly evolving over the last few decades, and game-oriented research has always been a cornerstone of this success. Chess, Go [14], Poker [2], Skat, Contract Bridge, or Starcraft [21] are among the most famous ones.

Perfect information games (Chess, Go) — where all information is available for each player — have been the most studied, and many algorithms have been able to achieve a level far beyond the level of a human professional player. On the other hand, Imperfect Information Games (IIGs) (Poker, Skat, Bridge) — where some information is hidden — have been less studied, and only a few algorithms are capable of beating professional human player [20, 2, 13].

In IIGs, the complexity is heightened by the missing information, as one must try to infer the missing information of the opponents and, at the same time, be wary to not reveal private hidden information to opponents. Among the methods used in IIGs, determinization-based algorithms — where the hidden information is fixed according to a belief distribution — such as Perfect Information Monte Carlo (PIMC) [11], Recursive PIMC [7], Information Set MCTS [5] or AlphaMu [4] achieve state-of-the-art performance in many trick-taking card games (Contract-Bridge, Skat).

In the work cited above, the determinization operates by sampling the hidden information according to the private information of a given player, *i.e.* what has happened since the beginning, from the point of view of a given agent. However, by doing so, one can indirectly reveal private infor-

mation to opponents, which can lead to a highly exploitable performance.

Recently, the concept of public knowledge [9] — where a distinction is made between observations accessible to everyone and those accessible individually — has emerged in the IIGs. This concept has resulted in many breakthroughs thanks to the decomposition, which made the calculations feasible [12, 1]. Despite this large benefit, there are limitations to its use, especially in the context of belief distribution. By completely removing the knowledge observed by the acting player, one might wonder whether or not using the private information was useful.

In this work, we analyze the impact of using one method rather than another. We also present a new belief distribution, which is a mixture of both public and private belief distribution. We extend the study by analyzing different mixtures, depending on the position within the game. Our experiments are carried out on determinization-based algorithms, which use the belief distribution to fix the incertitude.

The paper is organized as follows : the second section presents notation and current determinization-based algorithms ; Section 3 explains the different belief distributions used with their advantages and drawbacks, and presents our new belief distribution ; Section 4 empirically shows that using the new belief distribution allows us to improve past performance and the last section summarizes our work and future work.

2 Notation and Background

2.1 Notation

We use the notation based on factored-observation stochastic games (FOSGs [9]). This formalism distinguishes between private and public observations.

A game G is composed with a set $\mathcal{N} = \{1, 2, \ldots, N\}$ agents ($N \in \mathbb{N}$). The state of the game is called a **world state** $w \in \mathcal{W}$ and, in each world state, the acting player i chooses an action $a \in \mathcal{A}(w)$, where $\mathcal{A}(w)$ denotes the legal actions at w. After an action a is chosen, we reach the next world state w' from the probability distribution of playing a in w.

During the transition from w to w' by playing a, two observations are received : a **public observation** and a **private observation**. Public observation is the observation visible by every player noted $o_{pub} \in \mathcal{O}_{pub}(w, a, w')$ where $\mathcal{O}_{pub}(w, a, w')$ refers to all the possible public observations. Private observation is the observation visible by a precise player i, noted $o_i \in \mathcal{O}_i(w, a, w')$ where $\mathcal{O}_i(w, a, w')$ refers to all the possible private observations.

A *history* is a finite sequence of legal actions and world states, denoted $h^t = (w^0, a^0, w^1, a^1, ..., w^t)$. For describing the point of view of an agent i of a history h, we introduce an **infostate** $s_i(h)$. An **infostate** for agent i is a sequence of an agent's observations and actions $s_i^t = (\tilde{o}_i^0, a_i^0, \tilde{o}_i^1, a_i^1, ..., \tilde{o}_i^t)$ where $\tilde{o}_i^k = (o_{pub}^k, o_i^k)$. A **public infostate** is a sequence of public observations $s_{pub}^t = (o_{pub}^0, o_{pub}^1, ..., o_{pub}^t)$.

Determinization refers to the fact that we sample one world state according to a belief distribution of the different world states possible. Determinizing the belief distribution is not new and a similar concept exists in other formalisms such as belief state in POMDPs problems [17], occupancy-state in Dec-POMDPs problems [6].

2.2 Determinization-based algorithms

Each determinization-based algorithm has its own characteristics. Nevertheless, they share some common features such as (i) sampling a world state according to a belief distribution over the possible world states, and (ii) using a perfect information algorithm for estimating the value of the sampled world state.

The algorithms are simple and, in practice, they achieve great results, mainly due to the use of perfect information algorithms that are fast and efficient. Among the most famous perfect information algorithms, there are AlphaBeta [8], MCTS [3] or Value Network [14, 15, 16].

In the following, we present two determinization-based algorithms that are baseline and will, at a later stage, be used in our experiments.

2.2.1 PIMC

Perfect Information Monte Carlo (PIMC) is the state of the art of many IIG problems such as Contract-Bridge, Skat, and others.

The algorithm is defined in Algorithm 1 and works as follows (i) samples a world state by using the player's private information ; (ii) plays every action of the sampled world state ; (iii) estimates the new world state by using an algorithm available in perfect information setting ; (iv) repeats until the budget is over ; (v) selects the action that produces the best result in average. In practice, PIMC often uses AlphaBeta as the perfect information evaluator.

2.2.2 IS-MCTS

Information Set Monte Carlo Tree Search (IS-MCTS) [5] uses Monte Carlo Tree Search (MCTS) [3] according to a sampled world state.

MCTS is a state-of-the-art tree search algorithm in perfect information games. It works as follows (i) **selection** — selects a path of nodes based on an exploitation policy ; (ii) **expansion** — expands the tree by adding a new child node ; (iii) **playout** — estimates the child node by using an exploration policy ; (iv) **backpropagation** — backpropagates the result obtained from the playout through the nodes chosen

Algorithm 1: PIMC

```
Function PIMC(s) :
    for m ∈ Moves (s) do
        score[m] ← 0;
    end
    while budget do
        w ← InfoSampling(s);
        for m ∈ Moves (w) do
            score [m] ← score[m] + PerfectAlgo (w,
              m);
        end
    end
    return Best action on average
```

during the selection phase. In practice, MCTS often uses random playout as the perfect information evaluator, and UCB1 in the selection phase.

IS-MCTS works in a similar way to MCTS, but instead of building a tree on world states, IS-MCTS builds a tree on infostate. Yet, at each new iteration, it samples a world state according to using the player's private information and determines the dynamics with this world state (the selection and playout are done on the sampled world state).

Algorithm 2: IS-MCTS

```
Function IS-MCTS(s):
    while budget do
        w ← InfoSampling(s);
        MCTS conditioned on w.;
    end
    return Normalise visit count for each action
Function MCTS(w):
    u ← Selection(w);
    u ← Expansion (u,w);
    u ← Simulation (u,w);
    Backpropagation(u);
```

3 Belief Distributions

To present the different belief distributions, with their advantages and drawbacks, we use the following example throughout the section to facilitate understanding.

The example is based on the famous game 'Liar's Dice' (an explanation of the game is given in Subsection 4.1.2). In our case, two players play, each with 1 die of 2 sides. We denote $\{P_1 : X; P_2 : Y\}$ for player 1 has X and player 2 has Y. There are four world states possible ($w_1 = \{P_1 : 1; P_2 : 1\}$, $w_2 = \{P_1 : 1; P_2 : 2\}$; $w_3 = \{P_1 : 2; P_2 : 2\}$, $w_4 = \{P_1 : 2; P_2 : 1\}$).

For each player, there are two infostates possible and one

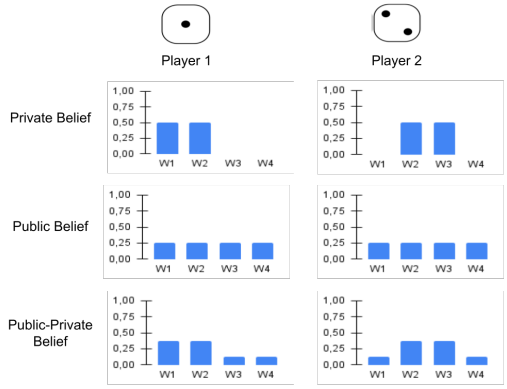


FIGURE 1 – Multiple belief distributions for the game Liar's Dice with 1 2-side die per player. Four world states possible $w_1 = \{P_1 : 1; P_2 : 1\}$, $w_2 = \{P_1 : 1; P_2 : 2\}$, $w_3 = \{P_1 : 2; P_2 : 2\}$ and $w_4 = \{P_1 : 2; P_2 : 1\}$. The Public-Private belief uses the mixture distribution with $\lambda = 0.5$.

public infostate $s_{pub} = \{o^1 = \emptyset, o^2 = \emptyset\}$ (no observation). For the player 1 we have $s_1 = \{\tilde{o}^1 = 1, \tilde{o}^2 = \emptyset\}$ or $s'_1 = \{\tilde{o}^1 = 2, \tilde{o}^2 = \emptyset\}$ (*i.e.* the player 1 observes the die it rolled but not the die rolled by the other player), and for the player 2, we have $s_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1\}$ or $s'_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2\}$ (*i.e.* the player 2 observes the die it receives but not the die received by the other player).

In the following, we suppose that the world state of this example is w_2. Therefore, for the player 1, the infostate is s_1 with two world states possible ($\{w_1, w_2\}$) and for the player 2, the infostate is s'_2 with two world states possible ($\{w_2; w_3\}$).

Figure 1 represents the different belief distributions presented throughout the section.

3.1 Private Distribution

As previously introduced, current determinization-based algorithms work by sampling world states according to the player's private information distribution, *i.e.* knowing a player's private and public observation, we sample a world state.

Let $S_j(s_i)$ be the set of possible infostate for player j conditioning to the infostate s_i of the player i.

In our example, the infostate possible for the player 2 when the player 1 has s_1 is $S_2(s_1) = \{s_2; s'_2\}$. In other world, having the die 1 for the player 1 does not exclude the player 2 to have a 1 or a 2. Yet, depending on the game, this can be restrictive, *e.g.* in trick-taking card games, if the player i has the card 'Queen of Hearts', no opponent can have it.

Definition 1 (Private Belief State) *Let $S_j(s_i)$ be the set of possible infostate for player j conditioning to the infostate s_i. Let $\Delta S_j(s_i)$ denotes the probability distribution*

over the elements of $S_j(s_i)$. *We define the private belief state as* $\Delta_i(s_i) = (\Delta S_1(s_i), \ldots, \Delta S_i(s_i), \ldots, \Delta S_N(s_i)) = (\Delta S_1(s_i), \ldots, s_i, \ldots, \Delta S_N(s_i))$.

In Figure 1, using Player 1's private belief state provides the following belief distribution $\Delta_1(s_1) = (\{s_1 : 100\%\}, \{s_2 : 50\%; s'_2 : 50\%\})$, which results in two equiprobable world states (w_1, w_2).

When using this distribution for determinization, the algorithm samples a world state (w_1 or w_2) consistent with the current player's information (s_1) and, as the state of the art in trick-taking games shows, great performance is obtained. Yet, by doing so, 3 problems arise.

(i) It is not consistent with the other player's belief, *e.g.* if we use it with the first player, the algorithm samples w_1 or w_2 but never w_3, which is nevertheless, a world state from the point of view of the player 2.

(ii) It is not able to mislead others. In our example, two actions are possible for the first player, 'I have a one' and 'I have a two'. The action 'I have a two' is a lie, however, one may want to play this action with the aim of deceiving the opponent. However, in our case only w_1 or w_2 can be sampled and, in each world, the action 'I have a two' results in a defeat because the second player will say 'This is a lie'. Therefore, lying is never an option, as it never succeeds.

(iii) It, indirectly, allows the opponents to infer our private information, *e.g.* after playing multiple matches, the second player understands that, if the first player plays 'I have a two', it is because he really has a two as it can not lie, and therefore, play to counter it.

Trying to infer the missing information is one of the key components of IIGs, and using the private belief distribution could result in a highly exploitable performance. To remove this problem, one can use public belief distribution, as presented in the next section.

3.2 Public Distribution

Recently in IIGs, many algorithms [12, 1] have been using the concept of public observation. This concept has resulted in many breakthroughs thanks to decomposition, which made the calculations feasible. One application of public observation is the creation of a public belief distribution over the world states possible according to the public observations observed so far.

Definition 2 (Public Belief State [1]) *Let* $S_j(s_{pub})$ *be the set of possible infostate for player* j *conditioning to the public infostate* s_{pub}. *Let* $\Delta S_j(s_{pub})$ *denote the probability distribution over the elements of* $S_j(s_{pub})$. *We define the public belief state as* $\Delta_{pub}(s_{pub}) = (\Delta S_1(s_{pub}), ..., \Delta S_N(s_{pub}))$.

In our example, using the public belief state from the point of view of the player 1 or player 2 would result in the same belief distribution $\Delta_{pub} = (\{s_1 : 50\%; s'_1 : 50\%\}, \{s_2 : 50\%; s'_2 : 50\%\}$. Indeed, the public infostate does not contain any information, therefore every world state is possible and equiprobable.

Using a public belief distribution instead of a private belief distribution removes the problem defined in Section 3.1.

(i) It is consistent with the other player's doubts, *e.g.* it samples the world w_3 which is a world state possible of the second player.

(ii) It is capable of misleading others, *e.g.* when sampling w_3 or w_4 the action 'I have a two' does not result in a defeat for the first player, therefore, allows the first player to play the action 'I have a two'.

(iii) It no longer reveals private information, *i.e.* as the reasoning is no longer biased toward the private information, it can not be used against it.

Nevertheless, using public distribution has a significant drawback. It does not consider a player's private information, and one might wonder whether it is useful to not use private information. In Figure 1, when using the public distribution, every world has the same probability, and this, for each player.

It is straightforward to consider that the extent to which private information should be kept hidden depends on the game being played and ; in certain games, it is not necessary to keep the information concealed.

In addition, by using public distribution, one must be cautious that there are more world states possible (*e.g.* by using private distribution, we have two world states possible and by using public distribution, we have four world states possible), which can be intractable in large games.

3.3 Mixture between public and private distribution

To solve both of the problems defined in Section 3.1 and in Section 3.2, we propose to use a mixture of private and public distribution.

Definition 3 (Public-Private Belief State (PPBS)) *Let* s_{pub} *be the public infostate associated with the infostate* s_i. *We define the public-private belief state as* $\Delta_\lambda(s_i) = (1 - \lambda)\Delta_i(s_i) + \lambda\Delta_{pub}(s_{pub})$

When $\lambda = 0$, we obtain the private belief distribution, and when $\lambda = 1$, we obtain the public belief distribution.

Using PPBS allows us to be consistent with the problem encountered. When care must be taken not to reveal information, one can increase λ. In contrast, when it is not appropriate to withhold information, one can decrease λ.

In our example, when using PPBS with $\lambda = 0.5$ for the player 1, we obtain the following belief distribution $\Delta_{0.5}(s_1) = (\{s_1 : 75\%; s'_1 : 25\%\}, \{s_2 : 50\%; s'_2 : 50\%\},$

so that w_1 and w_2 are more probable (37.5% each) than w_3 and w_4 (12.5% each). Nevertheless, their probabilities are not zero, which makes it consistent with the other player's belief.

It is possible to expand this concept by considering that λ depends on the progress of the game. As an example, in trick-taking card games, it may be important to keep the private information hidden at the beginning of the game (so as not to reveal information) but, as the game progresses, the focus shifts to accumulating points before the end, where the importance of concealing this information may decrease.

With the aim of using the public and the mixture distribution, one must take care to adapt the algorithm. In particular, if an algorithm starts at an infostate s_i, it must be adapted to start at the s_{pub}, where s_{pub} is the public infostate associated with the infostate s_i. Adaptations of PIMC and IS-MCTS are given in the appendix.

4 Experimentation

4.1 Benchmarks

For our experiments, the following benchmarks are tested 'Liar's Dice' (LD) and 'Leduc Poker' (LP). Each of them is described below.

4.1.1 Trick-Taking card game

For the purpose of the experimentation, we use a smaller version of classic trick-taking card games. The game is played with two players with N cards divided into four suits (Diamond, Spade, Club, Heart).

The playing phase is decomposed into tricks, the player starting the trick is the one who won the previous trick. The starting player of a trick can play any card in his hand, but the other players must follow the suit of the first player. If they can not, they can play any card they want but, without the possibility of winning the trick. The winner of the trick is the one with the highest ranking card.

At the end of the game, the points of each player are counted. The count is defined by the number of tricks won (plain version of trick-taking card game). A player wins if it has at least half of the points.

4.1.2 Liar's Dice

Liar's dice is a dice game played with two or more players, where each player possesses N K-sided dice, in which a player must deceive and be able to detect an opponent's deception.

In the beginning, each player rolls his dice and observes the values. After that, players take turns guessing the number of dice of a particular type held by everyone. The game continues until a player accuses another of lying. If the player who made the assumption is right, he wins the game, on the opposite, if the challenged player did not lie, the challenged player wins. During the game, a player can not bid less than previously, *i.e.* he must at least bid more dice than the previous player's bid, or the same number of dice, but with a higher value. Lastly, the highest face is a wild card, *i.e.* the value can be used to count for any other face.

4.1.3 Leduc Poker

Leduc Poker, as described in the work [19], is a variation of poker that uses a deck with only two suits, each containing three cards.

The game consists of two rounds. In the first round, each player is dealt a single private card. In the second round, a single board card is revealed. The maximum number of bets allowed is two, with the first round allowing raises of 2 and the second round allowing raises of 4. Both players begin the first round with 1 already in the pot.

4.2 Experimentation

In our experiments, our objective is (i) to observe the extent to which an algorithm X reveals information according to mixture belief distribution ; (ii) to analyze how the mixture belief distribution impacts the performance against an opponent that uses the information revealed ; (iii) to analyze how the mixture belief distribution impacts the performance against an opponent that does not use the revealed information.

Our code is based on OpenSpiel [10]. This is a collection of environments and algorithms for research in general reinforcement learning and search/planning in games.

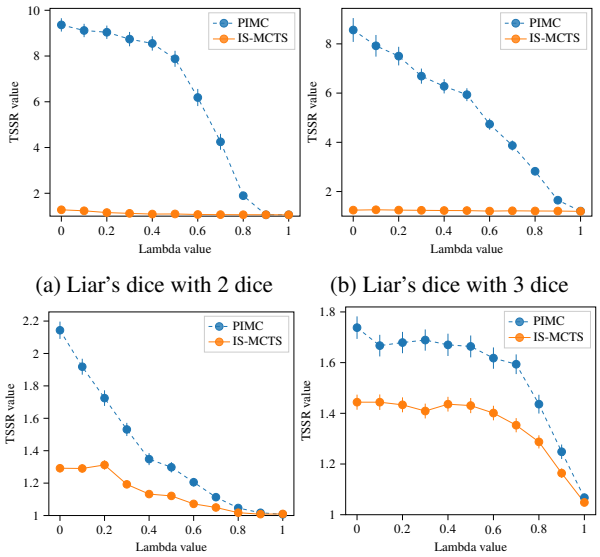
PIMC and IS-MCTS are used with their basic version, *i.e.* PIMC uses AlphaBeta and IS-MCTS uses random rollouts as the perfect information evaluator. For IS-MCTS, the exploration constant is fixed at 0.7. For both, we sample 1000 world states.

To achieve a stable policy (as PIMC and IS-MCTS are online algorithms), we run the algorithm multiple times for every infostate until the policy obtained has less than 1% of variation.

The experiments were conducted according to the player's playing position (each position can reveal more or less information). In the following part, the experiments are carried out for the first player and in the appendix for the second player.

4.2.1 How much information is revealed according λ

We analyze the impact of the information revealed according to the distribution used (public versus private distribution). We use the formula called True State Sampling Ratio (TSSR) [18], which measures how much more likely



(a) Liar's dice with 2 dice (b) Liar's dice with 3 dice

(c) Leduc poker (d) Card Game with 10 cards

FIGURE 2 – Average TSSR according to λ of the mixture distribution when playing at the first position.

it is for the opponent to guess the current world state when using an algorithm X than using a uniform function.

The formula is $TSSR(w) = \eta(w \mid s_i) \cdot |S_i(s_i)|$ where s_i is the infostate corresponding to w, $\eta(w \mid s_i)$ is the probability that the true state is guessed given the information set s_i. The closer the result is to 1, the less likely it is to know the real world state. Figure 2 presents the TSSR value obtained according to λ of the mixture distribution.

As expected, playing closer to the public belief distribution greatly reduces the probability of knowing the real-world state. In 'Liar's Dice' with 2 dice with PIMC, we observe that the opponent is up to 10-fold more likely to guess the real world state when using the private belief distribution instead of the public belief distribution.

'Liar's Dice' reveals more information than 'Leduc Poker'. With 'Liar's Dice', the algorithm reveals up to 10 times more than random, whereas in 'Leduc Poker', it is up to 2 times more than random.

In addition, we observe that PIMC reveals more information than IS-MCTS in every experiment. In 'Liar's Dice' with 2 dice, that is up to 10 times more likely to deduce the true state with PIMC at $\lambda = 0.0$ whereas, with IS-MCTS, it is 'only' 1.5 times more likely to deduce the true state.

4.2.2 How does the mixture impact the performance against the best response

To measure how the mixture impacts the performance, we compute the expected utility against the best responder. The best responder is the worst possible adversary of all algorithms, *i.e.* it knows exactly the policy our algorithm will execute, and therefore, can infer the infostate and plays the best possible action against it.

The results are available in Table 1 where the values represent the expected utility of the best responder and must be minimized. The results obtained are exact utility (without variation), as the best responder computes the best strategy knowing all the distributions in every infostate of the game.

We observe that the private belief distribution tends to perform better than the public belief distribution, *i.e.* for all benchmarks and algorithms (better results are obtained when $\lambda = 0.0$ than when $\lambda = 1.0$).

In 'Liar's Dice' with PIMC, the best performances are obtained when λ is close to 0.5 (with 2 dice, we obtain the best value when $\lambda = 0.6$). These results were expected, as PIMC reveals a lot of information with Liar's Dice, especially when $\lambda < 0.5$, which is then exploited by the best responder.

On the other hand, when the algorithm reveals less information (as observed in 'Leduc Poker' or IS-MCTS), it is preferable to use the private belief distribution or very close, as it is not sufficient for the best responder to exploit the information revealed.

4.2.3 Can the use of multiple mixture belief distributions throughout the game improve performance

We analyze the use of multiple mixtures throughout the game in order to improve the performance. For this purpose, we compute the expected utility against the best responder with multiple λs.

Figure 3 represents several heatmaps for 'Leduc Poker' and 'Liar's Dice' according to the position throughout the game when using PIMC (resp. IS-MCTS). For both games, we have a λ for the first action and another λ for the second action.

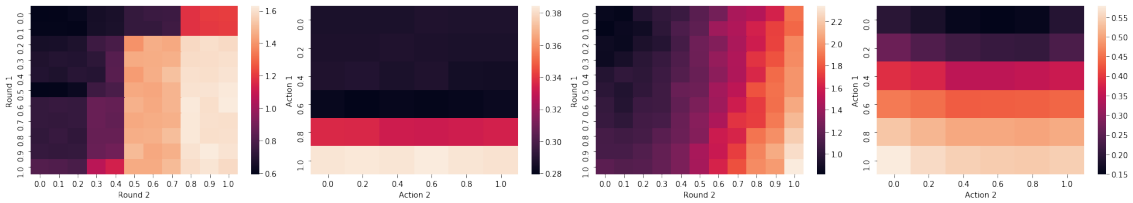
In all figures, we observe that using multiple λ throughout the game has an impact on the performance. In 'Leduc Poker' for both algorithms, not using our private belief distribution is more punished in the second round than in the first round (*e.g.* $(0.0, 1.0)$ has a value of 1.17 whereas $(1.0, 0.0)$ has a value of 1.88 for IS-MCTS).

For 'Liar's Dice', we observe that the first round is the most important one (changing the value of λ in the second round does not have a significant effect on the value obtained).

Furthermore, playing multiple λ can improve performance. In 'Liar's Dice', the best value for IS-MCTS is obtained when we have $\{0.0, 0.6\}$ and for PIMC when we have $\{0.0, 0.6\}$.

Algo	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 2D	0.300	0.298	0.297	0.292	0.294	0.288	**0.281**	0.290	0.336	0.382	0.382
	LD 3D	0.313	0.276	0.265	0.269	**0.235**	0.283	0.324	0.356	0.359	0.393	0.458
	LP	0.622	**0.616**	0.660	0.767	0.797	1.481	1.626	1.480	1.532	1.599	1.611
IS-MCTS	LD 2D	0.513	**0.512**	0.517	0.528	0.539	0.547	0.552	0.554	0.555	0.562	0.562
	LP	**0.797**	0.890	0.966	0.959	1.158	1.226	1.402	1.673	1.786	2.083	2.326

TABLE 1 – Expected utility against best responder when playing at the first player position.



(a) Leduc Poker with PIMC (b) Liar's Dice 2 dice with PIMC (c) Leduc Poker with IS MCTS (d) Liar's Dice 2 dice with IS MCTS

FIGURE 3 – Heatmap of the best response when playing at the first position. Y-axis refers to λ value used when playing the first action, and the X-axis refers to λ used when playing the second action.

4.2.4 How does the mixture impact the winning rate

As observed in the previous experiments, when using a λ closer to the public belief distribution, we obtain a distribution of action less relevant but with the advantage of disclosing less information. Therefore, against an opponent that does not use our private information revealed, it is expected to lose the utility of using λ closer to the public belief distribution.

Nevertheless, using a λ closer to the public belief distribution not only reveals less information but allows it to be more consistent with the hesitation of the other player.

To measure the impact of being more consistent with the other player's hesitation, we evaluate the performance against an algorithm that does not try to infer our private information. To do this, we compute the winning rate against 'PIMC' over 1000 games which results in 3.1% variation (95% of confidence interval). The scores are available in Table 2.

As before, we observe that it is preferable to use the private belief distribution instead of the public belief distribution. For example, in 'Liar's Dice' with 3 dice with PIMC, we observe a drop of 20.8 in the winning rate.

In addition, we observe that in every benchmark tested and for both algorithms, using a λ between 0.0 to 0.5 does not produce a drop in performance, but provides equivalent results. These results are surprising, as we could have expected a drop in performance as the actions are less relevant to the current infostate (as we have sampled less often the true infostate). This implies that being more consistent with the hesitation of the other players compensates for the loss of the player's private information.

5 Conclusion

In this paper, we study the strengths and weaknesses of probability distributions (private and public) in which particular attention has been paid to the information revealed and the impact of this revealed information on performance.

We complete the study by proposing a new probability distribution, a mixture of the two previous ones, which solves problems encountered by other distributions.

We show that using the mixture is beneficial to reduce the information revealed and improve performance. We also show that using multiple mixtures throughout the game improves performance. In addition, we observed that using the mixture against an opponent that does not use our private information revealed can also be used to improve performance, as we are being more consistent with the other player's doubt.

An avenue for improvement would be to extend the utilization of using multiple λ, especially by using a λ at each public infostate. We could also consider using a different lambda for the opponent player.

Another area for improvement would be to extend the study of algorithms that do not use determinization or even, without probability distributions but bearing in mind that one should not always use one's private information at the

Our	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 3D	48.6	**50.4**	47.9	47.4	44.6	42.6	39.9	37.5	36.1	28.9	27.8
	LD 5D	43.1	**43.4**	42.2	**43.4**	42.5	41.4	39.8	36.3	37.1	29.8	23.6
	CG 10C	**48.2**	47.9	47.7	47.7	47.6	47.4	46.7	46.	45.5	39.8	31.6
	CG 20C	53.7	53.8	54.2	**54.5**	53.9	53.2	52.8	52.	47.4	36.3	23.5
IS MCTS	LD 3D	23.7	23.7	24.7	**27.**	23.1	23.1	21.7	20.	19.3	15.4	16.4
	LD 5D	22.	20.9	21.9	**22.2**	21.9	20.8	21.6	21.	16.9	15.5	13.4
	CG 10C	45.3	**46.3**	45.4	45.1	43.8	45.1	45.	43.1	42.7	37.6	30.
	CG 20C	36.5	**38.5**	38.2	36.2	36.4	36.6	35.5	34.9	33.3	33.1	20.8

TABLE 2 – Winning rate when the opponent uses 'PIMC' according to λ of the mixture belief distribution when playing at the first player position.

risk of revealing information and, on the contrary, that one should not always use one's public information in order to be more consistent to one's private knowledge.

Lastly, it would be interesting to extend the results at a larger scale, either by using more games or by using larger games, especially for the calculation of the best responder.

A Appendix

A.1 Adaptation of algorithms

PIMC and IS-MCTS have been created with private belief distribution in mind. Therefore, care must be taken to adjust the algorithms to utilize the public belief distribution or the public-private belief distribution.

To do so, one must use a probability distribution over the infostates possible $S_i(s_{pub})$ before using the algorithm.

A.1.1 PIMC

PIMC requires a distinct algorithm to be applied for each possible infostate, and the final result is obtained by aggregating the scores using the distribution of possible infostates.

A single algorithm is not feasible as the score is calculated based on the actions that are possible for a specific infostate, and not all actions are possible in different infostates.

In the example described in the main article (in Section 3), for the first player, two infostates are possible (s_1 and s'_1). If w_2 is sampled, the algorithm used is the one defined in the infostate corresponding (s_1). In the end, if s_1 has been visited 75% (corresponding to the mixture belief distribution with $\lambda = 0.5$), the action chosen in s_1 will have more impact than the action chosen in s'_1.

A.1.2 IS-MCTS

With IS-MCTS, a singular algorithm is feasible as, IS-MCTS creates a tree where the nodes represent infostates,

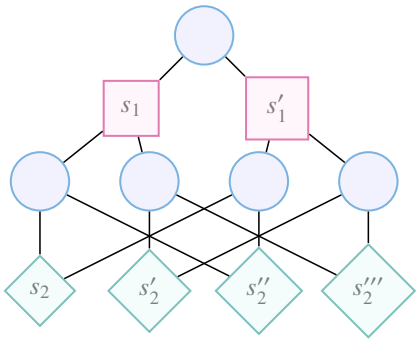


FIGURE 4 – Example of the MCTS tree constructed by IS-MCTS with public belief distribution, when playing the example described in the main article (Section 3). The first player is acting in the red square, the second player is acting in the green diamond and the blue circle refers to the chance node.

and an infostate for player j may come from several infostates of player i

An example is provided in Figure 4. In the example, two infostates are possible for the first player (s_1 and s'_1) and four infostates are possible for the second player after the first player's action ($s_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1, \tilde{o}^3 = a_1\}$, $s'_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 1, \tilde{o}^3 = a_2\}$, $s''_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2, \tilde{o}^3 = a_1\}$ or $s'''_2 = \{\tilde{o}^1 = \emptyset, \tilde{o}^2 = 2, \tilde{o}^3 = a_2\}$). For the second player, all infostates are achievable through any infostate of the first player, for example, s_2 is achievable when sampling w_1 (from s_1) or when sampling w_2 (from s'_1) and playing the action a_1.

A.2 Complementary experiments

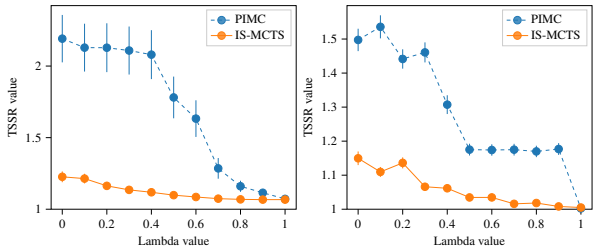
The following experiments are identical to those in the primary paper, with the exception that they are conducted for the second player position.

Algo	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 2D	**0.678**	0.695	0.703	0.707	0.716	0.718	0.711	0.741	0.779	0.836	0.836
	LP	**0.398**	0.400	0.459	0.612	0.796	1.461	1.450	1.509	1.593	1.615	1.632
IS-MCTS	LD 2D	0.697	**0.687**	0.697	0.716	0.727	0.732	0.740	0.751	0.759	0.768	0.787
	LP	**0.784**	0.784	0.898	0.800	1.017	1.078	1.186	1.324	1.561	1.728	2.002

TABLE 3 – Expected utility for best responder against our algorithm being the second player.

Our	Game	λ										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PIMC	LD 3D	51.9	**53.**	51.3	49.8	49.7	51.3	48.7	48.6	46.9	46.1	42.7
	LD 5D	**56.7**	55.5	56.	56.2	54.8	56.1	55.3	53.	51.9	44.7	42.3
IS MCTS	LD 3D	48.4	**51.3**	49.9	49.	50.1	51.	47.4	44.	39.7	36.9	33.3
	LD 5D	**48.4**	47.1	48.	46.7	47.8	45.	46.5	40.7	34.4	23.2	14.7

TABLE 4 – Winning rate when the opponent uses 'PIMC' according to λ of the mixture belief distribution when playing at the second player position.



(a) Liar's dice with 2 dice (b) Leduc poker

FIGURE 5 – Average TSSR for IS-MCTS and PIMC on multiple benchmarks according to λ value of the mixture distribution.

Références

[1] Brown, Noam, Anton Bakhtin, Adam Lerer et Qucheng Gong: *Combining Deep Reinforcement Learning and Search for Imperfect-Information Games*. Dans *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc., ISBN 978-1-71382-954-6. event-place : Vancouver, BC, Canada.

[2] Brown, Noam et Tuomas Sandholm: *Superhuman AI for multiplayer poker*. Science, 365 :885 – 890, 2019.

[3] Browne, Cameron, Edward Jack Powley, Daniel Whitehouse, Simon M. M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez Liebana, Spyridon Samothrakis et Simon Colton: *A Survey of Monte Carlo Tree Search Methods*. IEEE Transactions on Computational Intelligence and AI in Games, 4 :1–43, 2012.

[4] Cazenave, Tristan et Véronique Ventos: *The $\alpha\mu$ Search Algorithm for the Game of Bridge*. Dans *Monte Carlo Search at IJCAI*, Communications in Computer and Information Science, 2021.

[5] Cowling, Peter I., Edward Jack Powley et Daniel Whitehouse: *Information Set Monte Carlo Tree Search*. IEEE Transactions on Computational Intelligence and AI in Games, 4 :120–143, 2012.

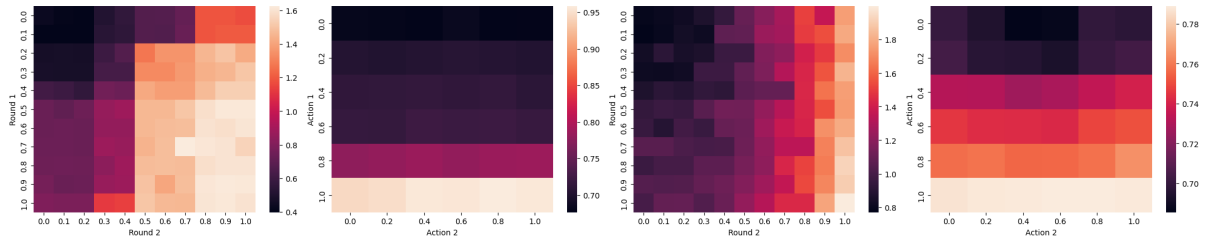
[6] Dibangoye, Jilles Steeve, Christopher Amato, Olivier Buffet et François Charpillet: *Optimally Solving Dec-POMDPs as Continuous-State MDPs*. Journal of Artificial Intelligence Research, 55 :443–497, février 2016.

[7] Furtak, Timothy et Michael Buro: *Recursive Monte Carlo search for imperfect information games*. 2013 IEEE Conference on Computational Inteligence in Games (CIG), pages 1–8, 2013.

[8] Knuth, Donald E. et Ronald W. Moore: *An analysis of alpha-beta pruning*. Artificial Intelligence, 6(4) :293–326, 1975, ISSN 0004-3702.

[9] Kovařík, Vojtěch, Martin Schmid, Neil Burch, Michael H. Bowling et V. Lisý: *Rethinking Formal Models of Partially Observable Multiagent Decision Making*. Artif. Intell., 303 :103645, 2022.

[10] Lanctot, Marc, Edward Lockhart, Jean Baptiste Lespiau, Vinícius Flores Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin



(a) Leduc poker with PIMC (b) Liar's dice 2 dice with PIMC (c) Leduc poker with IS-MCTS (d) Liar's dice 2 dice with IS-MCTS

FIGURE 6 – Heatmap of the best response when playing at the second position. Y-axis refers to λ value used when playing the first action, and the X-axis refers to λ value used when playing the second action.

Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas W. Anthony, Edward Hughes, Ivo Danihelka et Jonah Ryan-Davis: *OpenSpiel : A Framework for Reinforcement Learning in Games*. ArXiv, abs/1908.09453, 2019.

[11] Long, Jeffrey Richard, Nathan R. Sturtevant, Michael Buro et Timothy Furtak: *Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search*. Dans *AAAI*, 2010.

[12] Moravcík, Matej, Martin Schmid, Neil Burch, V. Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, K. Waugh, Michael Bradley Johanson et Michael H. Bowling: *DeepStack : Expert-level artificial intelligence in heads-up no-limit poker*. Science, 356 :508 – 513, 2017.

[13] Perolat, Julien, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis et Karl Tuyls: *Mastering the game of Stratego with model-free multiagent reinforcement learning*. Science, 378(6623) :990–996, dec 2022.

[14] Silver, D., Aja Huang, Chris J. Maddison, A. Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, S. Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel et Demis Hassabis: *Mastering the game of Go with deep neural networks and tree search*. Nature, 529 :484–489, 2016.

[15] Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan et Demis Hassabis: *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. ArXiv, abs/1712.01815, 2017.

[16] Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, L. Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan et Demis Hassabis: *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*. Science, 362 :1140 – 1144, 2018.

[17] Smith, Trey: *Probabilistic Planning for Robotic Exploration*. Thèse de doctorat, Carnegie Mellon University, Pittsburgh, PA, July 2007.

[18] Solinas, Christopher, Douglas Rebstock et Michael Buro: *Improving Search with Supervised Learning in Trick-Based Card Games*. ArXiv, abs/1903.09604, 2019.

[19] Southey, Finnegan, Michael P Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings et Chris Rayner: *Bayes' bluff : Opponent modelling in poker*. arXiv preprint arXiv :1207.1411, 2012.

[20] Tammelin, Oskari, Neil Burch, Michael Bradley Johanson et Michael Bowling: *Solving Heads-Up Limit Texas Hold'em*. Dans *IJCAI*, 2015.

[21] Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama,

Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps et David Silver: *Grandmaster level in StarCraft II using multi-agent reinforcement learning*. Nature, pages 1–5, 2019.

Actes JIAF-JFPDA 2023

Analysis of planning instances without search

Martin C. Cooper Arnaud Lequen Frédéric Maris

IRIT, Université Toulouse 3, France

Martin.Cooper@irit.fr, Arnaud.Lequen@irit.fr, Frederic.Maris@irit.fr

Résumé

Les planificateurs classiques actuels détectent les instances de planification insolubles au travers d'une recherche dans l'espace d'états sous-jacent. Dans cet article, nous montrons cependant qu'il est quelquefois suffisant d'utiliser un critère incomplet, mais efficace d'un point de vue calculatoire. Nous proposons une méthode permettant de tirer parti de ce critère, basé sur des techniques de programmation linéaire et en nombres entiers, dans le cas où il ne permet pas de conclure. Ce critère est central aux méthodes que nous proposons pour préciser et enrichir le modèle STRIPS, dans l'optique de collecter de nouvelles informations à son propos. Dans le cas où les informations supplémentaires ne permettent pas de s'assurer de l'insolubilité de l'instance, elles peuvent être réinvesties dans un algorithme complet intervenant ensuite, afin de l'accélérer.

Abstract

In order to prove classical planning instances unsolvable, state-of-the-art planners resort to a state-space search. However, we show here that an incomplete, yet computationally efficient criterion is sometimes sufficient to immediately identify as unsolvable a wide range of planning instances. Based on linear and integer programming, we show in this paper how it can be leveraged, were it to fail at first. This criterion is the keystone of various techniques we propose to rewrite and enhance the STRIPS model, so as to gather new information about it. In case the newly-found bits of information are not sufficient to identify the instance as unsolvable, they can be reinvested later to speed up a complete algorithm.

1 Introduction

Current classical planners resort to a search, with the goal of finding a solution-plan. They often start with the assumption that such a plan exists, and for the past few decades, significant work has been done on designing more and more efficient techniques to find solution-plans. However, various reasons may lead an instance not admitting any solution. Search-based planners will then explore the state-space in its entirety, potentially cutting branches of the search tree, until they realize no plan can be found. The detection of states that can not lead to any solution is often a byproduct of the heuristics used during search: an infinite heuristic value for an admissible heuristic is synonym of a dead-end state.

This is why in the recent years, there has been a renewed interest in detecting unsolvable planning instances, as illustrated by the 2016 Unsat IPC (International Planning Competition). Various techniques have been developed in the last couple of decades, such as dead-end formulas [4], traps [10, 14], and so on. However, all of these methods are based on the exploration of the state-space.

In this article, we propose to leverage a linear programming- and integer programming-based criterion to iteratively refine a planning model, to show its unsolvability. The criterion we use is fast to compute, and allows us to quickly recognize a wide range of unsolvable planning instances. However, it is not complete, in the sense that it may not recognize some unsolvable instances as such. Nevertheless, we show how to use it to iteratively refine the planning model, and keep gathering additional information about the instance with the aim that our procedure can detect that it is unsolvable.

Most of our techniques to gather information are based on a simple schema: after testing the solvability of planning instances Π' that are derived from the initial planning problem Π given as input, we deduce additional information about the problem Π if Π' is unsolvable. For instance, if the instance Π', which is Π where operator a was removed, is proven to be unsolvable, then it means that a appears in all solution-plans of Π. In the case where one can efficiently detect some unsolvable planning instances, then lots of such derived instances Π' can be tested successfully. As the criterion we use is incomplete but fast, even though it often fails to detect unsolvable instances, it still manages to help gather new information, as lots of tests can be made in reasonable time. As more and more information is known

about the planning instance, the mathematical program on which the criterion is based can also be enriched with the new knowledge, so that it can detect additional unsolvable instances.

More generally, being able to detect planning instances that have no solution can have various applications in itself. For instance, consider the case where an instance models the attacks of a malicious user may perform on a system, with the goal of accessing restricted data. Finding that no sequence of actions may achieve this shows that the system is secure.

The paper is organized as follows. In Section 2, we introduce our formalism and notations for classical planning. In Section 3, we present the mathematical-programming-based criterion we use throughout this paper. In Section 4, we show how to design tests to gather new information about a planning model. In Section 5, we report our experimental trials on standard sets of benchmarks. Section 7 is devoted to a discussion and perspectives on our findings.

2 Background

STRIPS planning instance A STRIPS planning instance is a tuple $\Pi = \langle F, I, O, G\rangle$ such that F is a set of propositional variables called *fluents*, and I is a set of fluents of F, called the *initial state*. G is a set of *literals* of F, such that no literal appears at the same time as its negation, and is called the *goal*. We will denote G^+ the set of positive literals of G, and G^- the set of negative literals. Finally, O is a set of *operators*: operators $a \in O$ are of the form $a = \langle \mathsf{pre}(a), \mathsf{eff}(a)\rangle$. $\mathsf{pre}(a)$ is the *precondition* of a and $\mathsf{eff}(a)$ is the *effect* of a, which are both sets of literals of F. We will denote $\mathsf{eff}^+(a) = \{f \in F \mid f \in \mathsf{eff}(a)\}$ the set of *positive* effects of a, and $\mathsf{eff}^-(a) = \{f \in F \mid \neg f \in \mathsf{eff}(a)\}$ its *negative* effects. We will use similar notations to define $\mathsf{pre}^+(a)$ and $\mathsf{pre}^-(a)$.

Note that we define a version of STRIPS with negative preconditions. However, we are not any more general than the original formulation of STRIPS. Indeed, any STRIPS instance with negative preconditions can be translated into an equivalent instance without negative preconditions in linear time, and the converse is immediate [6]. The same goes for negative goals: the original STRIPS formulation only specified positive goals. We nonetheless allow negative goals in our formulation of STRIPS, and we progressively take them into account. But one should keep in mind that most planning instances (and in particular, the ones used in our set of benchmarks) come with positive goals only: this is why we assume G^- is empty unless otherwise specified.

Without loss of generality, we assume that for all operators a, $\mathsf{pre}^+(a) \cap \mathsf{pre}^-(a) = \emptyset$. We also assume that $\mathsf{eff}^+(a) \cap \mathsf{eff}^-(a) = \emptyset$, otherwise we can remove from $\mathsf{eff}^-(a)$ any fluent also in $\mathsf{eff}^+(a)$. In addition, we will also suppose that $\mathsf{eff}^+(a) \cap \mathsf{pre}^+(a) = \emptyset$, and $\mathsf{eff}^-(a) \cap \mathsf{pre}^-(a) = \emptyset$, otherwise the redundant fluents from the effects can be removed. Any planning instance which does not satisfy these criteria can be transformed, in polynomial time, into an equivalent instance that complies with them.

States and plans A state s is an assignment of truth values to all fluents in F. For notational convenience, we associate s with the set of fluents of F which are true in s. An operator a can be applied to states of Π that verify its preconditions. More formally, for any state s, if $\mathsf{pre}^+(a) \subseteq s$ and $\mathsf{pre}^-(a) \cap s = \emptyset$, then we define the result of the application of a to s as $s[a] = (s \setminus \mathsf{eff}^-(a)) \cup \mathsf{eff}^+(a)$.

Given an instance $P = \langle F, I, O, G\rangle$, a *plan* is a sequence of operators $\pi = a_1, \ldots, a_k$ from O such that there exists a sequence of states $s_0, \ldots, s_k$, such that, for all $i \in 1, \ldots, k$, the operator a_i is applicable in s_{i-1}, so that $s_i = s_{i-1}[a_i]$. A plan is a *solution-plan* if we have, in addition, $s_0 = I$ and $G \subseteq s_k$. We denote $\mathcal{S}_\Pi$ the set of all solution-plans to Π. We say that a fluent f is established (resp. deleted) by some occurrence of an operator $a \in O$ in π if f is false (resp. true) in some state s_i, but true (resp. false) after the application of a, in state $s_{i+1} = s_i[a]$. In the rest of this paper, we will refer to solution-plans as simply plans.

3 Detecting unsolvable instances by LP

This section introduces two equivalent criteria that we use, and extend, to detect a planning instance's unsolvability. These criteria are incomplete, in the sense that they can not detect all unsolvable planning instances by themselves. However, they require very limited computational resources, and are fast to run, as they are based on linear programming, or mathematical programming in general. We will show later how to leverage those properties in order to make the most of these criteria when they are not able to detect an instance's unsolvability by themselves.

3.1 Potential-based argument

The first linear programming formulation that we worked with is based on the following argument. Suppose that we have a numerical function $\Phi : F \to \mathbb{R}^+$, that associates a *potential* to each fluent. We can then naturally define the potential of a state $s \subseteq 2^F$ as $\Phi(s) = \sum_{f \in s} \Phi(f)$. If one can prove that all goal states have a higher potential than the initial state, but the application of any operator a to any state s leads to a state s' of lesser (or equal) potential, then the planning instance has no solution-plan.

Such a function Φ can be found thanks to the following observation. In any plan, the potential of a state s' solely depends on the previous state s, and on the operator a that was applied such that $s[a] = s'$. In this case, we will say that a induced an increase in potential of $\Delta\Phi_a(s) =$

$\Phi(s') - \Phi(s)$. One can remark that there exists an upper bound for $\Delta\Phi_a(s)$, which does not depend on s but only on a. Indeed, in the limit case, all fluents $f \in \mathsf{eff}^+(a)$ are effectively established by a, but no fluent $f' \in \mathsf{eff}^-(a)$ is destroyed, except when $f' \in \mathsf{eff}^-(a) \cap \mathsf{pre}^+(a)$. Recall that we assume, without loss of generality, that $\mathsf{eff}^+(a) \cap \mathsf{pre}^+(a) = \mathsf{eff}^-(a) \cap \mathsf{pre}^-(a) = \emptyset$.

More formally, let us consider four[1] sets of operators, with regard to some fluent f: on the one hand, the operators that will *surely add* and *surely delete* f when applied, that we denote respectively SA_f and SD_f; on the other hand, the operators that could *possibly add* and *possibly delete* f when applied, respectively PA_f and PD_f. The latter are operators that may establish (resp. delete) f in the resulting state s' depending on whether f is false (resp. true) in the previous state s or not. More formally, the sets are defined as follows:

- $SA_f = \{a \mid f \in \mathit{eff}^+(a) \cap \mathit{pre}^-(a)\}$
- $SD_f = \{a \mid f \in \mathit{eff}^-(a) \cap \mathit{pre}^+(a)\}$
- $PA_f = \{a \mid f \in \mathit{eff}^+(a) \setminus \mathit{pre}^-(a)\}$
- $PD_f = \{a \mid f \in \mathit{eff}^-(a) \setminus \mathit{pre}^+(a)\}$

This leads to the following inequality, which models the limit case previously presented. This effectively gives us an upper bound on the change of potential induced by a from any state s, which we denote $\Delta\Phi_a(s)$. Remark that the right-hand side is independent of s.

$$\Delta\Phi_a(s) \leq \sum_{f \text{ s.t. } a \in PA_f} \Phi(f) + \sum_{f \text{ s.t. } a \in SA_f} \Phi(f) - \sum_{f \text{ s.t. } a \in SD_f} \Phi(f)$$

Now suppose that, for all operators a, the right-hand side of the previous inequation is negative. It means that applying any operator makes the potential of the state decrease. As a consequence, states that have a higher potential than the initial state cannot be reached. Note that, as the potential of a state is only determined by the potential of the fluents that are true in this state, and all potentials are positive, $\Phi(G)$ is a lower bound for the potential of any goal-state. Thus, if we also have that $\Phi(G) > \Phi(I)$, then the planning instance has no solution.

The only remaining issue is to check whether such a potential function Φ exists. As Φ is only determined by its values on the various fluents, this can be done with the following set of equations, with the set of variables $V = \{x_f \mid f \in F\}$. Intuitively, x_f corresponds to the potential $\Phi(f)$ of f.

Linear Program 1.
Variables: $V = \{x_f \mid f \in F\}$
Constraints:

$$\sum_{f \in G} x_f - \sum_{f \in I} x_f > 0 \tag{1}$$

$$\sum_{f \in \mathit{eff}^+(a)} x_f \; - \sum_{f \in \mathit{eff}^-(a) \cap \mathit{pre}^+(a)} x_f \leq 0 \; (a \in O) \tag{2}$$

$$x_f \geq 0 \;\; (f \in F) \tag{3}$$

The following proposition follows from the discussion above.

Proposition 1. *Let Π be a STRIPS instance. Suppose that there exists a solution for the Linear Program 1. Then Π has no solution.*

Note that the converse is not true: not all unsolvable planning instances are detected by the criterion we propose.

3.2 Dual linear program

The linear program presented in the previous section is hard to interpret, as the concept of potential we introduced has no reality outside of the criterion. However, we show in this section how to transform it into another program that can equivalently allow us to detect some unsolvable instances, but whose result is easier to interpret.

To this effect, we resort to Farkas's lemma. Farkas's lemma is related to the well-known fact that in linear programming, the primal problem is feasible iff the dual problem is feasible. One version of this lemma states that exactly one of the following sets of equations has a solution: either (1) $Ay \geq d$ where $y \geq 0$, or (2) $A^t x \leq 0$ and $d^t x > 0$ where $x \geq 0$, where A is a matrix and x, y and d vectors of the appropriate sizes. Let us consider the set of equations previously mentioned. Applying Farkas's lemma, it has a solution iff the following system has no solution:

Linear Program 2.
Let $\Pi = \langle F, I, O, G \rangle$ a planning instance. We define $\mathcal{L}^{op}_{\Pi}(V, C)$ as follows:
Variables: $V = \{y_a \mid a \in O\}$
Constraints C:

$$\sum_{a \in SA_f} y_a + \sum_{a \in PA_f} y_a - \sum_{a \in SD_f} y_a \geq \delta^-_f \;\; (f \in F) \tag{4}$$

$$y_a \geq 0 \;\; (a \in O) \tag{5}$$

where $\delta^-_f = \mathbb{1}_G(f) - \mathbb{1}_I(f)$ ($\mathbb{1}_S(x)$ being the indicator function of set S: $\mathbb{1}_S(x) = 1$ if $x \in S$, and 0 otherwise). In this context, the variable y_a corresponds to the number of times operator a is executed in some sequence of actions. Note that y_a is positive, but not necessarily integral: this allows us to obtain a polynomial-time relaxation of the STRIPS instance. Inequality (4) states that the number of (possible) establishments of f minus the number of sure destructions of f must be greater than or equal to δ^-_f. For

[1] Even though only three sets out of the four are needed here, we introduce all four sets as they will be useful later in the paper.

instance, any fluent that appears positively in the goal but not in the initial state must be established as least once. This dual version of our original linear program provides an alternative insight into the meaning of Proposition 1.

Lemma 1. *Let $\Pi = \langle F, I, O, G\rangle$ be a planning instance, $\mathcal{L}^{op}_{\Pi}(V, C)$ as defined in Linear Program 2, and π a solution-plan for Π. Let us define $c_\pi : O \to \mathbb{N}$ the number of occurrences of operators of O in π. Then the assignment $Y : V \to \mathbb{N}$ such that, for all $a \in O$, $Y(y_a) = c_\pi(a)$, is a solution for $\mathcal{L}^{op}_{\Pi}$.*

Proof. Let Y be as defined above. We will show that Y is a solution for $\mathcal{L}^{op}_{\Pi}$. For each fluent f, let us denote e_f the number of times a fluent is established during the execution of π, and d_f the number of times it is destroyed. Recall that a fluent f is established (resp. deleted) by some occurrence of an operator $a \in O$ in π if f is false (resp. true) before the application of the operator, but true (resp. false) after. As π is a solution plan, we have that:

$$\mathbb{1}_{G^+}(f) - \mathbb{1}_I(f) \leq e_f - d_f \leq 1 - \mathbb{1}_I(f) - \mathbb{1}_{G^-}(f)$$

which can be shown by case disjunction on whether f is in I, G^+ or G^- [2]. We denote the inequations above in a more concise way:

$$\delta^-_f \leq e_f - d_f \leq \delta^+_f$$

In addition, in the extreme case, f is established in π at most as many times as there are occurrences of operators a with $f \in \mathsf{eff}^+(a)$. Remark that SA_f and PA_f form a partition of the set $\{a \mid f \in \mathsf{eff}^+(a)\}$ [3]. Hence

$$e_f \leq \sum_{a \in SA_f} Y(y_a) \;+\; \sum_{a \in PA_f} Y(y_a)$$

Similarly, the only operators $a \in O$ whose applications are guaranteed to destroy f are such that $f \in \mathsf{pre}^+(a) \cap \mathsf{eff}^-(a)$. Thus,

$$d_f \geq \sum_{a \in SD_f} Y(y_a)$$

By combining both inequations above, we have

$$\begin{aligned} \delta^-_f &\leq e_f - d_f \\ &\leq \sum_{a \in SA_f} Y(y_a) + \sum_{a \in PA_f} Y(y_a) - \sum_{a \in SD_f} Y(y_a) \end{aligned} \quad (6)$$

which means that Y satisfies the constraints of the form of inequation (4) of $\mathcal{L}^{op}_{\Pi}$. As a consequence, as Y is also positive, Y is a solution to $\mathcal{L}^{op}_{\Pi}$. □

The contrapositive of Lemma 1 is an alternative proof that, if $\mathcal{L}^{op}_{\Pi}$ has no solution, then neither has Π. But it allows us to show more than that, as we have the following corollaries, that we use later on:

[2] Even though we suppose G^- empty now, we introduce the notation and argument here, for later use.

[3] Recall that we suppose that, for all $a \in O$, $\mathsf{pre}^+(a) \cap \mathsf{pre}^-(a) = \mathsf{eff}^+(a) \cap \mathsf{pre}^+(a) = \mathsf{eff}^-(a) \cap \mathsf{pre}^-(a) = \varnothing$

Corollary 1. *If $\mathcal{L}^{op}_{\Pi}$ has no* integral *solution, then the associated planning instance Π has no solution.*

Proof. The proof is immediate, as each operator appears an integral number of times in any solution-plan π. □

Corollary 2. *Optimising the value of y_a within $\mathcal{L}^{op}_{\Pi}$ leads to a bound on the number of times $a \in O$ must occur in a plan.*

Linear Program 2 is, in fact, a linear programming formulation of the state equation heuristic [2], as previously shown in [12]. Its efficiency for detecting unsolvable planning instances has been shown before, as it is part of the Aidos planner, which won the Unsat IPC in 2016 [13]. The planner uses the LP formulation of the operator counting heuristic to detect dead-ends during search, working on a finite domain representation (FDR) of the instance. We, however, do not resort to search, but show how to rewrite the model directly, potentially changing the linear program when doing so.

Even though we introduced $\mathcal{L}^{op}_{\Pi}$ as a linear program, we showed with Lemma 1 that one can also see it as an integer program. Solving an integer program is notoriously harder and slower than solving a linear program. As the integral solutions of the set of equations form a subset of its set of rational solutions, testing the solvability of the program over integral solutions is more likely to prove that the associated planning instance has no solution. Note that Farkas's lemma does not apply in the integral case, hence the need for Lemma 1.

In the next section, we show that, in the case where the criterion introduced here fails, it can still be leveraged to gather additional information about Π.

4 Enhancing the planning problem

This section is dedicated to extending and adding information to the initial planning instance, mainly with the goal of proving it unsolvable. Through various methods, we either add or remove elements from the input model Π, or add information about Π that is not directly encodable into the model, but that can nevertheless still be included in the linear program or to make deductions. In order to do so, we will resort to two kinds of methods. In the first ones, we build variations of Π so that, if one of these variations can be deemed unsolvable through the previous linear program, then some additional information about Π can be deduced. In the second method, we do not consider *per se* a variation Π' of Π, but we directly modify the linear program $\mathcal{L}^{op}_{\Pi}$ associated to Π, so that if it is unsolvable, we can deduce new specific information about Π.

In the following, we call *operation* any such method. In the specific case where the operation answers a boolean question (e.g. Is an action removable?), we call it a *test*.

In the rest of this section, we illustrate the previous general principles through various operations, that allow us to find new information about the planning instance given as input. As our goal is to detect unsolvable instances, in the following, we assume that the criterion could not detect, at first, that the instance is unsolvable and that we have to gather additional information in order to do so.

4.1 Operator counts and landmarks

Landmark detection An operator $a \in O$ is a landmark for Π if a occurs at least once in every solution-plan. We maintain through our procedure a set $L \subseteq O$ of landmarks. With regard to our framework, we can test if an operator is a landmark by removing it from the model and testing if the instance can be deemed unsolvable. More formally,

Lemma 2. *Let $\Pi = \langle F, I, O, G\rangle$ and $a \in O$. If $\Pi_{|a} = \langle F, I, O \setminus \{a\}, G\rangle$ is unsolvable, then a is a landmark.*

This leads us to defining the landmark detection test, as introduced below, where $\Pi_{|a}$ is defined in the lemma above.

LMDet
> **If** $\Pi_{|a}$ is unsolvable
> **then** add a to the set of landmarks L

Operator count One can generalize the notion of landmark, by counting the least number of times an operator appears in any solution-plan. This is why we maintain a function $n^- : O \to \mathbb{N}$, such that $n^-(a)$ is (a lower bound on) the least number of occurrences of action a in any plan. Likewise, we define $n^+(a)$ as (an upper bound on) the maximum number of times a appears in any plan. With these notations, $a \in O$ is a landmark iff $n^-(a) \geq 1$.

Reasoning on the number of occurrences of some operator $a \in O$ can be done through Linear Program 2. Indeed, as the variables are associated to the number of occurrences of each operator in some sequence of actions, one only has to find lower and upper bounds for each variable y_a in a solution of LP 2. This is why one can compute approximate values for $n^+(a)$ and $n^-(a)$ through an integral variation of our linear program, that we present below:

Integer Program 1.
Let $\Pi = \langle F, I, O, G\rangle$ a planning instance, with $O = \{a_1, \ldots, a_m\}$, and $\mathcal{L}^{op}_{\Pi}(V, C)$ the associated Linear Program 2. For $a \in O$, let us define $\mathcal{L}^{opt}_{\Pi}(V, C)(a)$ such that:
Variables *$V = \{y_a \mid a \in O\}$*
Constraints *C: Same as $\mathcal{L}^{op}_{\Pi}$*
Objective function *$g : \mathbb{N}^m \longrightarrow \mathbb{N}$:*

$$g : y_{a_1}, \ldots, y_{a_m} \longmapsto y_a$$

Lemma 3. *Let Π a planning instance, $a \in O$ an operator and consider integer program $\mathcal{L}^{opt}_{\Pi}(V, C)(a)$ with objective function g. Then minimizing (resp. maximizing) g yields a lower (resp. an upper) bound on the value of $n^-(a)$ (resp. $n^+(a)$).*

Proof. The proof is a consequence of Lemma 1. Let us show the case where g is minimized, as the proof for the other case is mostly identical. We denote $n^-_{\mathcal{L}}$ the value obtained by minimizing g in $\mathcal{L}^{opt}_{\Pi}(V, C)(a)$, where $a \in O$ is fixed. Suppose for a contradiction that $n^-(a) < n^-_{\mathcal{L}}$. Then there exists a plan π_a where a occurs exactly $n^-(a)$ times, by definition. By Lemma 1, there exists a solution Y_{π_a} for $\mathcal{L}^{op}_{\Pi}$ where $Y_{\pi_a}(a) = n^-(a) < n^-_{\mathcal{L}}$, which contradicts the optimality of $n^-_{\mathcal{L}}$. Consequently, we have $n^-_{\mathcal{L}} \leq n^-(a)$. □

OpCount$^+(a)$
> **If** the value $n^+_{\mathcal{L}}$ obtained by *maximizing* g over $\mathbb{N}$ in $\mathcal{L}^{opt}_{\Pi}(V, C)(a)$ is bounded
> **then** set the current value of $n^+(a)$ to $n^+_{\mathcal{L}}$

OpCount$^-(a)$
> **If** the value $n^-_{\mathcal{L}}$ obtained by *minimizing* g over $\mathbb{N}$ in $\mathcal{L}^{opt}_{\Pi}(V, C)(a)$ is non-zero
> **then** set the current value of $n^-(a)$ to $n^-_{\mathcal{L}}$

In the rest of this paper, we will often use the notation OpCount(a) to refer to the successive application of OpCount$^-(a)$ and OpCount$^+(a)$. As experimental trials show that OpCount$^-$ does not find all landmarks found by the test LMDet, OpCount$^-$ does not make it redundant.

Using operator counts Once non-trivial values for some $n^+(a)$ or some $n^-(a)$ has been found (i.e. a finite or non-zero value, respectively), one can reintroduce it into the linear program in the form of additional constraints. These constraints can be introduced in either $\mathcal{L}^{op}_{\Pi}$ or $\mathcal{L}^{opt}_{\Pi}$, as both programs use the same sets of variables and constraints. As the variables of the linear programs correspond to the number of occurrences of operators in some plan, adding these constraints is straightforward for every $a \in O$:

$$y_a \leq n^+(a)$$
$$y_a \geq n^-(a)$$

4.2 Detection of removable actions

This section is concerned with finding operators $a \in O$ that never appear in any solution-plan. Even though some such operators can be detected statically by the parser of Fast Downward, some others require additional computation. We present various techniques that allow us to detect if an operator can be immediately removed from the planning instance, without altering its set of solutions.

Through a modification of the linear program We start by extending $\mathcal{L}^{op}_{\Pi}$ into $\mathcal{L}^{ro}_{\Pi}(a)$ through the addition of the constraint $y_a \geq 1$. If $\mathcal{L}^{ro}_{\Pi}(a)$ has no solution, then Π has

no solution where a occurs at least once, and a can thus be removed from the model.

We do not elaborate on this argument further, as it is a special case of the technique seen in Section 4.1. Indeed, it is equivalent to show that $n^+(a) = 0$, as it ensures that a does not occur in any solution-plan. However, this argument allows us to find removable operators that are not detected by a test proposed later in this subsection.

Unreachable preconditions A simple way to prove that some operator a will never be part of any plan, is to prove that no reachable state satisfies its precondition. This can be done by testing that the planning instance $\Pi_a^{\mathsf{pre}} = \langle F, I, O, \mathsf{pre}(a)\rangle$ is unsolvable.

Removing some operators relaxes the linear program $\mathcal{L}_\Pi^{op}$, by the deletion of some of the associated variables and constraints. As a consequence, it can help prove some instances unsolvable. We introduce below the notation for the associated test:

PreImp
> **If** Π_a^{pre} is unsolvable
> **then** remove a from the set of operators O

Dead-end operators As it is possible to test whether or not there exists a reachable state where a can be applied, it is natural to ask the opposite: does a always lead to a dead-end, where no goal state can be reached?

This paragraph is dedicated to finding such operators, called *dead-end operators*. In order to do so, we need to restrict ourselves to the few fluents that appear in all states resulting from the application of a, that is to say, the fluents that are true after a is applied either because of the effects of a, or by inertia. Indeed, these fluents are the only ones for which we have enough information about their truth value to reason about. Let $F_a = \mathit{fluents}(\mathsf{pre}(a)) \cup \mathit{fluents}(\mathsf{eff}(a))$. For any set S of literals of F, and $E \subseteq F$, we note $S_{|E}$ the projection of S over the fluents E. Likewise, we denote $a_{|E} = \langle \mathsf{pre}(a)_{|E}, \mathsf{eff}(a)_{|E}\rangle$ the projection of operator a over E. For any $O' \subseteq O$, we also note $O'_{|E} = \{a_{|E} \mid a \in O'\}$. This leads us to the following lemma, for which the proof is skipped due to space limitations:

Lemma 4. *Let $\Pi = \langle F, I, O, G\rangle$ be a planning instance and $\Pi^{post} = \langle F_a, I_a^{post}, O_{|F_a}, G_{|F_a}\rangle$, where $I_a^{post} = ((\mathsf{pre}^+(a) \setminus \mathsf{eff}^-(a)) \cup \mathsf{eff}^+(a)) \cap F_a$. If Π_a^{post} is unsolvable, then a is a dead-end operator in Π.*

ActDLock
> **If** Π_a^{post} is unsolvable
> **then** remove a from the set of operators O

4.3 Extended preconditions and goals

In this section, we propose various methods to find more precise preconditions for operators. More precisely, we try to add new fluents to operators' positive or negative preconditions. Suppose for instance that some fluent f can only be true if some other fluent f' is true. Then any operator a such that $f \in \mathsf{pre}^+(a)$ can be extended by adding also f' to $\mathsf{pre}^+(a)$. These more precise preconditions make the program richer and hence more likely to detect unsolvable instances. Similarly, the negative preconditions of operators can be extended, and by the same reasoning, so can the goal. In addition to that, we introduce negative goals: fluents that have to be false in any goal state.

In the rest of this section, we propose several ways to extend preconditions and goals.

Extending the goal The previous argument can also be applied to the goal, and help us add new fluents to the goal. Indeed, let $f \in F$, and $\Pi_{+f}^G = \langle F, I, O, G \cup \{f\}\rangle$. If Π_{+f}^G is unsolvable, then f can be added to the negative goals of Π. Indeed, no goal state s_G such that $s_G \models f$ is reachable: necessarily, in any goal state s_G, we have $s_G \models \neg f$. Conversely, let $\Pi_{-f}^G = \langle F, I, O, G \cup \{\neg f\}\rangle$. If Π_{-f}^G is unsolvable, then f can be safely added to the goals of Π without changing the set of solutions.

We define below the test that allows us to detect if a fluent can be added to the negative goals.

FNegGoal
> **If** Π_{+f}^G is unsolvable
> **then** add f to the negative goals of Π

Taking negative goals into account The linear programs we presented earlier do not make use of the negative goals of the planning instance. Indeed, they usually do not appear in the STRIPS model, as they can be avoided by rewriting the instance during parsing time. However, the previous argument allows us to find such negative goals, and it would be costly to rewrite the whole instance to convert them into positive goals. As such, we show how to take these negative goals directly into account in our linear program.

The key elements have already been introduced in the proof of Lemma 1, where we defined for each $f \in F$ the value $\delta_f^+ = 1 - \mathbb{1}_I(f) - \mathbb{1}_{G^-}(f)$. δ_f^+ serves as an upper bound on the difference on the number of times f is established and the number of times it is destroyed, in any plan.

With a proof that is very similar to the one that leads to Equation 6 in the proof of Lemma 1, one can show that the following equation holds, for any fluent f:

$$\sum_{a \in SA_f} y_a - \sum_{a \in PD_f} y_a - \sum_{a \in SD_f} y_a \leq \delta_f^+ \qquad (7)$$

Note that the above equation is symmetrically equivalent to Equation 4, found in the original Linear Program 2, that we recall below. In the initial formulation, the significant

number of positive preconditions allows us to have non-empty sets of the form SD_f, thus adding negative variables in the left-hand side of the inequation. These negative variables penalize the whole sum, and make it harder to reach the threshold of δ_f^- given in the right-hand side. As our goal is to make the linear program unsatisfiable, the more positive preconditions we have, the better.

$$\sum_{a \in SA_f} y_a \; + \sum_{a \in PA_f} y_a \; - \sum_{a \in SD_f} y_a \; \geq \delta_f^-$$

The same case can be made for negative preconditions and Equation 7: negative preconditions contribute to populating sets of the form SA_f, which in turn further constraint the inequation. In addition, note that having negative goals also contributes to making the inequation harder to satisfy, by lowering the bound δ_f^+ on the right-hand side. As negative goals only appear in variables δ_f^+, without negative preconditions, there would be little interest in seeking to detect them. In addition, negative precondition do not affect the final expression of Equation 4, but only affect Equation 7. As such, negative goals and negative preconditions are closely intertwined.

4.4 Fluent mutexes and unreachable fluents

A fluent mutex is a set of fluents $M \subseteq F$ for which all states s accessible from the initial state I are such that $s \not\models M$. Some tests presented previously can be seen as testing whether some subset $M \subseteq F$ is a fluent mutex. Let us consider for instance the PreImp test presented in Section 4.2: for some operator $a \in O$, checking that $\Pi_a^{\mathsf{pre}} = \langle F, I, O, \mathsf{pre}(a) \rangle$ is unsolvable (and thus that operator a can be removed from the instance) is equivalent to checking that $\mathsf{pre}(a)$ is a mutex. However, our criterion allows us to check if any set of fluents $F' \subseteq F$ is a mutex, by testing the unsolvability of $\Pi_{F'}^{\mathsf{mut}} = \langle F, I, O, F' \rangle$.

FMut
> **If** $\Pi_{F'}^{mut}$ is unsolvable
> **then** F' is a fluent mutex

The criterion does not detect all fluent mutexes, and each candidate set of fluents has to be tested individually. Thus, not all fluent mutexes can be detected in reasonable time, as there exists an exponential number of candidates. Finding which sets are interesting to test is a problem in itself; even more so since one has to know how to make use of the newly-found information that some $M \subseteq F$ is a mutex.

In the general case, we could not find a way to reinvest into the linear program the knowledge that a set of fluents is a mutex. Indeed, Linear Program 2 reasons over the number of times operators (have to) occur in a plan. As a consequence, we do not have any obvious way to reason about properties concerning states, which is precisely what fluent mutexes are. For that reason, we do not include in our routine a computation of mutexes through our linear program, even though we can detect a range of fluent mutexes.

However, some fluents are always false, in the sense that no plan will ever establish them. We call the fluents *unreachable fluents*, and they can be detected with the same argument as above:

FReach(f)
> **If** $\Pi_{\{f\}}^{mut}$ is unsolvable
> **then** f is a an impossible fluent

Even though these fluents appear very rarely, as will be shown in the experimental trials, it remains linear to test for all fluents whether they are unreachable or not: thus, the computational burden is significantly lower than for other fluent "mutexes". When an unreachable fluent is detected, one can project the whole instance on fluents $F \setminus \{f\}$. Theoretically, one could also remove operators that have f in their positive preconditions: however, any such operator a would also be detected by test PreImp(a), which is more likely to succeed.

5 Experimental evaluation

Our implementation was done in Python 3.10, basing ourselves on the Fast Downward parser [8]. For linear programs, we resorted to the GLOP solver [11], while integer programs were solved with Gurobi [7]. We also used Google ORTools [11] to interface between our program and the solvers. We ran our experiments on a machine running Rocky Linux 8.5, powered by an Intel Xeon E5-2667 v3 processor, with a 30-minutes cutoff and using at most 16GB of memory per instance. Our code is available online [4].

In addition to the evaluation of the linear program, we also implemented a procedure based on the observations of Section 4 . The main loop of this procedure consists in executing sequentially a predetermined list of operations and tests, until the instance is detected as unsolvable or the list is depleted. We elaborate further on this in Section 5.2.

We wished to evaluate our program on two different aspects: first, its ability to detect unsolvable instances, and second, its ability to find additional information when it could not conclude.

Our set of benchmarks consists of the unsolvable instances from the unplannability track of the International Planning Competition 2016 (Unsat IPC), which consists of unsolvable instances. The Unsat IPC also included solvable instances, which we tested our program on, as a sanity check, with success.

[4] https://github.com/arnaudlequen/MPRefinement

Set	Unsat	Total
bag-transport	19	29
bottleneck	25	25
cave-diving	1	25
chessboard-pebbling	23	23
over-tpp	2	30
pegsol-row5	14	15
tetris	20	20
Remaining	0	180
Total	104	347

Table 1: Summary of the results returned by the LP-based criterion, run on the Unsat planning competition benchmark set. Each line corresponds to a domain: a set of instances modelling similar problems. The first column reports instances on which our criterion succeeds, while the second column reports the total number of instances in the benchmark set. Domains for which no instance could be solved are summed up in the last line labeled *Remaining*.

5.1 LP-based criteria

In this section, we show that our LP-based criterion suffices to detect a wide range of unsolvable planning instances. Our results are reported in detail in Table 1.

In essence, about 30% of all instances of the Unsat IPC are almost immediately found to be unsolvable by the sole use of the criterion. These results however vary greatly from one domain to the other, in a very dichotomous fashion: either the domain is (almost) entirely solved through the criterion, either few to no instances can be deemed unsolvable. In the case of domain bag-transport, which seems to be in-between, all instances the criterion has been tried on are actually found to be unsolvable: however, as the last 10 instances are too big to be parsed, we could not run the test on them. We can also note that both linear- and integer-programming-based criteria yield the same results, and that solving the IP-based program did not allow us to improve our results.

Both programs are however very lightweight: in every case, building and solving the program required less than a few seconds. In most cases, the criteria required little more than a few tenths of a second to complete. This further justifies our use of the program in the iterative procedure that we present in the next section.

Our program fails entirely on some domains, where no instance can be solved. While this is often because our criterion simply fails to detect the instance's unsolvability, this can also be due to the size of the model. This is the case of bag-gripper, where the first instance has 5681 fluents and 60604 operators, which prevents us from building the associated linear program. In our assessments of the performances of the criteria, the limitation always came from memory.

5.2 Iterative refinement of the model

In the case where the criterion did not immediately detect that an instance Π is unsolvable, one can resort to the several operations previously introduced. In addition, the order in which operations are executed is also critical. Consider for instance an operator a that is both recognized as a landmark and as a removable operator by our operations. In the case where the operator is first removed, then it can not be detected as a landmark, and we thus missed an opportunity to return that the instance is unsolvable. In the case where a is first detected as a landmark, then our routine terminates successfully by detecting that the instance is unsolvable.

5.2.1 Sequences of operations

We present below the different lists of operations that we chose. Note that all sequences start and end with a simple test of solvability with the criterion: initially with only the information contained in the STRIPS model, and then with all information that could be gathered after all operations.

Linear This sequence comprises all tests and operations that are linear in the size of the instance, i.e. that only require one argument. We tried to put first the tests that were the most likely to succeed, so that the followings tests and operations that come after have more information to work with. We successively apply the following tests on all relevant elements, in that order: LMDet, PreImp, OpCount, FReach, and FNegGoal. By that, we mean that we run LMDet(a) for all $a \in O$, then PreImp(a) for all $a \in O$, etc.

OperatorPreImpossible As will be reported later, the PreImp tests that check an operator's reachability are our most successful ones. We wished to gauge the time it requires and its possible impact on the model by itself.

OperatorDeadLocks Even though we choose this name to contrast with the OperatorPreImpossible sequence, this sequence tests both the reachability (through PreImp) and co-reachability (through ActDLock) of an operator. In our trials, no operator could be shown to be a deadlock, even when we tested after the Linear sequence: as a consequence, we only include this sequence for the sake of completeness.

OperatorCount This sequence consists in finding lower, then upper bounds on the number of times each operator has to appear in any plan. It aims to show that a linear number of integer programs to optimize can be done in reasonable time, while also providing interesting information.

5.2.2 Results

We present our results below. As we prune out instances that can be immediately identified as unsolvable, domains

		Operators			Others		
Set	**Diff.**	**PreImp**	**OpCount**	**Removed**	**LMDet**	**FReach**	**FNegGoal**
cave-diving (14)	**+9**	10.0%	14.1%	10.4%	1.1%	4.8%	3.0%
diagnosis (19)	0	0%	57.0%	11.6%	18.3%	4.6%	17.6%
doc-transfer (5)	0	13.0%	26.4%	27.9%	1.7%	0.0%	39.8%
over-nomystery (2)	0	33.4%	25.7%	34.8%	2.1%	0%	7.4%
over-rovers (8)	0	27.9%	17.2%	29.3%	0%	<0.1%	0%
over-tpp (8)	0	7.4%	54.8%	24.7%	0.3%	0.3%	0%
pegsol (24)	**+24**	13.6%	N/A%	13.6%	0.8%	N/A%	N/A%
sliding-tiles (20)	0	0%	0%	0%	0%	0%	69.2%

Table 2: Statistics for the Linear sequence. The first column with the name of the domain also reports the total number of instances for which the procedure terminated entirely within the time and memory limits. The "Difference" (Diff.) column shows the number of instances that could be found unsolvable during the execution of the procedure, compared to the single use of the criterion reported in Table 1. The next set of columns shows stats for operations related to the deletion of operators. The first pair of columns show the percentage of success of each test, while the last column of the set shows the average total percentage of operators pruned at the end of the sequence of tests. The last three columns show the percentage of success of three other tests. N/A values indicate that no such test was performed as the program terminated before.

that are immediately found unsolvable by the criterion are not reported.

Linear sequence Table 2 shows statistics for the Linear sequence. The main goal of our routine is to extract additional information from the model, so that another procedure that comes after can more easily show it unsolvable. However, we could notice that our algorithm was sometimes enough to detect unsolvable instances that are otherwise not detected as such by the criterion. There are few examples of such instances (about 9.5% of the entire benchmark set), and they are grouped in only two domains (cave-diving and pegsol). Nonetheless, they suffice to show that a well-chosen sequence of operations can sometimes replace a search, and that our work paves the way for further research in that regard.

In the cases where our procedure could not conclude, it still manages to gather valuable information about the planning instance. For example, on some domains, almost a third of all operators are pruned on average, among instances on which our procedure terminates.

The termination of our procedure is, however, the main issue of this sequence of operations, which is too computationaly costly, and often stops early because of the time and memory limits imposed. In some domains, very few instances could be run through the entire sequence of operations: such domains include over-nomystery, where this sequence terminated on only 2 instances out of the 24 that could be parsed.

Individual tests Table 3 summarizes the statistics for the other sequences, that mostly consists of series of one or two of the same operations. However, it does not report comprehensive results for all remaining sequences: indeed, in the case of the OperatorDeadLock sequence, no test answered positively. Thus, no dead-end operator could be found.

Nonetheless, the results for the other sequences of operations are encouraging. Be it for the sequence centered on PreImp or the one focused on OpCount operations, a significant proportion of operators could be removed. In some cases, it suffices to show that the instance was not solvable, as is the case for the cave-diving or pegsol domains. However, the time required for the computation is significant, which is discussed in the next section.

Note that these sequences of tests are not as powerful as the Linear sequence, when it comes to detecting unsolvable instances. This seems to indicate that the combination of different kinds of operations is crucial to draw conclusions, and studying their interactions is crucial in designing more powerful sequences.

6 Related work

The surge in interest for unsolvability detection, in the last decade, has been embodied by the first Unsolvability Planning Competition in 2016. The competition saw various adaptations of techniques that have shown themselves efficient for finding plans, in a state-space search. Such methods include heuristics specifically tailored for unsolvability detection, such as a Merge & Shrink-based heuristic [9] (which precedes the competition). Such heuristics rely on abstractions that do not preserve distance, but merely solvability.

Another heuristic that was successfully adapted was the operator-counting heuristic [2, 12, 18]. The heuristic is based on a relaxation of the orderings of the operators. Previous works showed that it admits a linear programming formulation, similar to the Linear Program 1 that we propose.

	OperatorPreImpossible				OperatorCount					
Set	**Cpt**	**Rem.**	**Diff.**	**Time**	**Cpt**	**OpCount$^-$**	**OpCount$^+$**	**Rem.**	**Diff.**	**Time**
bag-barman	4	77.2%	0	1177.8	0	.	.	.	.	.
cave-diving	17	6.5%	**+4**	147.8	17	0.9%	28.0%	7.0%	0	329.3
diagnosis	20	0%	0	6.4	20	16.9%	96.3%	19.5%	0	91.6
document-transfer	13	0%	0	475.7	8	1.7%	50.7%	29.8%	0	643.2
over-nomystery	10	18.8%	0	587.8	3	1.4%	87.2%	3.9%	0	746.2
over-rovers	11	21.9%	0	370.2	9	0%	62.4%	5.2%	0	455.1
over-tpp	14	<0.1%	0	268.1	9	0.3%	65.4%	20.2%	0	428.8
pegsol	24	16.4%	**+6**	0.6	24	0%	8.2%	3.0%	**+22**	0.51
sliding-tiles	20	0%	0	5.6	20	0%	0%	0%	0	19.4

Table 3: Performances of the individually run operations. The *Completed* (Cpt) columns show the number of instances the sequence terminated on, the *Diff.* columns show the number of instances solved thanks to the iterative refinement, and the *Time* columns show, in seconds, the average time per instance. The *Rem.* columns show the average percentage of operators that could be removed thanks to the operation. OpCount$^+$ and OpCount$^-$ columns report the average percentage of success of their respective operations.

However, while we only optimize the variable associated to the count of a single operator, the objective function that they minimize is the total cost of the plan. The adaptation of the linear program to the case of unsolvability detection, was carried out by the Fast Downward-based unsolvability planner Aidos [13]. It consists in checking the existence of a solution, in the same way as for Linear Program 2. However, Aidos uses this component in a state-space search, in order to detect dead-ends.

More generally, be it in unsolvable or in solvable planning tasks, the early detection of states that can not lead to a goal makes can help prune out whole branches of the search space. In the case of dead-end detection [4], various works have focused on the elaboration of formulas that can be efficiently evaluated, and whose only models are states that can not lead to a goal state. The notion of dead-end formula has been generalized with the notion of traps [10]: a formula ϕ such that, once it's verified in a state s, all states reachable from s will satisfy it too. A formula ϕ that is inconsistent with the goal then shows that the current branch is not worth exploring.

In the case where our algorithm does not manage to find that the task is unsolvable, it still manages to remove unnecessary elements from the planning model, to make the task easier for the next algorithm. Various other methods prune the model in a preprocessing step: in [1], the authors show that invariants in the form of mutexes can be leveraged to remove operators that will never be part of a plan. In [5], it is shown how to combine symmetries of the planning task and operator mutexes to find operators that are redundant, in the sense that removing them preserves at least one solution-plan.

Our algorithm also learns information that is not explicitly expressible in a STRIPS planning instance. In [16], the authors draw inspiration from a well-known technique in SAT solving, to learn clauses that recognize dead-ends, through a conflict-driven approach during search. They also show how to learn traps online [15]. Learning is ubiquitous in generalized planning, which is a domain concerned with the synthesis of generalized plans, which are procedures that solve multiple instances. For instance, previous work [17] proposed to learn heuristics in the form of logical formulas, out of a set of small examples instances, so as to recognize unsolvable planning instances.

In [3], another polynomial criterion is proposed to immediately detect a class of unsolvable instances without resorting to search. The authors synthesize a function that separates the initial state from all goal states, through a linear combination of features valued in a finite field. Akin to our criterion, their technique is incomplete, but it is very efficient at detecting parity arguments.

7 Discussion and conclusion

Section 5 showed that, when our criterion failed to show an instance unsolvable, it was still possible to extract additional information from the model by leveraging the criterion. Even more so, in some cases, otherwise undetected unsolvable instances could be identified as such by this means. Yet, there is still a lot of room for improvement: a more in-depth study of our operations, as well as their interactions, could help us fine-tune the algorithm. Indeed, not all sequences of tests are equal in all aspects, and finding a sequence that avoid unnecessary computations is a way to optimize our algorithm, and to boost its detection power.

In our tests, we choose to simply run pre-determined sequences of operations and tests. This means that, regardless of how tests succeed or fail, the algorithm will linearly go through the same sequence of operations, except if it can show preemptively that an instance is unsolvable. However, the outcome of some test may help in finding which step to take next. For instance, after finding that an operator is a

landmark, it might be interesting to check right away if it can be removed.

One of the main weaknesses of our iterative refinement algorithm is its computational cost. Even the most lightweight sequences, such as the OperatorPreImpossible sequence, takes significant time to complete. Our program builds each linear program from scratch each time a test is performed. However, very few constraints differ from one linear program to the other; thus, one could modify only these constraints from one test to the next, in order to save significant time.

As a conclusion, we showed that a simple criterion was sometimes enough to prove that a planning instance is unsolvable. Even though our program is non-optimised, we have still managed to show that resorting to a search is not always necessary, as reasoning on the model directly can suffice. Even when our procedure fails, it still gathers valuable information about the instance, that can help a complete procedure terminate faster.

References

[1] Alcázar, Vidal and Alvaro Torralba: *A reminder about the importance of computing and exploiting invariants in planning*. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, pages 2–6, 2015.

[2] Bonet, Blai: *An admissible heuristic for SAS+ planning obtained from the state equation*. In *IJCAI*, 2013.

[3] Christen, Remo, Salomé Eriksson, Florian Pommerening, and Malte Helmert: *Detecting unsolvability based on separating functions*. In Kumar, Akshat, Sylvie Thiébaux, Pradeep Varakantham, and William Yeoh (editors): *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling, ICAPS 2022*, pages 44–52, 2022. https://ojs.aaai.org/index.php/ICAPS/article/view/19784.

[4] Cserna, Bence, William Doyle, Jordan Ramsdell, and Wheeler Ruml: *Avoiding dead ends in real-time heuristic search*. In *AAAI*, 2018.

[5] Fišer, Daniel, Alvaro Torralba, and Alexander Shleyfman: *Operator mutexes and symmetries for simplifying planning tasks*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7586–7593, 2019.

[6] Geffner, Hector and Blai Bonet: *A concise introduction to models and methods for automated planning*. Morgan & Claypool Publishers, 2013.

[7] Gurobi Optimization, LLC: *Gurobi Optimizer Reference Manual*, 2023. https://www.gurobi.com.

[8] Helmert, Malte: *The fast downward planning system*. JAIR, 26:191–246, 2006.

[9] Helmert, Malte, Patrik Haslum, Jörg Hoffmann, and Raz Nissim: *Merge-and-shrink abstraction: A method for generating lower bounds in factored state spaces*. J. ACM, 61(3):16:1–16:63, 2014. https://doi.org/10.1145/2559951.

[10] Lipovetzky, Nir, Christian Muise, and Hector Geffner: *Traps, invariants, and dead-ends*. In *ICAPS*, pages 211–215, 2016.

[11] Perron, Laurent and Vincent Furnon: *Or-tools*. https://developers.google.com/optimization/.

[12] Pommerening, Florian, Gabriele Röger, Malte Helmert, and Blai Bonet: *LP-based heuristics for cost-optimal planning*. In *ICAPS*, pages 226–234, 2014.

[13] Seipp, Jendrik, Florian Pommerening, Silvan Sievers, Martin Wehrle, Chris Fawcett, and Yusra Alkhazraji: *Fast Downward Aidos*. Unsolvability International Planning Competition: planner abstracts, pages 28–38, 2016.

[14] Steinmetz, Marcel, Jörg Hoffmann, Alisa Kovtunova, and Stefan Borgwardt: *Classical planning with avoid conditions*. In *AAAI*, pages 9944–9952, 2022.

[15] Steinmetz, Marcel and Jörg Hoffmann: *Search and learn: On dead-end detectors, the traps they set, and trap learning*. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4398–4404, 2017. https://doi.org/10.24963/ijcai.2017/614.

[16] Steinmetz, Marcel and Jörg Hoffmann: *State space search nogood learning: Online refinement of critical-path dead-end detectors in planning*. Artificial Intelligence, 245:1–37, 2017, ISSN 0004-3702. https://www.sciencedirect.com/science/article/pii/S0004370216301448.

[17] Ståhlberg, Simon, Guillem Francès, and Jendrik Seipp: *Learning generalized unsolvability heuristics for classical planning*. In Zhou, Zhi Hua (editor): *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4175–4181, 2021. https://doi.org/10.24963/ijcai.2021/574.

[18] Van Den Briel, Menkes, J Benton, Subbarao Kambhampati, and Thomas Vossen: *An LP-based heuristic for optimal planning*. In *Principles and Practice of Constraint Programming–CP 2007: 13th International Conference*, pages 651–665. Springer, 2007.

Actes JIAF-JFPDA 2023

Comment rendre des comportements plus prédictibles

Salomé Lepers Vincent Thomas Olivier Buffet

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
prénom.nom@loria.fr

Résumé

Dans cet article, nous nous intéressons à des problèmes de prédictibilité, c'est à dire pour lesquels un agent doit choisir sa stratégie dans le but d'optimiser les prédictions que pourrait faire un observateur extérieur. Nous abordons ces problèmes en tenant compte des incertitudes sur la dynamique de l'environnement et sur la politique de l'agent observé. Dans ce but, nous faisons l'hypothèse que l'observateur 1. cherche à prédire l'action ou l'état future de l'agent à chaque pas de temps, et 2. suppose que l'agent agit selon une politique stochastique calculée à partir d'un problème sous-jacent connu, et nous nous appuyons sur le cadre des processus de décision markoviens conscients d'un observateur (OAMDP). Nous considérons différents critères de performance candidats pour la prédictibilité à travers des fonctions de récompense construit sur la croyance de l'observateur concernant la politique de l'agent ; montrons que ces OAMDP prédictibles induits peuvent être représentés par des MDP orientés but ou actualisés ; et analysons les propriétés des fonctions de récompense proposées à la fois théoriquement et empiriquement sur deux types de mondes grilles.

Abstract

In this paper, we are interested in predictability problems, wherein an agent must choose its strategy in order to optimize the predictions that an external observer could make. We address these problems while taking into account uncertainties on the environment dynamics and on the observed agent's policy. To that end, we assume that the observer 1. seeks to predict the agent's future action or state at each time step, and 2. models the agent using a stochastic policy computed from a known underlying problem, and we leverage on the framework of observer-aware Markov decision processes (OAMDPs). We consider several candidate predictability performance criteria through reward functions built on the observer's belief about the agent policy; show that these induced *predictable* OAMDPs can be represented by goal-oriented or discounted MDPs; and analyze the properties of the proposed reward functions both theoretically and empirically on two types of grid-world problems.

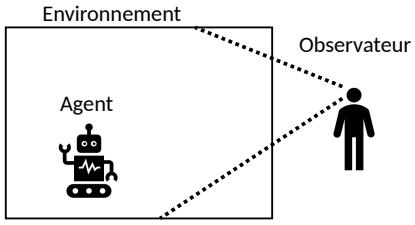


FIGURE 1 – Agent dans son environnement et un observateur passif

1 Introduction

Dans des situations de collaboration homme-robot, certaines propriétés du comportement du robot peuvent être appréciées de l'humain, voire permettre une meilleure collaboration. Divers travaux récents ont porté sur l'obtention automatique de comportements dotés de telles propriétés, en particulier dans le cas où l'humain ne fait qu'observer l'agent dans son environnement, et où l'agent, conscient de cet observateur, cherche à adopter un comportement qui permette de contrôler au mieux les informations acquises par l'humain (cf. figure 1).

CHAKRABORTI, KULKARNI, SREEDHARAN et al. [1] proposent une taxonomie des différents concepts rencontrés dans ces travaux, certains cherchant 1. à transmettre de l'information, tels que la *lisibilité* (lorsque l'agent essaye de communiquer son but à travers ses choix d'actions), l'*explicabilité* (un comportement explicable est conforme aux attentes de l'observateur), et la *predictabilité* (un comportement est prédictible si il est facile de deviner la fin d'une trajectoire en cours), ou 2. d'autres à cacher de l'information, par exemple l'*obscurcissement*, quand le comportement vise à cacher la tâche réelle de l'agent. Ils formalisent aussi ces différents problèmes de manière unifiée sous l'hypothèse que les transitions sont déterministes, raisonnant donc principalement sur des plans (une séquence d'actions

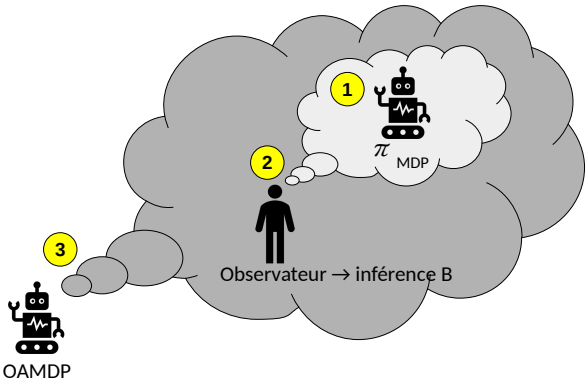


FIGURE 2 – Un agent OAMDP (3) fait l'hypothèse que l'observateur s'attend (2) à ce que l'agent se comporte de manière à accomplir une certaine tâche (1).

induisant une unique séquence d'états). Dans leur approche, le robot modélise l'humain comme ayant un certain modèle du système robot+environnement (y compris de la ou les tâches possibles du robot), et pouvant ainsi anticiper les comportements possibles du robot. Chacune de ces propriétés peut être intéressante dans certaines situations et transmet différentes informations à l'observateur. CHAKRABORTI, KULKARNI, SREEDHARAN et al. [1] expliquent qu'un plan explicable peut être imprévisible, notamment dans le cas où il existe plusieurs plans explicables. FISAC, LIU, HAMRICK et al. [2] suggèrent que, lorsqu'un agent agit de façon lisible, il est possible d'inférer son but mais pas forcément la façon dont il va atteindre ce but (auquel cas il aurait un comportement prédictible).

MIURA et ZILBERSTEIN [3], pour leur part, proposent un formalisme générique analogue (voir figure 2), mais sous l'hypothèse de transitions stochastiques, d'où le nom de *processus de décision markovien conscient d'un observateur* (OAMDP pour *observer-aware Markov decision process*). Entre autres choses, ils travaillent aussi sur l'explicabilité, la lisibilité et la prédictibilité. Comme MIURA et ZILBERSTEIN l'exposent, DRAGAN, LEE et SRINIVASA [4] ont proposé, sous hypothèse de transitions déterministes, de modéliser la prédictibilité d'une trajectoire comme proportionnelle à sa valeur, ce qui peut se traduire dans le cadre OAMDP par la maximisation de la récompense sous-jacente. FISAC, LIU, HAMRICK et al. [2], pour leur part, ont proposé de modéliser des agents t-prédictibles comme maximisant $P(a_{t+1}, \ldots, a_T | a_1, \ldots, a_t)$, ce qui peut être adapté dans le cas stochastique, mais avec un fort coût computationnel [3]. L'objectif de cet article est donc de proposer une nouvelle façon de modéliser la prédictibilité raisonnant non pas sur des séquences d'actions complètes, comme peuvent y inciter les travaux dans des cadres déterministes, mais sur les choix d'actions dans chaque état rencontré. Cela implique de raisonner sur des types dynamiques, ce qui requiert d'introduire une variante du formalisme OAMDP. En outre, nous ne considérons pas que des problèmes avec facteur d'actualisation, mais aussi des problèmes de type "chemin stochastique le plus court" (orientés vers des buts), étendant ainsi le cadre OAMDP.

La section 2 introduit des pré-requis sur le processus de décision markoviens (MDP) et les MDP conscients d'un observateur. Notre approche général et quelques fonctions de récompense candidates l'implémentant sont décrites en sec. 3. Des expérimentations sont décrites en sec. 4 pour illustrer et analyser davantage les comportements obtenus dans différentes configurations avant de conclure en sec. 5.

2 Pré-requis

Nous présentons d'abord brièvement les processus de décision markoviens avant de passer au cadre des MDP conscients d'un observateur.

2.1 Processus de décision markoviens

Un *processus de décision markovien* (MDP) est un 6-uplet $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma, \mathcal{S}_f \rangle$ où :

- $\mathcal{S}$ est l'ensemble des états ;
- $\mathcal{A}$ est l'ensemble des actions ;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0; 1]$, la fonction de transition, donne la probabilité $T(s, a, s')$ d'aller dans un état s' depuis un état s en exécutant l'action a ;
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, la fonction de récompense, donne la récompense reçue $R(s, a, s')$ lors d'une transition (s, a, s') ;
- $\gamma \in [0, 1]$ est le facteur d'actualisation ; et
- $\mathcal{S}_f \subset \mathcal{S}$ est l'ensemble des états terminaux : pour tout $s, a \in \mathcal{S} \times \mathcal{A}$, $T(s, a, s) = 1$ et $R(s, a, s) = 0$.

Une politique $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ détermine un comportement en associant à chaque état une action à effectuer. Elle peut éventuellement être stochastique, $\pi(a|s)$ étant alors la probabilité d'effectuer a dans l'état s. Considérant un *MDP actualisé*, c'est-à-dire tel que $\gamma < 1$, la valeur d'une politique π en un état s est l'espérance de la somme des récompenses actualisées sur un horizon infini :

$$V^\pi(s) \stackrel{\text{def}}{=} \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) | S_0 = s \right].$$

Il existe toujours au moins une politique π^*, dite optimale, telle que, pour tout s, $V^{\pi*}(s) = \max_\pi V^\pi(s)$. L'algorithme d'*itération sur la valeur* (VI) calcule cette fonction de valeur optimale, notée V^*, en itérant le calcul suivant jusqu'à atteindre une précision suffisante (où k désigne l'itération courante) :

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s, a, s') \cdot (R(s, a, s') + \gamma V_k(s')) .$$

On peut alors dériver une politique déterministe optimale en agissant de "manière gourmande" dans tout état s avec :

$$\pi^*(s) \leftarrow \arg\max_a \sum_{s'} T(s,a,s') \cdot (R(s,a,s') + \gamma V^*(s')) \,.$$

On interrompt les calculs quand le *résidu de Bellman* est inférieur à un seuil fonction de l'erreur ϵ souhaitée et de γ :

$$\underbrace{\max_s |V_{k+1}(s) - V_k(s)|}_{\text{résidu de Bellman}} \leq \frac{1-\gamma}{\gamma}\epsilon.$$

Les propriétés ci-dessus restent valident avec $\gamma = 1$ si

1. $\mathcal{S}_f$ non vide ; et
2. R est telle qu'il existe des politiques atteignant $\mathcal{S}_f$ avec probabilité 1 depuis tout état s, et que la valeur des autres politiques diverge vers $-\infty$ dans les états depuis lesquels on ne peut pas être sûr de pouvoir atteindre un état terminal.

On parle alors de problème de type *chemin stochastique le plus court* (SSP). On a un SSP en particulier, si, pour tout $(s,a,s') \in (\mathcal{S} \setminus \mathcal{S}_f) \times \mathcal{A} \times \mathcal{S}$, $r(s,a,s') < 0$, c'est-à-dire si on cherche à atteindre un état terminal à "moindre coût" (en moyenne).

Note : On peut transformer tout MDP actualisé en un SSP dans lequel, à chaque instant, on a une probabilité $1-\gamma$ de transiter vers un état terminal. Le cas SSP est donc plus général.

2.2 MDP conscient d'un observateur

Un *MDP conscient d'un observateur* (OA-MDP pour *observer-aware MDP*) [3] décrit une situation dans laquelle un agent interagit avec son environnement en ayant conscience de la présence d'un observateur, et en cherchant à maximiser un critère de performance lié aux croyances de cet observateur. Il est défini formellement par un 8-uplet $\langle S, A, T, \gamma, \mathcal{S}_f, \Theta, B, R\rangle$ où :

— $\langle \mathcal{S}, \mathcal{A}, T, \gamma, \mathcal{S}_f \rangle$ est un MDP sans fonction de récompense ;

— Θ est un ensemble fini de *types* possibles de l'agent, représentant une caractéristique de celui-ci telle que sa tâche réelle ou ses capacités ;

— $B : H^* \to \Delta^{|\Theta|}$ donne la croyance que l'observateur a sur le type de l'agent en fonction de l'historique des états et des actions ($H = \mathcal{S} \times \mathcal{A}$) ;

— $R : \mathcal{S} \times \mathcal{A} \times \Delta^{|\Theta|} \to \mathbb{R}$ est la fonction de récompense.

Dans la plupart des cas considérés par Miura et Zilberstein, B est obtenue en s'appuyant sur la définition de la mise-à-jour de croyance bayésienne BST de Baker, Saxe et Tenenbaum, c'est à dire en considérant que, du point de vue de l'agent, l'observateur modélise le comportement de l'agent pour une tâche donnée à travers un MDP en :

1. utilisant une fonction de récompense R_{MDP} approprié
2. résolvant le MDP $\langle \mathcal{S}, \mathcal{A}, T, \gamma, \mathcal{S}_f \rangle$ (où tout les composants excepté la fonction de récompense R_{MDP} émane de la définition de l'OAMDP) pour obtenir V^*_{MDP} ;
3. construisant une politique 'softmax' tel que pour chaque couple (s,a),

$$\pi_{\text{MDP}}(a|s) = \frac{e^{\frac{1}{\tau}Q^*_{\text{MDP}}(s,a)}}{\sum_{a'} e^{\frac{1}{\tau}Q^*_{\text{MDP}}(s,a')}}, \text{ où}$$
$$Q^*_{\text{MDP}}(s,a) = \sum_{s'} T(s,a,s') \cdot \left(r(s,a,s') + \gamma V^*_{\text{MDP}}(s')\right),$$

avec $\tau > 0$ représentant le niveau de rationalité de l'agent (considéré par l'observateur) pouvant être utilisé pour travailler avec des politiques plus ou moins optimale.

La croyance de l'observateur sur les types peut ensuite être obtenue par inférence bayésienne en utilisant π_{MDP}.

Miura et Zilberstein formalisent ainsi, entre autres, des problèmes de lisibilité, d'explicabilité, et de prédictibilité. Pour la prédictibilité, sur laquelle nous nous concentrons maintenant, Miura et Zilberstein proposent deux approches. La première repose sur les travaux de Dragan, Lee et Srinivasa, où la prédictibilité d'une trajectoire est modélisée comme étant proportionnelle à sa valeur (définit comme son coût négatif) [4]. Cela revient à optimiser la fonction de récompense R_{MDP}, donc à agir de manière gloutonne par rapport à Q^*_{MDP} (plutôt que de suivre π_{MDP}. La seconde approche repose sur la t-prédictibilité de Fisac, Liu, Hamrick et al. [2], laquelle maximise $Pr(a_{t+1}, \ldots, a_T | a_1, \ldots, a_t)$ dans des contextes déterministes en utilisant un type pour chaque trajectoire possible, c'est-à-dire un nombre exponentiel de types.

Dans la suite, nous proposons une approche alternative pour la prédictibilité et discutons ses propriétés.

3 Contribution

Kolobov, Mausam, Weld et al. [6] considèrent uniquement des OAMDP actualisés. Comme pour les MDP, on distinguera ici deux classes d'OAMDP : les OAMDP actualisés, et les OASSP (en utilisant $\gamma = 1$). En particulier, on se demandera sous quelles conditions un problème orienté but permet de construire un SSP valide.

3.1 MDP conscient d'un observateur et prédictible

Les deux approches de la prédictibilité proposées par Miura et Zilberstein s'inspirent de travaux dans des situations déterministes où il est naturel de raisonner sur les trajectoires. Parce qu'aussi bien la politique softmax π_{MDP} et la dynamique du système peuvent être stochastiques, on propose d'essayer de prédire soit l'action, soit l'état de l'agent,

chacune des deux alternatives pouvant amener à des résultats différents. Cependant, les types pour les OAMDP sont des variables statiques (comme les types des jeux bayésiens) alors que les actions et les états sont dynamiques. Cela nous amène à introduire les pOAMDP (OAMDP prédictibles), où le type (dynamique) est maintenant une fonction de la transition courante : $\theta_t = \tau(s_t, a_t, s_{t+1})$. Cela 1. ne permet pas d'encoder les problèmes où le type est statique et caché de l'observateur comme pour la lisibilité ou l'explicabilité, 2. mais permet toujours de définir et de résoudre le MDP de l'observateur (car le type n'influence pas la dynamique du système), et d'utiliser la mise à jour de croyance bayésienne (à cause de la nature markovienne des types dynamiques).

La section suivante décrit respectivement, pour la prédictibilité sur les actions et pour la prédictibilité sur les états, 1. comment dériver B et comment résoudre le pOAMDP étant donnée une fonction de récompense R, et 2. différentes fonctions de récompense candidates.

3.2 Fonction de croyance et propriété du pOAMDP

Pour la prédictibilité sur les actions, $\Theta = \mathcal{A}$, $\tau(s, a, s') = a$, et B est :

$$B : \begin{array}{ccc} H^* & \to & \Delta^{|\mathcal{A}|}, \\ (s_0, a_0, \ldots, s_t) & \mapsto & \pi_{\text{MDP}}(A_t|s_t). \end{array}$$

Pour la prédictibilité sur les états, $\Theta = \mathcal{S}$, $\tau(s, a, s') = s'$, et B est

$$B : \begin{array}{ccc} H^* & \to & \Delta^{|\mathcal{S}|}, \\ (s_0, a_0, \ldots, s_t) & \mapsto & \sum_{a'} \pi_{\text{MDP}}(a'|s_t) \cdot T(s, a', S_{t+1}). \end{array}$$

Dans les deux cas, B dépend uniquement de l'état courant, s_t, et on peut alors redéfinir la fonction de récompense du pOAMDP comme $R'(s_t, a_t) \stackrel{\text{def}}{=} R(s_t, a_t, B(s_t))$, et la croyance sur θ en s comme $b_s(\theta)$. Le problème de planification de l'agent peut alors être défini comme un MDP $\langle \mathcal{S}, \mathcal{A}, T, R', \gamma, \mathcal{S}_f \rangle$ qui peut être résolu par un algorithme comme Itération sur la valeur. En conséquence, la complexité de résolution correspond à la complexité de résolution de deux MDP : celui "du MDP de l'observateur", puis celui "du MDP induit par le pOAMDP". Dans le cas général [3], il n'est pas possible d'obtenir un tel MDP, et résoudre un OAMDP demande d'utiliser des algorithmes spécifiques dans lesquels le choix d'action est lié à l'historique état-action entière.

3.3 Possibles fonctions de récompense

Nous présentons ici 4 fonctions de récompenses candidates considérées qui peuvent être définies sur les états ou sur les actions.

[Confiance] $R^{\Theta}_{\mathbf{max}}(s, a, s') \stackrel{\text{def}}{=} \max_\theta b_t(\theta)$ **:** Cette première fonction récompense l'agent proportionnellement à la plus grande croyance sur les types possibles dans l'état courant s_t : $\max_\theta b_t(\theta)$. En d'autres termes, elle favorise les états où la règle de décision immédiate prévue par l'observateur est plus déterministe. Comme on le verra dans les expérimentations, cette définition conduit à des comportements qui ne répondent pas à notre besoin, essentiellement parce que le choix de l'action courante n'influe pas sur la récompense immédiate. L'agent a tendance à préférer rester dans des états où la croyance de l'observateur est plus déterministe, ce qui peut conduire en pratique à des comportements peu prévisibles.

[Probabilité] $R^{\Theta}_{\mathbf{pr}}(s, a, s') \stackrel{\text{def}}{=} b_t(\tau(s, a, s'))$ **:** Cette deuxième fonction de récompense favorise les états où la prochaine action ou le prochain état est plus prédictible. En d'autres termes, elle favorise le fait d'agir comme le prévoit l'observateur. Toutefois, cette fonction étant à valeurs positives, elle n'est pas appropriée pour les pOASSP, d'où les propositions suivantes.

[Regret] $R^{\Theta}_{\mathbf{regret}}(s, a, s') \stackrel{\text{def}}{=} b_t(\tau(s, a, s')) - \max_\theta b_t(\theta)$ **:** Cette fonction de récompense repose sur le concept de regret (de faire un choix sous-optimal C à la place du choix optimal C^*). On notera que cette fonction de récompense est strictement négative, sauf quand $b_s(\theta)$ est maximale pour l'état s courant, auquel cas elle est nulle. Dans le cas de la prédictibilité sur les actions, les solutions optimales résultant de cette fonction de récompense sont les solutions optimales du MDP résolu par l'observateur. Ce résultat correspond à l'adaptation par Miura et Zilberstein de l'approche de Dragan, Lee et Srinivasa. Dans le cas de la prédictibilité sur les états, il existe des situations où les solutions optimales sont différentes. Un tel cas est illustré dans la partie expérimentale. Une question ouverte reste de savoir si R^S_{regret} induit toujours un SSP valide avec $\gamma = 1$

[Coût] $R^{\Theta}_{\mathbf{cost}}(s, a, s') \stackrel{\text{def}}{=} b_t(\tau(s, a, s')) - 1$ **:** Cette seconde fonction de récompense négative est équivalente à R^{Θ}_{pr} dans le cadre des pOAMDP actualisés (qui sont, comme les MDP actualisés, invariants à l'ajout d'une constante à la fonction de récompense, et ont donc les mêmes solutions optimales).

Une première observation, détaillé dans les propositions suivante, est que cette fonction de récompense induit des pOASSP valides.

Proposition 1. *Supposons que (i) $\gamma = 1$, (ii) le MDP considéré par l'observateur est un SSP valide, et (iii) R^A_{cost} est la fonction de récompense du pOAMDP. Alors le pOAMDP est un problème bien défini car il induit un SSP valide.*

Démonstration. Raisonnons par l'absurde en supposant qu'il existe une politique optimale π^* pouvant atteindre

un sous-ensemble d'états $\mathcal{S}' \subset (\mathcal{S} \setminus \mathcal{S}_f)$ et y rester indéfiniment à "coût nul" (la fonction de récompense étant à valeurs négatives ou nulles). Or, agir à coût nul signifierait ici choisir, dans tout état $s \in \mathcal{S}'$, une action a telle que, pour tout s' possible, $R^A_{\text{cost}}(s, a, s') = 0 = \pi_{\text{MDP}}(a|s) - 1$, c.-à-d. $\pi_{\text{MDP}}(a|s) = 1$. Cela signifierait que, à l'intérieur de $\mathcal{S}'$, l'agent n'effectue que des actions optimales pour le MDP de l'observateur. Pourtant, comme le MDP de l'observateur est un SSP valide, ces actions optimales devrait faire sortir l'agent de $\mathcal{S}'$ (qui ne contient pas d'états terminaux) avec probabilité 1. □

Le même résultat peut être obtenu pour la prédictibilité sur les états. La preuve diffère en ce que la propriété $\pi_{\text{MDP}}(a|s) = 1$ est remplacée par le déterminisme de la fonction de transition T pour les actions échantillonnées.

Proposition 2. *Supposons que (i) $\gamma = 1$, (ii) le MDP considéré par l'observateur est un SSP valide, et (iii) R^S_{cost} est la fonction de récompense du pOAMDP. Alors le pOAMDP est un problème bien défini car il induit un SSP valide.*

Pour la prédictabilité sur les actions, nous pouvons aussi donner une interprétation du critère de performance obtenu. En effet, cette fonction de récompense est *l'opposé de la probabilité de* ne pas *prendre l'action qui serait échantillonnée si l'on suivait* π_{MDP}, ce qui s'écrit formellement $R^A_{\text{cost}}(s, a, s') = -P(A_{\text{MDP}} \neq a|S_t = s)$. Dans le cas $\gamma = 1$, la somme des récompenses sur une trajectoire s'écrit donc :

$$\sum_t R^A_{\text{cost}}(s_t, a_t, s_{t+1}) = -\sum_t P(A_{\text{MDP}} \neq a_t|S_t = s_t).$$

Ainsi, pour une politique π donnée qui atteint un état terminal avec probabilité 1 et pour un état s, $-V_\pi(s)$ est, quand on exécute π de s à un état terminal, *l'espérance du nombre d'actions échantillonnées en utilisant π_{MDP} (c.-à-d. les prédictions de l'observateur) en désaccord avec les actions réellement effectuées.*

Pour résumer cette section, les quatre fonctions de récompenses considérées sont

$$\begin{aligned}
R^A_{\max}(s, a, s') &\stackrel{\text{def}}{=} \max_{a'} \pi_{\text{MDP}}(a'|s),\\
R^S_{\max}(s, a, s') &\stackrel{\text{def}}{=} \max_{s''} \sum_{a'} \pi_{\text{MDP}}(a'|s) T(s, a', s''),\\
R^A_{\text{Pr}}(s, a, s') &\stackrel{\text{def}}{=} \pi_{\text{MDP}}(a|s),\\
R^S_{\text{Pr}}(s, a, s') &\stackrel{\text{def}}{=} \sum_{a'} \pi_{\text{MDP}}(a'|s) T(s, a', s'),\\
R^A_{\text{regret}}(s, a, s') &\stackrel{\text{def}}{=} \pi_{\text{MDP}}(a|s) - \max_{a'} \pi_{\text{MDP}}(a'|s),\\
R^S_{\text{regret}}(s, a, s') &\stackrel{\text{def}}{=} \sum_{a'} \pi_{\text{MDP}}(a'|s) T(s, a', s')\\
&\qquad - \max_{s''} \sum_{a'} \pi_{\text{MDP}}(a'|s) T(s, a', s''),\\
R^A_{\text{cost}}(s, a, s') &\stackrel{\text{def}}{=} \pi_{\text{MDP}}(a|s) - 1, \text{ and}\\
R^S_{\text{cost}}(s, a, s') &\stackrel{\text{def}}{=} \sum_{a'} \pi_{\text{MDP}}(a'|s) T(s, a', s') - 1.
\end{aligned}$$

La section suivante va les étudier sur des exemples simples.

4 Résultats expérimentaux

Le but des expérimentations est d'illustrer et de mieux comprendre les politiques obtenues à partir des fonctions de récompense. En particulier, on souhaite déterminer si ces politiques peuvent être considérées comme prédictible.

4.1 Protocole

Pour décrire les deux types de pOAMDP considérés dans nos expériences, détaillons les MDP correspondant pris en compte par l'observateur :

— un SSP, nommé *labyrinthe*, dans lequel l'agent se déplace dans un monde grille pour atteindre un état but terminal ;

— un MDP actualisé (sans état terminal), nommé *pompier*, dans lequel l'agent utilise pour éteindre des feux.

Pour faciliter les analyses, la plupart des problèmes ont une dynamique déterministe.

Problème MDP *labyrinthe* : Un *labyrinthe* (voir figure 3.a) est défini par un monde à grille avec des murs (cases grises), des cellules normales (en blanc), des cellules glissantes (en cyan), et des cellules terminales (disques roses). Plus formellement, dans ce SSP :

— chaque état s dans $\mathcal{S}$ indique les coordonnées (x, y) de l'agent dans une case normale, glissante ou terminale ;

— $\mathcal{S}_f$ est un sous-ensemble non-vide (mais aussi éventuellement non-singleton) de $\mathcal{S}$;

— $\mathcal{A} = \{up, down, left, right\}$;

— $T(s, a, s')$ encode les mouvements de l'agent : l'agent dans une cellule normale bouge dans la direction indiquée par son action si aucun mur ne l'empêche ; dans une cellule glissante, l'agent a une probabilité p (0.5 dans nos expérimentations) de faire un mouvement de 2 cellules plutôt qu'une (si possible) ; dans une cellule terminale, l'agent ne bouge pas ;

— R_{MDP}, la fonction de récompense, retourne (i) une pénalité par défaut de $-0,04$ pour chaque action, (ii) -1 si l'agent touche un mur, (iii) $+1$ s'il atteint un état terminal s_f, et (iv) 0 quand l'agent reste dans un état terminal.

Ce problème définit bien un SSP puisque toutes les récompenses ne menant pas un état terminal sont strictement négatives. La politique stochastique π_{MDP} est calculée avec l'algorithme d'itération sur la valeur avec $\gamma = 1$.

Problème MDP *pompier* : Le problème *pompier* utilise des grilles similaires, mais sans états terminaux, et avec des cellules représentant des feux et des sources d'eau (voir figure 8). L'agent a maintenant un réservoir d'eau, qui est vidé quand un feu (inextinguible) est attenit, et rempli quand une source d'eau (jamais vide) est atteinte. Plus formellement, dans ce MDP actualisé :

— chaque état s de $\mathcal{S}$ est représenté par un triplet (x, y, w) avec (x, y) les coordonnées de l'agent et w un booléen indiquant si le réservoir est plein ou vide ;

— $\mathcal{A} = \{up, down, left, right\}$;

— $T(s, a, s')$ est similaire au problème *labyrinthe*, sauf que w devient faux quand un feu est atteint, et vrai quand une source d'eau est atteinte ;

— R_{MDP}, la fonction de récompense, retourne (i) une pénalité par défaut de -0.04 pour chaque action, (ii) -1 quand l'agent touche un mur, et (iii) $+1$ quand l'agent atteint un feu alors qu'il transporte de l'eau (w =vrai).

Les politiques MDP optimales consistent en des allers-retours incessant entre une source d'eau et un feu. La politique softmax π_{MDP} est obtenue en utilisant l'algorithme d'itération sur la valeur avec $\gamma = 0.99$ pour garantir la convergence.

Modèle pOAMDP : Pour les deux types de problèmes, des pOAMDP sont dérivés en utilisant les fonctions de récompense précédemment proposées pour la prédictabilité. Nous construisons 8 pOAMDP (un par fonction de récompense) pour chaque environnement grille. Puisque chaque pOAMDP peut être considéré comme un MDP, les pOAMDP sont résolus en utilisant à nouveau l'algorithme d'itération sur la valeur avec un facteur d'actualisation approprié (détails dans la section suivante). Pour une fonction de récompense R_X^Θ, la politique solution du pOAMDP est notée π_X^Θ.

4.2 Résultats

Les figures représentent les politiques softmax π_{MDP} (avec des flèches dont le niveau de gris dépend de la probalitité d'action) et les politiques pOAMDP π_X^Θ (avec des flèches noires qui indiquent les actions ϵ optimales). Du fait de l'intérêt limité des politiques π_{max}^Θ, π_{Pr}^Θ, et $\pi_{\text{regret}}^\Theta$, celles-ci ne sont montrées que dans le premier problème (labyrinthe).

4.2.1 Problème *labyrinthe* :

grilles utilisées : Les labyrinthes sont principalement constitués de couloirs et de pièces (vides). Pour la prédictibilité sur les actions, nous nous attendons à ce que les politiques pOAMDP préfèrent les couloirs aux pièces (qui permettent plus d'actions optimales possibles).

— Le labyrinthe de la fig. 3 consiste en 1 état terminal qui peut être atteint par 1 couloir ou 1 salle.

— Le labyrinthe de la fig. 4 consiste en 2 états terminaux qui peuvent être atteints respectivement par un couloir de 2 cellules de large et par un couloir d'1 cellule de large.

— Le labyrinthe de la fig. 5 consiste en 1 état terminal qui peut être atteint soit par un couloir suivi d'une salle, soit par une salle suivie d'un couloir. Le but de ce labyrinthe est d'observer l'influence de γ.

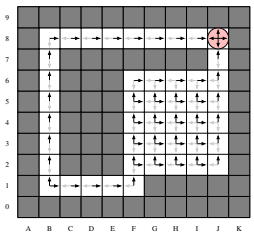
— Le labyrinthe de la fig. 6 consiste en 2 couloirs qui conduisent à un état terminal. Un de ces couloirs contient des cellules glissantes, mais le temps moyen de traversée est le même pour les deux. Le but de ce labyrinthe est d'observer les différences entre R_{cost}^A et R_{cost}^S.

Softmax MDP policy π_{MDP} Chaque SSP est résolu avec $\gamma = 1$ et $\epsilon = 0.001$. Une politique softmax π_{MDP} est ensuite obtenue en utilisant $\tau = 0.1$. Comme attendu, dans une salle, il y a de multiples actions optimales.

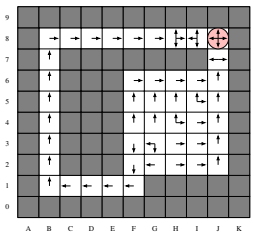
Note : Dans la suite, nous nous concentrons sur la prédictibilité des actions parce que les politiques solutions s'avèrent identiques pour le cas de la prédictibilité sur les états. Ce phénomène est favorisé par les environnements déterministes, où prédire le prochain état est souvent équivalent à prédire la prochaine action.

Analyse de π_{max}^A (et π_{max}^S) Les deux fonctions de récompenses π_{max}^A et π_{max}^S sont positives et sont à l'origine de comportements qui empêchent la convergence dans le cas ou $\gamma = 1$. On résout donc le pOAMDP avec $\gamma = 0.99$.

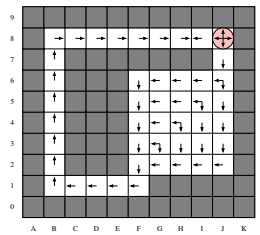
Comme attendu, la politique π_{max}^A tente d'atteindre des états dans lesquels l'observateur a une plus grande confiance, c.-à-d. que sa croyance est plus déterministe. L'agent peut choisir une action pour rester dans un état



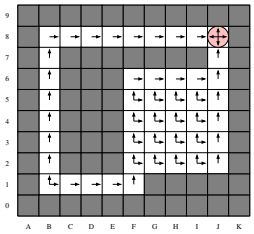
a) π_{MDP} ($\gamma = 1, \tau = 0.1$)



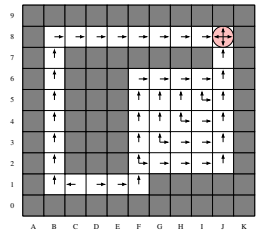
b) π^A_{max} ($\gamma = 0.99$)



c) π^A_{Pr} ($\gamma = 0.99$)

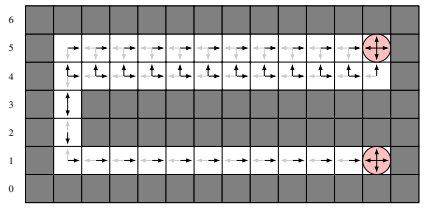


d) π^A_{regret} ($\gamma = 1$)

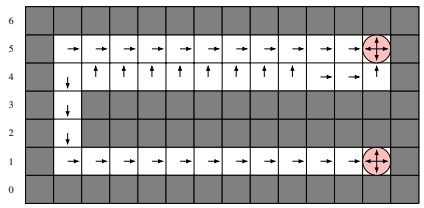


e) π^A_{cost} ($\gamma = 1$)

FIGURE 3 – Résultats de la prédictibilité sur les actions pour le premier labyrinthe, en commençant par la politique stochastique attendue par l'observateur (a) avec une température de $\tau = 0.1$, et en montrant ensuite toutes les actions optimales pour les 4 fonctios de récompenses considérées (b–e), avec $\gamma < 1$ quand le problème n'est pas un SSP valide.

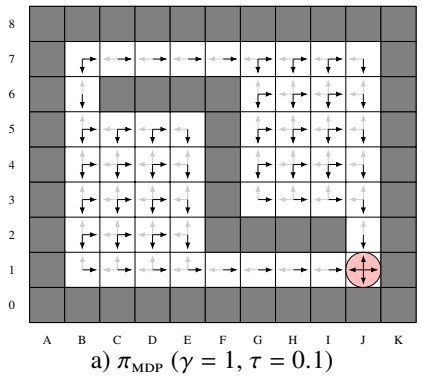


a) π_{MDP} ($\gamma = 1, \tau = 0.1$)

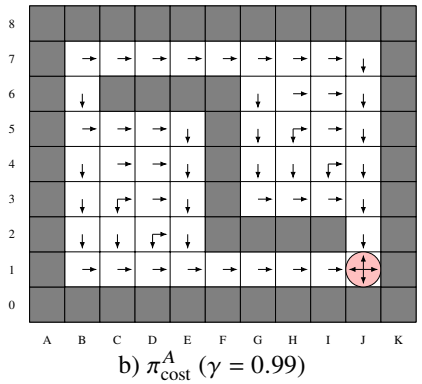


b) π^A_{cost} ($\gamma = 1$)

FIGURE 4 – Résultats pour le deuxième labyrinthe



a) π_{MDP} ($\gamma = 1, \tau = 0.1$)



b) π^A_{cost} ($\gamma = 0.99$)

FIGURE 5 – Résultats pour le troisième labyrinthe

plus déterministe même si cette action n'est pas prédictible (voir fig. 3.b, cellules $(H, 8)$, $(I, 8)$ et $(J, 7)$). Il évite aussi les états terminaux pour continuer d'accumuler des récompenses positives.

Analyse de π^A_{Pr} (et π^S_{Pr}) Pour des raisons identique au cas précédent, on utilise $\gamma = 0.99$ pour résoudre le pOAMDP.

Pour la prédictibilité sur les actions, donc avec R^A_{Pr}, les états terminaux ne sont toujours pas récompensés, ce qui, à nouveau, dissuade l'agent de l'atteindre. π^A_{Pr} diffère de π^A_{max} parce que la récompense encourage à prendre les actions

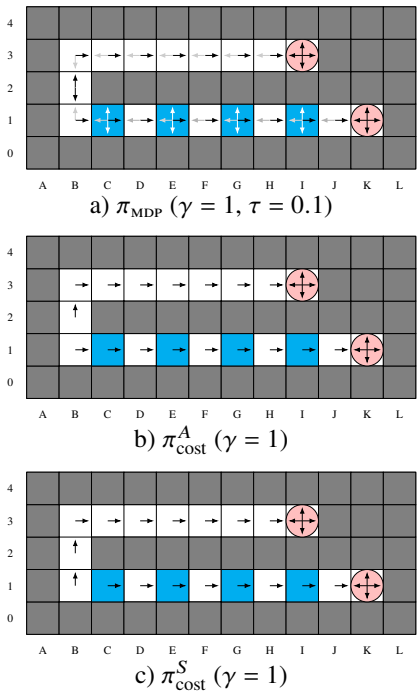


FIGURE 6 – Résultats pour le quatrième labyrinthe

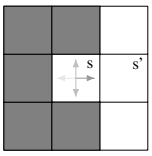


FIGURE 7 – Partie de labyrinthe dans laquelle π^S_{regret} peut différer de π^A_{regret} dans l'état s (et donc de la politique MDP optimale) car il est plus probable pour l'observateur (à cause de π_{MDP}) que le prochain état soit s au lieu de s'.

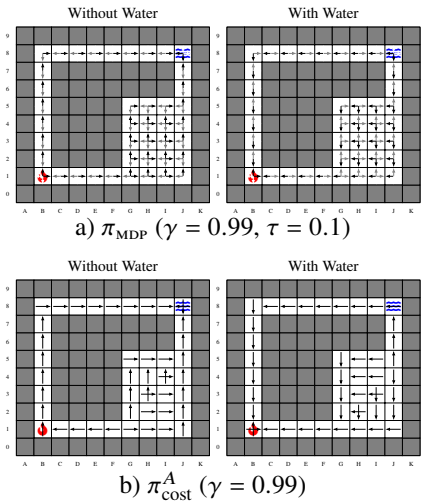


FIGURE 8 – Résultats pour le premier problème pompier

les plus prédites par l'observateur, comme on le voit sur la fig. 3.c, cellules $(H, 8)$, $(J, 8)$, $(J, 7)$, et $(J, 8)$.

Analyse de π^A_{regret} et π^S_{regret} Les fonctions de récompense "regret" induisent ici des SSP valides. Comme attendu (voir sec. 3.3), $R^A_{\text{regret}}(s, a, s')$ conduit aux solutions optimales du SSP de l'observateur, comme on le voit en comparant les actions les plus probables de la fig. 3.a et les actions (toutes optimales) de la fig. 3.d.

Comme précédemment, π^A_{regret} et π^S_{regret} s'avèrent identiques sur le premier problème. Mais regardons, sur la fig. 7, un motif de labyrinthe qui conduit à des comportements locaux dans π^S_{regret} différents de ceux de la politique optimale du MDP de l'observateur (et donc différents de π^A_{regret}). Ici, l'action optimale du MDP est de sortir de cette impasse s en allant vers la droite. Pourtant, en supposant une température assez élevée τ et des pénalités assez petites quand on touche le mur, la politique softmax pourrait être telle que $\pi_{\text{MDP}}(left|s) + \pi_{\text{MDP}}(up|s) + \pi_{\text{MDP}}(down|s) > \pi_{\text{MDP}}(right|s)$, de sorte que 1. le prochain état le plus probable est s plutôt que s', et 2. π^S_{regret} choisira n'importe quelle action autre que $right$.

Analyse de π^A_{cost} et π^S_{cost} Nous observons plusieurs comportements intéressants avec $R^A_{\text{cost}}(s, a, s')$:

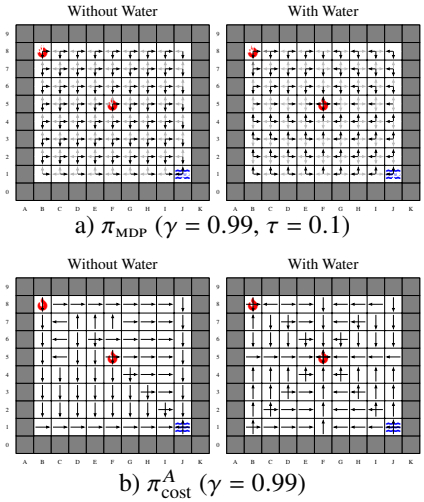
1. L'agent préfère un long chemin à travers un couloir étroit à un chemin plus court passant par une salle (fig. 3.e) ou un couloir large (fig. 4.b). Il y a en effet moins de choix d'actions dans les couloirs étroits, de sorte que les actions sont plus prédictibles.
2. Dans les salles, l'agent va souvent vers le mur le plus proche pour le suivre, comme sur la Fig. 3.e et la fig. 5.b.
3. Dans la fig. 5, l'agent peut choisir entre (i) un couloir conduisant à une pièce, et (ii) une pièce conduisant à un couloir. Quand $\gamma < 1$, l'agent préfère passer par le couloir d'abord parce que le facteur d'actualisation met plus d'importance sur les récompenses proches (voir cellule $(B, 7)$).
4. Dans la fig. 6, cellule $(B, 2)$, π^A_{cost} préfère monter plutôt que descendre. Cela est dû à la dynamique moins régulière le long du chemin du bas, lequel, à travers les Q-valeurs, conduit à de petites différences dans les prédictibilités des actions.

R^S_{cost} conduit à un résultat différent de R^A_{cost} dans le labyrinthe de la figure 6, parce que π^S_{cost} préfère monter dans la cellule $(B, 1)$, ce qui va à l'encontre des prédictions de l'observateur, pour suivre le chemin sans cellules glissantes.

4.2.2 Problème du *pompier*

Grilles utilisées Les grilles suivantes ont été utilisées pour tester les fonctions de récompense :

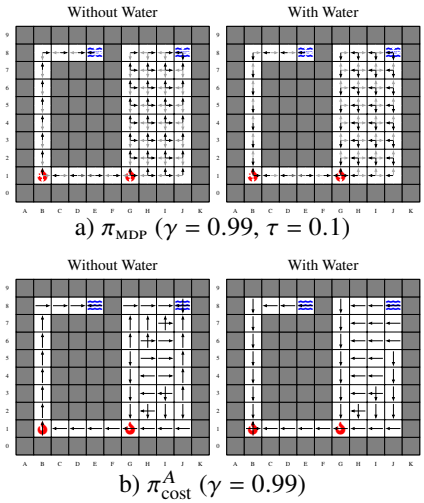
1. la grille de la fig. 8 contient 1 feu et 1 source d'eau reliés par une salle et un couloir ;



a) π_{MDP} ($\gamma = 0.99$, $\tau = 0.1$)

b) π^A_{cost} ($\gamma = 0.99$)

FIGURE 9 – Résultats pour le deuxième problème pompier



a) π_{MDP} ($\gamma = 0.99$, $\tau = 0.1$)

b) π^A_{cost} ($\gamma = 0.99$)

FIGURE 10 – Résultats pour le troisième problème pompier

2. la grille de la fig. 9 est une salle avec 2 feux et 2 sources d'eau ;
3. la grille de la fig. 10 contient 2 feux et 2 sources d'eau ; une partie de la carte est une pièce et l'autre un couloir ; le but de cette carte est d'observer si l'agent pOAMDP préfère apporter de l'eau au feu de la zone déterministe même si le feu situé sur la salle est plus proche.

Comparaison des politiques Les MDP sous-jacents ne sont plus des SSP, de sorte ue nous employons des pOAMDP $\gamma = 0.99$-actualisés. Comme pour le problème du labyrinthe : 1. R^A_{max} et R^S_{max} créeent des politiques qui préfère "boucler" qu'effectuer la tâche ; 2. R^A_{regret} et R^S_{regret} induisent la même politique que la politique optimale du MDP de l'observateur, et ne sont donc pas utiles ; et 3. dans la plupart des cas, les prédictibilités sur les actions et les états donnent des résultats similaires. Aussi, R^A_{Pr} et R^A_{cost} (ou R^S_{Pr} et R^S_{cost}) induisent les mêmes politiques solutions dans ce cadre actualisé. Pour toutes ces raisons, nous n'étudions que R^A_{cost}.

Analyse des politiques π^A_{cost} et π^S_{cost} Un comportement similaire au cas du problème labyrinthe peut être observé. Dans la fig. 8, π^A_{cost} préfère le couloir à la salle vide. Dans de telles salles, l'agent pOAMDP cherche à atteindre un mur pour le longer (figs. 8 and 10). Dans la fig. 9, l'agent pOAMDP tente d'être plus prédictible en marchant le long des murs ou en atteignant la ligne 5 ou la colonne F pour réduire le nombre de chemins optimaux pour atteindre le feu au milieu. Dans la fig. 10, l'agent pOAMDP préfère le feu situé en $(B, 1)$ et la source d'eau située en $(E, 8)$ même si un autre feu ou une autre source d'eau est plus proche. C'est particulièrement visible sur le côté "sans eau" de la figure, où π^A_{cost} va de $(G, 5)$ à $(E, 8)$ pour se remplir.

5 Conclusion et perspectives

Nous avons introduit un nouveau formalisme, celui des OAMDP prédictibles (pOAMDP), lequel permet de dériver des politiques dans lesquelles la prochaine action ou le prochain état est plus prédictible, et proposé de prendre en compte non seulement les problèmes actualisés, mais aussi les chemins stochastiques les plus courts (ce qui requiert de s'assurer que des politiques solutions valides peuvent être trouvées). Différentes fonctions de récompense ont été considérées et analysées à travers leurs propriétés théoriques et des illustrations des politiques résultantes sur deux mondes grilles. La fonction de récompense "coût" $R^\Theta_{\text{cost}}(s, a, s') \stackrel{\text{def}}{=} b_t(\tau(s, a, s')) - 1$ ($\Theta \in \{A, S\}$) s'avère être le meilleur choix, puisqu'elle est valide à la fois pour les problèmes actualisés et orientés-but, et rend effectivement les actions ou états plus prédictibles (mais des alternatives sont possibles). Dans certains cas, des actions contre-intuitives sont sélectionnées pour augmenter la prédictibilité ultérieure. Un propriété remarquable est que le complexité de résolution des pOAMDP est comparable à celle des MDP, et bien moindre que celle des OAMDP.

Une première perspective serait de conduire des expérimentations avec de vrais observateurs humains pour voir si les politiques pOAMDP sont effectivement perçues comme plus prédictibles, et de raffiner les exigences pour un agent prédictible. Par exemple, on pourrait s'attendre à ce que les humains arrêtent de faire confiance à l'agent si son comportement est temporairement non-prédictible.

Par ailleurs, pour revenir au travail précurseur de MIURA et ZILBERSTEIN [3], nous souhaiterions étendre la discussion des problèmes orientés-buts aux OAMDP, par exemple pour déterminer lesquels de leurs scénarios conduisent à des SSP valides. Aussi, nous avons dû nous éloigner de leur formalisme original et de leurs types statiques, mais une perspec-

tive importante est de généraliser les deux formalismes pour obtenir une théorie plus unifiée de la prise de décision séquentielle consciente d'un observateur Nous pensons que, pour se faire, un point clef est de restreindre l'observabilité des états et actions par l'observateur, de sorte que le type, qu'il soit statique ou dynamique, puisse être une variable d'état (même pour la prédictibilité sur les actions). En outre, cette observabilité partielle permettre aussi de couvrir plus de scénarios du monde réel. Dans ce cadre, nous envisageons d'étudier les propriétés de continuité de la fonction de valeur optimale pour éventuellement proposer des approximateurs minorant et majorant, et dériver des solveurs à base de points (comme cela a été fait pour les POMDP et des modèles apparentés [7]-[14]).

Références

[1] T. Chakraborti, A. Kulkarni, S. Sreedharan, D. E. Smith et S. Kambhampati, "Explicability ? Legibility ? Predictability ? Transparency ? Privacy ? Security ? The Emerging Landscape of Interpretable Agent Behavior," in *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling (ICAPS)*, Berkeley, CA, USA : AAAI Press, 2019. adresse : https://ojs.aaai.org/index.php/ICAPS/article/view/3463.

[2] J. F. Fisac, C. Liu, J. B. Hamrick et al., "Generating plans that predict themselves," in *Algorithmic Foundations of Robotics XII : Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, 2020.

[3] S. Miura et S. Zilberstein, "A unifying framework for observer-aware planning and its complexity," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, C. de Campos et M. H. Maathuis, éd., sér. Proceedings of Machine Learning Research, t. 161, PMLR, juill. 2021, p. 610-620. adresse : https://proceedings.mlr.press/v161/miura21a.html.

[4] A. D. Dragan, K. C. T. Lee et S. S. Srinivasa, "Legibility and predictability of robot motion," 2013, p. 301-308.

[5] C. L. Baker, R. Saxe et J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, t. 113, n° 3, p. 329-349, déc. 2009. doi : 10.1016/j.cognition.2009.07.005.

[6] A. Kolobov, Mausam, D. S. Weld et H. Geffner, "Heuristic Search for Generalized Stochastic Shortest Path MDPs," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS'11)*, 2011.

[7] T. Smith et R. G. Simmons, "Point-Based POMDP Algorithms : Improved Analysis and Implementation," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 2005, p. 542-549.

[8] M. T. Spaan et N. Vlassis, "Perseus : Randomized Point-based Value Iteration for POMDPs," *Journal of Artificial Intelligence Research*, t. 24, p. 195-220, 2005. adresse : http://www.aaai.org/Papers/JAIR/Vol24/JAIR-2406.pdf.

[9] H. Kurniawati, D. Hsu et W. S. Lee, "SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics : Science and Systems IV*, 2008.

[10] J. Pineau, G. Gordon et S. Thrun, "Anytime point-based approximations for large POMDPs," *Journal of Artificial Intelligence Research*, t. 27, p. 335-380, 2006.

[11] G. Shani, J. Pineau et R. Kaplow, "A survey of point-based POMDP solvers," *Journal of Autonomous Agents and Multi-Agent Systems*, t. 27, n° 1, 2013. doi : 10.1007/s10458-012-9200-2. adresse : http://dx.doi.org/10.1007/s10458-012-9200-2.

[12] J. Dibangoye, C. Amato, O. Buffet et F. Charpillet, "Optimally Solving Dec-POMDPs as Continuous-State MDPs," *Journal of Artificial Intelligence Research*, t. 55, p. 443-497, 2016. adresse : http://www.jair.org/papers/paper4623.html.

[13] K. Horák, B. Bošanský et M. Pěchouček, "Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, p. 558-564.

[14] K. Horák et B. Bošanský, "Solving Partially Observable Stochastic Games with Public Observations," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, p. 2029-2036. doi : 10.1609/aaai.v33i01.33012029.

Actes JIAF-JFPDA 2023

Opponent-model search in games with incomplete information*

Junkang Li[1,2] **Bruno Zanuttini**[2] **Véronique Ventos**[1]

[1]NukkAI, Paris, France

[2]Normandie Univ.; UNICAEN, ENSICAEN, CNRS, GREYC, 14 000 Caen, France

junkang.li@nukk.ai bruno.zanuttini@unicaen.fr vventos@nukk.ai

Résumé

Les jeux à information incomplète sont des jeux qui modélisent des situations dans lesquelles les joueurs n'ont pas connaissance commune du jeu auquel ils jouent, comme par exemple dans des jeux de cartes tels que le poker ou le bridge. Des modèles de l'adversaire peuvent revêtir une importance cruciale pour la prise de décision dans de tels jeux. Nous proposons des algorithmes pour calculer des stratégies optimales et/ou robustes dans les jeux à information incomplète, étant donné différents types de connaissances à propos des modèles de l'adversaire. En guise d'application, nous décrivons un cadre pour raisonner sur le raisonnement de l'adversaire dans de tels jeux, cadre dans lequel des modèles de l'adversaire apparaissent naturellement.

Abstract

Games with incomplete information are games that model situations where players do not have common knowledge about the game they play, e.g. card games such as poker or bridge. Opponent models can be of crucial importance for decision-making in such games. We propose algorithms for computing optimal and/or robust strategies in games with incomplete information, given various types of knowledge about opponent models. As an application, we describe a framework for reasoning about an opponent's reasoning in such games, where opponent models arise naturally.

1 Introduction

Most algorithmic studies in game theory focus on the computation of exact or approximate Nash equilibria, which leaves much to be desired for many reasons. Firstly, large games usually have more than one equilibrium. Which equilibria should be considered rational or reasonable has long been a subject of study: refinements of Nash equilibria [5], epistemic game theory [20], etc. More importantly, Nash equilibrium as a solution concept has two implicit assumptions: both players have unlimited computational power for computing Nash equilibria, and each knows which equilibrium the other will choose. Needless to say, both are difficult to justify in real-life situations.

These two assumptions are also opponent models in disguise, which are models that describe or predict how an opponent reasons in a game. In this paper, we are interested in more general opponent models than those behind Nash equilibria. Such opponent models have been explicitly incorporated into game tree search algorithms (e.g. minimax, $\alpha\beta$ search, MCTS) for games with perfect information, for instance by [10, 11]. The knowledge of opponent models can result in both acceleration of game tree search (e.g. by pruning branches not considered by the opponents) and increase of the performance of strategies computed (e.g. by exploiting the weakness of the opponents).

In this paper, we apply the same idea to games with incomplete information, where opponent models yield even more interesting results than in games with perfect information. We propose different ways of taking opponent models into account, and give algorithms for computing the corresponding robust and optimal responses. We further propose a principled method to take into account the probability that the opponent does not behave according to any of the given models. Finally, we show an application of these models to the recursive modelling of opponents, where a level-k player assumes that their opponent reasons at some level lower than k, and recursively down to level 0.

2 Related work

Equilibrium concepts in games with perfect or imperfect information have long been studied; in particular, they have been related to models of knowledge and beliefs (for each player about the others' reasoning and strategies) via the concept of rationalisability in the field of epistemic game

*A long version with proofs of the claims is available at https://hal.science/hal-04100646.

theory. For a thorough treatment, the reader may refer for instance to the textbooks by [20] or [2].

When no opponent model is available, one typically considers all possible (pure or mixed) strategies. In this case, [13] and [14, 22] study the complexity of computing maxmin strategies under a variety of settings; in particular, for mixed strategies, they give polynomial-time algorithms based on linear programming for two-player extensive-form games with perfect recall, a more general setting than ours. [18] propose a double-oracle algorithm for computing optimal mixed strategies for Markov decision processes with adversarial cost functions, which can also be regarded as a polynomial-time algorithm for computing the maxmin strategies of a normal-form game. [3] propose an algorithm that combines the ideas of linear programming and double-oracle for zero-sum extensive-form games with perfect recall, and experimentally demonstrate that it is more efficient than other algorithms when optimal mixed strategies have small supports.

Opponent models can come in diverse forms. [10, 11] propose opponent models for games with perfect information, where models are given by the evaluation function and the search depth of the opponent. A recent survey of opponent modelling approaches is provided by [1]. Our work is related to these in the sense that we assume opponent models to be given (called "type-based reasoning" by [1]). However, an important stream of work also studies the *learning* of opponent models; we refer the reader to the survey by [19].

Among opponent models, an important class is that of *recursive* models, where MAX searches a strategy (at level k) assuming that MIN themselves searches a strategy (at level $k-1$) assuming that MAX searches..., etc, down to level 0. Such models have been essentially studied to capture human reasoning in games. [4] propose a *cognitive hierarchy model*, where an opponent's level is modelled by a Poisson distribution on levels $k-1, \dots, 0$, and validate this model against empirical data. [24] assess the relevance of various modelling assumptions for level 0. [23] assess the efficiency of reasoning with recursive models by simulation. Such recursive models are also used in epistemic game theory to define notions such as common belief in rationality [20].

Finally, a line of work closely related to ours is the study of interactive POMDPs for collaborative decision-making in partially observed environments [9, 6]. In this model, a level-k agent optimizes their behaviour given a distribution over (partially observed) physical states and over other agents' models at level $k-1$. An interesting feature of this model is that optimal behaviours at level k can be computed iteratively as a sequence of optimal policies for POMDPs, where at each iteration the other agents' model can be considered as part of the environment.

3 Background

In this paper, we focus on games in extensive form, i.e. represented by a tree. We also focus on zero-sum games with two players (MAX and MIN), but our study can be easily extended to more players and general-sum.[1] We briefly describe our setting and refer the reader to textbooks [17] for details.

In an extensive-form game with no chance, each internal node n of the game tree is owned by a player. To each terminal node, an *outcome* (or *value*) is attached, typically a real number, which denotes the payoff for MAX (and MIN's payoff is the opposite).

We denote MAX and MIN by $+$ and $-$, respectively. Under imperfect information, an *information set* for a player $i \in \{+, -\}$ is a set of their nodes that they cannot distinguish. A *pure strategy* for i, denoted by s_i, maps each information set IS of i to an action available at IS; in particular, the same action must be chosen at all nodes in the same IS. A *mixed strategy* for i, denoted by σ_i, is a probability distribution over the set of all pure strategies of i, with the interpretation that i plays a pure strategy randomly chosen according to this distribution at the beginning of a game.

We write Σ_i^{P} (resp. Σ_i^{M}) for the set of all pure (resp. mixed) strategies of player i in a game. We also write $p_1 s_i^1 + \cdots + p_k s_i^k$ for a mixed strategy of i with support $\{s_i^1, \dots, s_i^k\}$ and probabilities $p_1, \dots, p_k$; in particular, a pure strategy can be regarded as a mixed strategy with singleton support. In a game with no chance, a *profile* of pure strategies $(s_+, s_-) \in \Sigma_+^{\mathrm{P}} \times \Sigma_-^{\mathrm{P}}$, uniquely determines a terminal node to be reached. The *payoff* (for MAX) under this profile, written as $u(s_+, s_-)$, is defined to be the value of this terminal node. The *expected payoff* (for MAX) under a profile of mixed strategies $(\sigma_+, \sigma_-) \in \Sigma_+^{\mathrm{M}} \times \Sigma_-^{\mathrm{M}}$ is the expectation of MAX's payoff over drawings of pure strategies.

In general, games include *chance* nodes, which can be seen as being owned by a player called Nature, who uses a behaviour strategy that is common knowledge.

Games with incomplete information

In this paper, we study games with incomplete information, where players do not have common knowledge about the game they play. For example, a player can be uncertain about the payoff or available actions of other players, or whether other players are themselves uncertain about the game, etc. Notable examples of such games are poker, bridge, and mahjong, where the initial distribution of cards/tiles is not common knowledge.

A game with incomplete information can be modelled as a game with imperfect information via the Harsanyi model,

[1] With the exception of the lexicographic setting, for which the definition of the problem does not trivially generalise.

which uses the notion of *types* to define the knowledge of a player. For example, in a game of poker or bridge, the type of players is their hand. More concretely, at the beginning of a game, there is a chance node that selects a type for each player according to a common prior. Every player learns their own type but not the types of the other players. Then all players participate in a game where the form of the game tree and the outcomes can depend on each player's type, but the actions of every player only depend on their type.[2]

Best-defence model

We are interested in decision-making in games with incomplete information, where all actions except the selection of each player's type by the chance node at the root are public. We also assume that there is no other chance node. In other words, we are interested in two-player zero-sum games with incomplete information where had the types been common knowledge, the game would be of perfect information without chance node; we call them combinatorial games with incomplete information (CGII). However, the results in this paper can be extended to any game with incomplete information with minor modifications.

Given a CGII, our goal is to find maxmin-like strategies of MAX. Such strategies are usually computed by backward induction (most typically minimax-like depth-first search) in games with perfect information. However, for a game with incomplete information, traditional backward induction is impossible since there is no non-trivial subgame: each proper subtree is connected to another one via at least one information set. Instead, an approximation of maxmin strategies can be found using the *best-defence model* [7], which, by assuming MIN knows MAX's type, simplifies a game where both players have incomplete information into a game where MIN has perfect information.

Throughout the paper, we study CGIIs under the best-defence model. In general, available actions of MIN depend on their type, so not every terminal node is reachable by a given type of MIN. However, one can assume that the payoff for MAX at such an unreachable terminal node is $+\infty$, which does not change the maxmin value of the game [7]. Therefore, we assume without loss of generality that MIN's set of actions is independent of their type, and all terminal nodes are reachable by any type of MIN.

Formally, a CGII under the best-defence model is specified by a game tree (the nodes of which are partitioned into terminal nodes, MIN's decision nodes, and MAX's decision nodes), an integer $t \geq 1$ (which denotes the number of MIN's types), a common prior $\vec{q}$ over MIN's types, and a payoff function $u : L \to \mathbb{R}^t$ that to each terminal node $n \in L$ of the game tree, assigns a vector of length t written as $\vec{u}(n) = (u(n)_1, \ldots, u(n)_t)$,[3] where $u(n)_i$ is the payoff for MAX at node n if MIN is of type i. Notice that a game with perfect information and no chance is a CGII under the best-defence model with $t = 1$, i.e. only one type of MIN.

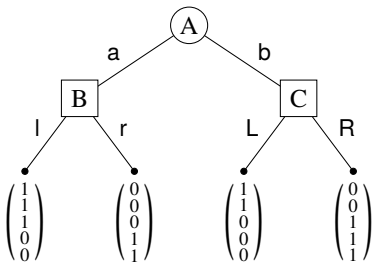


Figure 1: A CGII with 5 possible worlds.

Example. *A CGII under the best-defence model with* 5 *types of MIN is given in Figure 1, where we use squares and circles to denote MAX's and MIN's nodes, respectively. Unless stated otherwise, we assume in all our examples that the common prior over MIN's types is uniform (hence each type occurs with probability* $1/5$ *in this CGII). Here is an example of playout: if MIN plays* a *at A and MAX plays* l *at B, MAX's payoff vector will be* $(1, 1, 1, 0, 0)$*, which means MAX's gain is* 1 *if MIN is of one of the first three types, and* 0 *otherwise.*

4 Maxmin values without opponent models

In this section, we give an overview of algorithms from the literature for computing maxmin values without opponent models, which will be the basis of our algorithms for opponent-model search.

We are interested in computing the *maxmin value*

$$\underline{v}_+ := \max_{\varsigma_+ \in \Sigma_+} \min_{s_- \in \Sigma_-^{\mathrm{P}}} u(\varsigma_+, s_-), \tag{1}$$

where Σ_+ is Σ_+^{P} or Σ_+^{M}, depending on context. We recall that u denotes the expected payoff (for MAX) with respect to type distribution and mixed strategies. Since u is linear in MIN's mixed strategies, replacing Σ_-^{P} by Σ_-^{M} in (1) would not change the value defined, hence we define the maxmin value to be against all pure strategies of MIN.

The maxmin value $\underline{v}_+$ is the largest payoff MAX can guarantee by any strategy from Σ_+, no matter how MIN plays. MAX's strategies achieving this value are called *maxmin strategies*. Depending on whether we allow MAX to use mixed strategies, two notions of maxmin arise: pure maxmin and mixed maxmin. By definition, it is clear that the mixed maxmin value is no smaller than the pure maxmin

[2] An equivalent model called the *Aumann model* uses Kripke structures where an equivalence class for player i corresponds to a type of i in the Harsanyi model. For more details, we refer the reader to the textbook by [17].

[3] In the following, we use the notation v_i for the i-th component of a vector $\vec{v}$.

value. As we will see, it is in general more difficult to compute the pure than the mixed maxmin value for a CGII. Still, in some situations, pure maxmin is more desirable or even the only viable solution concept, e.g. when outcomes are only partially ordered, or when mixed strategies are not allowed due to their probabilistic nature. Hence, we will study algorithms for both notions, with a focus on pure maxmin since algorithms for mixed maxmin only require minor modifications in the presence of opponent models.

A generic minimax algorithm

We will focus on algorithms for computing the maxmin *value*, but they can be easily modified to compute the corresponding maxmin *strategies*. The maxmin value of games with perfect information is typically computed by the *minimax* algorithm, a generic version of which is shown in Algorithm 1.

Algorithm 1: Generic minimax algorithm

```
def MiniMax(node n):
    if n is a terminal node:
        return eval(n)
    else:
        find the set of n's children C(n)
        if n is MAX's decision node:
            return ⋁_{n'∈C(n)} MiniMax(n')
        else:
            return ⋀_{n'∈C(n)} MiniMax(n')
```

This depth-first search algorithm has four parameters, which we will use to capture different algorithms in the following sections:

- V is a set of objects called *situational values*;
- eval is an evaluation function which maps each terminal node n to a value $\mathsf{eval}(n) \in V$;
- $\vee, \wedge : V \times V \to V$ are two associative binary operators, referred to as MAX's and MIN's operator, respectively.

With eval as boundary conditions, this algorithm recursively defines a situational value $\mathsf{val}(n)$ for every node n. For an instantiation of this algorithm to compute the maxmin values, one should choose the parameters as a function of the class of games under consideration, in such a way that there is a polynomial-time computable mapping from the situational value of the root $\mathsf{val}(r)$ to the maxmin value of the game.

For example, for games with perfect information, it is well-known that Algorithm 1 run on the root yields the pure/mixed maxmin value (1) with $V := \mathbb{R}$, $\mathsf{eval}(n) := u(n)$, $\vee := \max$, and $\wedge := \min$.

This algorithm has several advantages: returned values for internal nodes are readily interpretable; the algorithm is extremely efficient on memory since the recursion depth is the depth of the game tree, which in general is exponentially smaller than the tree; the search can be combined with other techniques, such as heuristic functions and $\alpha\beta$ pruning (which is possible whenever $(V, \vee, \wedge)$ forms a lattice [16]), move ordering, Monte Carlo techniques such as MCTS, etc.

In the following, we will present various algorithms for computing pure and mixed maxmin values, with or without opponent models. Whenever possible, we will describe them succinctly as a particular instantiation of Algorithm 1.

Pure maxmin

[7] show that the pure maxmin value is NP-hard to compute for CGIIs. The first exact algorithm was proposed by [8], and it can be reframed as follows.

Proposition 1. *For a CGII with root r, t types of MIN, and common prior $\vec{q}$ over them, consider the instantiation of Algorithm 1 where: situational values are finite sets of vectors in $\mathbb{R}^t$; for all terminal nodes n, $\mathsf{eval}(n) := \{\vec{u}(n)\}$; MAX's operator is set union $\cup$; MIN's operator is $\sqcap$, defined for all situational values f and g by:*

$$f \sqcap g := \left\{ \left(\min(v_i, v_i') \right)_{1 \le i \le t} \mid \vec{v} \in f, \vec{v}\,' \in g \right\}.$$

Then it holds that

$$\underline{v}_+ := \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \min_{s_- \in \Sigma_-^{\mathrm{P}}} u(s_+, s_-) = \max_{\vec{v} \in \mathsf{val}(r)} \vec{q} \cdot \vec{v}.$$

Example. *For the CGII in Figure 1, we get*

$$\begin{aligned} \mathsf{val}(B) &= \{(1,1,1,0,0), (0,0,0,1,1)\}; \\ \mathsf{val}(C) &= \{(1,1,0,0,0), (0,0,1,1,1)\}; \\ \mathsf{val}(A) &= \{(1,1,0,0,0), (0,0,1,0,0), \\ &\qquad (0,0,0,0,0), (0,0,0,1,1)\}. \end{aligned}$$

This algorithm actually recursively enumerates all strategies of MAX: each vector in $\mathsf{val}(n)$ implicitly represents one or several strategies of MAX in the subtree rooted at n. At the root A, given the uniform prior over MIN's types, the best vectors are $(1,1,0,0,0)$ (corresponding to MAX's strategy (l, L), by which MAX chooses l at B and L at C) and $(0,0,0,1,1)$ (MAX's strategy (r, R)); both achieve the pure maxmin value $(\frac{1}{5}, \ldots, \frac{1}{5}) \cdot (1,1,0,0,0) = (\frac{1}{5}, \ldots, \frac{1}{5}) \cdot (0,0,0,1,1) = 2/5$.

Importantly, in a CGII, the expected payoff of the strategies of the subtree rooted at n may depend on that of strategies of a subtree rooted at another node n', which can be far away from n. In our example, l and R are locally optimal with respect to the uniform prior. However, (l, R) is not optimal at the root, since it is MIN who chooses, with

perfect information, either a or b as a function of their type. In other words, it is not correct to use the common prior to evaluate strategies locally at nodes B and C: the conditional probabilities of MIN's types at both B and C depend on MIN's strategy and can be different from the prior.[4]

Reduction of situational values Even with non-locality, situational values, which are sets of vectors, can be reduced to accelerate the computation. If in $\mathsf{val}(n)$ a vector $\vec{v}$ is weakly dominated by another vector $\vec{v}\,'$, then we can discard $\vec{v}$ from $\mathsf{val}(n)$. This reduction corresponds to the elimination of weakly dominated strategies. For example, if A is an internal node of a larger CGII, then $(0,0,0,0,0)$ (corresponding to MAX's strategy (r, L)) can be discarded from $\mathsf{val}(A)$ without effect on the pure maxmin value of the larger game: MAX never does worse by playing, say, the strategy represented by $(1,1,0,0,0)$ in the subtree rooted at A.

In general, any reduction other than the elimination of dominated vectors is unsound, i.e. would yield incorrect results for at least one game. However, we will see that more reductions become sound if opponent models are available.

Mixed maxmin

The mixed maxmin value, defined by

$$\underline{v}_+ \coloneqq \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \min_{s_- \in \Sigma_-^{\mathrm{P}}} u(\sigma_+, s_-), \tag{2}$$

can be computed in polynomial time with the linear programming (LP) algorithm proposed by [13]. This LP algorithm relies on two insights:

- The set of all mixed strategies of MAX can be represented by a system L of linear equalities, with linearly many (in the size of the game tree) variables and equalities.
- For any threshold v and any mixed strategy σ_+ of MAX represented as a solution to L, it can be verified in linear time whether $\min_{s_- \in \Sigma_-^{\mathrm{P}}} u(\sigma_+, s_-) \geq v$ holds by computing MIN's best responses to σ_+. This computation serves as the separation oracle in the LP.

Then the LP consists of maximising the variable v (which will yield the mixed maxmin value in (2)) under the constraints in L and the separation oracle. For more details, we refer the reader to [13].

Example. *In the game in Figure 1, the optimal mixed strategy is the uniform strategy, i.e. a uniform distribution over all* 4 *pure strategies of MAX. This strategy yields an expected payoff of at least* $1/2$*, which is the mixed maxmin value and is better than the pure maxmin value* $2/5$.

[4]This phenomenon, called *non-locality* by [7], is the culprit behind the NP-hardness of pure maxmin.

The above algorithm has been improved by [22, 14]. However, for simplicity, we only show modifications of the initial algorithm for taking opponent models into account. Adapting them to the improved algorithms is straightforward.

5 Opponent-model search

We now come to our main contributions, which are algorithms for finding maxmin strategies when given opponent models (OM). We will be interested in the maxmin value against a restricted set of opponent's strategies:

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \min_{\omega_- \in \Sigma_-^{\mathrm{O}}} u(\varsigma_+, \omega_-),$$

where Σ_+ is the set of all pure or all mixed strategies for MAX, Σ_-^{O} is the set of strategies of MIN considered to be possible by the OMs, and ω_- is an arbitrary strategy from Σ_-^{O}.

In general, OMs are models of the opponent's reasoning, which can come in various forms (see section 2). As a quite general setting, we consider that an OM describes a behaviour strategy of MIN. A *behaviour strategy* for a player i maps each information set IS of i to a probability distribution over i's actions at IS. All mixed strategies can be expressed as behaviour strategies in games with perfect recall[5], and *a fortiori* in CGIIs since CGIIs are games with perfect recall. For a strategy represented by a mixed strategy or another linear representation (like sequence form [13] or evaluation function [10]), its equivalent behaviour strategy can also be computed in time linear in the size of the game tree.

Algorithmically, we assume that each OM is specified by an oracle O such that, for any decision node n of MIN and any type i of MIN, $O(n, i)$ is the strategy at n of MIN of type i, specified as a probability distribution over MIN's actions available at n. We also assume that the OMs are given in the input and each call to the oracles takes constant time.

In this section, we consider situations where MAX is certain that MIN only considers strategies described by these OMs. This assumption will be relaxed in section 6.

Single OM

We first present the simplest case, with only one OM ω_-, which means that MAX has complete knowledge of MIN's strategy. Then the game becomes a single-player game with perfect information [13], and the pure/mixed maxmin value

[5]*Perfect recall* means players never forget what they knew or did in the past. For the formal definition of perfect recall and the equivalence between mixed and behaviour strategies in games with perfect recall, see [15].

becomes

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} u(s_+, \omega_-) = \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} u(\sigma_+, \omega_-),$$

where the last equality is due to the linearity of u. This value can be computed by a bottom-up (i.e. depth-first) procedure, which recursively computes MAX's best strategies at each of their information set.

More precisely, since MIN's strategy is known perfectly, all MIN's decision nodes become chance nodes. As a consequence, even though MAX still does not know MIN's type, they can compute the exact probability of reaching a node under each of MIN's types and, using Bayesian updates, deduce the conditional probability of MIN's types at every node. Then MAX can choose the actions that maximise the payoff with respect to this conditional probability at any MAX's decision node.

Example. *Consider again the game on Figure 1, with ω_- defined as follows: MIN plays a if of type* 1 *or* 2*, b if of type* 4 *or* 5*, and $\frac{1}{2}a + \frac{1}{2}b$ if of type* 3*. Against ω_- and the uniform prior over MIN's types, MAX can compute the vector $(\frac{1}{5}, \frac{1}{5}, \frac{1}{10}, 0, 0)$ at node B, which we call the* non-normalised belief state *(NBS) at B. For instance, the first component means that the probability of the combined event that MIN is of type* 1 *and B is reached is* $1/5$*. Observe that normalising the NBS would give the posterior probability over MIN's types (for instance,* $2/5$ *for type* 1*, and* 0 *for type* 5*). Therefore, by maintaining an NBS, MAX implicitly performs Bayesian inference on MIN's types using ω_-.*

Given the NBS at B, action l *yields a higher (non-normalised) payoff of* $1/2$ *than* r *(with a payoff of* 0*) at B. Similarly, at C the NBS is $(0, 0, \frac{1}{10}, \frac{1}{5}, \frac{1}{5})$ and prescribes action* R *(with a payoff of* $1/2$*). At node A, MAX's payoff can be simply computed as the sum of their payoff at B and C, which yields* 1*. One can check that* 1 *is indeed the best MAX can get when playing against MIN with this particular OM, and this payoff is obtained by the strategy* (l, R)*, which gives MAX a payoff of* 1 *independent of MIN's actual type.*

In general, every MAX's node n is the result of a series of MAX's actions and MIN's actions. MAX's NBS at n, written as $\overrightarrow{\mathsf{nbs}}(n)$, is computed component-wise: the i-th component is computed as the product of the probability of MIN being of type i and the probability that MIN of type i takes those actions leading to n at each of MIN's nodes that are an ancestor of n. In particular, the NBS at the root is the common prior over MIN's types. With the NBS of terminal nodes thus computed, we can then compute the best payoff for MAX.

Proposition 2. *For a CGII with root r and a single opponent model ω_-, consider the instantiation of Algorithm 1 where: $V \coloneqq \mathbb{R}$; for all terminal nodes n, $\mathsf{eval}(n) \coloneqq \overrightarrow{\mathsf{nbs}}(n) \cdot \vec{u}(n)$; MAX's operator is* max*; MIN's operator is* +*. Then it holds that $\underline{v}_+ = \mathsf{val}(r)$, and the algorithm is polynomial-time.*

This algorithm can be seen as a generalisation of the OM search proposed by [10], which only consider games with perfect information for which OMs are described by MIN's evaluation functions.

Probabilistic OMs

We now consider the case when MAX has several OMs $\omega_-^1, \ldots, \omega_-^m$ of MIN, and a probability distribution $\vec{p} = (p_1, \ldots, p_m)$ over them: MIN plays the strategy ω_-^1 with probability p_1, ω_-^2 with probability p_2, etc. In particular, the pure/mixed maxmin value is given by

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \sum_{j=1}^{m} p_j u(s_+, \omega_-^j) = \max_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \sum_{j=1}^{m} p_j u(\sigma_+, \omega_-^j),$$

This setting is not much different from the previous one, due to the linearity of u: these OMs can be merged into one single OM describing the mixed strategy $\omega_- \coloneqq p_1\omega_-^1 + \cdots + p_m\omega_-^m$.[6] In principle, one can traverse the game tree once and compute the behaviour strategy corresponding to ω_-, then run the single-OM algorithm from Proposition 2. Instead, we present a one-pass algorithm for probabilistic OMs, without the need to explicitly compute and store the strategy ω_-. The key is to maintain not just one, but m NBSs, one $\overrightarrow{\mathsf{nbs}}_j$ for each OM ω_-^j.

Proposition 3. *For a CGII with root r and opponent models $\omega_-^1, \ldots, \omega_-^m$ distributed according to $p_1, \ldots, p_m$, consider the instantiation of Algorithm 1 where: $V \coloneqq \mathbb{R}$; for all terminal nodes n, $\mathsf{eval}(n) \coloneqq \sum_{j=1}^{m} p_j \big(\overrightarrow{\mathsf{nbs}}_j(n) \cdot \vec{u}(n)\big)$; MAX's operator is* max*; MIN's operator is* +*. Then it holds that $\underline{v}_+ = \mathsf{val}(r)$, and the algorithm is polynomial-time.*

Lexicographic OMs

An important subcase of search with multiple OMs is the case when MAX holds a lexicographic belief over MIN's OMs $\omega_-^1, \ldots, \omega_-^m$. For example, MAX deems that MIN most probably follows ω_-^1. Otherwise, with an infinitesimally smaller probability (compared to ω_-^1), MIN follows ω_-^2. Otherwise, with an infinitesimally smaller probability (compared to ω_-^2), MIN follows ω_-^3, etc. We define the pure/mixed maxmin value in this case to be the vector of length m

$$\begin{aligned}\mathbb{R}^m \ni \overrightarrow{\underline{v}_+} &\coloneqq \operatorname*{lexmax}_{s_+ \in \Sigma_+^{\mathrm{P}}} \big(u(s_+, \omega_-^1), \ldots, u(s_+, \omega_-^m)\big) \\ &= \operatorname*{lexmax}_{\sigma_+ \in \Sigma_+^{\mathrm{M}}} \big(u(\sigma_+, \omega_-^1), \ldots, u(\sigma_+, \omega_-^m)\big),\end{aligned}$$

where lexmax is lexicographic maximum over vectors of length m. In other words, if there is a unique optimal

[6] We abuse notation by writing ω_-^i both for the given behaviour strategy and for the equivalent mixed strategy.

strategy against ω_-^1, then this strategy is chosen; otherwise, ties are broken according to their values against ω_-^2, and so on.

This lexicographic belief can in fact be regarded as an instance of probabilistic OMs, where the distribution over OMs is $\overrightarrow{p_\varepsilon} = (1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{m-1})$ with ε an indeterminate interpreted as an infinitesimally small value.

Here, we give a direct algorithm. As introduced for probabilistic OMs, for each node n, let $\overrightarrow{\mathsf{nbs}}_j(n)$ be the NBS for ω_-^j at n, which is a $t\times 1$ vector (each component corresponds to one of the t types of MIN). We can write all m NBSs as an $m \times t$ NBS matrix $\mathsf{NBS}(n) \coloneqq \left(\overrightarrow{\mathsf{nbs}}_1(n)^\intercal, \ldots, \overrightarrow{\mathsf{nbs}}_m(n)^\intercal\right)$. For any leaf node n, the matrix product $\mathsf{NBS}(n) \times \vec{u}(n)$ yields a $m \times 1$ matrix, or equivalently a vector of size m.

Proposition 4. *For a CGII with root r and opponent models $\omega_-^1, \ldots, \omega_-^m$ with a lexicographic interpretation, consider the instantiation of Algorithm 1 where: $V \coloneqq \mathbb{R}^m$; $\mathsf{eval}(n) \coloneqq \mathsf{NBS}(n) \times \vec{u}(n)$; MAX's operator is* lexmax*; MIN's operator is the component-wise addition of vectors $+$. Then it holds that $\overrightarrow{v_+} = \mathsf{val}(r)$, and the algorithm is polynomial-time.*

Nondeterministic OMs

The last case that we consider is when MAX does not have a probability distribution over MIN's OMs: MIN's strategy is only known to be among $\omega_-^1, \ldots, \omega_-^m$. This situation is very similar to planning under adversarial cost functions [18]. The maxmin value is then

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \min_{1 \le j \le m} u(\varsigma_+, \omega_-^i),$$

which in general is different depending on whether Σ_+ is Σ_+^{P} or Σ_+^{M}. MIN now has (*a priori*) more agency than in the case of probabilistic OMs, since they can choose from a larger (but still limited) set of strategies.

Proposition 5. *For a CGII with root r and opponent models $\omega_-^1, \ldots, \omega_-^m$ with a nondeterministic interpretation, consider the instantiation of Algorithm 1 where: situational values are finite sets of vectors in $\mathbb{R}^m$; for all terminal nodes n, $\mathsf{eval}(n) \coloneqq \{\mathsf{NBS}(n) \times \vec{u}(n)\}$; MAX's operator is set union $\cup$; MIN's operator is $\uplus$, defined for all situational values f and g by $f \uplus g \coloneqq \{(v_j + v'_j)_{1 \le j \le m} \mid \vec{v} \in f, \vec{v}\,' \in g\}$. Then the pure maxmin value satisfies*

$$\underline{v}_+ \coloneqq \max_{s_+ \in \Sigma_+^{\mathrm{P}}} \min_{1 \le j \le m} u(s_+, \omega_-^j) = \max_{\vec{v} \in \mathsf{val}(r)} \min_{1 \le j \le m} v_j.$$

The algorithm above is exponential time in the worst case; it can actually be shown that this problem is NP-complete, even if MAX has perfect information (i.e. MIN only has 1 type) and there are only 2 OMs of MIN.

Compared to Proposition 1, it can be seen that the knowledge of OMs transforms MAX's incomplete information about MIN's *type* into their incomplete information about MIN's *strategy*. Situational values are now sets of vectors of length m (instead of t). Each such vector implicitly represents a strategy of MAX by its expected payoff against each OM. In contrast with the case of probabilistic OMs, we cannot further collapse each vector to a single real number, since we have no distribution over the OMs. Still, reduction by weak dominance can be used just as for pure maxmin without any opponent model.

It follows that at the root, the remaining vectors are the non-dominated strategies of MAX against MIN's OMs. In other words, the algorithm computes the normal form of the game restricted to MIN's fixed m strategies, which justifies the correctness of Proposition 5 for pure maxmin.

As for mixed maxmin, one can modify the separation oracle in the LP algorithm of [13]: now the oracle only computes MIN's best responses from the m OMs.

6 Opponent models with uncertainty

We now come to our second contribution, about the case where a set of OMs of MIN is available, but MAX is not certain that MIN will behave as one of them. Without loss of generality, we focus on the case when there is a single OM ω_-, which encompasses as well the case of several OMs with a probability distribution, as discussed in section 5.

We assume that with a probability p^∞, which is known to MAX, MIN does not follow ω_-, in which case their behaviour is arbitrary and unpredictable, and that with probability $1 - p^\infty$ MIN follows ω_-. Intuitively, p^∞ quantifies MAX's uncertainty about MIN's behaviour. This may arise for instance when MAX tries to estimate MIN's gameplay level: with $1-p^\infty$, MIN is of a certain level with a behaviour predictable by some OM; otherwise, they have an unknown level and nothing can be assumed about their play.

This model yields a conflict between robustness and performance, well-known in the literature of linear programming with uncertain parameters or MDP planning under uncertain cost functions. MAX desires to be cautious and robust against MIN's unpredictable behaviour occurring with probability p^∞, and at the same time to improve their performance by exploiting their knowledge of the OM, which correctly predicts MIN's strategy with probability $1 - p^\infty$. Formally, we define the following modified maxmin value:

$$\underline{v}_+ \coloneqq \max_{\varsigma_+ \in \Sigma_+} \left((1 - p^\infty) u(\varsigma_+, \omega_-) + p^\infty \min_{s_- \in \Sigma_-^{\mathrm{P}}} u(\varsigma_+, s_-)\right),$$

where Σ_+ is either Σ_+^{P} or Σ_+^{M}.

Example. *Consider again Figure 1 and the OM ω_- "MIN plays* a *if of type* 1 *or* 2*,* b *if of type* 4 *or* 5*, and $\frac{1}{2}$a + $\frac{1}{2}$b if of type* 3*". The best strategy of MAX against ω_- is* (l, R) *with a payoff of* 1*. However, this strategy does not fare so well if MIN's strategy is not ω_- (or when p^∞ is close to*

1*): in the worst case, MIN plays* b *if of type* 1 *or* 2*, and* a *if of type* 4 *or* 5*. Against this strategy, MAX's expected payoff by playing* (l, R) *is only* $1/5$*. On the other hand, the pure maxmin strategy* (l, L) *only has a payoff of* $1/2$ *against* ω_-*, and so does the mixed maxmin strategy (which is the uniform strategy), hence neither is optimal when* p^∞ *is close to* 0*.*

It is clear from the example that the modified maxmin and optimal strategies depend on p^∞. We now show how to modify algorithms from the last sections to compute them.

Mixed strategies

We first consider the mixed strategies of MAX. The LP algorithm from [13] can compute the modified mixed maxmin value and an optimal strategy of MAX, with a minor modification of the separation oracle. Concretely, given a threshold v and a mixed strategy σ_+ for MAX, the separation oracle should now, apart from computing MAX's payoff v_{BR} with strategy σ_+ under MIN's best responses, also compute MAX's payoff against the OM $v_{\mathsf{OM}} = u(\sigma_+, \omega_-)$, then check whether $(1 - p^\infty)v_{\mathsf{OM}} + p^\infty v_{\mathsf{BR}} \geq v$ holds.

Example. *In the game of Figure 1 with* ω_- *as above, one can use this algorithm to verify that MAX's optimal strategy is* (l, R) *for* $p^\infty \leq 5/8$*, otherwise it is the uniform strategy. This confirms that when nondeterministic behaviour happens with a small enough probability, it is worth deviating from maxmin strategies in order to exploit the OM.*

Pure strategies

For pure strategies, we build on the algorithm for a single OM (Proposition 2). To cope with non-locality (because of MIN's partially unpredictable behaviour), we use situational values which are sets of ordered pairs $\langle s, \vec{v}\rangle$, with $s \in \mathbb{R}$ and $\vec{v} \in \mathbb{R}^t$, where t is the number of types of MIN. We call such a pair an *annotated vector*; it implicitly represents a strategy for MAX for which the payoff against ω_- is s, and the worst payoff against unpredictable behaviour is given by $\vec{v}$. We also maintain an NBS $\overrightarrow{\mathsf{nbs}}(n)$ for each node n, over MIN's types, as in section 5.

Proposition 6. *For a CGII with root* r*,* t *types of MIN with common prior* $\vec{q}$*, opponent model* ω_-*, and probability* p^∞ *that MIN does not behave according to* ω_-*, consider the instantiation of Algorithm 1 where: situational values are finite sets of annotated vectors; for all terminal nodes* n*,* $\mathsf{eval}(n) := \{\langle \overrightarrow{\mathsf{nbs}}(n) \cdot \vec{u}(n), \vec{u}(n)\rangle\}$*; MAX's operator is set union* $\cup$*; MIN's operator is* $\sqcap'$*, where, for all situational values* f *and* g*,* $f \sqcap' g$ *is defined to be*

$$\{\langle s + s', (\min(v_i, v_i'))_{1 \leq i \leq t}\rangle \mid \langle s, \vec{v}\rangle \in f, \langle s', \vec{v}\,'\rangle \in g\}.$$

Then the modified pure maxmin value satisfies

$$\underline{v}_+ = \max_{\langle s, \vec{v}\rangle \in \mathsf{val}(r)} \big((1 - p^\infty)s + p^\infty(\vec{q} \cdot \vec{v})\big).$$

Notice that when combining two annotated vectors at a MIN's node, the scalar part is additive; this reflects the fact that when following the (single) OM, MIN has no agency, just as in the case without uncertainty.

Example. *Using the algorithm above for the game in Figure 1 with the aforementioned OM* ω_-*, we find that MAX's optimal strategy is* (l, R) *for* $p^\infty \leq 5/7$*, otherwise* (l, L) *or* (r, R)*. Again, this shows that it may be worth deviating from maxmin strategies in order to exploit an OM.*

Reduction of situational values The algorithm in Proposition 6 generalises the one in Proposition 1, which can be regarded as the case $p^\infty = 1$. We have seen that, in the latter case, the only sound reduction of situational values is the elimination of weakly dominated strategies. Interestingly, when an OM is available, further reductions become sound.

Let n be a node, and $\langle s, \vec{v}\rangle, \langle s', \vec{v}\,'\rangle \in \mathsf{val}(n)$ be two annotated vectors. Discarding $\langle s', \vec{v}\,'\rangle$ because of $\langle s, \vec{v}\rangle$ is sound if MAX is never worse-off in the whole game if they choose $\langle s, \vec{v}\rangle$ instead of $\langle s', \vec{v}\,'\rangle$ at n.

Since scalar parts are summed up, if $s > s'$ holds, then $\langle s, \vec{v}\rangle$ has an advantage $s - s'$ over $\langle s', \vec{v}\,'\rangle$ in terms of contribution to the final value at the root. Contrastingly, for the vectorial part, components for which $\vec{v}$ is larger than $\vec{v}\,'$ might be erased by the combination (via component-wise min) of vectors at an ancestor of n. In other words, $\vec{v}$'s advantage with respect to $\vec{v}\,'$ can be annihilated at the root. On the other hand, the components for which $\vec{v}$ is smaller than $\vec{v}\,'$ may never get erased so that $\vec{v}$'s disadvantage with respect to $\vec{v}\,'$ can survive intact at the root.

Hence, in the worst case, $\vec{v}\,'$ can keep all advantages it has compared to $\vec{v}$, while $\vec{v}$ can lose all its advantages. Hence, to safely discard $\langle s', \vec{v}\,'\rangle$, the advantage of $\vec{v}\,'$ over $\vec{v}$ must be no larger than the advantage of s over s'. More formally, we can safely discard $\langle s', \vec{v}\,'\rangle$ when the following holds:

$$(1 - p^\infty)(s - s') \geq p^\infty \sum_{1 \leq i \leq t} \big(q_i \max(v_i' - v_i, 0)\big), \quad (3)$$

Notice that without the scalar part (e.g. when $p^\infty = 1$), the pruning condition (3) reduces to $v_i \geq v_i'$ for all i, which is exactly the pruning condition shown in section 4.

7 Application to recursive opponent models

We now propose an application of the algorithms presented before to the computation of optimal strategies with recursive opponent models. We formulate a quite general setting, where various types of opponent models naturally arise.

Limitations of the best-defence model

In general, in a game with incomplete information, both players have incomplete information, rather than just MAX.

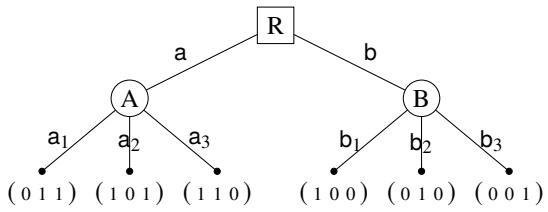


Figure 2: A CGII with 3 possible types of MAX.

As a result, the best-defence model usually gives MIN too much power.

Example. *Consider the game in Figure 2, where MAX has* 3 *types and MIN has only* 1 *(hence MIN has incomplete information). If MAX reasons according to the best-defence model, then both actions* a *and* b *have a value of* 0*: MAX of type i reasons that MIN will play* a_i *at node A, and* b_j *at node B for some* $j \neq i$*. The culprit is that under the best-defence model, MAX assumes MIN is aware of MAX's type, therefore can adapt their strategy to it. However, if MAX realises MIN is unaware of their type, then MAX will prefer* a *since under uniform common prior over MAX's types,* a *yields an expected payoff of* $2/3$*, compared to* b*'s* $1/3$*.*

On the other hand, computing maxmin strategies for the original game tree without using the best-defence model is not ideal either, for these strategies fail to exploit any assumption one may have about their adversary, such as that they have limited computational power or reasoning depth, or that they have a predictable behaviour pattern. Such assumptions make sense in particular when playing against humans [10, 21].

Proposed framework

The framework which we propose can be seen as a generalisation of the cognitive hierarchy model [4] and at the same time as a counterpart of interactive POMDPs [6] for competitive games. The general idea is to define *level-k* strategies to be the optimal strategies against an adversary of level $k-1$, and recursively down to level-0 strategies. We however give a general and parametrizable definition about (1) how level-0 strategies are defined, (2) how optimal strategies at a given level are aggregated, and (3) how strategies of various levels are aggregated. Moreover, using our results in section 6, the framework leaves the possibility for players to assign a non-zero probability to the event that their opponent has an unknown strategy/level.

As a consequence, this framework serves as a compromise between the best-defence model and the full game, and can be used to find better strategies against non-omnipotent and non-omniscient players; in particular, it generalises the best-defence model. Moreover, this framework can be used to explain real-life human psychological gameplay in games with incomplete information such as bridge, as we illustrate at the end of this section.

For the formal definition, consider a two-player zero-sum game. Let Σ^0_+, Σ^0_- be non-empty sets of strategies of MAX and MIN, respectively. Moreover, let $\oplus : 2^\Sigma \to \Sigma$ be a function that maps any set of (pure or mixed) strategies to a single (pure or mixed) strategy, and $\mathrm{BR} : \Sigma^* \to 2^\Sigma$ be a function which maps any tuple of strategies to a set of strategies; $\oplus$ will be used to aggregate strategies of a player at a given level, and BR to compute the set of optimal strategies given a tuple of opponent models (one per lower level).[7]

Definition 7 (level-k strategies)**.** *Let* Σ^0_+, Σ^0_-, $\oplus$, BR *be defined as above, and let* $i \in \{+,-\}$*. The set of* level-0 *strategies for player* i *is defined to be* Σ^0_i*. For* $k \geq 1$*, the set of* level-k strategies *for player* i*, denoted by* Σ^k_i*, is defined to be* $\mathrm{BR}\left(\oplus(\Sigma^{k-1}_{-i}), \oplus(\Sigma^{k-2}_{-i}), \ldots, \oplus(\Sigma^0_{-i})\right)$*.*[8]

In short, the level-k strategies of player i are the best responses (computed by BR, the *best-response function*) against an opponent using the strategy $\oplus(\Sigma^{k'}_{-i})$ (computed by $\oplus$, the *intra-level aggregation*) at each level $k' \leq k$. The boundary conditions, i.e. the level-0 strategies, are given by Σ^0_+ and Σ^0_-, which can come from maxmin strategies under the best-defence model, randomly chosen strategies [18], modelling assumptions for human players [24], etc.

Example. *The Poisson-CH model in [4] is captured by choosing* Σ^0_+ *and* Σ^0_- *to be the set of all pure strategies of MAX and MIN, the intra-level aggregation* $\oplus$ *to map any set of strategies to the uniform mixture of the set, and the best-response function* BR *to map a tuple of strategies* $(\sigma^{k-1}_{-i}, \ldots, \sigma^0_{-i})$ *to the set of all pure best responses to the mixed strategy* $p_{k-1}\sigma^{k-1}_{-i} + \cdots + p_0\sigma^0_{-i}$*, where* $p_{k-1}, \ldots, p_0$ *follow some Poisson distribution.*

An interesting choice for the intra-level aggregation $\oplus : 2^\Sigma \to \Sigma$ is given by the uniform mixture, as in the example above. Under this $\oplus$, many other situations can be modelled by using different best-response functions BR for the inter-level aggregation. For games with incomplete information, if a function BR computes the best responses per type of player[9], then such BR can be implemented by the algorithms presented in the last sections. Some examples follow.

Probabilistic model If each player i at level k has a subjective probability over their opponent's reasoning levels in the form of a vector $(p^{k-1}_{i,k}, p^{k-2}_{i,k}, \ldots p^0_{i,k})$, then we can define BR to compute, for each player

[7] The framework could be easily adapted to more general functions, e.g. an aggregation of the strategies at the same level into a set or a tuple of strategies. It could also be easily adapted to general games, beyond the two-player and zero-sum assumptions.

[8] For a player $i \in \{+,-\}$, we write $-i$ for the other player.

[9] In other words, for each player, BR computes the best strategies for each type of this player.

i and level k, the best responses against the mixture $p_{i,k}^{k-1} \oplus (\Sigma_{-i}^{k-1}) + \cdots + p_{i,k}^{0} \oplus (\Sigma_{-i}^{0})$ (which can be implemented by the algorithm in Proposition 3). This model amounts to assuming that a player at level k reasons as if their opponent places themselves at a reasoning level drawn from the above distribution; such a distribution can be obtained by empirical studies, for instance by fitting a model against a population of possible opponents in an open tournament.

Iterative model By setting $p_{i,k}^{k-1} = 1$ for all i and k in the previous model, we can model situations where each player at level k assumes their opponent reasons at exactly level $k-1$, which corresponds to Proposition 2.

Lexicographic model BR can also be defined to compute the best responses against the tuple of opponent models $(\oplus(\Sigma_{-i}^{k-1}), \ldots, \oplus(\Sigma_{-i}^{0}))$ under the lexicographic interpretation (which can be implemented by the algorithm in Proposition 4); this amounts to assuming that the opponent reasons at level $k-1$, and to tie-break equivalent strategies by level $k-2$, and so on.

Nondeterministic model With a BR as in Proposition 5, we can model situations where each player at level k assumes the opponent reasons at a level lower than k but without assuming a distribution over their levels. In such cases, the incomplete information about the opponent's types is transformed into the one about their reasoning levels, which are in general much fewer.

Partially unknown opponent model If in addition to the probabilistic or the lexicographic model above, we consider probabilities $p_{i,k}^{\infty}$ that the opponent of player i at level k is not reasoning at any level lower than k, then we can use the approaches under uncertainty from section 6.

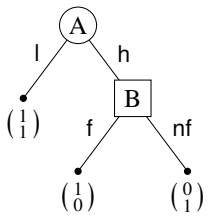
Let us also emphasise that the straightforward generalisation of this framework to general games allows, for instance, to take into account one's partner's incomplete information in multiplayer games, akin to interactive POMDPs.

A real-life example

We now give an example application of our formalism, which captures the psychological strategies of a contract bridge deal played in a bridge tournament. We present the abstract version of the game on Figure 3 (left); for the bridge deal itself, see [12].

In this game, the common prior about MIN's types is given by $p_1 = 0.4$ and $p_2 = 0.6$. For the recursive reasoning, $\oplus$ is given the uniform mixture, BR is given by the lexicographic model, and the level-0 strategies for both players are their pure maxmin strategies.

The first few levels of the recursive reasoning proceed as in Figure 3 (right). In the following, we write $\sigma_-^1|\sigma_-^2$ for MIN's strategy if type-1 MIN plays σ_-^1 and type-2 MIN plays σ_-^2; and $\frac{\mathsf{l}+\mathsf{h}}{2}$ for the uniform mixed strategy $\frac{1}{2}\mathsf{l} + \frac{1}{2}\mathsf{h}$.



k	MAX	MIN 1	MIN 2
0	{nf}	{l, h}	{l, h}
1	{nf}	{h}	{l, h}
2	{f}	{h}	{l, h}
3	{f}	{h}	{h}

Figure 3: Recursive reasoning in a CGII with 2 types of MIN. For each $k \leq 3$, the set of level-k strategies (as defined in Definition 7) for MAX, MIN of type 1, and MIN of type 2 are given in the table.

k = 0: MAX prefers nf, which achieves a maxmin value of 0.6, against 0.4 for f; both types of MIN are indifferent between l and h since both yield a minmax value of 1.

k = 1: Against $(\oplus(\Sigma_-^0)) = (\frac{\mathsf{l}+\mathsf{h}}{2}|\frac{\mathsf{l}+\mathsf{h}}{2})$, MAX's best strategy is still nf; however, against $(\oplus(\Sigma_+^0)) = (\mathsf{nf})$, type-1 MIN prefers h which yields a value of 0.

k = 2: Against $(\oplus(\Sigma_-^1), \oplus(\Sigma_-^0))$, MAX now prefers f, which is strictly better than nf against $\oplus(\Sigma_-^1) = \mathsf{h}|\frac{\mathsf{l}+\mathsf{h}}{2}$ since the NBS of MAX at node B judges MIN is more likely to be of type 1 than of type 2 if MIN plays $\mathsf{h}|\frac{\mathsf{l}+\mathsf{h}}{2}$;

k = 3: At level-3, type-1 MIN still prefers h: h and l are equivalent against $\oplus(\Sigma_+^2) = \mathsf{f}$, but h is preferred against $\oplus(\Sigma_+^1) = \mathsf{nf}$; but now type-2 MIN also prefers h!

As it turns out, this recursive reasoning perfectly captures what happened during the bridge deal, where MAX was at level 2 and therefore chose f (rather than the maxmin strategy nf) while MIN, being of type 2, reasoned at level 3 and used strategy h to defeat MAX.

Admittedly, in the game of Figure 3, MIN's strategy h weakly dominates l, and therefore MIN should never play l. However, this game is an extreme abstraction of the real game, which has a huge number of strategies; it is not at all obvious that h is weakly dominant. In addition, many other real-life examples of recursive reasoning, which we cannot give here for space reasons, yield risky strategies, i.e. those that are neither maxmin nor dominant and as a result could perform worse if the opponent's reasoning level is incorrectly estimated. Indeed, in our example, this is the case for MAX's level-2 strategy f: against MIN of level 1, it indeed performs better (0.7) than the maxmin strategy nf (0.6), but against MIN of level 3 it performs worse (0.4).

8 Conclusion

We have proposed a number of ways to take into account opponent models in games with incomplete information.

For each type of opponent model, we have formally defined the maxmin value and proposed an algorithm to compute it. We have also considered the case where the opponent, with some probability, does not follow any model, and the goal is to be robust against any possible adversarial strategy while maximally exploiting the knowledge of opponent models.

As an application, we have proposed a general framework of recursive opponent models. This parametrizable framework can model, by using appropriate intra-level aggregations and best-response functions, a wide range of situations of recursive reasoning, including the possibility that an opponent does not follow any model. Illustrated by an example from the game of Bridge, we have shown how this framework captures real-life strategic reasoning, and how our algorithms can be used for models defined in the economy literature [4].

Two main directions are worth pursuing for future work: To consider games represented compactly (e.g. by game rules) instead of explicitly by their game tree; and to formally define our recursive framework in doxastic logic, which is similar to the notion of rationalisability in epistemic game theory but allows false beliefs (about the others' level, for instance). For the latter direction, the intuition is that level-k strategies can be seen as strategies optimal for an agent with a depth of knowledge of k in the Kripke structure over the players' types. For instance, assuming the players' distributed knowledge of the actual combination of types, it can be shown that under this definition, level-0 strategies are optimal strategies against the best-defence model.

References

[1] Albrecht, Stefano V. and Peter Stone: *Autonomous agents modelling other agents: A comprehensive survey and open problems*. Artif. Intell., 258:66–95, 2018. https://doi.org/10.1016/j.artint.2018.01.002.

[2] Bonanno, Giacomo: *Game Theory*. Kindle Direct Publishing, 2018.

[3] Bosanský, Branislav, Christopher Kiekintveld, Viliam Lisý, and Michal Pechoucek: *An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information*. J. Artif. Intell. Res., 51:829–866, 2014. https://doi.org/10.1613/jair.4477.

[4] Camerer, Colin F., Teck Hua Ho, and Juin Kuan Chong: *A cognitive hierarchy model of games*. The Quarterly Journal of Economics, 119(3):861–898, August 2004, ISSN 0033-5533.

[5] Damme, Eric van: *Stability and Perfection of Nash Equilibria*. Springer Berlin Heidelberg, 1991.

[6] Doshi, Prashant, Piotr J. Gmytrasiewicz, and Edmund H. Durfee: *Recursively modeling other agents for decision making: A research perspective*. Artif. Intell., 279, 2020. https://doi.org/10.1016/j.artint.2019.103202.

[7] Frank, Ian and David A. Basin: *A theoretical and empirical investigation of search in imperfect information games*. Theor. Comput. Sci., 252(1-2):217–256, 2001. https://doi.org/10.1016/S0304-3975(00)00083-9.

[8] Ginsberg, Matthew L.: *GIB: imperfect information in a computationally challenging game*. J. Artif. Intell. Res., 14:303–358, 2001. https://doi.org/10.1613/jair.820.

[9] Gmytrasiewicz, Piotr J. and Prashant Doshi: *A framework for sequential planning in multi-agent settings*. J. Artif. Intell. Res., 24:49–79, 2005. https://doi.org/10.1613/jair.1579.

[10] Iida, Hiroyuki, Jos W. H. M. Uiterwijk, H. Jaap van den Herik, and I. S. Herschberg: *Potential applications of opponent-model search, part 1: The domain of applicability*. J. Int. Comput. Games Assoc., 16(4):201–208, 1993. https://doi.org/10.3233/ICG-1993-16403.

[11] Iida, Hiroyuki, Jos W. H. M. Uiterwijk, H. Jaap van den Herik, and I. S. Herschberg: *Potential applications of opponent-model search, part 2: Risks and strategies*. J. Int. Comput. Games Assoc., 17(1):10–14, 1994. https://doi.org/10.3233/ICG-1994-17103.

[12] Karpin, Fred L.: *Psychological strategy in contract bridge: The techniques of deception and harassment in bidding and play*. Dover Publications, 1977.

[13] Koller, Daphne and Nimrod Megiddo: *The complexity of two-person zero-sum games in extensive form*. Games and Economic Behavior, 4(4):528–552, 1992, ISSN 0899-8256. https://www.sciencedirect.com/science/article/pii/089982569290035Q.

[14] Koller, Daphne, Nimrod Megiddo, and Bernhard von Stengel: *Efficient computation of equilibria for extensive two-person games*. Games and Economic Behavior, 14(2):247–259, 1996, ISSN 0899-8256. https://www.sciencedirect.com/science/article/pii/S0899825696900512.

[15] Kuhn, H. W.: *11. Extensive Games and the Problem of Information*, pages 193–216. Princeton University Press, Princeton, 1953, ISBN 9781400881970. https://doi.org/10.1515/9781400881970-012.

[16] Li, Junkang, Bruno Zanuttini, Tristan Cazenave, and Véronique Ventos: *Generalisation of alpha-beta search for AND-OR graphs with partially ordered values*. In Raedt, Luc De (editor): *Proc. Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022)*, pages 4769–4775. ijcai.org, 2022. https://doi.org/10.24963/ijcai.2022/661.

[17] Maschler, Michael, Eilon Solan, and Shmuel Zamir: *Game Theory*. Cambridge University Press, 2nd edition, 2020.

[18] McMahan, H. Brendan, Geoffrey J. Gordon, and Avrim Blum: *Planning in the presence of cost functions controlled by an adversary*. In Fawcett, Tom and Nina Mishra (editors): *Proc. Twentieth International Conference on Machine Learning (ICML 2003)*, pages 536–543. AAAI Press, 2003. http://www.aaai.org/Library/ICML/2003/icml03-071.php.

[19] Nashed, Samer B. and Shlomo Zilberstein: *A survey of opponent modeling in adversarial domains*. J. Artif. Intell. Res., 73:277–327, 2022. https://doi.org/10.1613/jair.1.12889.

[20] Perea, Andrés: *Epistemic Game Theory: Reasoning and Choice*. Cambridge University Press, 2012.

[21] Stahl, Dale O. and Paul W. Wilson: *On players' models of other players: Theory and experimental evidence*. Games and Economic Behavior, 10(1):218–254, 1995, ISSN 0899-8256. https://www.sciencedirect.com/science/article/pii/S0899825685710317.

[22] von Stengel, Bernhard: *Efficient computation of behavior strategies*. Games and Economic Behavior, 14(2):220–246, 1996, ISSN 0899-8256. https://www.sciencedirect.com/science/article/pii/S0899825696900500.

[23] Weerd, Harmen de, Rineke Verbrugge, and Bart Verheij: *How much does it help to know what she knows you know? an agent-based simulation study*. Artif. Intell., 199-200:67–92, 2013. https://doi.org/10.1016/j.artint.2013.05.004.

[24] Wright, James R. and Kevin Leyton-Brown: *Level-0 models for predicting human behavior in games*. J. Artif. Intell. Res., 64:357–383, 2019. https://doi.org/10.1613/jair.1.11361.