CONTRIBUTIONS TO MULTIAGENT DECISION MAKING UNDER UNCERTAINTY AND PARTIAL OBSERVABILITY

Aurélie Beynier

SORBONNE UNIVERSITÉ

CONTRIBUTIONS TO MULTIAGENT DECISION MAKING UNDER UNCERTAINTY AND PARTIAL OBSERVABILITY

Aurélie BEYNIER

Mémoire d'habilitation à diriger des recherches

Sorbonne Université

Soutenu publiquement le **16 novembre 2018**

Coordinateur	Nicolas Maudet	Professeur Sorbonne Université
Rapporteurs	Craig Boutilier	Principal Scientist Google Moutain View, USA
	Jérôme Lang	Directeur de Recherche CNRS Université Paris-Dauphine
	Francesca Toni	Professeur Imperial College London, UK
Examinateurs	Amal El Fallah-Seghrouchni	Professeur Sorbonne Université
	Bruno Zanuttini	Professeur Université Caen-Normandie

CONTENTS

RÉSUMÉ

Ce document présente les principales activités de recherche que j'ai menées depuis l'obtention de mon doctorat en novembre 2008. Mon travail de recherche porte sur la coordination, la planification et la prise de décision distribuée dans les systèmes multiagents. Je m'intéresse plus particulièrement aux interactions entre agents leur permettant de planifier et d'exécuter des actions de manière coordonnée dans des environnements partiellement observables et incertains.

Le chapitre 1 introduit le domaine des systèmes multiagents et présente les problématiques liées à la coordination dans des environnements partiellement observables et incertains.

Le chapitre 2 porte sur la prise de décision distribuée dans le cadre de l'allocation de ressources. En partant d'une distribution initiale d'un ensemble de ressources entre des agents, le travail présenté dans ce chapitre vise à étudier et mettre en œuvre des procédures basées sur des échanges locaux de ressources permettant aux agents d'améliorer leur satisfaction. Je m'intéresse tout d'abord au cas des "house-markets" dans lesquels chaque agent possède une seule ressource. Bien que les procédures distribuées puissent engendrer des pertes d'efficacité importantes par rapport aux procédures centralisées, je montre que les procédures d'échanges bilatéraux possèdent des propriétés intéressantes en termes de qualité des solutions. Je m'intéresse ensuite à des cadres plus complexes où les agents peuvent posséder plusieurs ressources et doivent décider quel agent rencontrer et quelles ressources échanger, en n'ayant qu'une connaissance partielle des ressources possédées par les autres. J'étudie plus particulièrement l'équité des solutions calculées en se basant sur la notion "d'absence d'envie". Enfin, j'aborde les problèmes d'allocation équitable lorsque les relations entre agents sont définies par un graphe social et que chaque agent n'est en contact qu'avec un sous-ensemble d'agents.

Le chapitre 3 traite de la planification multiagent dans des environnements partiellement observables et incertains. Je m'intéresse plus particlièrement aux Processus Décisionnels de Markov Décentralisés (Dec-MDPs et Dec-POMDPs) qui offrent un modèle mathématique adapté à la prise de décision distribuée sous incertitude. Toutefois, ces modèles souffrent de certaines limitations telles qu'une représentation du temps et des actions restreinte. Ils font de plus l'hypothèse que les données du problème sont stationnaires (elles ne changent pas au cours du temps). Enfin, il a été démontré que résoudre de manière optimale des Dec-MDPs et des Dec-POMDPs constitue un problème très difficile (NEXP-Complet), ce qui limite leur applicabilité à des problèmes réels. Les travaux présentés dans ce chapitre visent tout d'abord à améliorer la modélisation du temps et des actions dans les Dec-POMDPs. Je m'intéresse d'autre part à la mise en place d'approches de résolution efficaces permettant de traiter des problèmes de grandes tailles. Les approches que j'ai développées se basent sur la recherche d'une solution approchée et l'exploitation de la structure du problème afin d'en décomposer la résolution. Enfin, je m'intéresse à la modélisation et à la résolution de problèmes de décision dans des environnements non-stationnaires.

Le chapitre 4 aborde l'argumentation stratégique, c'est-à-dire la planification de stratégies en théorie de l'argumentation. La théorie de la décision permet d'améliorer les comportements des agents dans des systèmes argumentatifs, où agents humains et logiciels débattent entre eux. Par ailleurs, l'argumentation offre des outils permettant de résoudre des problèmes de décision distribuée en aidant à régler des situations de non-coordination entre agents. Dans ce chapitre, je mets en évidence l'apport des modèles markoviens dans deux types de problèmes : les débats stochastiques et les problèmes de médiation face à des agents dont les comportements sont non-stationnaires. Je montre ainsi comment des problèmes de planification de stratégies argumentatives ou de médiation peuvent être représentés et résolus par les modèles développés dans le chapitre précédent.

1 INTRODUCTION

The purpose of *Artificial Intelligence* (AI) is to set up autonomous systems endowed with smart cognitive functions enabling them to perform complex tasks while interacting with their environment and with other entities (software systems or humans). As evidenced by different recent reports (Stone et al., 2016; INRIA, 2016; Artificielle, 2017), Artificial Intelligence now becomes a reality in everyday life. Autonomous vehicles, smart buildings and cities, service robots and many other applications may now be used by an increasing audience.

Multiagent Systems (MAS) fit naturally in the development of AI systems. A multiagent system is composed by a set of autonomous entities (software or human entities) acting and interacting in a same environment in order to achieve their objectives (Russell and Norvig, 2003; Weiss, 1999). A large range of AI applications are inherently composed of a set of autonomous and interacting agents. We can cite for instance autonomous vehicles traveling in a city, rescue rovers operating in a disaster area to find and rescue some victims, mobile robots patrolling sensitive area or assisting people in public area, wireless sensor networks in smart cities...

In order to exhibit smart behaviors, an agent must be able to make autonomous decisions based on her knowledge about the environment and the other agents. In fact, an agent continuously executes an observation-action cycle (see Figure 1) where she observes her environment and then decides which action to execute. The execution of the action modifies the environment and the agent obtains new observations leading to another decision about the next action to execute and so on. All the information contained in the environment that is relevant to the decision-making process of the agent constitutes the *state of the environment*[1]. As an agent obtains new observations of the environment, she maintains an *internal state* that contains all the knowledge of the agent about the system. This knowledge may consist in built-in knowledge, observations of the system, information received by communication or evidences deduced from knowledge reasoning. The current internal state is then used by the agent to make decisions.

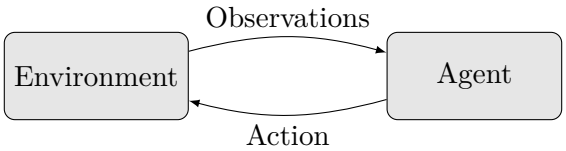


Figure 1: Observation - action cycle of an agent

Since all the agents act in the same environment, each agent must account for the possible *interactions* with the others when taking decisions. An interaction situation occurs when several agents are brought into a dynamic relationship through their individual actions. Interactions can be classified according to the compatibility of the agents' goals, the available resources and the skills of the agents (Ferber, 1999). In a cooperative setting where the agents have compatible goals, interactions can enhance the capacities of the agents by enabling new tasks that could not be executed by a single agent. Interactions may also consist in direct communication between the agents to exchange some

[1] This state could also be referred to as the state of the world or the state of the system. From the point of view of an agent, the other agents will be considered as being part of the environment. The relevant information about the other agents will thus be endowed in the state of the environment.

knowledge, resources or tasks. When resources are limited or the agents have conflicting goals, interactions may have however negative effects. For instance, the actions of an agent may consume some limited resources and thus prevent another agent from executing her own actions. Some actions may also invalidate some conditions required for the execution of other agents' actions. For instance, a mobile robot moving in a corridor may block the paths of other robots. To develop efficient behaviors, the agents have to foresee the actions of the other agents and take *coordinated decisions*.

The field of MAS covers a broad range of applications where the agents may have various capabilities and the environment may have different characteristics. Each MAS can be characterized by various properties raising different issues and influencing the way the MAS will be designed. Among the most significant features of MAS, the issues raised in this document as well as our contributions are more specifically concerned with the following characteristics:

- *Discrete vs. Continuous environment:* if the number of states of the environment and the number of actions that can be executed are finite, the environment is said to be discrete. Otherwise, the environment is continuous.

- *Dynamics environment:* a *static* environment does not evolve over time except through the effects of the actions performed by the agents (Woolridge, 2001; Vlassis, 2007). In a *dynamic* environment some changes occurring in the environment are out of control of the agents. Much work dealing with single-agent decision-making has been interested in static environments. However in MAS, due to the presence of multiple agents, the environment appears as dynamic from the point of view of an agent. In fact, the environment is not only modified by the action of the agent herself but the environment is also modified by the actions of the other agents.

- *Deterministic or Stochastic environment:* an environment is said to be *deterministic* if the outcome of an action cannot be predicted with certainty. In a *stochastic* environment, an action may lead to different outcomes. As pointed out by Woolridge (2001), stochasticity is closely related to the dynamicity of the environment. In a dynamic environment, many events may interfere with the actions of an agent, leading to different possible action outcomes. Depending of the degree of observability, it may also be difficult for the agent to distinguish some states with different characteristics influencing differently the outcome of an action. The sources and models of uncertainty will be detailed further below.

- *Degree of observability:* if an agent has access to all the information contained in the environment and relevant to her decision, the environment is said to be *fully observable*. Otherwise, the environment is *partially observable*.

- *Ability to communicate:* as an agent acts in the environment, she may be able to communicate with the other agents by sending messages. This kind of communication is referred to as *direct communication*. Due to physical constraints (distance between the agents, topology of the environment, communication technology used), limited resources (bandwidth, energy) or security reasons (for instance, an adversary listening to the messages), an agent may not be able to communicate at all time with the other agents to exchange some knowledge, coordinate or negotiate.

- *Cooperative or Self-interested agents:* agents may be self-interested or cooperative. A *cooperative agent* is interested in maximizing the performance of the group whereas a *self-interested agent* has her own preferences on the state of the system and will try to attempt the state she likes the most.

These characteristics of MAS are not exhaustive, we only reviewed the main features studied in this document. For a more complete characterization of MAS, we refer the reader to (Sycara, 1998; Ferber, 1999; Weiss, 1999; Woolridge, 2001; Vlassis, 2007; Russell and Norvig, 2003; Shoham and Leyton-Brown, 2008; Vidal, 2009; Bordini et al., 2014).

1.1 Illustrative example

For illustrative purpose, we consider a multi-robot rescue scenario such as the ones investigated in the Robocup Rescue[2] (Kitano and Tadokoro, 2001; Skinner and Ramchurn, 2010; Annibal B. M. da Silva, 2000). A set of rescue agents (robots) has to operate in a city after an earthquake occurs. Roads are blocked by rubble, some buildings are burning and injured civilians must be rescued and driven in safe places. Different types of agents are involved to manage the crisis situation:

- Police forces: they are responsible for removing blockades from the roads.
- Ambulance teams: they are responsible for rescuing humans and taking them to a safe place (a refuge for instance).
- Fire brigades: they are responsible for extinguishing fires.

The environment is inherently dynamic since fires evolve inside the buildings and can spread from one building to another. Moreover, the health of humans may decrease over time. Each agent has limited perception of the environment and only observes things in her (limited) line of sight. Thus, agents cannot determine the exact state of the whole environment. For instance, they cannot localize all injured civilians or determine the state of all buildings. Moreover, changes of the environment are uncertain because it is not possible to foresee accurately how the fires would evolve and how the health of each civilian may deteriorate.

Agents can exchange messages using radio communication or direct communication but direct communication has a limited range and messages are only received by the agents within a radius from the sender. Radio communication reaches all the agents but is limited in the number of messages and bandwidth. Moreover, it is not reliable and may fail.

The objective of the agents is to rescue as much civilians as possible and prevent from damage as much property as possible. Therefore, agents need to work together. Indeed, police forces must remove blockades so that ambulance teams and fire brigades can access some sites. Fire brigades must have extinguished fire in a building before an ambulance team can rescue civilians in this building. Coordination is crucial issue for developing effective behaviors. Nonetheless, agents must act in real-time and have limited processing time to make decisions.

Robot rescue scenarios highlight a large set of issues studied in multiagent systems. Among them, we will particularly focus on the following topics that will be discussed with more details in the document:

- *MultiAgent Resource Allocation (MARA):* MARA consists in assigning resources to agents. In the Robocup rescue context, resources are in fact tasks to allocate among rescue entities. Depending of her capabilities and state (location, remaining power...), an agent will have different preferences among the tasks. In a rescue context, the allocation cannot be computed by a central entity because of partial observability and limited communication. When agents meet, they may exchange tasks to execute, in order to obtain more preferred tasks that would lead to more efficient behaviors.

[2] http://roborescue.sourceforge.net/web/

- *Distributed cooperative sequential decision making:* Each task requires the agents to execute, in a coordinated way, a sequence of individual and joint actions. For instance, rescuing some civilians in the city hall requires that an ambulance unit drives to the city hall, then enters the building, removes debris, evacuates civilians and finally drives them to a safe place. The decisions of the agents must fulfill dependency constraints between these tasks. Agents thus have to make individual but coordinated decisions based on their partial observations of the system.

- *Multiagent Planning under uncertainty:* In order to improve their performances, agents must anticipate the effects of their actions and future decisions. Planning methods have thus to be developed. For instance, moving to the city hall requires path planning. Moreover, since the environment is uncertain, agents must consider the possibility that some moves fail because of unforeseen actions or events. For instance, ambulance units may not succeed to reach their target because of unforeseen blockades or fires. Due to agent interactions, each individual plan is highly dependent of the plans of the other agents. This is why specific methods are needed to solve planning problems in multiagent settings. Moreover, plans must fulfill resource, space and temporal constraints. For instance, fire brigades have limited water capacities and water tanks can only be refilled at dedicated places.

- *Information sharing:* Rescue entities can exchange some information by sending messages. Since communication is constrained, each agent must carefully assess which information must be communicated and when. The environment being dynamic and uncertain, agents may have inconsistent or conflicting knowledge. For instance, one agent may believe that some civilians to rescue in the city hall must be given the priority whereas another one may believe that the priority should be given to extinguishing the fire in the old library. In this case, agents will have to exchange information and eventually to debate about their representation of the environment or about the appropriate decisions to take.

The robot rescue domain is used here to illustrate the discussion but our work is not restricted to a single application domain and considers more general problems among which resource allocation, multiagent planning or abstract argumentation. Our work has also been motivated by applications dealing with cooperative multi-robot exploration and multiagent patrolling.

In this document we will focus more specifically on issues arising from partial observability and uncertainty in distributed decision-making.

1.2 Partial observability

In realistic settings, agents are often unable to observe the whole state of the environment. Indeed, agents have limited sensors that cannot give a full and accurate picture of the environment state at any time. The sensors of the agents may have limited range of perception and the environment may be too large regarding the perception range. Sensors may also return inaccurate measurements resulting in noisy perceptions. Indeed, even the most sophisticated robots have noisy and inaccurate sensors: pictures captured by the camera are influenced by the luminosity of the scene, data from infrared sensors depends of the light reflection from the surface...

Several degrees of observability are commonly studied in the literature to characterize the observability of the environment (Goldman and Zilberstein, 2004; Becker et al., 2004; Pynadath and Tambe, 2011):

- *Non-observable*: none of the agents observes any useful information about the environment.

- *Partially-observable*: the agents only observe part of the useful information about the environment.
- *Fully-observable*: the useful information is observed by the agents who can determine exactly the state of the environment.

In a multiagent context, the notion of observability of the environment state can be refined by distinguishing individual observability from joint obsevability. *Individual observability* denotes the observability of an agent alone. *Joint observability* is the union of the observations of the agents. It corresponds to the observability of the whole set of agents if they could gather all their observations. This is also referred to as collective observability.

It has to be noticed that joint observability is relevant when the agents can communicate and communication is free, instantaneous and reliable. In such settings, there is no loss nor distortion of the messages and the agents are able to communicate at all time without cost. The agents must also be willing to disclose their knowledge or some private information to the other agents.

The degrees of observability may concern individual observability as well as joint observability. For instance, in jointly fully observable environments, the agents are able to deduce exactly the state of the environment if they exchange all their observations. In a jointly partially observable environment, some relevant information of the environment is not observable by any agent: even gathering all their observations does not allow the agents to deduce the state of the environment.

It is well known that partial observability mainly impacts the complexity of making optimal decisions. Even in the single agent case, it has been proved that optimal decision-making under partial observability is a hard problem (Papadimitriou and Tsitsiklis, 1987).

In multiagent systems, the problem is even more difficult because of interactions between agents. Since agents are spread among the environment and have limited range of perception, each agent obtains different observations of the system. Unless the agents communicate all their observations at all time, an agent cannot determine exactly the set of observations made by the other agents at each decision step. Since strategies of action depend of these observations, the actions of the other agents are difficult to determine. It is thus a difficult problem for an agent to predict the actions of the other agents and possible interactions. Making coordinated decisions is then challenging. The degree of observability of the agents mainly influences the computational complexity of the multiagent decision-making problem. Intuitively, in a MAS, the most complex settings are those where the system state is jointly partially observable.

1.3 Uncertainty

Decision making under uncertainty has been studied for a while in AI and economics. It has recently been extended to distributed systems where each agent has to make decisions, in an autonomous way, from incomplete information. In most domains, agents have indeed to account for uncertainty about the environment. In fact, most of the time, the agents do not have enough information to get full knowledge about the current state of the environment and to predict exactly how the system will evolve (Parsons and Wooldridge, 2002).

Partial observability caused uncertainty on action outcomes since the agents are, most of the time, unable to exactly determine the state of the environment and the internal states of the other agents, i.e. the knowledge of the other agents, their preferences and their strategies. Uncertainty arising from partial observability is referred to as *state uncertainty* (Kochenderfer et al., 2015).

In addition, uncertainty may arise from imperfect modeling of the environment dynamics. Indeed, the environment may be too complex or unpredictable to determine in a deterministic way the issues

of an action. This type of uncertainty is referred to as *outcome uncertainty*. It has to be distinguished from *model uncertainty* as defined by Kochenderfer et al. (2015) where the dynamics and the rewards are not known and should be learned.

Different models have been proposed to represent uncertainty (Pearl, 1988). The most widely accepted and used framework is based on probabilities but other representations can be considered such as Dempster-Shafer belief functions, possibility measures or ranking functions (Halpern, 2003). In this document, we will focus on probabilities since they provide a powerful framework to reason about possible states of the world. Probabilities can be used to formalize *outcome uncertainty* by assigning a probability distribution to each couple state / action. Given a couple (s, a), a probability is assigned to each possible outcome of action a when it is executed from s. The probabilities of all possible outcomes for a couple (s, a) have to sum to 1.

Under partial observability, probabilities also provide a convenient way to represent and update the uncertainty about the state of the system. As an agent executes actions in the environment and gets new observations, she obtains new evidences about the possible states of the system and about the likelihood of each state. A *belief* of an agent can be represented as a probability distribution over the states of the system. The value associated to a state s represents the agent's assessment of the likelihood that the state s is the current state of the system.

1.4 Distributed decision-making under uncertainty and partial observability

In this document, we consider *rational agents*, i.e. agents that select, at any given time, the action maximizing their performance measure (Russell and Norvig, 2003; Kochenderfer et al., 2015). This implies that the agents are able to evaluate each available action and chooses the best one.The performance measure has to be designed regarding the objectives of the agents and the characteristics of the system.

In Economics, the utility theory proposes to represent the preferences of the agents in a numerical way (Neumann and Morgenstern, 1953; Fishburn, 1970). A utility function maps each possible state to a real number describing the satisfaction of the agent for this state. A rational agent would thus take, at any time, the action maximizing her utility.

In a MAS, several agents have to make decisions at the same time and the decisions of an agent i are influenced by the decisions of the other agents $j \neq i$. The decisions of the other agents $j \neq i$ are in turn influenced by the decision of i. Thus, each agent must reason about the other agents in order to anticipate their actions and to choose the best coordinated action to execute. Nonetheless, predicting the actions of the other agents is not that simple. Because of partial observability and uncertainty, it is a difficult problem to determine exactly the state of the environment and the internal states of the other agents. First, each agent is often uncertain or even unaware of the sequences of observations by the others. Second, each agent may not know the preferences of the other agents. In a cooperative setting, the agents share the same objectives. The preferences of the agents are thus common knowledge since the agents try to maximize a common performance measure. On the other hand, in the case of self-interested agents, each agent has her own preferences on the states. For privacy reasons, these preferences may not be known to the other agents. In such settings, it is even more difficult for an agent i to determine the actions of an agent $j \neq i$ since i is unable to predict the choices of $j \neq i$ even if her state is known.

One way to address coordination issues is to allow a central entity to plan the actions of the agents. Given a probabilistic model of outcome uncertainty, this central entity should be able to compute

coordinated strategies for the agents. Then, each agent would receive her individual strategy from the central coordinator and would be able to make autonomous decisions following this strategy (Ferber, 1999). Such approaches are usually used in multiagent resource allocation problems or multiagent decision making under uncertainty. Nonetheless, the computational complexity faced by the central coordinator remains high and computing optimal strategies is often untractable. In addition, the use of a central entity may not be possible or desirable. First, self-interested agents may doubt of the neutrality of the coordinator. Indeed, the coordinator could favor some agents. Second, such an approach requires many messages to be exchanged between the coordinator and the agents. Such communication may not be possible because of time and resource constraints or limitations of the communication infrastructure. Moreover, such approaches create a bottleneck in the system and a weak point. A failure of the coordinator would prevent all the agents to execute their actions. Finally, such systems are less flexible to system variations or unforeseen events. If some agents need to update their plans, the coordinator must be called upon.

An alternative approach is to provide the agents with sophisticated decision processes allowing them to take coordinated decisions in an autonomous and distributed way. This is the common approach developed in multiagent systems. In this document, we will more specifically focus on issues related to *partial observability* and *uncertainty* in *distributed decision-making*. In this context, we will study the extent to which agents can coordinate with other agents and make effective decisions based on partial observations of an uncertain environment.

We will investigate more specifically issues dealing with:

- *Formalizing the decision-making problem:* in order to solve real-world problems, decision-making frameworks must be able to deal with large and complex systems. The components of the decision problem must be adequately modeled. As mentioned previously, we will focus on probabilistic representations of the uncertainty on action outcomes. However, we will discuss appropriate representations of states, actions, observability and preferences. We will also study how *constraints on action execution* such as space, time and resource constraints, can be represented. Finally, we will focus on the *scalability of the models*, i.e. on the ability of the models to deal with large sets of agents, actions, states and observations.

- *Representing and inferring knowledge from observations:* different approaches can be envisioned to model the knowledge acquired by an agent along her observations of the environment. A possible approach consists in storing the whole history of observations. Nonetheless, such a representation quickly becomes untractable in real-world contexts. One major issue is thus to design compact models of knowledge that adequately summarize the information obtained by the agent. When a probabilistic representation of the uncertainty is available, belief states may be used to summarize the knowledge of an agent about the state of the environment[3]. Ideally, a compact representation should provide sufficient and complete information to make optimal solutions. However, the compactness of the model may comes at the price of a loss of optimality.

  Once the model of knowledge defined, efficient mechanisms to update the representation have to be developed. Updating the information about the environment may not be straightforward. In dynamic environments, evidences obtained from past observations may become inaccurate due to the evolution of the state of the environment. Given a new observation, an agent may be able to infer that a piece of her current knowledge is now incorrect without further details about the correct value of the information. For instance, a rescue-robot may observe that four civilians must be rescued in a building and then observes an ambulance unit leaving the building with an unobserved number of persons. The rescue robot should update the number of civilians to

[3] As we will explained later, this is the case of belief states in POMDP settings (see Chapter 3)

rescue in that building but she does not how. There are probably less than four civilians to rescue but we do not know exactly how many persons still remain.

- *Making efficient decisions under uncertainty:* we will also focus on algorithms and protocols allowing an agent to make coordinated and effective decisions based on her knowledge about the system. We will first consider myopic decision-making for distributed resource allocation. We will then turn to planning over finite horizon in uncertain and partially observable environments. Distributed decision making under uncertainty and partially observability is a hard problem (Bernstein et al., 2002b). Efficient algorithms exploiting the characteristics of the problems and/or searching for approximate solutions have thus to be developed. Various requirements may arise from the characteristics of the problems such as the need to compute strategies on-line (i.e. during action execution), to handle non-stationary environments (i.e. environments with evolving dynamics), to learn environment characteristics... Although these issues have already been the subject of much research work, we will investigate new directions and domains which have been the subject of little interest until now.

- *Exchanging knowledge about the system:* when the agents can communicate, they may share some information in order to increase their knowledge about the system. Although communication could drastically improve coordination and performances of decision-making, it often comes at a cost. The agents must therefore carefully decide when to communicate, to whom and which information should be sent. There is thus a need to evaluate the relevance of communicated information (in terms of the induced gain in performance for the agents, for instance) compared to the communication cost incurred.

- *Reaching consensus:* in uncertain and partially observable environments, agents may acquire different views of the system that may not be coherent (Bourgne et al., 2009). In a dynamic environment, agents may observe different values for a same information at different time steps. Moreover, agents may have obtained different pieces of knowledge. In such situations, abstract argumentation is a natural framework for the agents to debate about their knowledge and try to reach a consensus about the state of the environment or the strategy to undertake. When communication is limited, the number of time steps of the debate may not be sufficient for the agents to exchange all their knowledge. The agents have thus to strategically select the information (in this case the arguments) to exchange.

1.5 Overview of the document

Chapter 1 focuses on *distributed multiagent resource allocation*, i.e. resource allocation problems without any central coordinator. This widespread problem in multiagent systems has been mainly addressed from a centralized point of view. On the opposite, we will assume that each agent is initially endowed with a set of indivisible resources and the agents try to perform local swaps of resources in order to improve their satisfaction. Since agents have partial observability of the system, they are uncertain about the resources held by the other agents and their willingness to make some exchanges. The aim of this chapter is to study the consequences of distributing the allocation process. In particular, we will study the efficiency of the allocation outcomes. Different performance criteria will be investigated such as the utilitarian social welfare, the egalitarian social welfare or the envy-freeness. We will also investigate new notions of envy taking into account the partial observability of the agents. Finally, a special attention will be paid to computational complexity issues.

Chapter 2 is devoted to *planning issues for distributed decision-making under uncertainty and partial observability.* Decentralized Markov Decision Processes under Partial Observability (Dec-

POMDPs) provide a powerful mathematical model to formalize and solve such multiagent decision problems. However, they propose limited models of time and actions and they suffer from high complexity. In this chapter, we will address some of these limitations and we will propose models formalizing time, space and resource constraints. Because of the high complexity of planning under uncertainty and partial observability, we will investigate different techniques to handle the complexity of decision making such as searching for an approximate solution or exploiting the structure of the problem. Finally, we will consider non-stationary environments and study how observations can be exploited to take appropriate and effective decisions in such settings.

Chapter 3 focuses on *abstract argumentation* as a way for the agents to exchange information and to reach a consensus. In distributed systems, agents should be able to debate about their knowledge that may consist of local observations or private information. Such debates should result in better coordination between the agents. This entails that the agents make decisions on the arguments to put forward in the debate. Given a probabilistic model of the debating partners, the problem can be viewed as a sequential decision problem under uncertainty and partial observability. We will describe suited models to formalize these problems and we will study methods to efficiently compute argumentation strategies.

Each chapter starts with a brief recall of the relevant background required to understand the notions and issues addressed in the chapter. For more details, the reader may refer to the various pointers given in background sections. Each chapter ends with a discussion about the research directions opened in the chapter.

2 DISTRIBUTED RESOURCE ALLOCATION

Allocating some resources among a set of agents is crucial issue in many multiagent systems. Indeed, a wide range of application domains gives raise to MultiAgent Resource Allocation (MARA) problems. These include multi-robot task allocation (MRTA) in robotics scenarios such as rescue missions (Lerman et al., 2006; Scerri et al., 2005; Ferreira et al., 2010), allocation of schools, courses or rooms to students (Budish, 2011; Abraham et al., 2007b; Othman et al., 2010), division of goods in inheritance or divorce settlement (Brams and Taylor, 1996), management of computational resources in grid-computing (Galstyan et al., 2005), scheduling of manufacturing tasks (Sousa et al., 2000)... This topic has received a lot of attention from both Economics and Computer Science communities. The later is particularly interested in the computational aspects of MARA (Chevaleyre et al., 2006; Bouveret and Lang, 2008). The objective is to develop efficient allocation protocols. A particular interest is given to the study of the computational complexity. On the one hand, it is necessary to evaluate the computational resources required to represent and solve MARA problems. On the other hand, protocols and algorithms must be defined in order to efficiently compute an allocation. In fact, the designer often seeks for guarantees on the properties of the outcomes such as Pareto-efficiency or fairness while limiting the computational complexity of the protocol.

Solving MARA problems can be envisioned from a *centralized* or a *distributed* perspective. Centralized approaches rely on the existence of a central coordinator responsible for organizing the allocation of resources between the agents. In distributed approaches, agents autonomously negotiate over the resources and locally agree on deals. The outcome of the resource allocation problem then emerges from the sequence of local deals. Multiagent resource allocation has been mainly investigated from a centralized point of view. An important interest has been put on the design of centralized allocation procedures and on the study of the computational complexity (Shapley and Scarf, 1974; Beviá, 1998; Bansal and Sviridenko, 2006; Bouveret and Lang, 2008; de Keijzer et al., 2009; Asadpour and Saberi, 2010; Lesca and Perny, 2010; Dickerson et al., 2014; Bouveret and Lemaître, 2014; Aziz et al., 2016a; Bouveret and Lemaître, 2016). These procedures have the advantage of providing optimality guarantees or at least bounds on the quality of the solution. However, there are a number of arguments in favor of distributed approaches. First, the system may be inherently distributed and the use of a central coordinator may not be possible or desirable. A global coordinator must indeed be able to communicate with all the other agents, which is not always possible because of limitations in the communication infrastructure. In addition, the use of a central coordinator induces a weak point in the system: the coordinator causes a bottleneck whose default leads to the failure of the whole allocation process. Centralized approaches further requires that all agents communicate to the coordinator their preferences and their bundles of resources. This has a significant communication cost and may not be desirable for privacy reasons. Although its is typically more difficult to provide guarantees on the outcomes of distributed approaches, they exhibit a greater robustness and allow the agent to make autonomous decisions regarding the deals while having incomplete knowledge about the system.

Among the few existing approaches to distributed resource allocation, we can cite the approach of Netzer et al. (2016) based on distributed constraint optimization for minimizing the envy in the system. Another line of research initiated by Sandholm (1998) consists in departing from an initial allocation and allowing the agents to negotiate in order to improve their welfare (Endriss et al., 2003, 2006; Chevaleyre et al., 2007a, 2017). The agents then autonomously agree on local deals in order to improve their own utilities. Our work follows this line of research. Each agent is assumed to be

initially endowed with a set of indivisible resources and agents autonomously negotiate rational swaps of resources.

In this chapter, we consider rational and self-interested agents. Each agent thus takes her decisions in order to maximize her own satisfaction given her preferences about the states of the system. Here, the state of the system consists in the allocation of resources among the agents. The satisfaction value is referred to as the *welfare* of the agent (Moulin, 2003). Although the agents aim at maximizing some individual utility measures, the outcome of the allocation must also be assessed from a global point of view also called "social" point of view (Chevaleyre et al., 2017). We investigate various collective utility functions that aggregate individual utilities into a collective measurement. Notably, we will be interested in fairness measurements. Indeed, following some microeconomic principles and social choice theory (Sen, 1970; Rawls, 1971; Moulin, 1988), we assume that all agents should be considered as equal and should receive equal treatments.

Building on existing negotiation approaches, we study how distribution influences the efficiency of the allocation process and the desirable properties of the outcomes. When considering distributed resource allocation, agents naturally have partial observability of the system. They may obtain additional information as they encounter other agents but the individual knowledge of each agent often remains imperfect. We show that partial observability of the system leads to defining new measures of envy. Finally, we explore how the agents can improve their decisions with smart use of the observations about the system.

2.1 Research context

The work presented in this chapter arises from several collaborations. I started working on multiagent resource allocation with Sylvia Estivie and Nicolas Maudet. This work has been continued with Nicolas Maudet. In this context, we co-supervised master internships and the PhD of Anastasia Damamme. Complexity issues discussed in the chapter and the work dealing with social networks result from collaborations with colleagues from the LAMSADE (Université Paris-Dauphine): Yann Chevaleyre, Laurent Gourvès, Julien Lesca and Anaëlle Wilczynski. Finally, the work related to the relationships between our framework and picking sequences has been done in collaboration with Sylvain Bouveret and Michel Lemaître.

2.2 Background on resource allocation

2.2.1 *Resource allocation of indivisible goods*

Resource allocation problems can deal with either *divisible* or *indivisible* resources. A divisible resources can be divided into pieces allocated to different agents while an indivisible resource cannot. The problem of allocating divisible resources to agents is usually referred to as the *cake cutting* problem (Steinhaus, 1948; Brams and Taylor, 1996; Procaccia, 2009; Chen et al., 2013; Kurokawa et al., 2013).

In this document, we are interested in distributed decision making and negotiation between the agents. The objective is for the agents to negotiate exchanges of resources in order to improve their satisfaction. We will thus concentrate on indivisible resources and investigate how the agents can improve their satisfaction by exchanging resources.

Definition 1. ***MultiAgent Resource Allocation Problem - MARA***

An instance of a MultiAgent Resource Allocation (MARA) problem is defined as a tuple $\langle \mathcal{N}, \mathcal{R}, \mathcal{P} \rangle$ *where:*

- $\mathcal{N} = \{1, \cdots, n\}$ *is a set of agents,*
- $\mathcal{R} = \{r_1 \cdots, r_m\}$ *is a set of resources (or items),*
- $\mathcal{P}$ *is a profile of preferences representing the interest of each agent* $i \in \mathcal{N}$ *towards the resources.*

An allocation A is mapping of the resources in $\mathcal{R}$ among the agents in $\mathcal{N}$. A_i stands for the set of items held by agent i. We assume that a resource cannot be shared by several agents ($A_i \cap A_j = \emptyset$, $\forall i \in \mathcal{N}, j \in \mathcal{N}$ such that $i \neq j$). Moreover, each resource is allocated to a least one agent ($\cup_{i \in \mathcal{N}} A_i = \mathcal{R}$). An allocation is thus a partitioning of $\mathcal{R}$ among $\mathcal{N}$.

When $|\mathcal{N}| = |\mathcal{R}|$ (*i.e.* $n = m$) and each agent receives exactly one resource, the allocation problem corresponds to a *matching* or *house-allocation* problem (Shapley and Scarf, 1974; Roth and Sotomayor, 1992; Abraham et al., 2005).

2.2.2 *Preference representation and domain restriction*

There are different ways to represent the preferences each agent has over resources:

- *ordinal preferences*: the preferences of each agent i consist of a binary relation $\succ_i$ over the bundles of resources. $\mathcal{L}_1 \succ_i \mathcal{L}_2$ means that agent i prefers the bundle $\mathcal{L}_1$ to the bundle $\mathcal{L}_2$. This relation is transitive and reflexive. The resulting ordering of all possible bundles by agent i may be partial or complete. The preference profile $\mathcal{P}$ of the agents is then defined as $\{\succ_1, \cdots, \succ_n\}$.
- *cardinal preferences*: the preferences of each agent i are modeled using a utility function $u_i : 2^{\mathcal{R}} \rightarrow \mathbb{R}$ mapping each possible bundle of resources to a real value. The preference profile $\mathcal{P}$ of the agents is then defined as $\{u_1, \cdots, u_n\}$.

A simple way to translate a linear order over the bundles into cardinal preferences is to use Borda scores (Baumeister et al., 2014). A utility is thus assigned to each bundle based on its position in the preference ordering. Let $|\mathcal{L}|$ be the number of possible bundles. The scores are integer values ranging in $[1, |\mathcal{L}|]$. The most preferred bundle of resources obtains the highest score ($|\mathcal{L}|$), and the worst bundle of resources is valued to 1.

Preference representation languages

Since the number of possible bundles is exponential in the number of resources, it is often unrealistic to enumerate the values of each bundle or to give a full ordering of the bundles. A wide range of researches have focused on providing compact preference representation languages. The idea is to provide a language to represent the preferences over the bundles of items in a reasonable size (Brandt et al., 2016). This topic is not restricted to the resource allocation domain and, more broadly, deals with representing preferences over a combinatorial set of alternatives. Among the most widespread languages, we can mention bidding languages initially developed for combinatorial auctions, graphical models, logic-based languages and additive utility functions. For a more detailed discussion on these languages, we refer the interested reader to Chapter 12 of (Brandt et al., 2016).

In this document, as soon as we consider cardinal utilities, we will focus on *additive preferences* (Lipton et al., 2004; Procaccia and Wang, 2014; Dickerson et al., 2014; Caragiannis et al., 2016; Bouveret and Lemaître, 2016). Additive functions provide a compact and commonly used representation that can be easily elicited to formalize the agent's preferences over bundles of resources. The utility of an agent i for an allocation A_i is then defined as the sum of the utilities over the resources forming A_i:

$$u_i(A_i) = \sum_{r_i \in A_i} u_i(r_i)$$

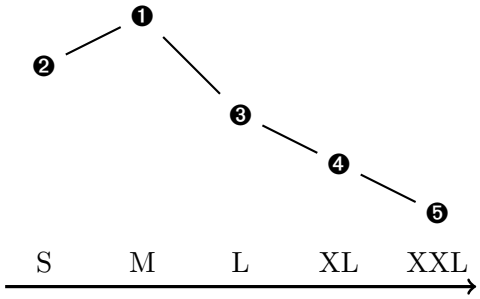


Figure 2: Single-peaked preferences

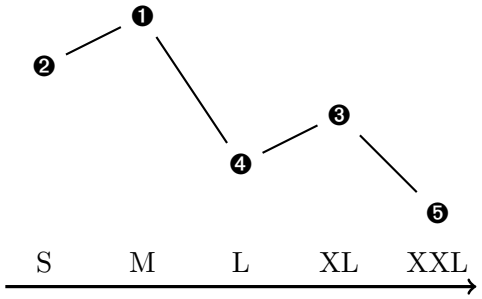


Figure 3: Non single-peaked preferences

Here, we assume that there is no externalities i.e. the utility of an agent does not depend on the distribution of the remaining resources on the other agents. We can thus alleviate the notation $u_i(A_i)$ using $u_i(A)$ without misunderstanding.

It has to be noticed that additive functions do not allow for representing synergies between the resources such as super-modularity and sub-modularity. *k-additive* functions extends additive utility functions to formalize such synergies between set of resources of size at most k (Grabisch, 1997; Chevaleyre et al., 2008). As the value of k increases the value function becomes more and more expressive but less and less concise (as proved by Chevaleyre et al. (2008), any utility function can be represented as a k-additive function with $k = |\mathcal{R}|$).

Domain restriction

In some domains, assumptions about the agents' preferences can be identified, thus restricting the set of possible preference orderings. The most studied domain restriction is probably *single-peaked* preferences. Single-peaked preferences were initially described in social choice for studying voting rules (Black, 1948; Arrow, 1951). This restriction deals with ordinal preferences on single resources. Let $\rhd = r_1 \rhd r_2 \rhd \cdots \rhd r_m$ be an axis over the resources i.e. a linear order. Let $top(i)$ denote the most preferred resource of agent i. In single-peaked domains, the preferences of an agent i are related to the relative positions of the resources to $top(i)$ on the axis $\rhd$.

Definition 2. *Single-peaked preferences (Escoffier et al., 2008; Endriss, 2017)*

The preferences $\succ_i$ of an agent i are said to be single-peaked with respect to $\rhd$ if $\forall r_j, r_k \in \mathcal{R}$ such that r_j is closer to $top(i)$ than r_k in $\rhd$, we have $r_i \succ_i r_j$. More formally, $\succ_i$ is said to be single-peaked with respect to $\rhd$ if $\forall r_j, r_k \in \mathcal{R}$ such that $top(i) \rhd r_j \rhd r_k$ or $r_k \rhd r_j \rhd top(i)$ we have $r_j \succ_i r_k$.

Definition 3. *Single-peaked preference profiles*

A preference profile $\mathcal{P} = \{\succ_1, \cdots, \succ_n\}$ is said to be single-peaked if all the agents share a common axis $\rhd$ on resources such that each agent's preferences are single-peaked with respect to $\rhd$.

This restriction arises naturally in various preference domains where some characteristics of the resources inherently define a common axis: political ideologies for candidates to an election (Bruner and Lackner, 2015), distances to downtown for hotels in a city, storage capacities for hard-disks, sizes for clothes...

Figures 2 and 3 illustrate two preference orders for tee-shirt sizes. The preferences described in Figure 2 are single-peaked with size M being the peak. The preferences described in Figure 3 are not single-peaked since the peak is size M but size XL is preferred to size L.

Culture and preference generation

In order to perform some experimental evaluations of resource allocation protocols, it is required to consider various preference profiles for the agents. These preferences can be extracted from real data or be automatically generated.

PREFLIB Several tools have been developed over the last decade to facilitate experiments in computational social choice. Notably, the PrefLib library provides a large number of preference data-sets extracted from real data (Mattei and Walsh, 2013). PrefLib gathers election, matching and rating data. Matching data provide problems where the agents have preferences over some items (and vice-versa). The objectives is to pair agents to items. Preferences from the rating data-set and the election data-set contains various forms of ranking. In our context, we are more specifically interested in profiles providing a complete order over the items.

CULTURE AND RANDOM GENERATION When profiles have to be automatically generated, preferences are randomly drawn from a particular distribution. This probability distribution is usually referred to as the *culture* (Mattei, 2011).

If we consider ordinal preferences, a linear order of the resources has to be drawn. The less restrictive and simplest culture is the *impartial culture*. It was initially introduced in social choice theory to characterize the preferences of voters among candidates (Black et al., 1958; Gehrlein and Fishburn, 1976). Under the *impartial culture*, the uniform probability distribution is used: all preference orders are equally likely and are chosen independently. Under the single-peakedness assumption, different methods can be envisioned to generate single-peaked preferences. Conitzer (2009) proposed to first randomly draw the position of the resources on the axis, all alternatives being equally likely. Then, for each agent, a peak is randomly chosen with equal probability. The second-highest ranked resource is chosen with equal probability from the two adjacent alternatives, and so on until a full order is obtained. Given an axis, Walsh (2015) proposed a recursive procedure building single-peaked preferences from the end (i.e. the worst resource of the agent) to the top resource.

In order to generate cardinal preferences, we need to assign a utility value to each bundle of resources. If we consider additive value functions, this process consists in the assignment of a utility value to each resource. Given an interval $[min, max]$ defining acceptable utility values of a resource, Dickerson et al. (2014) described two ways of generating a profile of preferences $\mathcal{P}$:

- Uniform distribution: for each agent, the utility value of a resource is drawn in the interval $[min, max]$ from a uniform distribution. Preferences under the impartial culture are thus obtained.

- Correlated distribution: for each resource r_i, an intrinsic value μ_i is drawn in the interval $[min, max]$ from a uniform distribution. For each agent, the utility value of r_i is then drawn from the truncated normal distribution with μ_i being the mean of the distribution. The variance σ of the distribution is defined as an input of the method. This method allows for representing dependencies between the preferences of the agents for a same resource. In the extreme case, if $\sigma = 0$, all the agents have the same utility μ_i for a same resource r_i.

2.2.3 *Negotiation of rational deals*

Given an instance of a MARA problem, the objective is to find an allocation A that satisfies some desired properties or maximizes a performance criterion. As mentioned previously, we are interested in distributed procedures to allocate the resources among the agents. In such settings, each agent is

assumed to be initially endowed with a set of resources. The agents then negotiate some deals in order to improve their welfare.

Definition 4. ***Deal***

A deal is defined as a pair $\delta = (A, A')$ *where* A *and* A' *are some allocations of resources such as* $A \neq A'$.

In the following, $\mathcal{N}^\delta$ denotes the agents involved in the deal δ.

TYPES OF DEALS Different types of deals have been investigated in the literature (Sandholm, 1998). Endriss et al. (2006) proposed to characterize the set of deals that are allowed to the agents by both structural and rationality constraints.

Structural constraints define the number of agents and the number of resources involved in the deal. A type of deal can thus be characterized by a pair (na, nr) where na denotes the number of agents involved in the deal and nr is the maximum number of resources exchanged in a deal by each of the na agents. Note that it is often assumed that all the agents exchange the same number of resources so, nr is the exact number of resources exchanged by each agent.

The most studied types of deals are:

- *1-deals* or $(1, 1)$-deals where a single resource is passed from one agent to another,
- *swap deals* or $(2, 1)$-deals where two agents exchange one of their resources,
- *bilateral deals* or $(2, *)$-deals where two agents exchange a subset of their resources, the number of resources exchanged being unfixed[1],
- *k-deals* or $(k, 1)$-deals where k agents are involved in the deal and each agent exchanges only one resource.

Bilateral and swap deals are easy to implement since they involve only two agents and thus do not require the coordination of many agents. Moreover, they often fit with constraints of real-world environments where agents cannot negotiate with all the other agents at all time because of space constraints (exchanging physical resources requires the agents to be close to each other) or limited communication range (agents have to be close enough to each other in order to communicate and to negotiate some deals).

When more than two agents are involved in a deal, it may be desirable to be able to decompose the deal into a sequence of bilateral deals. The agents involved in the deal are then ordered such as the first agent of the sequence gives her item to the second one, the second one gives her item to the third one, and so on until the last agent gives her item to the first one. Such deals are called *cycle-deals*. Each step of a cycle-deal requires the presence and the coordination of only two agents. These exchanges are thus easier to implement. The relevance of such exchanges can be exemplified by barter-exchange markets or by kidney exchanges where patients can obtain compatible donors by exchanging their own wiling but incompatible living donors with other patients (Abraham et al., 2007a).

In this context, a *k-cycle-deal* refers to a cycle-deal involving k agents where each agent of the cycle exchanges a resource with the next agent in the sequence. *k-cycle-deals* can be generalized to (k, l) cycle-deals where each agent exchange at most l resources. A long cycle may be undesirable in practice since it significantly increases the likelihood of failure of the whole deal. The length of the cycle is thus often bounded by a small integer value.

[1] Swap-deals are thus a restriction of bilateral deals.

MONETARY SIDE PAYMENTS The transfer of a resource from one agent to another can be balanced by another resource and/or a monetary side payment. Such monetary transfers can compensate disadvantageous deals. MARA problems with monetary payments have been studied under cardinal preferences (Sandholm, 1998; Endriss et al., 2003; Estivie et al., 2006; Chevaleyre et al., 2007a, 2017). However, such monetary transfers may be impossible or not desirable. For instance, they may be forbidden by the law as in the case of kidney exchanges (Abraham et al., 2007a). Moreover, as pointed out by Endriss et al. (2003), this framework assumes that each agent has an unlimited amount of money allowing her to pay for any deal.

In the following of the document, we will not allow for monetary compensation. The welfare of an agent will thus only rely on the utility of her resources. In such contexts, when an agent gives a resource, she must receives another resource to compensate the loss of utility.

RATIONAL DEALS In order a (self-interested) agent to decide if a deal is acceptable or not, the first requirement is the deal to be *individually rational* i.e. the agent is better off if the deal is performed. Notions related to rationality can be defined under both cardinal and ordinal preference relations. In this document, we will focus on cardinal settings and we will omit definitions under ordinal preference relations.

Definition 5. ***Strictly Improving Deals***
A deal $\delta(A, A')$ is said to be strictly improving for an agent i iff $u_i(A) < u_i(A')$.

This definition can be relaxed to consider "neutral" deals i.e. deals that do not change the utility of the agent.

Definition 6. ***Weakly Improving Deals***
A deal $\delta(A, A')$ is said to be weakly improving for an agent i iff $u_i(A) \leq u_i(A')$.

The definition of a rational deal follows directly from previous definitions.

Definition 7. ***Individual Rational Deals***
A deal $\delta(A, A')$ is said to be individually rational (IR) for an agent i iff the deal is strictly improving.

Individual rationality is a self-interested concept and does not account for the impact of the deal among the other agents. Given an allocation A, several individual rational deals may be possible from the point of view of an agent i (even if we limit the types of deals allowed). The agent will thus have to make a choice between the different possible individual rational deals. One requirement is the deal to be rational for all the agents involved in the deal otherwise the other agents will refuse to make the exchange. For a deal to be accepted by all the agents involved, it must be *cooperatively rational*.

Definition 8. ***Cooperative Rational Deals***
A deal $\delta(A, A')$ is said to be cooperatively rational (CR) iff it is weakly improving for all the agents and strictly improving for at least one agent.

More formally, a deal $\delta(A, A')$ is said to be cooperatively rational iff $\forall i \in \mathcal{N}^\delta$, $u_i(A) \leq u_i(A')$ and $\exists j \in \mathcal{N}^\delta$ such that $u_j(A) < u_j(A')$.

It has to be noticed that the definition of rationality is *myopic* since the agents only consider the immediate effects of the deal on their utilities and do not take into account the effects of the deal on future opportunities. The agents thus do not plan ahead the deals they will accept in future steps.

Moreover, individual and cooperative rationality are *self-interested* notions where the agents try to maximize their own *individual* utility. When an agent proposes or accepts a deal, she does not take into account the consequences of the deal on the agents not involved in the deal. Some deals may prevent the other agents from exchanging their resources and may lead to sacrify some of the agents. In order to evaluate MARA procedures, it is thus important to consider the well-being of the society of agents i.e. to study the quality of the final allocation (outcome) from a global point of view.

2.3 Well-being of the society

The *social welfare* formalizes the well-being of a society given the individual preferences of its members. Social welfare comes from Welfare Economics and Social Choice (Sen, 1970; Moulin, 1988; Arrow et al., 2002). Different measures of social welfare can be envisioned according to the principles of the society under consideration (Endriss and Maudet, 2004). Two different perspectives can be considered to assess the well-being of a society regarding an allocation of resources: the *efficiency* perspective is interested in maximizing a global utility function, while the *fairness* perspective is governed by egalitarian principles.

Under cardinal preferences, a common way to define social welfare consists in defining a collective utility function that aggregates the individual utilities of the agents.

2.3.1 *Efficiency of the allocation*

Following the utilitarian theory[2], an efficient allocation is an allocation maximizing the sum of the utility. The welfare of a society is thus measured by the sum of the individual utilities.

Definition 9. ***Utilitarian Social Welfare***

The utilitarian social welfare $sw_u(A)$ of an allocation A is defined as:

$$sw_u(A) = \sum_{i \in \mathcal{N}} u_i(A)$$

Utilitarian social welfare is sensitive to scale differences in utilities and assumes that the individual preferences of the agents are defined on the same scale. Such metrics are said to be scale dependent.

Under additive utility functions, optimizing the utilitarian social welfare in a centralized way is quite easy and consists in assigning each resource to the agent that values it the most. However, it may lead to very unbalanced allocations where a subset of the agents get all the resources and the others get no resource. Capacity constraints can then be envisioned to fix the number of resources per agent. When the number of resources equals the number of agents and each agent must receive a resource, one can simply translate the problem to a (weighted) matching problem in a bipartite graph. Agents are only matched to objects they prefer to their current assignment, with weights corresponding to their utility for each resource. This can be solved by standard techniques in $O(n^3)$. When each agent must be assigned exactly k resources, the problem can also be reduced to a matching problem by duplicating each agent into k agents with the same preferences. In the general case (no restriction on the additivity of the utility functions), optimizing the utilitarian social welfare in a centralized way is NP-complete (Dunne et al., 2005; Chevaleyre et al., 2008).

It is important to note that, in decentralized systems, the optimal allocation may not be reachable by a sequence of cooperative bilateral rational deals: every sequence of deals allowing to obtain the optimal solution may incur a loss in utility for at least one agent.

Example 1. ***Reachability of the allocation maximizing social welfare***

Let's consider an instance involving 4 agents and 4 resources (one resource per agent). Boxes identify the resource held by each agent.

	r_1	r_2	r_3	r_4
agent 1	6	3	2	1
agent 2	1	4	3	2
agent 3	2	1	4	3
agent 4	4	2	1	3

[2] Earliest principles of utilitarianism were developed by Jeremy Bentham (1748–1832).

The underlined allocation maximizes the utilitarian social welfare. However, it is not reachable from the current (boxed) allocation by a series of rational deals. In fact, agent 4 should give r_1 to agent 1 which is not individually rational from her point of view for any resource she would receive (recall that monetary side payments are not allowed).

2.3.2 *Pareto-optimality*

When the optimal solution regarding the utilitarian social welfare cannot be reached, a weaker requirement consists in searching for a solution where no agent can improve her allocation without incurring a loss on at least another agent. This notion corresponds to *Pareto-optimality* (Moulin, 1988; Arrow et al., 2002).

An allocation is said to be Pareto optimal if and only if there is no other allocation A' that is not worse for all the agents and is strictly better for at least one agent.

Definition 10. *Pareto-optimality*
An allocation A is said to be Pareto optimal iff there is no other allocation A' such that:

$$\forall i \in \mathcal{N}, u_i(A') \geq u_i(A) \text{ and}$$
$$\exists j \in \mathcal{N}, u_i(A') > u_i(A))$$

Endriss et al. (2003) proved that any sequence of cooperative rational deals will eventually result in a Pareto optimal allocation of resources. This result supposes that there is no restriction on the types of deals that can be implemented by the agents. In fact, they also proved that all possible cooperatively rational deals must be allowed in order to be able to guarantee a Pareto-optimal outcome of a negotiation.

The main disadvantage of this criterion is that it is not very selective. Indeed, the number of Pareto-optimal solutions can be large and there may be significant disparities between the agents according to the envisioned Pareto-optimal solution. Under additive preferences, testing whether an allocation is Pareto-optimal is CoNP-complete (de Keijzer et al., 2009). Moreover, Aziz et al. (2016a) proved that in the presence of an initial endowment, finding an individually rational and Pareto optimal improvement is NP-hard.

2.3.3 *Egalitarian Social Welfare*

Egalitarian theories state that all people should be treated as equal. The issue of redistributing resources in a society has been widely studied in egalitarianism. The egalitarian social welfare function (also called Rawlsian welfare function) stems from the theory of justice proposed by Rawls (1971). Rawls defined the equality principle to characterize fairness. This equality principle states that every member of the society should be given the same utility. However, there may not exist feasible solution satisfying this principle. Rawls thus also introduced the difference principle that aims at reducing the inequalities. This principle states that the welfare of a society should be measured by the welfare of the worst-off member of the society. The egalitarian social welfare of an allocation A is thus defined as the minimum of the utility of the agents.

Definition 11. ***Egalitarian Social Welfare***
The egalitarian social welfare $sw_e(A)$ of an allocation A is defined as:

$$sw_e(A) = \min_{i \in \mathcal{N}} u_i(A)$$

Following Rawls' theories, an egalitarian allocation should maximize the egalitarian social welfare. In computational social choice, this solution is usually referred to as a *maximin* share. Like the utilitarian social welfare, this measure is scale dependent and requires all the agents to use the same utility scale. Furthermore, the egalitarian social welfare does not allow for deciding among several allocations where the worst-off agent has the same utility. In order to discriminate allocations leading to the same egalitarian social welfare, the *leximin ordering* can be used (Moulin, 1988).

2.3.4 *Nash social welfare*

Under positive utility functions, the Nash collective utility function (Nash, 1950) provides a good trade off between the utilitarian and the egalitarian social welfare. The Nash social welfare is defined as the product of the individual utilities of the agents.

Definition 12. ***Nash Social Welfare***
The Nash Social Welfare $sw_n(A)$ of an allocation A is defined as:

$$sw_n(A) = \prod_{i \in \mathcal{N}} u_i(A)$$

The Nash Social Welfare is appealing since it takes into account the average utility of the agents and also favors balanced utility distributions i.e. equality among the agents.

2.3.5 *Envy-freeness*

When the agents can compare their shares, it is relevant to consider other notions of fairness such as the absence of envy (Foley, 1967; Feldman and Kirman, 1974). An agent i would envy another agent j if she prefers the share of j to her own share. More formally,

Definition 13. ***Envy***
An agent i envies an agent j iff

$$u_i(A_j) > u_i(A_i)$$

An agent is said to be envious if she envies at least one other agent. This criterion is scale independent. Note that an agent does not need to know the others agents' utility functions in order to determine whether she is envious or not but she has to know the allocation A i.e. how the resources are allocated among the other agents. A completely fair allocation would thus be an *envy-free allocation* i.e. an allocation where no agent is envious. However, as soon as it is required to allocate all the resources of the system (completeness requirement), an envy-free allocation may not exist.

Example 2. ***Absence of envy-free allocation***
Let's consider a simple instance involving 2 agents and 2 resources where both agents have the same preferences

	r_1	r_2
agent 1	7	$\boxed{2}$
agent 2	$\boxed{7}$	2

Under completeness requirement there is no possible envy-free allocation: one of the agent will always envies the other one (the one that holds r_1).

The only way to avoid envy is to not allocate any resource to any agent but this is quite inefficient.

Lipton et al. (2004) proved that deciding whether an envy-free allocation exists under additive preferences is NP-complete. It is important to note that an envy-free allocation may be Pareto-dominated. The reverse is also true: a Pareto optimal allocation may not be envy-free. Given a MARA instance, deciding whether there exists an allocation that is both envy-free and Pareto-optimal, is $\sum_2^p$-complete (de Keijzer et al., 2009).

An alternative objective consists in minimizing the envy. Different metrics have been investigated to measure the degree of envy of a society (Lipton et al., 2004; Chevaleyre et al., 2007a; Caragiannis et al., 2009; Cavallo, 2012; Nguyen and Rothe, 2014). Chevaleyre et al. (2007a) define the degree of envy at three different levels:

- the *Bilateral envy* measures the envy of an agent i towards another agent j:

$$e_{ij} = \max(u_i(A_j) - u_i(A_i), 0)$$

  It is also possible to consider a boolean measure saying whether i is envious or not of j.

- the *Individual envy* measures the envy of an agent i towards all the other agents. It aggregates the bilateral envy of i among all the other agents. Different operators can be used such as the `sum` or the `max`:

$$e_i^{max} = \max_{j \in \mathcal{N}}(e_{ij}) \text{ or } e_i^{sum} = \sum_{j \in \mathcal{N}}(e_{ij})$$

  It is also possible to count the number of agents that i envies.

- the *global envy* of the society aggregates the individual envies of the agents in the society. A wide variety of operators can be considered. The most used ones are the `max` operator that focuses on the most envious agent of the society or the `sum` operator that considers the average envy among the society:.

$$e^{max} = \max_{i \in \mathcal{N}}(e_i) \text{ or } e^{sum} = \sum_{i \in \mathcal{N}}(e_i)$$

  where e_i corresponds to an individual envy measures (the measure being the same for all the agents). Note that this measure is scale dependent.

2.3.6 *Proportionality*

A less demanding fairness criterion is *proportionality*. It was first introduced in cake-cutting problems (Steinhaus, 1948). The allocation received by an agent i is said to be proportional if the utility of the agent is greater or at least equal to the utility it would receive from a virtual perfectly equitable allocation.

Definition 14. ***Proportionality***
An allocation A is proportional iff $\forall i \in \mathcal{N}$:

$$u_i(A) \geq \frac{u_i(\mathcal{R})}{n}$$

This measure can be computed in a distributed way and does not require the agent to know the bundles of the other agents. Nevertheless, a proportional allocation is not guarantee to exist. In Example 2, $u_i(\mathcal{R}) = 9$ for both agents and there is no complete allocation giving at least 4.5 to each agent. Moreover, deciding whether a MARA instance admits a proportional allocation is NP-complete (Bouveret and Lemaître, 2014).

2.4 Distribution and related issues

NEGOTIATION PROTOCOLS For the agents to be able to autonomously agree on local swap deals, there is a need for *negotiation protocols* defining how the agents interact to exchange resources. More specifically, a protocol describes the behaviors and the communication messages allowing each agent to come into contact with the other agents, to exchange information, to agree on deals and finally to perform the deals. Although the agents are assumed to perform cooperative rational deals, we can question whether the agents would reach a stable state and what would be the quality of this outcome whether in terms of efficiency or fairness.

Indeed, when designing a protocol for distributed resource allocation, a major requirement is to provide *termination guarantees*. In fact, it must be ensured that the agents will eventually converge to stable state where no more exchange is possible and where the agents can safely quit the allocation process.

Another requirement is the protocol to be efficient. The efficiency of the protocol concerns its *computational and communication complexity* but it has also to take into account the *quality of the final allocations* obtained using the protocol. Indeed, guarantees on the quality of the outcome are usually desired. One possible requirement is to guarantee convergence towards a Pareto-optimal solution. Efficiency or fairness of the outcome may also be required. Although, distributed protocols usually do not give optimality guarantee, some bounds on the gap between the worst-case outcome and the optimal one can be defined (Koutsoupias and Papadimitriou, 1999).

ASYNCHRONISM AND CONCURRENCY Although they consider distributed allocation mechanisms, most existing frameworks make the implicit assumption of synchronization i.e. only one deal is performed at a time in the system. In fact, in a distributed system, each agent has her own execution thread and the agents act in an asynchronous manner. The protocol must therefore handle concurrent encounters and offers, communication delay, etc. Similar issues have been investigated in extensions of the Contract Net Protocol to account for contingency contracts (Sandholm and Lesser, 2001; Aknine et al., 2004). The protocols that we will describe in the following will take into account concurrency issues.

PARTIAL OBSERVABILITY AND DYNAMICITY Since the agents exchange resources upon encounters, the allocation of resources evolves along the time. In a distributed allocation process, it is natural to assume that each agent has limited visibility of the system and only observes part of the whole allocation. Indeed, the agents can be physically distant and have limited perception capabilities. Furthermore, broadcasting the preferences of the agents and their individual allocations may not be desirable for privacy reasons or because of the high communication cost. In this document, we make minimal assumptions about the degree of observability of the agents. We consider that each agent only observes the resources held by another agent upon encounters. Preferences are assumed to be private and thus non-observable information.

Partial observability has been little studied in distributed resource allocation. In fact, common measurements of efficiency and fairness requires full and perfect knowledge of the allocation. The only notable exception is proportionality that is defined in terms of an agent's own value for her bundle. In the following, we will investigate how each agent can assess the fairness of an allocation based on an incomplete and may be incorrect view of the system.

2.5 Distributed house-allocation

We start our discussion about distributed resource allocation by addressing *one-sided matching problems* also known as house allocation problems, or house markets. In this context, each agent is assumed to initially (and all over the process) hold exactly one resource. A deal corresponds in the swap of a single resource against another single resource.

2.5.1 *Properties of the procedures*

The Top Trading Cycle (TTC) is a recognized procedure for solving house allocation problems when each agent starts with an initial endowment (Shapley and Scarf, 1974). TTC returns a unique solution belonging to the core. In other words, in the final allocation, no coalition of agents can make all of its members better off by exchanging the items they initially own in a different way. Moreover, the procedure is Pareto-efficient, individually rational, and strategy proof (no agent has an incentive to misrepresent her preferences). TTC is the only mechanism presenting all these guarantees (Ma, 1994). However, TTC is a centralized procedure that is not well suited in our context. Furthermore, the cycle-deals that must be performed to reach the final allocation may involve a large number of agents (in the worst case, all the agents). Such long cycles may not be desirable as they require the coordination of many agents which may be problematic. Although TTC is not well suited to distributed contexts, it provides appealing guarantees that would be desirable in a distributed processes. In this section, we investigate desired properties of distributed procedures.

Convergence guarantees

In a distributed context, we can question the guarantees provided by the protocols regarding the outcomes of the procedures.

In fact, convergence guarantees can be stated regardless of the number of resources per agent and of the number of agents involved in a deal.

Proposition 1. *If the number of deals is finite and each deal is cooperatively rational, the system will eventually reach a stable state where no more deal is possible.*

The proof of this proposition strictly follows the proofs of convergence provided in (Sandholm, 1998; Endriss et al., 2006).

The definition of stability can be refined to take into account restrictions on the types of deals allowed to the agents. Let first denote by C_k the class of cycle-deals involving at most k agents.

Definition 15. *An allocation is k-stable when no C_k rational deals are possible.*

Proposition 1 still holds when we restrict the class of possible deals to C_k in house-allocation problems. However, some outcomes may not be reachable anymore since they require the agents to perform deals that cannot be reduced to k-cycle-deals.

Efficiency and Fairness of the outcome

Once convergence of the distributed allocation protocol is guaranteed, it is relevant to study the efficiency and the fairness of this stable outcome.

We will be interested in the following problems:

- Can we guarantee Pareto-optimality of the outcome?

- Is the outcome efficient regarding utilitarian social welfare?
- What is the cost of distributing the allocation process?
- Is the outcome fair regarding egalitarian social welfare?

It is important to note that envy-freeness is of little interest in house allocation settings. In fact, an agent necessarily envies another agent as soon as she does not get her top object. More specifically, an agent envies all the agents holding a resource preferred to the resource she currently holds.

In this document, we will more specifically focus on the simplest version of cycle-deals: *bilateral deals* i.e. deals involving exactly two agents (also denoted as C_2). Besides being simple and not requiring the coordination of a large number of agents, we will show that this class of exchanges can achieve good performance.

2.5.2 *Dynamics of bilateral deals*

We start by investigating the dynamics based on rational bilateral swaps of resources. The agents thus make local bilateral exchanges until they reach a stable allocation. When several deals are possible, one of them is randomly chosen. Unlike centralized procedures (as TTC), the allocation process may lead to different outcomes, depending on the sequences of encounters among the agents. In fact, our theoretical results are independent of the decision rule used to select a possible deal. Nonetheless, we will have to implement a precise decision rule for experimental purpose.

2.5.3 *Pareto-optimality*

We first investigate the efficiency of an allocation in terms of Pareto-optimality. It is possible to test Pareto-optimality using TTC. Since TTC returns a Pareto-optimal solution and respects individual rationality, one can run TTC from an allocation A and compare it with the allocation A' returned by TTC. If $A \neq A'$ then, A is not Pareto-optimal.

A Pareto-optimal solution can be thought as an allocation where no more cooperative rational deal is possible. A Pareto-optimal allocation is thus k-stable. In house-allocation settings, the reverse is also true iff $k = |\mathcal{N}|$.

Proposition 2. *Any sequence of k-deals reaches a Pareto-optimal allocation.*

It has to be noticed that this proposition only applies in house-allocation problems. We will discuss its generalization later in this chapter.

Restricting the size of the cycles leads to less possible exchanges. If $k < |\mathcal{N}|$, it is thus obvious that a k-stable allocation may not be Pareto-optimal. More specifically, this observation applies to bilateral-deals ($k = 2$) if the system involves more than 2 agents.

Example 3. ***Reachability of a Pareto-optimal allocation***
Let's consider an instance involving 4 agents and 4 resources. Boxes identify the resource held by each agent.

	r_1	r_2	r_3	r_4
agent 1	$\underline{4}$	$\boxed{3}$	2	1
agent 2	1	$\underline{4}$	$\boxed{3}$	2
agent 3	2	1	$\underline{4}$	$\boxed{3}$
agent 4	$\boxed{3}$	2	1	$\underline{4}$

The allocation where each agent obtains her top-resource (underlined allocation) maximizes the utilitarian social welfare and is Pareto-efficient. Moreover, the boxed allocation is not Pareto-efficient since it is Pareto-dominated by the underlined allocation. However, the underlined allocation is not reachable (from the boxed allocation) by a series of bilateral rational deals. Indeed, the agent 1 *would only accept a deal where she obtains* r_1 *but the agent* 4 *holding* r_1 *would only accept a deal where she obtains* r_4*. The exchange is thus not possible. The underlined allocation can be reached by a cycle deal involving all the agents where each agent gives her resources to the next one in the sequence of agents* $(1, 2, 3, 4)$*.*

Nonetheless, this negative results may be alleviated if we restrict the preference domain. In (Damamme et al., 2015), we proved that a positive result can be obtained if the preferences are single-peaked.

Proposition 3. *In a single-peaked domain, any sequence of bilateral deals reaches a Pareto-optimal allocation.*

Although distributed procedures are not guarantee to converge to a Pareto-optimal allocation, a natural question is to estimate the proportion of 2-stable allocation which are Pareto-optimal. In (Damamme et al., 2015), we showed that under impartial culture, with $n = 10$, there is about 75% chance to reach a Pareto-optimal allocation, while with $n = 14$ there is still 50% chance to reach a Pareto-optimal allocation. From $n = 30$, it becomes almost impossible. For comparison, we also tested the Pareto-optimality on real world instances of PrefLib and we obtained better results. This seems to suggest that correlation between preferences is favorable to convergence to Pareto-optimal allocations.

2.5.4 *Utilitarian social welfare*

Although the probability of reaching a pareto-optimal allocation using swap-deals is quite low, we can question the effectiveness of the allocations and their fairness. We first focus on the worst-case study and then study average performance experimentally. Note that, often, this will involve to interpret cardinally the ordinal preferential information provided by agents. We shall simply use the *rank* as a measure of satisfaction, and thus assign —using a Borda count— some utility to ranks, *i.e.* $u_i(A(i)) = n$ when a_i gets her preferred object, $n-1$ when she gets her next preferred object, and so on.

Worst-case analysis

In order to conduct a worst-case analysis of distributed procedures, we used two measures that are the *price of anarchy* and the *price of short cycles*.

PRICE OF ANARCHY The price of Anarchy (PoA) is a performance metric that measures the cost of letting selfish agents negotiate the exchanges instead of invoking a central entity that computes an optimal allocation (Koutsoupias and Papadimitriou, 1999). The price of Anarchy is thus the ratio between the optimal solution and the worst reachable k-stable allocation. PoA is usually defined regarding social welfare.

Definition 16. ***Price of Anarchy***

Let $\mathcal{I}$ *the set of all MARA-instances and let* k *be the maximal authorized size of a cycle. The Price of Anarchy is defined as follows:*

$$PoA = max_{I \in \mathcal{I}} \frac{max_{A \in I} sw_u(A)}{min_{A \in C_k(I)} sw_u(A)}$$

where $C_k(I)$ is the set of k-stable allocations reachable from the initial allocation in the instance I.

It can first be demonstrated that the individual rationality constraint alone induces a high PoA.

Lemma 1. *Any procedure respecting individual rationality have PoA ≥ 2.*

We give the proofs of this lemma and of the following proposition in (Damamme et al., 2015). Moreover, for all procedures based on k-cycle-deals, the PoA is exactly 2.

Proposition 4. *All C_k procedures have PoA $= 2$.*

These results show that no procedure guaranteeing IR can provide better guarantees, and the size of the cycles does not change the worst-case efficiency of the final allocation.

It is important to note that this is a worst-case cost. In fact, it can be demonstrated that this cost is only observed in a special case where all the agents initially hold their middle ranking resource and they have circular preferences. In this case, the social welfare is $\frac{n(n+1)}{2}$.

Definition 17. *Circular preferences and middle-ranking allocation*

A profile of preferences is circular iff:

$$\forall i \in \mathcal{N}, r_j \in \mathcal{R}, \forall k \in \{1, \ldots, n\} : u_i(r_j) = (u_{i+k}(r_j) + k) \mod n$$

If n is odd, the allocation where each agent has her $\frac{n+1}{2}^{th}$ resource is called the middle-ranking *allocation. It has to be noticed that this allocation is k-stable.*

Example 4. *Circular preferences and middle-ranking allocation*

Let consider the following instance with 5 agents. The preference profile is circular and the white box allocation is the middle-ranking allocation.

agent 1	$: r_1$	$\succ$	r_2	$\succ$	$\boxed{r_3}$	$\succ$	r_4	$\succ$	r_5
agent 2	$: r_2$	$\succ$	r_3	$\succ$	$\boxed{r_4}$	$\succ$	r_5	$\succ$	r_1
agent 3	$: r_3$	$\succ$	r_4	$\succ$	$\boxed{r_5}$	$\succ$	r_1	$\succ$	r_2
agent 4	$: r_4$	$\succ$	r_5	$\succ$	$\boxed{r_1}$	$\succ$	r_2	$\succ$	r_3
agent 5	$: r_5$	$\succ$	r_1	$\succ$	$\boxed{r_2}$	$\succ$	r_3	$\succ$	r_4

If the initial allocation is randomly chosen, the probability to obtain the middle-ranking allocation is $\frac{1}{n!}$. The likelihood to observe the PoA is thus quite low. Nonetheless, other stable allocations may have a social welfare close to $\frac{n(n+1)}{2}$.

PRICE OF SHORT CYCLES Another natural question that arises is to evaluate the maximum cost of restricting the length of the cycles. We thus introduced a new notion, the "Price of Short Cycles" (PoSC), that measures the gap between the best allocation which may be reached with short cycles (of length k) and the worst allocation which may be reached with cycles of arbitrary length.

Definition 18. ***Price of short cycles***

Let $\mathcal{I}$ the set of all MARA-instances and let k be the maximal authorized size of a cycle. The Price of Short Cycles is defined as:

$$PoSC = max_{I \in \mathcal{I}} \frac{min_{A \in C_n(I)} sw_u(A)}{max_{A \in C_k(I)} sw_u(A)}$$

where $C_k(I)$ is the set of k-stable allocations reachable from the initial allocation in the instance I.

In (Damamme et al., 2015), we proved that the bilateral procedure (C_2) has $PoSC = 2$ but this proposition can be generalized to all C_k procedures with $k < n$ (see (Damamme, 2016)).

Proposition 5. *All C_k procedures with $k < n$ have $PoSC = 2$.*

Again, this worst case arises if the initial allocation is the middle-ranking allocation. As mentioned in the PoA study, this case has low likelihood to arise.

Average-case analysis

We run some experiments to study utilitarian welfare in the average case. We compared the social welfare obtained by the centralized procedure TTC, the optimal value which can be obtained (centrally) respecting the rationality constraint, the distributed procedure based on bilateral exchanges (C_2) and the distributed procedure allowing quite larger cycles (C_3). Figures 4 and 5 present average utilitarian social welfare obtained for different sizes of instances under Impartial Culture and Single Peaked preferences respectively. The values obtained by each procedure are depicted as a percentage of the theoretical maximal utilitarian welfare. For each instance size, a run is an instance (including an initial allocation) on which we apply the different methods mentioned, *i.e.* for C_2 deals are performed until a stable allocation is reached. Average values are computed over 2000 runs.

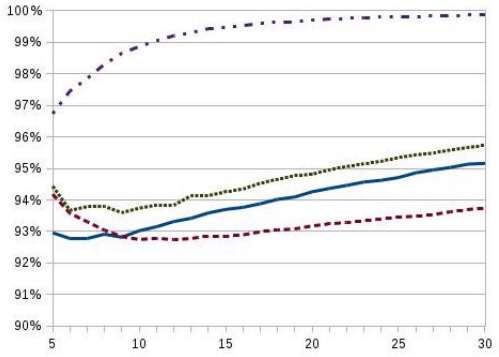


Figure 4: Mean value of sw_u under impartial culture

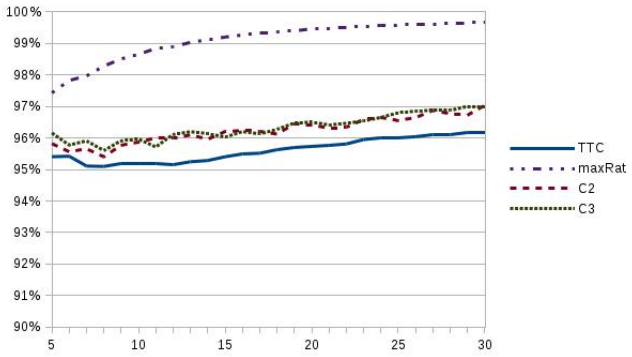


Figure 5: Mean value of sw_u under single-peaked preferences

It can first be observed that the outcomes provide fairly high values of social welfare (above 90% of the theoretically max value). Under impartial culture, it can also be noticed that C_3 provides no improvement over C_2 for small size instances, and then from $n = 10$ this improvement is rather small: an almost constant 1%. Under single-peaked preferences, we obtained similar results.

The results under single-peaked preferences also show that the social welfare values are on average higher (above 95%) than under impartial culture. These experiments also support the very good behaviour of C_2 under single-peaked preferences relatively to other procedures: for instance, we see that it slightly outperforms TTC (as mentioned before, both procedures are guaranteed to return Pareto-optimal allocations under this culture).

2.5.5 *Egalitarian social welfare*

Previous worst-case and average results studied the efficiency of the allocation. We can also investigate the fairness of the procedures and consider the egalitarian social welfare of the outcomes.

Worst-case analysis

The price of anarchy and the price of short cycles can be revised to study worst-case egalitarian social welfare. To avoid confusion, they will be denoted respectively PoA^{eg} and $PoSC^{eg}$.

In house allocation settings, there is no guarantee on the gap that can exist between the optimal and the worst egalitarian welfare. In fact, there exists some instances where the initial allocation A is such that $sw_e(A) = 1$ and no cycle-deals are possible whereas the allocation where all the agents get their top item is possible.

Example 5. *Consider the following instance and allocation:*

$$
\begin{array}{lllllll}
\textit{agent 1} & : \boxed{r_1} & \succ & \boxed{r_n} & \succ & \dots & \\
\textit{agent 2} & : \boxed{r_2} & \succ & \boxed{r_1} & \succ & \dots & \\
\textit{agent 3} & : \boxed{r_3} & \succ & \boxed{r_2} & \succ & \dots & \\
\textit{agent 4} & : \boxed{r_4} & \succ & \boxed{r_3} & \succ & \dots & \\
\vdots & & & & & & \\
\textit{agent n-1} & : \boxed{r_{n-1}} & \succ & \boxed{r_{n-2}} & \succ & \dots & \\
\textit{agent n} & : \boxed{r_n} & \succ & \dots & \succ & \dots \succ \boxed{r_{n-1}} \succ & r_{n-2}
\end{array}
$$

Take the initial allocation as the "white box" allocation A: it is 2-stable with $sw_e(A) = 2$, while the "yellow box" allocation A' is the only Pareto-optimal allocation, with $sw_e(A') = n$. It can be checked that a C_n reallocation of resources could lead from A to A'.

We thus obtain the following results:

Proposition 6. *All individually rational procedure have $PoA^{eg} = \Theta(n)$.*

Proposition 7. *All C_k procedure with $k < n$ have $PoSC^{eg} = \Theta(n)$.*

The proofs of these propositions are detailed in (Damamme et al., 2015; Damamme, 2016).

Average-case analysis

We run experiments like the ones described in the previous section but we studied the loss in egalitarian social welfare. Figures 6 and 7 present average egalitarian social welfare obtained for different sizes of instances under Impartial Culture and Single Peaked preferences respectively.

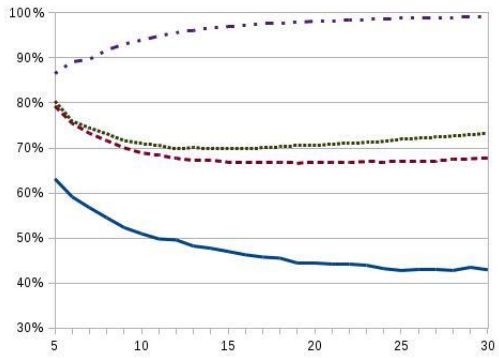


Figure 6: Mean value of sw_e under impartial culture

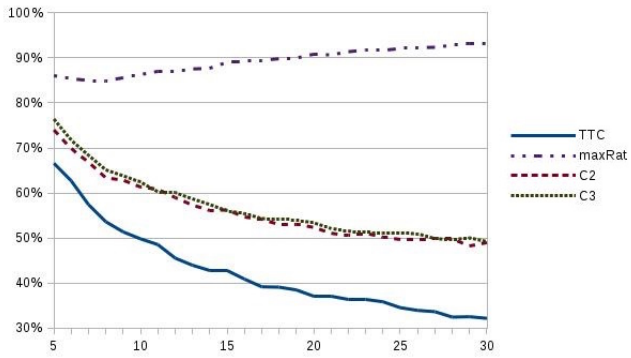


Figure 7: Mean value of sw_e under single-peaked preferences

It is important to observe that C_2 gives very good results especially under impartial culture. In fact, distributed procedures allow the agents to perform several exchanges. Agents with low-ranked

resources have thus more opportunity to exchange their resource. On average, the poorest agent will receive a resource ranked in the top-third of her preferences.

On the other hand, TTC gives poor performance. This is not surprising since the procedure implements the "best" cycles (i.e. maximizing the utility of the agents) and then discards the resources: this reduces a lot the range of possible cycles for other agents. Some agents may thus keep low-ranked (or even their initial) resources, leading to low individual utility.

Under single-peaked preferences, lower values are obtained and tend towards 50% on large instances. In fact, the likelihood that the agents have correlated preferences is more important. It is thus more difficult to reach an allocation where all the agents have highly valued resources.

2.5.6 *Discussion*

In this section, we investigated bilateral deals for distributed resource allocation in house market settings, under a very simple dynamics that allows the agents to improve their satisfaction without requiring complex coordination. We provide several new insights to assess the power of this approach in such settings. Pareto optimality is thus guaranteed under single-peaked preferences. Although worst-case solutions obtained using bilateral deals can have poor utilitarian welfare, it has been proved that no other individually rational mechanism (even those involving more than 2 agents) can ever provide better guarantee about the worst case social welfare. While the "price of short cycles" may be high in principle, our experimental findings show that in the average case, performances are fairly good in terms of social welfare.

Instead of allowing the agents autonomously negotiate bilateral deals, one could imagine that a centralized authority plan ahead a sequence of bilateral deals and the agents then apply exchange strategies in a distributed way. This is a common approach in multiagent planning. However, in (Damamme et al., 2015), we showed that this is not a realistic solution since we proved NP-completeness of deciding whether an allocation maximizing utilitarian or egalitarian welfare is reachable.

2.6 Fairness in dynamic and partially observable systems

We now investigate the more general setting where each agent holds k resources (with $k \geq 1$). As before, the agents autonomously negotiate rational deals and modify their allocation by exchanging resources. A deal consists in a cooperatively rational bilateral swap where one resource is exchanged against another resource. Agents are assumed to have additive preferences over the resources.

In the following, we will assume that each agent initially holds (and throughout the process) k resources. This restriction is commonly made as soon as fairness is under consideration (Brams et al., 2014; Aziz et al., 2016b; Segal-Halevi et al., 2017). It can be readily explained by the fact that assigning the same number of items to each agent is often regarded as a basic fairness feature. It is also a requirement in various domains such as course allocation. However, our approach is not limited to such settings. In fact, it generalizes easily as soon as each agent holds a finite and *a priori* known number of resources.

As discussed in Section 2.4, we consider distributed allocation protocols where each agent has limited visibility of the other agents. Starting from a situation where each agent has a set of items, and completely ignores how the rest of the resources are allocated, pairwise encounters occur as the result of the agents' decisions. As they do so, agents both observe the bundle currently held by the other agent, and try to agree on rational deals (swaps). Hence, agents have a partial and uncertain view of the entire allocation, that they maintain throughout the process, and which allows them to have different estimates of their envy.

The notion of fairness under incomplete information has been little studied so far. Bouveret et al. (2010) studied the complexity of fairly dividing a set of indivisible items under ordinal preferences when knowledge about the preferences of the agents is incomplete.

Recently, Chen and Shah (2017) introduced the notion of *Bayesian envy* where envy is considered in expectation over the possible allocations. In fact, each agent has no observation about the other agent's bundles but instead she has a prior distribution over the valuations of the other agents. A posterior probability on the allocations is thus estimated from this prior.

Aziz et al. (2018) studied the notion of *epistemic envy* where an agent is only aware of the set of resources allocated to her. An allocation is said to be *epistemic envy free* if, for each agent i, the items that are not allocated to her can be distributed among the other agents so that agent i is not envious of anyone. Epistemic envy-freeness does not exploit any observations about the other agents and corresponds to an optimistic estimate of the envy.

Departing from our study on distributed house allocation, we investigate how previous results can be extended to the partially observable context where each agent holds k resources. Whereas we previously did not make any hypothesis on the dynamics of the system, we now investigate how partial observability can be handled in the protocol. We will also study the impact of incomplete knowledge on the evaluation of the allocations. This raises the following issues:

- It is easy to see that the proof of *convergence* stated in Proposition 1 remains true in this context. Nonetheless, since the agents have partial observability of the system, distributed termination detection is an important issue. An efficient distributed procedure has to be designed in order the agents to be able to detect convergence to a stable state in a distributed way.

- Regarding the *efficiency of the allocation*, we can generalize previous (negative) optimality results. We already know that convergence to an allocation maximizing the utilitarian social welfare is not guaranteed. Without any restriction on the preference domain, a distributed protocol based on bilateral swap deals is not even guaranteed to reach a Pareto-optimal allocation. It can nevertheless be asked whether some domain restrictions would allow for providing optimality guarantees.

- In order to asses *fairness of the allocation*, we will use the notions of proportionality and envy. Indeed, in a context where each agent holds several resources, these notions are now fully significant. However, because of the partial observability and of the dynamicity of the system, the agents may have incomplete and incorrect knowledge about the global allocation. The envy actually experienced by an agent may thus be different from the envy computed by an omniscient agent knowing exactly the whole allocation. We will see that the usual notion of envy has thus to be adapted to distributed contexts. We will investigate new definitions of the envy in order each agent to asses the fairness of an allocation using her history of observations about the system.

- Finally, we will question whether the agents can take advantage of their observations about the system to guide their decisions and to *improve the efficiency of the distributed protocol.*

2.6.1 *Envy-freeness under incomplete knowledge*

We start by investigating knowledge incompleteness and incorrectness arising from partial observability. We then extend the notion of envy to this setting.

Since agents become aware of their respective bundles when they meet, each agent can maintain a set of observations describing her knowledge of the whole allocation of resources. Each time an agent encounters another agents, this set is updated given the new observations. However, agents have no way of knowing how the allocation evolves besides these encounters. Indeed, an agent is not aware of

the exchanges made by the other agents. In order to deal with this lack of observation, we shall use the following principle:

> *Unless proven otherwise, agents assume resources are still held where they were last observed.*

Let us denote by O_i^j the up-to-date set of resources that i assigns to agent j.
The update process is simple and captured by two rules:

1. upon encountering an agent j, an agent i observes the k items $\{r_1, \dots r_k\}$ that agent j currently holds, and thus updates $O_i^j \leftarrow \{r_1, \dots r_k\}$
2. if an item r_l is observed by i upon encountering j while it was supposed to be held by j' ($r_l \in O_i^{j'}$), then the observation set of this agent is updated: $O_i^{j'} \leftarrow O_i^{j'} \setminus \{r_l\}$

Now denote $O_i \cup_{j \in \mathcal{N}} O_i^j$ the set of resources i assigns to someone and $\overline{O}_i = R \setminus O_i$ be the set of items that agent i does not know where to allocate. This set $\overline{O}_i$ must not be confused with the set of items that agent i has *never* observed. Indeed, by virtue of (1), $\overline{O}_i$ does not necessarily grow monotonically.

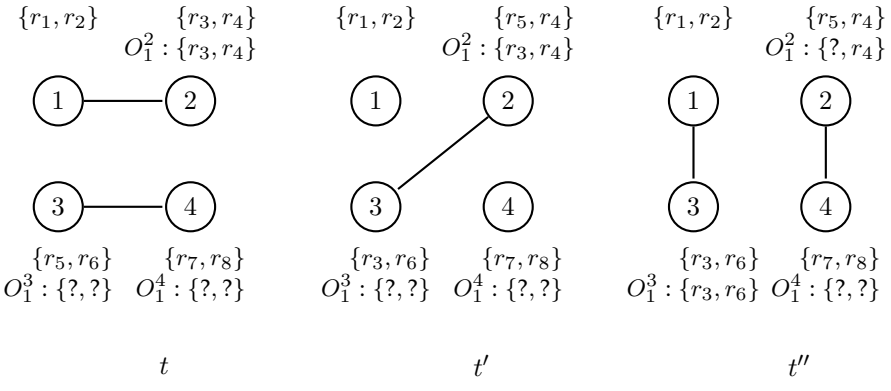
Knowledge incompleteness and incorrectness

Since an agent has no way of knowing how the allocation evolves besides her own observations, her view may not only be incomplete, but also *incorrect*. More precisely, for an agent i, we define:

- knowledge *incompleteness*, as the ratio of the number of items that agent's i view does not allocate to some agents:
$$Kincomp(i) = 1 - (|O_i|/(m-k))$$
- knowledge *incorrectness*, as the ratio of the number of items that agent's i view allocates incorrectly, over the number of items allocated:
$$Kincorr(i) = 1 - |\{o \in O_i^j \cap A_j | j \in \mathcal{N}\}|/|\cup_{j \in N} O_i^j|$$
where, implicitly, A_j stands for the bundle held by agent j in the actual allocation.

Note in particular that the incorrectness is a ratio over the set of items that agent i assigns to someone. Thus, it is possible for an agent to have perfectly correct but incomplete view.

Example 6. *We picture a scenario involving four agents, and three time-steps ($t < t' < t''$). We take the point of view of agent* 1*. Edges represent encounters between the agents at each time-step.*



At time t, agent 1 updates her observation set for agent 2. At time t', an encounter takes place between agent 2 and agent 3, but agent 1 is not aware of this. At this stage, $Kincomp(1) = 4/6$, and $Kinccorr(1) = 1/2$, since out of the two items that agent 1 can assign to someone in her observation set, only one is correct. Finally, at time t'', agent 1 encounters agent 3 and updates her observation sets for agent 3, but also for agent 2. At this stage, $Kincomp(1) = 3/6$, and $Kincorr(1) = 0$.

Evidence-Based envy

Extending the basic notion of envy introduced in (Lipton et al., 2004), we defined the Evidence-Based Envy (EBE) which stands for an estimate of the degree of envy relatively to a given set of observations.

Definition 19. ***Evidence-Based Envy***

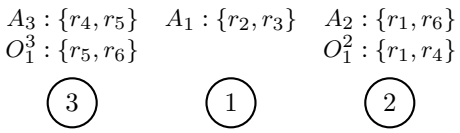
Given the set of observations O_i and an allocation A_i of items, the Evidence-Based Envy experienced by an agent i towards j is defined as:

$$e_{ij} = \max(0, u_i(O_i^j) - u_i(A_i))$$

A system is *evidence-based envy-free* (EBEF) when no agent is envious, based on her observations only.

Because of the possible incorrectness of knowledge, it is easy to see that "actual" envy-freeness (as would be evaluated by an omniscient agent observing the true and complete allocation) does not imply evidence-based envy, nor vice-versa.

Example 7. *For a simple example, take three agents, and six resources $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, with $u_1(r_1) = u_1(r_6) = 0, u_1(r_2) = u_1(r_3) = 1, u_1(r_4) = u_1(r_5) = 2$. Suppose agent 1 holds $\{r_2, r_3\}$, and thus enjoys a utility of 2. Suppose $O_1^2 = \{r_1, r_4\}$ and $O_1^3 = \{r_5, r_6\}$ while $A_2 = \{r_1, r_6\}$ and $A_3 = \{r_4, r_5\}$. This situation can be illustrated as:*



The agent is not EBEF but it would actually be EF for an omniscient agent. By swapping bundles of observations and actual allocations, we get the opposite. Note that this example does not involve any incomplete knowledge.

As agents may have incomplete knowledge, it is natural to consider different ways to complete the observations an agent may have regarding another agent in order to estimate the envy. In (Beynier et al., 2018b), we investigated different methods to estimate the utility of the other agents' bundles based on the utility of non-allocated resources. Let $\overline{O}_i\uparrow[q]$ (resp. $\overline{O}_i\downarrow[q]$) be the top-$q$ (resp. last-q) elements of $\overline{O}_i$, that is, the items not allocated with the q highest (resp. lowest) utility for agent i.

We then defined the following notions of envy:

- *optimistic* envy of agent j, obtained by completing the missing items by the least valuable:

$$e_{ij}^{OPT} = \max(0, u_i(O_i^j \cup \overline{O}_i\downarrow[k - |O_i^j|]) - u_i(A))$$

- *pessimistic* envy of agent j, obtained by completing the missing items by the most valuable:

$$e_{ij}^{PES} = \max(0, u_i(O_i^j \cup \overline{O}_i\uparrow[k - |O_i^j|]) - u_i(A))$$

- *average* envy of agent j, obtained by completing the missing items by the average value of the set $\overline{O}_i$:

$$e_{ij}^{AV} = \max(0, (u_i(O_i^j) + (k - |O_i^j|) \cdot \ avg(\overline{O}_i)) - u_i(A))$$

Clearly, for any j, it is the case that $e_{ij}^{OPT} \leq e_{ij}^{AV} \leq e_{ij}^{PES}$, and all notions coincide with classical envy when the observation set of an agent is complete and correct.

Note also that, for an agent i, this induces a partition of the other agents into: (i) agents that he envies (under optimistic envy), (ii) agents that he does not envy (even under pessimistic envy), and (iii) agents he may envy (*i.e* agents not belonging to either (i) or (ii)).

It is worth noticing that, in the absence of a cardinality constraint these notions are of very limited interested. Indeed, pessimistic envy would mean being envious of j assuming j would get *all* the remaining resources.

2.6.2 *Negotiation protocol*

The dynamics of our system is basically the same as the one used in house-allocation. Nonetheless, as each agent has several resources, a new decision level is introduced: besides choosing which agent to contact, the agent has also to decide for the exchange to propose to the other agent.

The dynamics is guided by the agents themselves, and is best described at two levels: (1) at the *global* level, agents decide to contact another agent ; and (2) at the *local* level –once a bilateral contact has been established– the agents try to exchange resources.

GLOBAL LEVEL: Assuming that each of the n agents holds k resources, an agent has to choose another agent to contact among $n-1$ agents and she has k^2 potential swap deals to consider per agent. The numbers of encounters and proposals before convergence may thus be large. Since, each agent has no knowledge about the preferences of the others and has partial knowledge on resource owners, she has not enough information to decide for the most promising agent to encounter nor for the most valuable exchange to propose. However, each agent can try to make the best use of her observations about the system.

We developed several informed-heuristics allowing an agent i to decide for the next agent to encounter. These heuristics exploit, in various ways, information about the time-stamp of the observations and the utility of the targeted resources. Time-stamps are used to estimate the degree of certainty of the agent's knowledge. Intuitively, oldest information should be less reliable than latest observations.

Heuristics range from random choices to more and more informed choices:

• **Random heuristic**: select one of the other agents using a uniform distribution.

• **Deterministic Time based heuristic**: the agent seen for the longest time is selected. In fact, this decision heuristics consists in randomly ordering the other agents and then following this order.

• **Probabilistic Time based heuristic**: select one of the other agents using a probability distribution based on the time-stamp of the last encounter with each other agent. The probability for agent i to select agent j is then defined as:

$$p_i(a_j) = \frac{age_{ij}}{\sum_{k \in \mathcal{A}, k \neq i} age_{ik}}$$

where age_{ij} is the age of the last encounter between agents i and j and is deduced from the time-stamp of the observations of i related to j.

• **Probabilistic Age and Utility based heuristic**: select one of the other agents using a scoring function. Each agent i maintains a matrix M_i of probabilities with m columns and n lines. $M_i[l][k]$ is

a value in $[0, 1]$ formalizing the likelihood that resource r_k is currently held by agent l from the point of view of agent i. Each time an agent i encounters another agent j, the matrices M_i and M_j of both agents are updated. Agent i (resp. j) updates the line related to agent j (resp. i) with values in $\{0, 1\}$. For each resource r_k held by agent j (resp. i), $M_i[j][k] = 1$ (resp. $M_j[i][k] = 1$). Otherwise, $M_i[j][k] = 0$ (resp. $M_j[i][k] = 0$). The other lines of the matrix are updated using a discount factor γ and the following equation:

$$M_i[l][k] = \gamma * M_i[l][k] + (1 - \gamma) * \frac{1}{n-2}$$

where $\frac{1}{n-2}$ is the probability that agent l holds resource r_k assuming that the $n-2$ other agents (different from i and j) have uniform likeliness to hold the resource. In fact, the process consists, for unobserved resources, to uniformly redistribute a part (tuned by γ) of the probabilities to all agents not involved in the current encounter. As time passes, probabilities related to unseen resources will tend to a uniform probability distribution.

In order to decide which agent to contact, agent i then computes a score for each other agent j using the matrix M_i, his utility function u_i and his interest information set:

$$score_i(j) = \sum_{r_k \in \mathcal{R} | interest_i(r_k) = AT} M_i[j][k] \cdot u_i(r_k)$$

The agent with the highest score is chosen at the end.

It has to be noticed that all time-based heuristics require that each agent has been encountered at least once. At first, the random heuristic can be used to schedule the first encounter between each pair of agents.

LOCAL LEVEL: When two agents encounter each other, they start negotiating some deals using a turn-based protocol. At her turn, an agent makes a proposal based on her preferences. A proposal is accepted by the other agent as soon as it is rational from her point of view. If both agents agree on a deal, the exchange is performed. An encounter is a *success* if at least one swap deal took place during the interaction, otherwise it is called a *failure*.

2.6.3 *Distributed convergence detection*

In order to improve their utility and decrease their envy, the agents try to agree on bilateral swap deals as long as some deals are still possible. Since each agent has limited observability of the system and does not know the preferences of the other agents, detecting the end of the exchanges in a distributed way is not easy. Moreover, each agent may not have full knowledge about the current owner of each resource. Agents are thus unable to individually infer the global allocation and detect whether some exchanges are still possible or not. We proposed a distributed approach allowing each agent to detect the end of the exchanges and providing guarantees on termination.

Interest information set

Each agent i maintains an *interest information set* about the resources of the system. In this set, each resource r_k is labeled by a level of interest. In this set, each resource r_k is labeled as:

- *unattractive (UN)*, meaning that resource r_k is not of interest for the agent, i.e. the value of r_k is less than the value of the worst resource currently held by the agent. Formally r_k is not interesting for the agent iff i.e. $\forall r_l \in A_i$, $u_i(r_l) > u_i(r_k)$.

For the other (attractive) resources, two further labels are used:

- *to-try (TT)* , meaning that the agent may try to obtain r_k;
- *wait-for-new-resources (WR)*, meaning that the agent has already tried to obtain r_k, and that the exchange failed. The agent must thus acquire new resources in order to propose (potentially) better exchanges to the agent holding r_k.

The interest information set of each agent is then initialized and updated using the following procedure:

- Initially, each agent distinguishes attractive resources to try -TT- from uninteresting resources -UN-
- Each time an agent i encounters another agent j, both agents update their interest information set.
  - If the encounter between agents i and j has lead to an exchange, the agents re-initialize their interest information set.
  - If the agent i (respectively j) realizes that he cannot obtain attractive resources held by agent j (resp. i). Labels on these resources are then turned to WR.

If the encounter between agents i and j has lead to an exchange, some resources may now not be of interest for the agents (because the value of the worst resource held by the agent has changed). Moreover, the agents may now re-try to get some resources previously labeled as WR. Labels of interest are then re-initialized using the initialization procedure.

Example 8. *Consider again the scenario depicted in Example 6 where the preferences of agents* 1 *and* 2 *are given by:*

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
agent 1	5	5	8	3	4	1	7	7
agent 2	6	6	3	5	1	4	6	6

Initially, agent 1 *holds* $\{r_1, r_2\}$ *and agent* 2 *holds* $\{r_3, r_4\}$*. The interest information set of each agent is initialized as:*

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
agent 1	$-$	$-$	TT	UN	UN	UN	TT	TT
agent 2	TT	TT	$-$	$-$	UN	TT	TT	TT

Note that resources held by an agent are not labeled.

At time t, agent 1 *encounters agent* 2 *and they exchange* r_1 *and* r_3*. The agents update their information set. Resources* r_3 *and* r_6 *become uninteresting for agent* 2 *since their utilities are less than the utility of the worst resource held by the agent (i.e.* r_4*). Later, agent* 2 *proposes to agent* 1 *to exchange* r_4 *against* r_2 *but agent* 1 *refuses since the exchange is not rational from her point of view. Agent* 2 *then updates the resources of interest with the label IT (here resource* r_2*).*

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
agent 1	TT	$-$	$-$	UN	UN	UN	TT	TT
agent 2	$-$	WR	UN	$-$	UN	UN	TT	TT

Soundness of the protocol

Thanks to their interest information sets and a limited amount of communication, agents are then able to efficiently detect termination in a distributed way. Our approach guarantees that termination is detected only when no more exchange is indeed possible (convergence is indeed reached). In fact, once an agent has no more resources labeled as TT (to-try), she has tried to obtain all resources preferred to the ones she currently holds. Some resources can still be of interest for the agent but she knows that she must wait to try again to obtain one of them. Actually, the agent has to wait for the other agents to be interested in her resources.

Each agent thus alternates between two execution modes: the *active mode* where the agent contacts some other agents and try to exchange resources and the *standby mode* where the agent waits for some contact requests. Initially, each agent starts in the active mode.When an agent has no more resource labeled as TT, she moves to the standby mode where she only waits for contact requests from the other agents. If another agent contacts her and an exchange is performed, the agent re-initialize her information set. If at least one resource is labeled as TT, the agent comes out of the standby mode. In order to allow each agent to individually detect convergence, each agent has to inform the other agents when she enters and exits the standby mode.

Proposition 8. *When all agents are in the standby mode, there is no more possible rational exchange of resources between the agents.*

The proof of this proposition is detailed in (Beynier et al., 2018b).

It has to be noticed that the reverse is not true. Indeed, while there is no more possible exchange (i.e. convergence is reached), some resources may still be labeled as TT. Although some agents have incomplete or incorrect knowledge, this does not prevent termination detection since the procedure only relies on interest information sets and does not account for knowledge on resource location.

Because fairness notions based on observations depend on individual knowledge, it is relevant to estimate knowledge incorrectness and incompleteness once termination is detected.

Proposition 9. *Upon termination, in a system of n agents holding k resources, the total amount of incorrectness is at most $kn(n-2)$, and the bound is tight.*

In fact, we experimentally showed that this upper bound is too generous and the agents often have very low incorrectness ratio upon convergence.

2.6.4 *Efficiency of the outcomes*

In house allocation settings, we showed that Pareto-optimality is guaranteed under single-peaked preferences (see Section 2.5.3). A natural question is whether such guarantee can still be provided when the agents hold several resources. In fact, the answer is no. It can be shown that even with two resources some allocations may be 2-stable but not Pareto-efficient.

Example 9. ***Non-Pareto efficiency***

We give a counter-example. Let consider the following instance involving 3 agents and 2 resources per agent. It can be checked that preferences are single-peaked.

	r_1	r_2	r_3	r_4	r_5	r_6
agent 1	**1**	**2**	3	4	5	6
agent 2	1	3	**4**	**5**	6	2
agent 3	1	2	4	5	**6**	**3**

The white-box allocation is 2-stable but it is not Pareto-optimal since it is Pareto-dominated by the allocation where agent 1 *holds* $\{r_1, r_6\}$*, agent* 2 *holds* $\{r_2, r_5\}$ *and agent* 3 *holds* $\{r_3, r_4\}$*. Indeed, we have:*

$$u_1(r_1, r_6) = 7 \geq u_1(r_1, r_2) = 3$$
$$u_2(r_2, r_5) = 9 \geq u_2(r_3, r_4) = 9$$
$$u_3(r_3, r_4) = 9 \geq u_3(r_5, r_6) = 9$$

SEQUENCEABILITY Nonetheless, the solutions obtained using swap deals can be related to the notion of *sequenceability*. This later notion relies on a particular allocation protocol: *sequences of sincere choices* (also known as *picking sequences*). This very simple protocol works as follows. A central authority chooses a sequence of agents before the protocol starts, having as many agents as the number of objects (some agents may appear several times in the sequence). Then, each agent appearing in the sequence is asked to choose in turn one object among those that remain. For instance, according to the sequence $\langle 1, 2, 2, 1\rangle$, agent 1 is going to choose first, then agent 2 will pick two consecutive objects, and agent 1 will take the last object. This protocol, actually used in a lot of everyday situations, has been studied for the first time by Kohler and Chandrasekaran (1971). Later, Brams and Taylor (2000) have studied a particular version of this protocol, namely alternating sequences, in which the sequence of agents is restricted to a balanced ($\langle 1, 2, 2, 1...\rangle$) or strict ($\langle 1, 2, 1, 2...\rangle$) alternation of agents. Bouveret and Lang (2011) have further formalized this protocol.

Definition 20. *Let* $I = \langle \mathcal{N}, \mathcal{R}, \mathcal{P}\rangle$ *be an MARA instance with additive preferences. A* sequence of sincere choices *(or simply* sequence *when the context is clear) is a vector of* $\mathcal{N}^m$*. We will denote by* $\mathcal{S}(I)$ *the set of possible sequences for the instance* I*.*

A sequence $\overrightarrow{\sigma}$ is said to *generate* allocation A iff A can be obtained as a possible result of a non-deterministic algorithm on input I and $\overrightarrow{\sigma}$ which simply makes agents chose one of their top object at their turn in the sequence.

Definition 21. *An allocation* A *is said to be* sequenceable *if there exists a sequence* $\overrightarrow{\sigma}$ *that generates* A*, and* non-sequenceable *otherwise. For a given instance* I*, we will denote by* $s(I)$ *the binary relation defined by* $(\overrightarrow{\sigma}, A) \in s(I)$ *if and only if* A *can be generated by* $\overrightarrow{\sigma}$*.*

Bouveret and Lemaître (2016) and Aziz et al. (2016b) proved that every Pareto-optimal allocation is sequenceable. Moreover, Bouveret and Lemaître (2016) proved that there exists, for a given instance, three classes of allocations: (1) non-sequenceable (therefore non Pareto-optimal) allocations, (2) sequenceable but non Pareto-optimal allocations, and (3) Pareto-optimal (hence sequenceable) allocations. These three classes define a "scale of efficiency" that can be used to characterize the allocations. What is interesting and new here is the intermediate level.

Pareto-optimality can be thought as a reallocation of objects among agents using improving deals (Sandholm, 1998). In distributed settings, *Trading cycles* or *cycle deals* constitute a sub-class of deals (see Section 2.2.3), which is classical and used, *e.g.*, by Varian (1974, page 79) and Lipton et al. (2004, Lemma 2.2) in the context of envy-freeness. Trying to link efficiency concepts with various notions of deals is thus a natural idea.

Intuitively, if it is possible to improve an allocation by applying an improving cycle deal, then it means that this allocation is inefficient. Reallocating the items according to the deal will make everyone better-off. It is thus natural to derive a concept of efficiency from this notion of cycle-deal.

Definition 22. *An allocation is said to be* $>$*-*(N, M)*-Cycle Optimal (resp.* $\geq$*-*(N, M)*-Cycle Optimal) if it does not admit any strictly (resp. weakly) improving* (k, M)*-cycle deal for any* $k \leq N$*.*

We begin with easy observations. First, $\geq$-cycle optimality implies $>$-cycle optimality, and these two notions become equivalent when the preferences are strict on shares. Moreover, restricting the size

of the cycles and the size of the bundles exchange yield less possible deals and hence lead to weaker optimality notions. Note that for $N' \leq N$ and $M' \leq M$ $>$-(N, M)-cycle-optimality and $\geq$-(N', M')-cycle-optimality are incomparable. These observations show that cycle-deal optimality notions form a (non-linear) hierarchy of efficiency concepts of diverse strengths. The natural question is whether they can be related to sequenceability and Pareto-optimality. Obviously, Pareto-optimality implies both $>$-cycle-optimality and $\geq$-cycle-optimality. An adaptation of the Proposition of Bouveret and Lemaître (2016) and Aziz et al. (2016b) leads to the following stronger result:

Proposition 10. *An allocation A is sequenceable if and only if it is $>$-n-cycle optimal (with $n = |\mathcal{N}|$).*

Interestingly, when preferences are single-peaked, the hierarchy of N-cycle optimality collapses at the second level:

Proposition 11. *If all the preferences are single-peaked (and additive), then an allocation A is $\geq$-n-cycle optimal iff it is swap-optimal.*

Together with Proposition 10, Proposition 11 gives another interpretation of sequenceability in this domain:

Corollary 1. *If all the preferences are single-peaked (and additive), then an allocation A is sequenceable if and only if it is swap-optimal.*

The proofs of these propositions and of this corollary are detailed in (Beynier et al., 2018).

2.6.5 *Fairness of the outcomes*

Several experiments have been developed to study the dynamics and the fairness of the distributed resource allocation protocol. Experiments first showed that our distributed protocol leads to low degrees of envy. We observed that agents get less envious as the numbers of agents and resources increases. In fact, the more agents there are in the system, the more opportunities there are for each agent to exchange her resources. Each agent has a greater likelihood to find another agent with whom she can make an exchange and obtain more preferred resources. We also observed that even for small numbers of agents, the ratio of proportional outcomes is quite high. The same trends can be observed while increasing the number of resources per agent, which is expected (Dickerson et al., 2014). Beyond 9 resources per agent, almost all outcomes are proportional. Recall that proportionality is less demanding than envy-freeness (Bouveret and Lemaître, 2016).

Even if agents have partial observation of the system, they obtain, at the end, high knowledge completeness and correctness ratios. We then experimented the influence of the informed heuristics on the fairness of the outcomes by looking at the final envy (computed by an omniscient agent) and the Evidence-Based Envy. We observed that the heuristics have little influence on the final envy. Indeed, since the agents keep meeting each other and exchanging resources until no more deal is possible, we obtain in average nearly equivalent allocations. However, heuristics influence the length of the process before convergence. We observed that time-based heuristics lead to smaller numbers of encounters and proposals since they provide more efficient contact decisions. The agents are more likely to contact some agents that are willing to exchange valuable resources. These heuristics allow the agents to converge more quickly without loss in fairness.

2.6.6 *Discussion*

In this section, we extended our work on house-allocation to the more general context where each agent holds several resources. Although the procedure based on bilateral deals is guaranteed to converge,

we showed that no guarantees on the optimality of the outcomes can be provided even in restricted domains of preferences. In fact, these negative results arise from the limitations of bilateral exchanges. Indeed, $k-$deals may not always be decomposed as a series of rational bilateral deals. This is the price to pay for the simplicity of our protocol which avoids to search over an exponential number of deals at the local level, and an exponential number of groups of agents at the global level.

We also studied issues dealing with partial observability. In particular, we introduced new notions of envy-freeness accounting for incomplete and incorrect knowledge. These measures allow the agents to asses locally the fairness of an allocation and give good estimates of the actual envy.

Finally, we provided a fully distributed protocol allowing to guarantee termination despite imperfect knowledge of the agents. We investigated experimentally the performance of this system, testing in particular several heuristics governing agents' decision-making both at the agent's selection and bilateral negotiation stage. Surprisingly, heuristics have little effect on the fairness of the outcomes. In fact, the dynamics of the system does not limit the number of resource exchanges and allows the agents to reach (at the end) equivalent solutions. Nonetheless, heuristics influence the numbers of encounters and exchanges performed before convergence. They are thus especially useful under time constraints or limitations on the number of exchanges. In such contexts, it has to be noticed that our distributed process has anytime properties.

2.7 Networked exchanges

Until now, it has been supposed that each agent can interact with any other agent. However, it is quite natural to consider that each agent can perceive only a subset of the agents and can only trade with these agents. Such acquaintances can be modeled by a *social network*.

This later notion was first introduced in social sciences. Indeed, the analysis of social networks and their impact on the behaviors of the individuals is a prominent topic of research especially in sociology and economics (Burt, 1982; Jackson, 2008; David and Jon, 2010). Among other issues, one topic of interest in these domains is how the structure of the network influence the exchanges between rational agents.

A social network is modeled as an undirected graph $G = (\mathcal{N}, E)$ where the nodes represent the agents and two agents i and j are directly connected in the graph if they can interact. It has to be noticed that the network is not necessarily limited to the representation of social or economical relationships but can also reflect other kinds of constraints on interactions such as spatial constraints (proximity constraints for instance).

In multiagent resource allocation, the edges of the graph may have different degrees of expressiveness. The network first models the limited visibility of the agents. Two agents are thus directly connected if they can observe each other set of resources. In cake-cutting settings, Abebe et al. (2017) and Bei et al. (2017) investigated fairness issues in MARA problems where an agent can only compare her share with the ones of her neighbors in the social network. In the context of fair allocation of indivisible resources, Aziz et al. (2018) extends the standard notion of envy-freeness to take the limited visibility of the agents on a social graph.

In distributed contexts, the network also structures the exchanges between the agents. Chevaleyre et al. (2007c) introduced the notion of *negotiation topology* which is equivalent to a social network. The graph formalizes visibility relations between the agents but also represents the possible trading interactions. A deal is possible if it only involves agents belonging to a same clique in the graph.

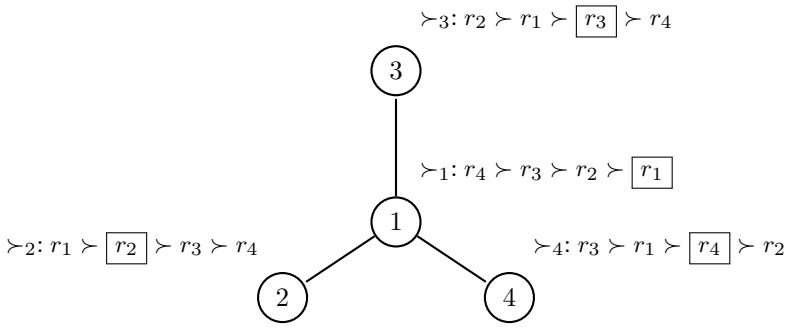
As social networks limit potential interactions, we can question how these restrictions affect the allocation process. One the one hand, each agent has less possible interactions to consider with the

other agents and the decision should be easier. On the other hand, the paths taken by the resources among the agents are much less numerous and the risk that a resource remains blocked at a given agent is more important. Reaching an efficient or fair allocation thus often requires a careful scheduling of the exchanges. In general, it is more difficult to reach an efficient state in a social network (Chevaleyre et al., 2007c).

As it can be observed in the following example, the network topology plays a crucial role on the complexity of the allocation process and on the efficiency and the fairness of the outcomes.

Example 10. *Reaching a Pareto-efficienct allocation on a star*

Let us consider a social network involving 4 agents where each agent is connected to only one central agent. This kind of topology is referred to as a "star".



Each agent may obtain her top resources if the exchanges follow a specific schedule: agent 2 *must first exchange* r_2 *with the center of the star (agent* 1*), then agent* 3 *must swap her resource* r_3 *with* r_2 *held by agent* 1 *and finally agent* 4 *must swap* r_4 *with* r_3 *now held by agent* 1*.*

If 1 *first exchanges the resource* r_1 *with agent* 3 *or agent* 4*, the outcome would not be Pareto-efficient.*

2.7.1 *Efficiency on social networks*

Gourvès et al. (2017) analyzed the computational complexity of decision problems related to bilateral swap-deals along a social network. They consider the same context as the one we investigated in Section 2.5 but they enrich the problem description with an undirected graph formalizing possible interactions among the agents. They consider several restrictions on the network topology such as the line or the star.

Besides providing complexity results in the general case, they also study how the topology influences the complexity of (i) deciding whether an object or an allocation is reachable, (ii) finding a reachable Pareto-efficient allocation. Notably, they proved that deciding if an allocation is reachable is **NP**-complete and computing a sequence of swaps to reach a Pareto-efficient allocation is **NP**-hard. Nonetheless, polynomial time algorithms can return a Pareto-efficient allocation for two specific topologies: the star and the path. Moreover, when the graph is a tree, the reachable assignment problem can be solved in polynomial time.

2.7.2 *Fairness on social networks*

As mentioned before, proportionality can be computed in a distributed way and does not require full knowledge of the whole allocation. In fact, an agent is able to decide if she gets a proportional share as soon as she is aware of her own share, of the number of agents in the system and of the set of resources $\mathcal{R}$ in the system. Nonetheless, in a social network, an agent may only be interested in the

shares of her neighbors. Abebe et al. (2017) thus introduced the notion of local proportionality where each agent compares the value of her share with the average value of her neighbors.

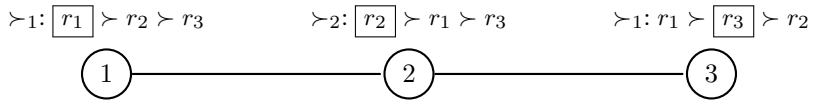
Envy-freeness on graphs

The envy is very sensitive to the information available to agents. The notion of envy can also naturally be extended to account for the topology of the social graph. Intuitively, an allocation will be *locally envy-free* if none of the agents envy her neighbors. This notion has also been referred to as *graph-envy-freeness* (Chevaleyre et al., 2017). In the case of a complete network, local-envy-freeness is equivalent to the notion of envy we have previously introduced.

In house allocation problems, an allocation is envy-free in a complete graph if and only if each agent gets her top object and this is obviously also a Pareto-optimal allocation in that case (recall our previous discussion on envy in the context of house-allocation). However, when an agent is only connected to a subset of the other agents, the envy becomes a much more interesting notion. Indeed, an agent may not need to get her top-resource to be envy-free. The locations of the resources on the graph as well as the connections between the agents are then crucial issues in order to compute a locally envy-free allocation.

Example 11. *Envy-free allocation on a line*

To see how the network can make a difference, consider consider the following scenario.



If the agents could observe all the other agents, none of the allocations is envy-free.

If each agent only observes the bundles of her neighbors, the white-box allocation is locally-envy-free. Note that a local envy-free allocation is not necessarily Pareto-optimal (take the same allocation, but the ranking of agent 1 *to be* $\succ_1$: $r_3 \succ r_1 \succ r_2$*) but giving her top item to each agent if possible will always be an envy-free Pareto-optimal allocation in* any *network.*

It is important to note that unlike evidence-based envy, local envy assumes full and correct knowledge of the resources held by the neighbors but the agents have no knowledge on their non-neighbors.

Complexity results

In (Beynier et al., 2018a), we investigated the problem of deciding if a central planner, who has a complete knowledge of the social network and the agents' rankings of the objects, can allocate the objects such that no agent will envy a neighbor[3]. In house-allocation problem, we identified intractable and tractable cases of this decision problem with respect to the number of neighbors of each agent, that is the degree of the nodes in the graph representing the social network.

It has been proved that deciding whether a locally envy-free allocation exists is **NP**-complete on a line, or on a circle, and more generally on graphs of maximum degree k for $k \geq 1$ constant. Nonetheless, if the social graph is dense enough, then deciding whether a locally envy-free allocation exists is solvable in polynomial time. In fact, whether a locally envy-free allocation exists in graphs of minimum degree $n-2$ is solvable in polynomial time.

Another relevant parameter is the size of a *vertex cover*. A subset of agents $\mathcal{N}'$ forms a vertex cover of the social network if every agent is either in $\mathcal{N}'$, or at least one of her neighbors is in $\mathcal{N}'$. We provided an algorithm which shows that deciding local envy-freeness is in **XP** (parameterized by the

[3] It has to be noticed that Bredereck et al. (2018) studied, at the same time, closely related complexity issues on *directed* social graphs.

size of a vertex cover) and a proof of **W[1]**-hardness. This means that this problem is unlikely to be fixed-parameter tractable with the vertex cover size as parameter.

We also considered optimization problems with two different perspectives. The first one consists in maximizing the number of locally envy-free agents, and the other one consists in maximizing the degree of non-envy of the society. Some approximation algorithms have been provided for both approaches. For maximizing the number of locally envy-free agents, our algorithm gives an $\frac{|I|}{n}$ approximation with $|I|$ the size of an independent set in the social network. For maximizing the degree of non-envy, a derandomization technique based on the minimization of a conditional expectation, has been proposed. This algorithm is a polynomial-time $\frac{5}{6} - o(1)$ approximation algorithm.

Finally, some experiments were drawn to test the likelihood to find a locally envy-free allocation as the degree of the graph augments. This likelihood clearly decreases (in the extreme case of a complete graph, recall that all agents must have a different preferred item). Under impartial culture, experiments showed that this decrease is sharp and from a degree equal to half of the agents, it actually becomes highly unlikely to find a locally envy-free allocation. On the other hand, for graphs of small degrees, it is often the case that a locally envy-free allocation can be found, and, as expected, it becomes even more so as the number of agents and items increases.

2.7.3 *Discusssion*

From Example 11 it may be objected that agent 3 may still be envious of agent 1, because she knows that this agent must have received the item agent 2 didn't get, i.e. r_1. In (Beynier et al., 2018a) and (Beynier et al., 2018b), we provided several counter-arguments for that. First, as a technical response, note that in general agents would not know exactly who gets the items they do not see. Envy is intuitively a notion which needs to be "targeted" to someone. Being envious "of someone" without further precision may seem at odd with the very notion of envy. The second reason is that it is actually difficult, for an agent, to decide whether she is envious of someone. Thus, although agents may know that they must be envious of *some* agents, they cannot identify which one, which makes a significant difference in the case of envy. Our second point is more fundamental and concerns the model and the motivation of the work. In fact, another interpretation of the meaning of links in the graph is that they may represent envy the central authority is concerned with. In other words, although there may theoretically be envy among all agents, the central authority may have reasons to only focus on some of these envy links. For instance, you may wish to avoid envy among members of the same team in your organization, because they actually work together on a daily basis (in that case links may capture team relationships).

Aziz et al. (2018) take a different point of view and extended the notion of epistemic-envy to graph-envy-freeness. An agent is graph-epistemic-envy-free (G-EEF) if she does not envy her neighbors in the current allocation and there exists an allocation of the unseen items among unobserved agents such that the agent does not envy none of them. In this framework, the graph restricts the visibility of the agents but the envy is not limited to the neighbors and may concern unobserved agents.

2.8 Perspectives

The work developed in this chapter has been mainly conducted over the last 5 years. This is still an ongoing work with various perspectives.

FAIRNESS IN SOCIAL GRAPHS The first immediate perspective of this work is to carry on our work on fair allocation in social networks and to extend our contributions on distributed procedures to this context. It would be interesting to investigate the fairness of the allocations obtained by allowing the agents to negotiate bilateral deals with their neighbors. Although, it can be expected that the Price of Anarchy would be high, we can hope that good results could be obtained on average. Further experiments need to be developed to support the power of swap deals in social networks.

If the observations about the other agents are costly (if the agents obtain observations from communication for instance) or are subjected to time or spatial constraints, an agent may not be able to observe the items of her neighbors at all time. It would thus be interesting to merge both notions of local-envy and evidence-based envy in order to introduce local-evidence-based envy. In fact, our notion of evidence-based envy can be viewed as an envy measurement in an underlying dynamic graphs where an edge between two agents is added to the graph once the agents has established a contact and this edge is removed at the end of the encounter. Nonetheless, in a social network, an agent may not observe all the other agents. In this context, our distributed procedure to detect termination needs to be adapted since an agent may not observe all the resources she would like to obtain. In fact, the agent may be unable to remove all *to-try* labels and to move into the standby mode.

Until now, we considered that the agents observe little information about the other agents. One can imagine that the agents would be able to obtain more knowledge about the other agents (or part of them). For instance, in a social network, an agent could be able to observe the preferences of her neighbors and/or the resources of the agents in a range of k edges. It would then be interesting to investigate how the value of k influences the complexity and the likelihood to find a fair allocation.

Finally, if no fair allocation can be found in a social graph, we can question which modifications of the graph would allow for finding an envy-free solution. The first approach consists in removing some of the connections between the agents. The problem can thus be defined as the minimization of the number of edges to remove in order an envy-free allocation to exist. Another approach consists in switching the positions of the agents in the graph. The problem could be defined as the minimization of the number of swaps (of agents) in order to allow for an envy-free allocation.

OTHER FAIRNESS NOTIONS In this chapter, we mainly focused on envy-free allocations. But as we have seen, the existence of an envy-free allocation cannot be guaranteed and even when an envy-free allocation exists, computing such an allocation can be computationally hard. Other fairness notions have been proposed such as max-min fair-share, min-max fair share, proportionality, competitive equilibrium from equal incomes. Bouveret and Lemaître (2016) proposed a hierarchy to characterize the level of fairness of these notions. On the other hand, several works proposed some approximations of the envy-freeness notion such as envy-freeness up to one good (Lipton et al., 2004), envy-freeness up to any good or pairwise max-min fair share (Caragiannis et al., 2016). Recently, Amanatidis et al. (2018) studied the relations between some fairness notions and their relaxations.

In our context, it would be interesting to consider these different fairness notions and relaxations, and to study whether some guarantees could be provided regarding the outcomes of our allocation protocols. Furthermore, to our knowledge, these notions have never been investigated in the context of social networks. This open up promising perspectives to relax the notion of local envy-freeness.

PREFERENCE MODELS In this chapter, the preferences of the agents were represented by additive functions. Another line of research is to consider more general preference functions allowing for synergies between the resources.

k-additive functions are useful to represent problems where positive or negative synergies between the resources are limited to bundles of at most k elements (Chevaleyre et al., 2007b). An important issue is to determine whether our theoretical results still hold under k-additive functions. It is expected that convergence to Pareto-optimal or fair allocation cannot be guaranteed under k-additive prefer-

ences. Nonetheless, we can question if some domain restrictions or some types of deals could provide efficiency and/or fairness guarantees. One promising type of deals to consider are cycle-deals. It has to be noticed that, in a distributed allocation context, Chevaleyre et al. (2017) provided convergence guarantees toward an efficient envy-free allocation when functions are super-modular and monetary side payments are allowed. To our knowledge, similar results without monetary side payments have never been described.

DISTRIBUTED DECISION MAKING Until now, the decisions of the agents have been only based on direct observations. It would be relevant to allow the agents to infer new knowledge from these observations. This coincides with our comment made earlier in Section 2.7.3. From the location of observed resources, an agent can indeed infer (even uncertain) knowledge on the location of unobserved resources. Some knowledge on the preferences of the other agents could also be deduced from the success or the failure of the exchanges. This additional knowledge could be then exploited to refine the decisions of the agents.

A related direction is to allow the agents to plan ahead the swaps of resources. So far, we have considered myopic agents performing rational deals that immediately improve their utility. Although we showed that the computational complexity of searching for a Pareto-optimal or a fair allocation is high in the general case, we could consider an intermediate approach where the agents plan ahead over h decision steps and select the best plan over this planning horizon. The agents would be able to accept non-rational deals to make better deals in the future.

In (Beynier and Estivie, 2013), we started investigating a multiagent planning approach to compute such policies using Decentralized Markov Decision Processes (see Section 3.2.2). One of the main difficulty is to model the possible outcomes of the encounters. Markovian models postulate the existence of a probabilistic representation of the uncertainty on action outcomes. In MARA problems, such probabilistic representations are not always available to each agent. As we will discuss in the next chapter, another difficulty is the high computational complexity of planning algorithms.

PROCEDURAL FAIRNESS In our work, the notion of fairness have been investigated from the point of view of the final allocation. A complementary approach is to assess the fairness of the allocation procedure. As argued in Experimental Economics, people are not only concerned with the final allocation but also take into account their perception of the allocation process. In Economics, the fairness of the allocation process is referred to as *procedural justice* and differs from *distributive justice* which deals with the fairness of the outcome (Thibaut et al., 1974). Experiments from economics suggest that empowering the agents in an allocation procedure leads to higher fairness perception. In fact, allocation processes giving more voice to the agents are perceived as more fair (Shor, 2009).

One interesting issue is to develop some experiments to evaluate the fairness perception of distributed procedures based on bilateral swaps compared to other procedures such as picking sequences or serial dictatorship (Kohler and Chandrasekaran, 1971; Bouveret and Lang, 2011). It is expected that such experiments would strengthen even more the relevance of this distributed protocol.

3

MULTIAGENT PLANNING UNDER UNCERTAINTY

In the previous chapter, we envisioned rationality from a myopic point of view. In fact, an action (in this case, a deal) was considered as being rational for an agent if and only if it immediately increases her utility. Nonetheless, when an agent has to make a sequence of decisions, it is important to look ahead to the consequences of the agent's current decision on future states and executable actions. Hence, an action can provide no immediate gain (or even incurs a cost) but could allow the agent to reach, at a later stage, more valuable states or to execute more valuable actions. In sequential decision making, the rationality criterion should take into account short-term and long-term impacts of decisions, whenever possible.

Nonetheless, anticipating the effects of an action is not that simple. Indeed, as explained in the Introduction of the document, agents acting in real-world environments often have partial observability of the system and may not have enough information to determine exactly the effects of the actions. Moreover, the dynamics of the environment is often uncertain and a single action may have different issues. When uncertainty can be modeled by probability distributions[1] (Neumann and Morgenstern, 1953; Bernoulli, 1954). This principle says that *a rational agent should choose the action that yields the maximum expected utility.* In fact, in such settings, the utility of a state is defined as the expected utility over the sequences of states that might follow. This sequence of states depends on the actions executed by the agent, i.e. the policy π of the agent. The utility of a state is thus defined as the expected sum of discounted rewards under policy π (Russell and Norvig, 2003; Bellman, 2003). The utility of state s under π is defined as the expected discounted sum of rewards of the next states when executing the policy π from state s:

$$U^{\pi}(s) = E\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s\Big] \tag{1}$$

where $R(s_t)$ denotes the immediate gain of the agent for being in s at t, γ is a discount factor ($\gamma \in [0, 1[$) weighting short-term and long-term rewards.

In multiagent settings, each agent is assumed to take her own decisions autonomously. An individual strategy π_i has thus to be defined for each agent i. Because of interactions, the optimal strategies of the agents cannot be computed independently (using a set of individual and independent planning algorithms for instance). In fact, the stochastic changes in the states of the system, and hence the expected utility of each agent, rely on the joint action, i.e. on the actions taken by all the agents.

In order to optimize her performance, each agent has thus to coordinate her actions with the ones of the other agents. Nonetheless, because of partial observability, each agent is not fully aware of the states of the other agents and may have limited information about the state of the environment. Combined with the uncertainty on action outcomes, this led each agent to be uncertain about the state of the other agents and about the actions taken by the others.

[1] In distributed MARA, building such a probabilistic model of uncertainty requires each agent to know, at least, the preferences of the other agents. In the previous chapter, preference functions were assumed to be private information. That's why we did not consider that each agent has a probabilistic model of the uncertainty.

In this chapter, we investigate *sequential multiagent decision-making under uncertainty* where a set of agents have to decide, in a distributed way, how to act given partial observations about the system. We focus on *cooperative systems* where the agents aim at maximizing a shared objective function. Multiagent planning consists in computing a plan (or strategy) for each agent such as each agent can take, at each decision step and given her sequence of local observations, the individual action optimizing the expected utility of the system.

Such multiagent planning problems can be represented by Decentralized Partially Observable Markov Decision Processes (DEC-POMDPs). This model extends POMDPs to multiagent settings where control is decentralized. The DEC-POMDP framework is very general and covers a broad spectrum of optimal distributed control problems. Notably, DEC-POMDPs allow for formalizing coordination problems among teams of mobile robots for exploring unknown environments (Bernstein et al., 2001; Beynier and Mouaddib, 2011a; Matignon et al., 2012a; Amato et al., 2016) or for patrolling among sensitive areas (Beynier, 2016). Among the wide range of applications covered by DEC-POMDPs, we can also cite sensor network management (Nair et al., 2005), load balancing in server networks (Beynier and Mouaddib, 2009) or distributed control on multi-access broadcast channels (Ooi and Wornell, 1996).

Nonetheless, despite the genericity of the model, applying DEC-POMDPs to real-world problems remains difficult in practice. Several reasons can explain the gap between DEC-POMDPs and real-world applications. First, solving a DEC-POMDP has a high computational complexity. Although significant advances have been made recently to improve the efficiency of solving methods, the scalability of DEC-POMDPs is still limited, making the framework difficult to be used to solve real-world problems. Moreover, most existing solving algorithms consists in centrally computing a joint policy before the execution (off-line centralized planning). As we will discuss later, such a planning entity may not be available in practice. Finally, time and action representations are very general and the standard DEC-POMDP model does not allow for formalizing constraints on action execution. If the optimal solution is computed without handling the constraints of the problem, the performances obtained in practice may be significantly lower than those provided theoretically.

The work presented in this chapter aims at improving the applicability of the DEC-POMDP framework. We first study issues dealing with modeling different kinds of constraints in DEC-POMDPs such as temporal and resource constraints or dependencies between actions. We then propose approximate approaches for solving DEC-POMDPs that take into account constraints on actions. Our approach consists in exploiting the structure of the interactions in order to scale up and to be able to consider sizes of problems such as those encountered in real-world applications. In particular, we focus on distributed planning methods and pay particular attention to the efficiency and scalability of our algorithms. Finally, we study non-stationary settings where the dynamics of the system change over time and the agents need to detect these changes to adapt their strategies accordingly.

The contributions presented in this chapter have been motivated by real-world applications. Several issues stem from multi-robot planning problems and has led to some implementations on real mobile robots.

3.1 Research context

The work presented in this chapter has been initiated during my PhD supervised by Abdel-Illah Mouaddib at the GREYC lab in the University of Caen - Normandie. I have then pursued this work over the last decade. The work on DEC-POMDP models and algorithms have initially been motivated by multirobot exploration scenarios and implemented on real-robots at the GREYC. Part of the work on Dec-MDP and MDP decomposition have been developed during the PhD co-supervision

of Guillaume Lozenguez (co-supervision with Abdel-Illah Mouaddib fom the GREYC, Lounis Adouane and Philippe Martinet from the LASMEA in Clermont-Ferrand). The results obtained during this PhD were implemented on Pioneer robots and deployed on the PAVIN platform in Clermont-Ferrand. Finally, the work on non-standard Markovian models (especially on non-stationary environment) has been developed in collaboration with Paul Weng and our PhD student Emmanuel Hadoux.

3.2 Background on Markov Decision Processes

In this section, we review the various Markov decision models and introduce solving methods to plan the decisions of the agents. We end this section with a discussion about the current limitations of these models.

Markov Decision Processes (MDPs) and Partially Observable Markov Decision Processes (POMDPs) are standard formal frameworks for modeling and solving single-agent sequential decision problems under uncertainty. The Decentralized Partially Observable Markov Decision Processes (DEC-POMDPs) extend these models to cooperative multiagent sequential decision making. These models make the Markov assumption that is, the probability distribution over the next states depends only upon the current state of the system and the action taken. Moreover, transition and reward functions are assumed to be stationary, i.e. they do not change over time.

3.2.1 *Single-agent decision making*

Markov Decision Processes extend Markov chains to decision-making problems where the current state of the system is exactly observed (Puterman, 1994). At each time-step, an agent chooses an action to execute that influences transition probabilities over the next states.

Definition 23. ***Markov Decision Processes***
A Markov Decision Process is defined as a tuple $\langle S, \mathcal{A}, T, R \rangle$ *with:*

- S*, a finite set of states,*
- $\mathcal{A}$*, a finite set of actions,*
- $T : S \times \mathcal{A} \to Pr(S)$*, a transition function over the states,*
- $R : S \times \mathcal{A} \to \mathbb{R}$*, a reward function.*

The transition function formalizes the probabilistic outcomes of the actions. $T(s'|a, s)$ *is the probability of reaching state* s' *from state* s *after performing action* a*.*

The reward function specifies the objectives of the agent. $R(s, a)$ *is the reward obtained by the agent when action* a *is performed from state* s*.*

A solution for an MDP is a *policy* π, i.e. a sequence $(\delta_0, \delta_1, \ldots, \delta_t, \ldots)$ of *decision rules* where $\delta_t : S \to \mathcal{A}$ dictates to the agent which action to take for each state at time-step t. The *horizon* of the decision problem defines the number of steps during which the agent has to take decisions. The horizon can be *finite* or *infinite*.

A policy π can be valued at time-step t by the following equation that computes the expected discounted total reward from state s:

$$V^{\delta_t}(s) = R(s, \delta_t(s)) + \gamma \sum_{s' \in S} T(s'|s, \delta_t(s)) \times V^{\delta_{t+1}}(s') \tag{2}$$

The value $V^{\delta_t}(s)$ is defined as the immediate reward obtained by the agent ($R(s, \delta_t(s))$) plus the weighted sum of the values of the next states s' when the policy $\pi = (\delta_0, \delta_1, \ldots, \delta_t, \ldots)$ is applied. V is referred to as the *value function* of π and Equation 2 is the *Bellman equation* (Bellman, 1957).

Optimally solving an MDP consists in identifying a policy π^* that maximizes the expected discounted sum of rewards:

$$\pi^*(s) = \arg\max_a \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) \times V^{\pi^*}(s') \right\} \tag{3}$$

Interestingly, it has been proved that the optimal policy of an MDP is *stationary*, i.e. for each time-step t, $\delta_t = \delta_0$. An optimal policy is thus a decision rule mapping each state s to an action a. The most widespread algorithms for optimally solving MDPs are the *Value Iteration* (Bellman, 1957) and the *Policy Iteration* (Howard, 1970) algorithms.

Decision Making under partial observability

Unfortunately, in many sequential decision problems, the agent is not able to have full knowledge about the state of the system at each decision step. Instead, after each action, the agent receives partial information (observations) about this state and must take decisions on the basis of these observations. The Partially Observable Markov Decision Process (POMDP) model (Puterman, 1994) extends MDPs to partially observable settings.

Definition 24. ***Partially Observable Markov Decision Processes***
A Partially Observable Markov Decision Process is defined as a tuple $\langle S, \mathcal{A}, T, R, O, \Omega \rangle$ *with:*

- $S, \mathcal{A}, T, R$ *as defined for MDPs,*
- O*, a finite set of observations,*
- $\Omega : S \times \mathcal{A} \to Pr(O)$*, an observation function.*

Since the agent cannot observe the state of the system, she has to choose for her next action depending on the sequence of observations $o_{1:t} = (o^1, o^2, \cdots, o^t)$ made so far (where o^t is the observation made at time-step t). However, it is not necessary to keep track of this history at each decision step. The agent can instead compute a probability distribution over states $P(s_t|s_0, \ldots, s_{t-1})$ called *belief state* that defines the likelihood that the system is in state s at t (Åström, 1965). It has been proved that maintaining this distribution is sufficient and complete information to make optimal decisions. A policy is thus a mapping from belief states to actions. A POMDP can be thought as an MDP where the states are in fact belief states. Nonetheless, it has to be noticed that the state space of such a belief-MDP is continuous.

Fortunately, in the finite-horizon case, the value function of a POMDP is piecewise-linear-convex (PWLC) on the space of belief states. Optimal algorithms have thus been proposed to solve POMDPs such as *Witness* (Kaelbling et al., 1998) and *Incremental Pruning* (Cassandra et al., 1997). Nevertheless, these algorithms fail to scale to large problems. In fact, finding optimal policies for finite-horizon POMDPs has been proved to be PSPACE-complete (Papadimitriou and Tsitsiklis, 1987). In the infinite-horizon case, Madani et al. (2003) proved the undecidability of POMDPs.

Research works have therefore focused on the development of approximate algorithms such as *Heuristic Search Value Iteration (HSVI)* (Smith and Simmons, 2004), *SARSOP* (Kurniawati et al., 2008) or *Point-Based Value Iteration* (Pineau et al., 2003).

The *Partially Observable Monte-Carlo Planning* (POMCP) algorithm (Silver and Veness, 2010) is currently one of the most efficient online algorithms to approximately solve large-sized POMDPs. To choose an action at a given time-step, POMCP runs an effective version of Monte-Carlo Tree

Search (MCTS), called UCT (Upper Confidence Bounds (UCB) applied to Trees) from the current belief state. This Monte Carlo search uses a black-box simulator of the environment thus avoiding to explicitly represent probability distributions. The simulator runs a fixed number of simulations in order to evaluate the actions before performing in the real environment the best one found in the search tree. Moreover, POMCP uses a particle filter in order to approximate belief states. POMCP is guaranteed to converge towards the optimal solution as the number of simulations performed by the simulator increases.

3.2.2 *Multiagent decision making*

Although POMDPs and MDPs provide powerful frameworks for decision-theoretic planning under uncertainty, they consider *single* agent decision problems. The Dec-POMDP[2] framework extends POMDPs and MDPs to multiagent cooperative settings where the agents aim at maximizing a common performance criterion[3]. In a Dec-POMDP, each agent receives an individual observation and makes her decision solely based on her local information about the system (this corresponds to the right part of Figure 8). In particular, each agent only observes her own actions and does not observe the actions of the other agents. At each decision step, the transition to the next state and the reward depend on the joint action taken by all the agents. Since agents are assumed to be cooperative, the immediate reward for executing a joint action a from a state s is awarded to the team.

Definition 25. ***DECentralized Partially Observable Markov Decision Processes***

A DECentralized Partially Observable Markov Decision Process (Dec-POMDP) (Bernstein et al., 2002a) is defined as a tuple $\langle \mathcal{N}, S, \mathcal{A}, T, O, \Omega, R, b_0 \rangle$ *where:*

- $\mathcal{N} = \{1, ..., n\}$ *is a finite set of* n *agents,*
- S *is the finite set of world states* s*,*
- $\mathcal{A} = \{\mathcal{A}_1 \times \cdots \times \mathcal{A}_n\}$ *is the finite set of possible joint actions* $a = \{a_1, ..., a_n\}$ *such as* $\mathcal{A}_i$ *is the action set of agent* i *and* $a_i \in \mathcal{A}_i$*,*
- T *is the transition function giving the probability* $T(s'|s, a)$ *that the system moves to state* s' *while executing the joint action* a *from state* s*,*
- $O = \{O_1 \times \cdots \times O_n\}$ *is the set of joint observations* $o = \{o_1, ..., o_n\}$ *where* o_i *is the individual observation of agent* i*,*
- Ω *is the observation function giving the probability* $\Omega(o|s, a)$ *of observing* o *when executing the joint action* a *from state* s*,*
- $R(s, a)$ *is the reward obtained when executing the joint action* a *from state* s*,*
- b_0 *is the initial probability distribution over the set of states at* $t = 0$*.*

As for POMDPs and MDPs, a planning horizon h defines the number of decision steps $t \in \{0, 1, \cdots, h-1\}$ until the problem terminates. In this document, we focus on finite-horizon Dec-POMDPs.

[2] The interested reader can refer to (Amato et al., 2013), (Oliehoek and Amato, 2016) and Chapter 7 of (Kochenderfer et al., 2015) for a detailed overview of researches related to this framework.

[3] In this chapter, we will focus on cooperative agents. Partially Observable Stochastic Games (POSG) generalize Dec-POMDP to settings where each agent receives an individual rewards (Hansen et al., 2004). This framework is not addressed in this document.

Optimally solving a Dec-POMDP consists in finding a joint policy $\pi = \{\pi_1, ..., \pi_n\}$ that maximizes the common performance measure of the agents. In finite horizon problems, the undiscounted expected sum of rewards is commonly used to define the performance measure to optimize. π_i is the individual cooperative policy of agent i and maps each possible history of observations of the agent i to an individual action a_i. In fact in the multiagent setting it is not sufficient for an agent to maintain a belief over the states, as it is done in POMDPs. Indeed, each agent is only aware of her own actions and observations. Hence, an agent does not have access to the actions and the observations of the other agents. In order to predict the actions of the other agents, a belief for an agent i should specify probabilities over histories / policies / types / beliefs of the other agents $j \neq i$ (Oliehoek and Amato, 2016).

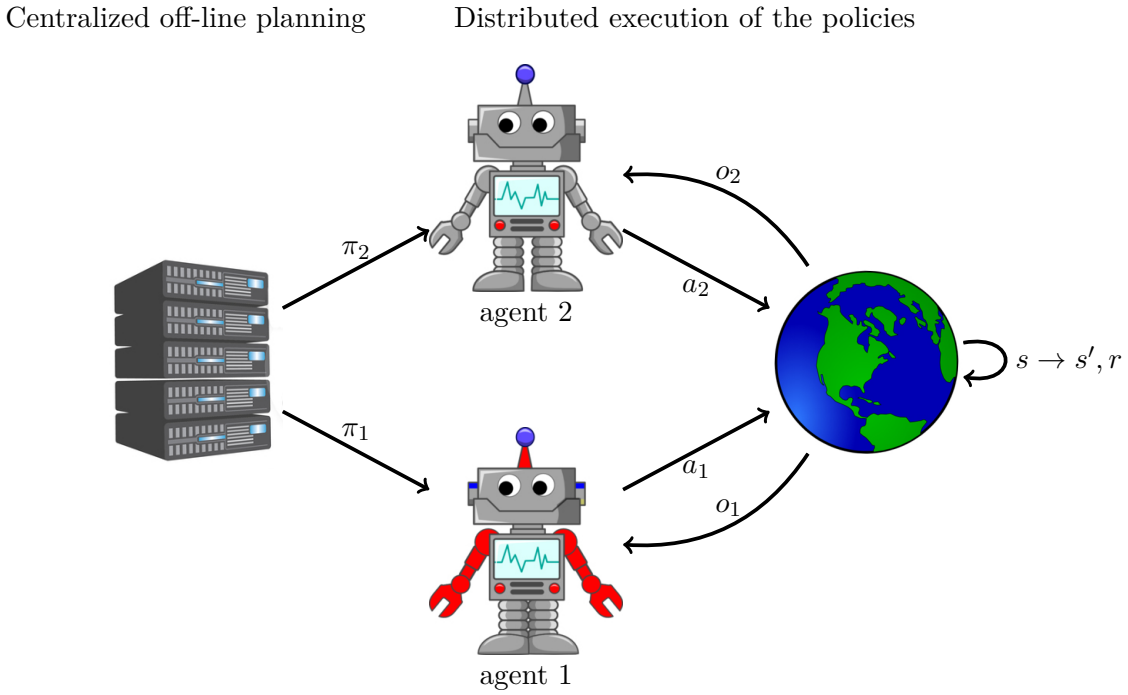


Figure 8: Centralized policy computation and distributed control in Dec-POMDPs

Planning for a Dec-POMDP over a *finite* horizon h involves searching for an optimal policy over the set of joint policies of length h. For each decision step $t \in [0, h-1]$, the individual policy of an agent i maps each possible observation history of length t to an action[4]. A deterministic individual policy can be represented as a decision tree. Figure 9 illustrates an individual policy for $h = 3$ as a tree. Each node corresponds to an action and each edge is labeled by a possible individual observation. At $t = 0$, the agent takes action a_1. If she observes o_1, the agent then takes at $t = 1$ action a_1. If she observes o_2, the agent then takes at $t = 1$ action a_2. A joint policy for n agents is thus a set of n policy trees. It has to be noticed that the number of joint policies grows doubly exponentially with the horizon h. Moreover, evaluating a joint policy takes exponential time in both the number of agents and the horizon h. It is thus not surprising that optimally solving a finite-horizon Dec-POMDP is NEXP-Complete (Bernstein et al., 2002a).

It is important to note that most existing algorithms for solving Dec-POMDPs compute a joint policy in a centralized way. Planning is performed before the execution (also called the "off-line" phase) by a central entity. Policies are then distributed among the agents and executed individually

[4] Note that we restrict our discussion to deterministic policy since it has been proved that every Dec-POMDP has at least one pure optimal policy (Bernstein et al., 2002a).

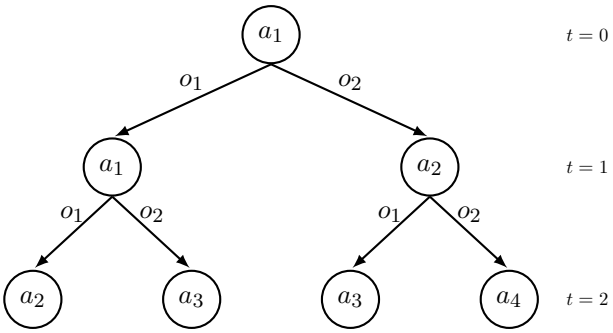


Figure 9: An individual policy tree for $h = 3$

by each agent. This approach is illustrated on Figure 8. The term *DECentralized* thus only refers to the control and most solving approaches consist in *centralized planning for decentralized control.*

Exact algorithms

Several exact algorithms have been proposed to optimally solve finite-horizon Dec-POMDPs. Three kinds of approaches can be identified: *dynamic programming* algorithms, *heuristic search* algorithms and approaches that *convert the Dec-POMDP into a single-agent POMDP.*

The dynamic programming algorithm for Dec-POMDP proposed by Hansen et al. (2004) iteratively generates the policies from the bottom-up. The policy of an agent i from decision step t is represented as a tree of depth $h - t$. For each decision step t (from the last one to the first one), the algorithm proceeds into two stages. It first constructs the possible sub-tree policies at t by adding an action at the root of the sub-tree policies computed for the decision step $t + 1$. The algorithm then eliminates dominated strategies among the set of sub-tree policies that have just been generated. Although dynamic programming computes an optimal policy for finite-horizon Dec-POMDPs, the elimination of dominated strategies is often not sufficient to maintain tractable sets of policies and the algorithm fails to scale well.

The optimal policy can also be computed in a top-down fashion using heuristic search algorithms. The Multiagent A* (MAA*) iteratively builds a search-tree where each node of depth t corresponds to a partial joint-policy for the first t decision steps (Szer et al., 2012). At each iteration, the algorithm selects a node to expand using a heuristic function and adds to the search tree all possible children of this node i.e. all policies extending the partial joint policy of the selected node to one further step. The MAA* algorithm returns an optimal solution but it may require to develop a very large number of nodes. In the worst case, the algorithm constructs the complete search tree and enumerates all possible joint policies.

Building on the fact that the optimal policy is computed by a central planner which is aware of the possible states and joint histories, Dibangoye et al. (2016) proposed to recast the Dec-POMDP problem as a continuous-state MDP with a piece-wise linear and convex optimal value function. Efficient POMDP and continuous-state MDP methods can then be used to solve this new formulation. Experiments show that this approach allows for considering significantly longer planning horizons than other exact approaches. For instance, the multiagent tiger problem (Nair et al., 2003) is currently solved up to horizon 10 whereas other approaches do not scale up to horizon 5. The recycling robot problem (Amato et al., 2007) is solved up to horizon 100 whereas other approaches do not scale up to horizon 5. This framework is currently the most efficient approach for optimally solving Dec-POMDPs.

Tractable sub-classes of Dec-POMDPs

Given the computational complexity of Dec-POMDPs, it is natural to wonder if one could identify some properties of the problems that would reduce the complexity of computing an optimal solution.

FULL JOINT OBSERVABILITY In analogy to the relationship between POMDPs and MDPs, Dec-POMDPs can be restricted to settings where the agents jointly fully observe the state of the system, i.e. where the aggregation of the agents' local observations allows for deducing the exact state of the system.

Definition 26. ***DECentralized Markov Decision Processes***

A Decentralized Markov Decision Process (Dec-MDP) is a Dec-POMDP where the state of the system is jointly fully observable.

More formally, the state of the system is jointly observable if the following condition holds:

$$\textit{If } \Omega(o = \langle o_1, \cdots, o_n \rangle | s, a) > 0 \textit{ then } Pr(s | \langle o_1, \cdots, o_n \rangle) = 1$$

It has to be noticed that each agent does not individually observe the state of the system and receives partial observations. As the agents do not share their observations about the system during the execution, they cannot deduce exactly the state of the system at each time-step. A Dec-MDP is thus defined (like Dec-POMDPs) as a tuple $\langle \mathcal{N}, S, \mathcal{A}, T, O, \Omega, R \rangle$ (components related to the observations cannot be omitted but the initial distribution on the states is known). The complexity of Dec-MDPs remains the same as the one of Dec-POMDPs (NEXP-complete (Bernstein et al., 2002a)).

INDEPENDENT TRANSITIONS AND OBSERVATIONS In various real-world applications, some independence between the agents. Although the agents aim at maximizing a common performance measure, they may have independent individual states, the outcomes of their actions may not interact or their observations may not depend on each other. Different levels of independence have been investigated and it has been shown that some properties reduce the complexity of the multiagent sequential decision problem (Goldman and Zilberstein, 2004).

Let first define Factored Dec-MDPs as introduced in (Oliehoek et al., 2008).

Definition 27. ***Factored Dec-MDP*** *A factored Dec-MDP is a Dec-MDP where the state space can be decomposed into $n+1$ distinct components such as $S = S_0 \times S_1 \times \cdots \times S_n$ where S_i (with $i > 0$) is the state space of agent i and S_0 is the set of properties of the environment that is not affected by the actions of the agents. A state s is an assignment of factors such as $s = (s_0, s_1 \cdots, s_n)$ with $s_i \in S_i$.*

A factored Dec-MDP is said to be **transition independent** if the probability that an agent i moves from a state s_i to a state s'_i only depends on the action taken by the agent i. More formally:

Definition 28. ***Transition-independent Dec-MDP***

A factored Dec-MDP is said to be transition independent if the transition function can be decomposed as a product of probabilities such as:

$$T(s'|s,a) = \prod_{i=1}^{n} T_i(s'_i|s_i, a_i)$$

This property holds when the agents' actions do not interact and the outcomes of each individual action are thus independent of the actions taken by the other agents.

Similarly, it is possible to consider independence of observations.

Definition 29. ***Observation-independent Dec-MDP***

A factored Dec-MDP is said to be observation independent if the observation function can be decomposed as a product of probabilities such as:

$$O(o|s,a) = \prod_{i=1}^{n} O_i(o_i|s_i,a_i)$$

This property holds when the observations of an agent i only rely on the action a_i taken by agent i and on the resulting local state s_i. These independence properties commonly hold when the agents have limited sensors and perform independent and distant tasks in the environment.

If a Dec-MDP has independent observations and transitions, then the Dec-MDP is locally fully observable. It has been proved that a Dec-MDP with independent transitions and observations is NP-complete (Goldman and Zilberstein, 2004). An optimal algorithm, the Coverage Set Algorithm (CSA), has been developed to solve such Dec-MDPs (Becker et al., 2004).

LOCAL INTERACTIONS In the general Dec-POMDP framework, each agent may interact with any other agent. Inspired by distributed sensor applications, Nair et al. (2005) proposed the Networked Distributed POMDP (ND-POMDP) framework to account for locality of interactions. The interactions are represented as a graph or an hypergraph where each agent i is represented as a vertice and only interacts with her neighbors in the graph. ND-PODMPs form in fact a sub-class of Dec-POMDPs with transition and observation independence but without the assumption on joint full observability. The reward function is decomposed into a sum of rewards over the sets of neighboring agents. ND-POMDPs have the same worst case complexity as Dec-POMDPs but exploiting the structure of the reward function leads, in practice, to more efficient algorithms.

Approximate algorithms

Since the scalability of exact algorithms remains quite limited, one can sacrify the optimality requirement and focus on finding a good approximate solution.

The main drawback of dynamic programming for Dec-POMDPs is memory requirement. In order to limit the necessary amount of space, alternatives to the exact dynamic programming (DP) algorithm have been proposed. Seuken and Zilberstein (2007) proposed the Memory Bounded Dynamic Programming (MBDP) that combines the bottom-up dynamic programming approach with top-down heuristics to prune the policy trees computed by DP. At each iteration step of the DP algorithm, heuristics are used to sample the most likely belief states at this step and select the best policy trees for these belief states. A parameter fixes the maximum number of policy trees that can be selected at each step of the algorithm. Different approaches have been proposed to improve memory bounded dynamic programming by tuning observation representations or replacing the full backup performed at each step of the policy computation (see (Amato et al., 2013) for more details).

Nair et al. (2003) proposed the Joint Equilibrium Based Search for Policies (JESP) algorithm, to solve transition and observation independent Dec-MDPs. JESP algorithm relies on an alternative improvement of individual policies: at each iteration, all the individual policies but one are fixed and a best response to these fixed policies is computed by the remaining agent. It is proved that JESP algorithm will eventually converge to a Nash equilibrium. Several improvements over JESP have been proposed to improve policy computation (Nair et al., 2003, 2005; Varakantham et al., 2007).

Mutiagent learning methods

The exact and approximate methods presented so far assume that the dynamics of the problem is known. In particular, the uncertainty about the system and the agents' objectives can be represented

by the functions $\mathcal{T}$, $\mathcal{O}$ and $\mathcal{R}$. In complex and/or (partially) unknown environments, defining these functions comprehensively can be difficult and even impossible. As soon as a simulator of the environment is available or the agents can train sampled policies in the environment, *reinforcement learning methods* can be investigated to solve Dec-POMDPs. One of the key challenges of these methods consists in providing compact representations of the policies learned (or compact representations of the value functions learned). Wu et al. (2013) developed a model-free learning approach based on Expected Maximization to solve infinite-horizon Dec-POMDPs. The approach iteratively improves Finite State Controllers (FSCs) by first drawing trajectories to estimate the probability on future states and rewards, and then improving FSCs so as to maximize the reward likelihood. Although EM methods scale well to large number of agents, they are very sensitive to initial conditions and may converge to poor local optima.

Very recently, a growing amount of work has been interested in developing Deep Reinforcement Learning methods to solve Dec-POMDPs (Foerster et al., 2016, 2017; Gupta et al., 2017; Omidshafiei et al., 2017). These approaches combine Reinforcement Learning and Deep Q-Networks. Experimental results showed that these approaches are able to handle large state and action spaces. Nonetheless, it is important to note that most deep RL approaches perform centralized off-line planning and require a large amount of data to be available (Amato, 2018).

One of the only distributed learning approach has been proposed by Peshkin et al. (2000) and consists of a gradient descent method for computing policies represented as finite state factored-controllers in Partially Observable Stochastic Games.

3.2.3 *Dec-POMDP limitations*

Despite the genericity and the amount of work dedicated to Dec-POMDPs, applying this model to solve real-world problems remains challenging. Difficulties mainly concern the expressiveness of the model on the one hand and the applicability of the algorithms to real settings on the other hand.

EXPRESSIVENESS OF THE MODEL Standard multiagent Markov models assume that all the actions have the same durations and the agents are thus fully synchronized. Indeed, at each time-step, all the agents are supposed to take a new decision. Nonetheless, the actions of the agents are often much more complex: they may last over several time-steps and their duration is usually stochastic. Hence, at each timestep, some of the agents take new decisions while the others keep on executing their current actions for an uncertain amount of time.

Different kinds of constraints may be related to the execution of the actions and all actions may not be executable at each time-step. In fact, the agents may have to respect constraints on the action execution such as temporal, resource or precedence constraints. Hence, an action may have to start before a given time-step and to finish before a given deadline. Moreover, some actions may have to be executed before others can start. Such constraints are not taken into account in the original Dec-POMDP model.

Finally, Markovian models run under the assumption that the environment is stationary, i.e. the transition and the reward functions do not evolve over time. In many real-world applications, this assumption does not hold and the sources of non-stationarity are diverse. For instance, the environment may change due to external events such as weather variations or because of human activities. In multiagent systems, from the point of view of an agent, a change of behavior of another agent (*e.g.*, due to learning for instance) may affect the stationarity of the environment. Several issues then arise from the non-stationarity of the environment among which we can mention topics dealing with the representation of non-stationary environments, the detection of changes in the environment dynamics and the policy computation.

APPLICABILITY OF SOLVING METHODS The computational complexity of optimally solving a Dec-POMDP furthermore limits the applicability of the model. Although improving the scalability of Dec-POMDPs has been a very active topics in the research community, optimally solving large size of problems such as the one encountered in real-world problems remains an open issue. In particular, the number of agents is a limiting factor. Existing optimal approaches are usually unable to solve problems involving more than two agents whereas real applications can count tens of agents.

Moreover, most existing algorithms consist in centralized off-line planning. Centralized approaches make coordination between individual policies easier but they may not be desirable or applicable in real-world problems. First, even solving small problems requires a huge amount of memory and quickly becomes untractable for a central planing entity. Distributed planning methods can then be preferred in order to divide the problem resolution among several entities. Second, when the environment is highly dynamic or unknown (even partially), the agents may have to adapt their strategies on-line to unforeseen situations and *distributed on-line planning* would thus be recommended.

3.2.4 *Bridging the gap between real-world and Dec-POMDPs*

In order to exemplify the limitations of Dec-POMDP models, we end this section by describing real-world problems of decentralized control in uncertain and partially observable environments. More specifically, we consider multi-robot exploring scenarios that have been investigated in our research work.

Cooperatively exploring an environment with a fleet of robots is a persistent topic in mobile robotics (Burgard et al., 2005). This problem is encountered in various applications like planetary rovers (Bernstein et al., 2001), robot rescue (Akin et al., 2013), warehousing (Amato et al., 2015), surveillance (Kochenderfer et al., 2015), etc. The goal of the robots is to cover an unknown environment and/or to perform a set of tasks among the environment. In this context, the use of several cooperative robots improves the capabilities, the reliability and the efficiency of the system.

COOPERATIVE EXPLORATION OF DISTANT PLANETS In (Beynier and Mouaddib, 2011b), we considered planetary rovers scenarios where a team of heterogeneous rovers aims at performing a set of exploring tasks (scientific measurements) on a distant planet. Since the environment is unknown, rovers must deal with uncertainty regarding the duration of actions and the consumption of resources. Once a day, a mission of a hundred tasks is sent from the Earth to the robots using a satellite. Due to orbital rotation of the satellite, the agents can communicate with the spatial centre on Earth only during a specific temporal window. For the rest of the day, the rovers must complete their tasks in autonomous way. In addition, because of bandwidth limitations and of distance between the robots and obstacles, the robots are often unable to communicate directly. Rovers are equipped with different tools and measuring devices so they have to coordinate to perform exploring tasks. Obviously, tasks may have different lengths and the length of a task may vary depending on execution conditions. To guarantee valuable task outcomes, temporal and precedence constraints on task execution have also to be respected. For example, on distant planets, pictures must be taken at sunset or sunrise because of lighting constraints.

COOPERATIVE EXPLORATION AMONG A SET OF POINTS OF INTEREST In (Lozenguez et al., 2012, 2016), we considered multi-rover exploration scenarios where a fleet of rovers has to visit a set of key-positions (also called "points of interest") identified by an UAV in an outdoor environment[5]. Such planning problems can be met in search and rescue applications (see Introduction of the document)

[5] This work has been developed as part of the ANR project R-Discover.

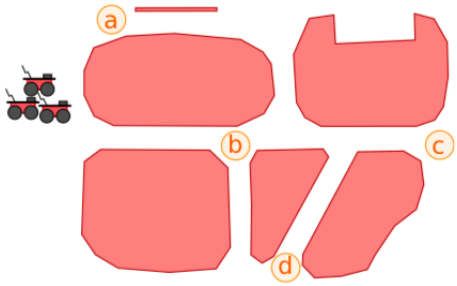


Figure 10: Exploration scenario involving 4 points of interest (a, b, c and d)

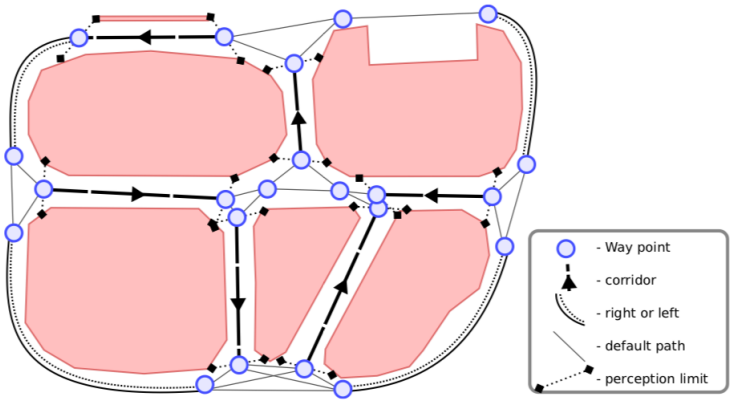


Figure 11: Road-Map of the exploration scenario at left

where an UAV (Unmanned Aerial Vehicle) can obtain an overview of the environment and send an abstract road-map to the robots. Figure 10 exemplifies such a scenario.

A Probabilistic Road-Map (Kavraki et al., 1996) is defined as a graph $\langle W, P \rangle$ where W is the set of way-points (nodes) and P is the set of possible paths in the environment (edges). The set of way-points W is composed of the points of interest I to visit in addition to the environment way-points (points around the known obstacles). Figure 11 describes the Road-Map of the scenario depicted in Figure 10. To each path $p \in P$ is associated a cost and a probability distribution formalizing the stochastic outcomes of moving through the path. Each time a rover successfully visits a point of interest, she obtains a reward related to this key position.

Because of imperfect actuators and of limited and noisy sensing capabilities, exploring robots act in *uncertain and partially observable environments*. In particular, moving from a way-point to another and visiting a point of interest takes an uncertain amount of time. Moreover, because of bandwidth limitations and limited range of communication, the robots cannot communicate unless they are close to each other. Hence, effective distributed control methods must be developed to allow each robot to *make cooperative decisions based on local knowledge and observations about the environment and the other robots.*

Such decentralized robotic decision problems naturally fall into the scope of Dec-POMDPs (Bernstein et al., 2001; Matignon et al., 2012a; Amato et al., 2015, 2016). However, this general framework does not handle various characteristics of multi-robot exploration problems such as asynchronous action execution and constraints on actions. In (Beynier and Mouaddib, 2011b), we identified three kinds of constraints that are frequently encountered in multi-robot exploration scenarios and that are not considered in the original Dec-POMDP framework: temporal constraints, precedence constraints and resource constraints. Temporal constraints limit time intervals where a task can be executed. Precedence constraints define dependencies between the tasks and result in dependencies between the agents. Finally, each agent is initially endowed with a limited amount of resources that are consumed upon task execution. Executing a task requires power, storage (storing pictures or measurements) or bandwidth (data communication). Resource constraints must therefore be respected in order to successfully complete a task.

3.3 Constrained Dec-POMDPs for multi-task planning

The above multi-robot exploration scenarios can be envisioned as *multi-task planning problems with constraints* where a set of agents $\mathcal{N}$ has to execute a set of tasks $\mathcal{T}$ respecting different kinds of

constraints. A task may consist in exploring a point of interest, moving between two way-points, making measurements on a site, taking photography, etc.

Based on the study of the application domains described above, we proposed the following characterization of a task:

Definition 30. *A **task** $\tau_i \in \mathcal{T}$ is characterized by:*

- *a probability distribution over the finite set of possible durations of the task,*
- *a probability distribution over the finite set of possible resource consumptions of the task,*
- *a **temporal window** $TC_{\tau_i} = [EST_{\tau_i}, LET_{\tau_i}]$ during which the task must be executed. EST_{τ_i} stands for the Earliest possible Start Time of the task and LET_{τ_i} is the Latest possible End Time of the task,*
- *a **set of predecessor tasks** $Pred_{\tau_i}$ that must be finished before τ_i can start,*
- *a **location** in the environment (Cartesian coordinates or a way-point in a toplogical map) describing spatial constraints related to the execution of the task. Since the number of possible locations can be infinite, we assume that the environment can be discretized into a finite number of possible locations. For instance, in a multi-rover scenario, the set of way-points identified in the road-map corresponds to the possible locations of the agents*[6]*,*
- *a **reward** r_{τ_i} related to the importance of the task for the agents.*

When an agent tries to execute a task τ_i at t, the execution of the task succeeds if all the constraints related to the task are respected. The agent then obtains the reward associated with the task and can turn to the execution of another task. When the constraints related to a task τ_i are violated, the agent fails to execute τ_i. We distinguish partial failures where the agent can retry to execute the task later, from permanent failures where the task cannot be executed anymore.

3.3.1 *Constraint modeling*

In order to solve multi-task planning problems with constraints and to compute valuable joint policies, we introduce *Constrained Dec-POMDPs* where the components of the model are defined so as to formalize constraints on action execution and to handle asynchronous execution of actions. Like standard Dec-POMDPs, a *Constrained Dec-POMDP* is defined as a tuple $\langle \mathcal{N}, S, \mathcal{A}, T, O, \Omega, R, b_0 \rangle$ with the following specific features:

STATE SET S: A state of the system contains all the information that may influence the decision of the agents. In multi-task execution problems, a state gives the set of already completed tasks. Because of resource constraints, the level of available resources of each agent must also be specified. In order to fulfill spatial constraints, the location of each agent has to be registered. Since a task can last over several time-steps and task durations are stochastic, a state must indicate, for each agent i, whether i has just completed a task and if not, the start-time of the ongoing task must be stipulated. In fact, the time already passed to execute a task influences the probability to complete the task at the next time-step, i.e. it influences transition probabilities over states at the next time-step. Hence, by specifying the start-times of the tasks, the model complies with the Markov assumption. Finally, since probabilities on action outcomes rely on the current time t (because of temporal constraints), the value of t is added to the state description in order the transition function to be stationary.

[6] The control of moves between two way-points is delegated to a reactive module of lower level in the robot architecture.

One can notice that the description of the states can be reduced when some kinds of constraints do not have to be considered. For instance, if a problem does not involve resource constraints, levels of resources can be omitted from the state description.

ACTION SET $\mathcal{A}$: At each time step t, some agents must take a decision about the execution of a new task while the others keep on executing their current task. Making a decision about a new task consists in deciding which task to execute next and when. Indeed, because of precedence constraints, an agent may prefer to wait before starting to execute a task in order to increase the likelihood that the predecessors of the selected task have been executed. An action thus consists in *"Executing task τ_i at time t'"* such that $t' \geq t$ (denoted by $E(\tau_i, t')$). Obviously, the envisioned start-time t' must comply with the earliest possible start-time and the latest possible end-time of the task. In order to fulfill the requirements of Dec-POMDP models where each agent takes a decision at each time-step, we also introduce a *nop* action which is used for the agents who do not take new decisions, i.e. who are still executing a task.

TRANSITION FUNCTION T: The transition function of the constrained Dec-POMDP has to take into account constraints on action execution. If precedence or location constraints are not respected when an agent starts executing a task τ_i, the execution of the task immediately fails. When precedence and location constraints are fulfilled (i.e. when the execution of the tasks successfully starts), the probabilistic outcomes of the action can be deduced from probability distributions on resource consumption and task duration. Indeed, these distributions allow for computing the probabilities on the end-times of the task and the probability that the agent does not lack of resources during the execution of the task.

OBSERVATION SET O AND FUNCTION Ω: The observation set and observation function are similar to the ones proposed in the original Dec-POMDP model. In our context, we assume that each agent only observes her own actions and is only aware of the outcomes of her tasks. An observation for an agent thus consists in observing whether her current task has been successfully completed or not.

In our definition of a constrained Dec-POMDP, each agent is assumed to know her location, the set of tasks she has completed, their related start-times and end-times and her available level of resources. The state of the system can be deduced by combining the agents' partial views of the system. The problem is thus jointly fully observable and falls into the class of the Dec-MDPs.

REWARD FUNCTION R: Each time a task is successfully executed, the agents receive the reward related to the task.

3.3.2 *Complexity of Constrained Dec-MDPs*

Although we consider jointly-fully observable problems, our constrained Dec-MDPs are not transition-independent nor observation-independent. Indeed, due to precedence constraints between the agents, the probability an agent succeeds to execute a task τ_i depends on the end-times of the predecessors tasks of τ_i. Since these predecessors may be executed by the other agents, transition probabilities of an agent rely on the other agents' actions and states. Moreover, the probability to observe the successful execution of an action depends of the other agents' actions. Our constrained Dec-MDPs thus inherit the high complexity of standard Dec-MDPs.

Proposition 12. *Optimally solving a Constrained Dec-MDP is NEXP-complete.*

Since the set of tasks to execute is finite and each task has to be executed only once, the planning horizon of a Constrained Dec-MDP can be upper bounded by the highest Latest End Time of the task[7]. It is known that the number of possible joint policies related to a Dec-MDP over a finite horizon h is:

$$O\Big(|\mathcal{A}^+|^{\frac{n(|\mathcal{O}^+|^h-1}{|\mathcal{O}^+|-1}}\Big)$$

where $|\mathcal{O}^+|$ and $|\mathcal{A}^+|$ respectively denote the largest individual observation set and the largest individual action set (Oliehoek, 2012). Moreover, the cost of evaluating such a joint policy is $O(|S| \times |\mathcal{O}^+|^{nh})$.

The size of the problems considered in multi-task decision problems are usually large. In fact, we aim at considering problems involving at least ten agents and a hundred of tasks. Since the set of completed tasks and temporal information are modeled in the states and are locally observable, large state and observation spaces are obtained. Moreover, the number of individual actions is $O(|\mathcal{T}| \times |ST^+|)$ where $|ST^+|$ is the largest number of possible start-times for a task (at worst $|ST^+| = h$). Although we considered finite planning horizon problems, the size of the horizon and the size of the Constrained Dec-MDPs make the problem untractable for optimal algorithms and most approximate approaches.

3.4 Constrained Dec-MDPs decomposition

Inspired by the success of decomposition techniques to improve the scalability of single-agent MDPs (Dean and Lin, 1995; Boutilier et al., 1999), we investigated how the structure of constrained Dec-MDPs can be exploited so that policies can be efficiently computed. The idea is to split the initial multiagent decision problem into loosely-coupled pieces that could be solved independently, at least to some extent.

In the following, we will depart from the hypothesis that tasks are allocated among the agents. In fact, in many application domains such as multi-rover exploration scenarios, the agents often have different capabilities. Combined with temporal, resource and spatial constraints, it is often possible to allocate the tasks among the agents before planning their execution (Hanna and Mouaddib, 2002; Abdallah and Lesser, 2005; Gerkey and Matarić, 2002; Esben et al., 2002). Given an allocation of the tasks among the agents, the problem is then to plan the start-times of the tasks such that the agents maximize their performance while respecting precedence, temporal and resource constraints. In (Lozenguez et al., 2013), we proposed a distributed approach to automatically allocate the tasks among a fleet of robots. This protocol consists of a series of simultaneous auctions where the agents evaluate their preferences among the tasks and exchange the tasks.

3.4.1 *Decomposition as a set of individual MDPs*

In (Beynier and Mouaddib, 2011b), we proposed splitting the initial multi-task and multi-agent planning problem formalized as a constrained Dec-MDP into a set of MDPs $\{m_1, m_2, \cdots, m_n\}$ where the MDP m_i formalizes the decision problem of the agent i. The main difficulty of such a decomposition arises from the definition of individual transition functions because of dependencies between the agents. The probability that an agent succeeds to execute a task τ_i depends on the end-times of the predecessors of τ_i. Since these predecessor tasks may be executed by the other agents, the transition probabilities of agent i from a state s_i rely on the other agents' actions and states, in particular on the

[7] Note that a tighter bound can be defined by propagating temporal constraints through the graph of tasks where each node corresponds to a task and edges represent precedence constraints between the tasks.

other agents' strategies. If these strategies are unknown, the transition function of the Constrained Dec-MDP cannot be directly decomposed as a set of independent individual transition functions.

Decomposing a constrained Dec-MDP consists in defining, for each agent i, an MDP $m_i = \langle S_i, \mathcal{A}_i, T_i, R_i \rangle$ with the following components:

- **Set of states S_i:** The states s of the constrained Dec-MDP are decomposed into a set of states $\{s_1, \cdots, s_i, \cdots, s_n\}$ where s_i contains all the information that is relevant to the decisions of the agent i. In our context, this information consists in the location of the agent i, the set of tasks completed by agent i, the current time-step t, the resources available to agent i, and the end-time of the last task executed by i. A state s_i also records partial failures of the tasks. In fact, this information gives insights about the other agents: if a task τ_i partially fails, the agent can deduce that at least one of the predecessors of τ_i has not been completed yet.

- **Set of actions $\mathcal{A}_i$:** The set of actions $\mathcal{A}_i$ of agent i contains all the actions related to the execution of the tasks allocated to agent i plus the *nop* action.

- **Transition function T_i:** The individual transition function T_i of an agent i gives the probability that the agent moves from one state to another when she executes an individual action ($T_i : S_i \times S_i \times \mathcal{A}_i$). Because of precedence constraints, these probabilities rely on the other agents' strategies. If the agent i knows the individual policies of the agents $j \neq i$, she can deduce probabilities on the end-times of the tasks executed by the other agents. Probabilities on the outcomes of the tasks allocated to agent i can then be estimated. We proposed to decompose the constrained Dec-MDP assuming a fixed set of policies for the agents allowing each agent i to estimate her transition probabilities. As we will explain below, this set of policies will be iteratively improved while solving the set of individual MDPs.

- **Reward function R_i:** The individual reward function R_i rewards the agent i each time she successfully completes a task.

Note that each MDP m_i formalizes temporal, precedence and resource constraints on task execution. The model also handles uncertainty on the duration for each task.

By decomposing a constrained Dec-MDP into a set of individual MDPs, we should break the high dimensionality of the multiagent decision problem into a set of smaller problems that could be solved in a distributed way. Instead of solving a Dec-MDP whose complexity is NEXP-complete, we aim at solving a set of MDPs. Nonetheless, solving these individual MDPs is not that simple. Because of dependencies between the agents, the individual MDPs cannot be solved independently. Indeed, the transition function of each individual MDP is defined assuming that the other agents follow some fixed policies. If these policies change (because the agents compute new strategies for their own MDP), individual transition functions must be updated.

3.4.2 *Distributed policy computation through Opportunity Cost*

We proposed a distributed solving approach allowing each agent to compute her own policy given the individual MDP formalizing her decision problem. This approximate approach consists in a series of simultaneous iterative improvements of individual policies.

From an individual point of view, each agent has to decide for the start-times of her allocated tasks. In order to respect precedence constraints, an agent may decide to delay, as far as possible, the execution of her tasks. Nonetheless, this may prevent the other agents from respecting temporal

constraints dealing with their own tasks. For purpose of coordinating the agents, we thus introduced the notion of Opportunity Cost (OC) to allow each agent to measure the effect of her decisions on the other agents.

Coordination based on opportunity cost

Opportunity cost is borrowed from economics where it refers to hidden indirect costs associated with a decision (Wieser, 1889). In our work, we used opportunity cost to measure the effect of an agent's decision on the other agents. When an agent i decides when to start the execution of a task τ_i, her decision influences all the successor tasks of τ_i. In order to obtain coordinated behaviors, i must therefore consider the influence of her actions on the execution of these tasks. In fact, i must consider the consequences of her decisions on the other agents' expected value (the agents which execute the successors of τ_i). In our approach, we defined the notion of expected opportunity cost to consider the expected loss in value provoked by the decision of an agent about a task τ_i on the agents j executing successor tasks of τ_i. The opportunity cost induced on an agent j when delaying by Δ_t the execution of her task τ_j measures the difference between the expect utility of j when she starts the execution of τ_j as soon as possible and the expected utility of j when the execution of τ_j is delayed by Δ_t.

We then modified the Bellman equation (Equation 4) to allow each agent i to select the best action to execute in a state s_i. The choice of the best action to execute from s_i results from a trade-off between the agent's expected utility and the expected opportunity cost provoked on the other agents:

$$\pi_i(s_i) = argmax_{E(\tau_{i+1},st),st\geq t}\Big(\overbrace{V(E(\tau_{i+1},st),s_i)}^{\text{Expected Utility}} - \overbrace{EOC(\tau_{i+1},st_{i+1})}^{\text{Expected Opportunity Cost}}\Big) \quad (4)$$

In order for each agent i to estimate the expected opportunity cost on the other agents, the agents have to communicate opportunity cost values along policy computation. In fact, as an agent computes her individual policy, she estimates her loss in expected utility induced by the other agents when they delay the execution of her allocated tasks. Loss values, i.e opportunity cost values, are then sent to the other agents $j \neq i$ that in turn use this information to update their policies and send new opportunity cost values. We refer the interested reader to (Beynier and Mouaddib, 2011b) for more explanation about opportunity cost definition and computation.

Distributed policy computation

In order to implement policy coordination based on expected opportunity cost, we proposed a distributed algorithm that iteratively improves the initial joint policy used to decompose the constrained Dec-MDP in order to compute an approximate solution to the multi-task planing problem. At each iteration k, the transition function of each agent is updated based on the joint policy computed during the previous iteration $k-1$. Each agent has therefore to be aware of all policy updates made by the other agents and these updates are broadcasted at the end of each iteration. The policy of each agent is then improved using a value iteration algorithm based on Equation 4. Hence, when an agent plans the actions related to the execution of a task τ_i, she coordinates her strategy by measuring the opportunity cost incurred by her decisions on the other agents.

Two versions of the algorithm have been proposed: a synchronized one where the actions related to only one task is revised at the same time and a parallel algorithm where the agents can revise the execution of different tasks at the same time. Policy improvements are iterated unless no more improvement is made by any agent. We proved the convergence of the synchronized version of our algorithm to a Bayesian Nash equilibrium whereas the desynchronized version is not theoretically guarantee to converge because of possible unaccurate estimates of opportunity cost. Experimentally, we did not find cases where the desynchronized version of our algorithm does not converge though.

Our approach is closely related to other co-alternative approaches such as Subjective MDPs developed by (Chadès et al., 2002) or the Joint Equilibrium based Search for Policies (JESP) (Nair et al., 2003). Although these methods could be executed in a distributed way, they have been developed from the point of view of a centralized planning and do not pay attention to limiting communication complexity. Hence, our distributed algorithm requires less communication. Indeed, these approaches revise the policy of only one agent at each iteration thus leading to more frequent policy communication. Moreover, these approaches do not handle constraints on action execution nor time extended actions.

Complexity analysis proved that our approach is polynomial time in the number of states and actions. Although we were not able to provide theoretical guarantees regarding the performance of the solutions in the general setting, we pointed out some relaxations of the constraints guarantying optimal policy computation (Beynier and Mouaddib, 2011b). Experimental results showed that our algorithms were able to efficiently coordinate multiagent systems with tens of agents executing hundreds of tasks. Our approach thus fulfills our initial ambitions since it can deal with large missions and compute good quality solutions respecting several kinds of constraints.

3.5 Hierarchical decomposition among space

In (Lozenguez et al., 2011, 2012), we considered a special case of the multi-task planning problem that we previously defined. This setting has been motivated by multi-robot scenarios developed in the ANR project R-Discover where we aimed at developing a fleet of mobile robots to visit a set of points of interest identified by an UAV flying over the area. The set of point of interest was assumed to be frequently updated and the robots had to responsively adapt their strategies to these changes. Hence, the robots were required to be able to compute their strategy in a distributed way while executing their mission. Computation time was particularly sensitive because we wanted the robots to be responsive to the UAV updates. In this setting, visiting a point of interest can be achieved by a single robot and we assumed that the visit of each point of interest is not influenced by the visit of the other sites.

Here, there is no precedence constraints between the tasks. Consequently, the multi-task decision problem can be modeled as a constrained Dec-MDP with independent observation, transition and reward functions. Such a Dec-MDP can be decomposed as a set of n individual and independent MDPs $\{m_1, \cdots, m_n\}$ and the complexity of computing an optimal solution for each MDP turns to be P-complete. Unfortunately, in our context, the number of states of each individual MDP m_i prevents on-line solving as soon as the problem involves too much points of interest and way-points. Indeed, the state space of an individual MDP increases exponentially with the number of points of interest allocated to the agent. More precisely, the state space size is $|W| \cdot 2^{|I|}$ where $|W|$ is the number of way-points in the road-map and $|I|$ is the number points of interest allocated to the robot.

We then proposed to exploit the road-map topology to decompose the individual MDP of an agent into a set of smaller and loosely-coupled sub-MDPs. Our approach consists in partitioning the individual MDP m_i of agent i (built for the whole set of points of interests allocated to the agent i) into a set of MDPs $\{m_i^1, m_i^2, \cdots, m_i^p\}$. The idea is to aggregate strongly connected states together in a sub-MDP m_i^j and to solve the sub-MDPs in a distributed way.

3.5.1 *Decomposition based on topological maps*

Generally, decomposing an MDP consists in building a partition of the state set as balanced as possible while minimizing connections between the sub-MDPs; that means minimizing the cuts on the transition set. In fact, minimizing the cuts allows for reducing coordination complexity between the sub-MDPs.

Such an approach is particularly efficient in problems with spatial constraints. Indeed, one can exploit the topological aspects of the environment to decompose the initial problem.

We defined a *greedy decomposition algorithm* that efficiently builds a partition of the road-map into a set of regions $\{\rho_1, \cdots, \rho_p\}$. The regions constitute a partition of the way-points identified in the road-map. The algorithm builds the regions one by one. For a given region, the most appropriate way-points to add to the region are iteratively selected: each way-point (not already selected) is assigned a score based on a ratio between (i) the number of transitions between this way-point and other way-points inside the region (ii) the number of transition between this way-point and other way-points outside the region. The way-point with the highest score is selected and added to the region. In order to get balanced regions, this ratio is weighted by the difference between the ideal size of the region and its actual size.

Once the partition of the road-map defined, a sub-MDP m_i^j is built for each region ρ_j. The state space and the action space of a sub-MDP are restricted to the way-points W^{ρ_j} and to the points of interest I^{ρ_j} belonging to the region ρ_j. The state space of a sub-MDP is thus reduced to $|W^{\rho_j}| \cdot 2^{|I^{\rho_j}|}$.

3.5.2 *Hierarchical solving*

Once the sub-MDPs defined, it would be expected to solve these local decision problems in a distributed way using standard MDPs algorithms. Nonetheless, as the agent may move from one region to another, some states of a region ρ_j may have successor states belonging to another region ρ_k. Following Bellman equation, the value of a state in a sub-MDP m_i^j related to an agent i and a region ρ_j may thus depend on states belonging to another sub-MDP m_i^k related to a neighboring region ρ_k. When solving the sub-MDP m_i^j, the values of the states belonging to the other sub-MDPs m_i^k $(k \neq j)$ have to be taken into account.

In order to coordinate the exploration strategy of an agent among the regions, we envisioned a hierarchical approach. More specifically, we defined a *high-level MDP* m_i^{HL} formalizing the exploration problem of a robot i over the regions.

High-Level MDP

The high-level MDP m_i^{HL} is defined as a tuple $\langle S_i^{HL}, \mathcal{A}_i^{HL}, T_i^{HL}, R_i^{HL} \rangle$ with S_i^{HL} a set of macro-states and $\mathcal{A}_i^{HL}$ a set of macro-actions over the regions. A macro-action consists in exploring the current region ρ_i^j $(explore(\rho_i^j))$ or moving from a region ρ_i^j to another region ρ_i^k $(move(\rho_i^j, \rho_i^k))$. A macro-state is defined as the current region of the robot and the set of already visited regions. A specific state called *blocked* is added to represent situations where an unknown obstacle prevents the robot from reaching a way-point and the exploration of a region fails. Part (a) of Figure 12 illustrates the decomposition of a road-map into 3 regions; i.e into 3 sub-MDPs. Part (b) of Figure 12 illustrates the transitions between the macro-states of the high-level MDP coordinating the exploration among the 3 regions.

The transition function of the high-level MDP gives the probability to move from a macro-state s_i^{HL} to another macro-state $s_i'^{HL}$ when executing a macro-action a_i^{HL}. Transitions and rewards for moving from a state to another in the high-level MDP depend on the initial state, the dynamics of the system in the current region and the strategy applied in the region. This strategy is influenced by the value of the input states of the neighbor regions. Optimally evaluating transitions and rewards would require to consider, for each region, the optimal local policy related to every possible set of values of these input states. In a resource-constrained setting where the robots have limited computational resources, optimal computation of transition and reward functions is not realistic.

We thus proposed to approximate these functions from the dynamics of the environment inside the regions. Hence, the probabilities related to a macro-action $move(\rho_i, \rho_j)$ are estimated by averaging

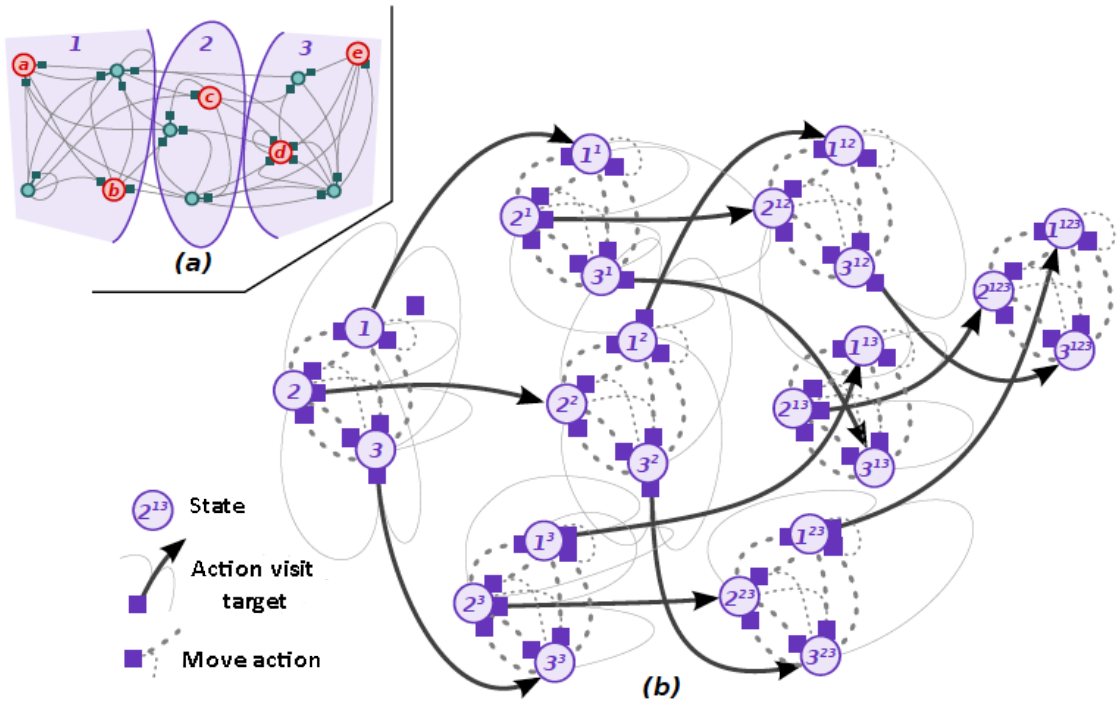


Figure 12: Partition of a road-map into 3 regions (a) and the related high-level MDP (b). A state 2^{13} means that the robot is in the region 2 and the regions 1 and 3 are explored

the probabilities related to a move action between two way-points w_i and w_j such that w_i and w_j are frontier way-points between ρ_i and ρ_j. Conversely, the probabilities related to an action $explore(\rho_i)$ are estimated by averaging the probabilities related to actions where the robot remains in the region ρ_i. In order to estimate rewards related to a region, an average reward is computed for each action of a region and then weighted by an estimate average number of actions performed in the region. We refer the interested reader to (Lozenguez, 2012) for more details about these approximation methods.

Lazy policy computation

In order to further limit computational resources, we also proposed a "lazy evaluation" approach regarding policy computation. This approach consists in delaying policy computation of each sub-MDP as much as possible. At first, the algorithm only computes the policy of the high-level MDP based on the approximate transition and reward functions. Then, each sub-MDP is solved only when the robot enters the associated region. Hence, the policy of the high-level MDP guides the exploration among the regions and local policies are computed only when it is necessary. This approach allows for limiting computation overhead. Moreover, in our application context where the UAV can send periodically updated road-maps to the ground robots, our approach fits initial requirements by limiting policy computation time and updates.

Experimental results

We developed some experiments to study the efficiency of the hierarchical approach on multi-rover exploration scenarios. Experiments were first implemented in simulated environments. The approach has then been successfully deployed on Pioneer robots operating on the PAVIN outdoor platform in Clermont-Ferrand (Figure 13).

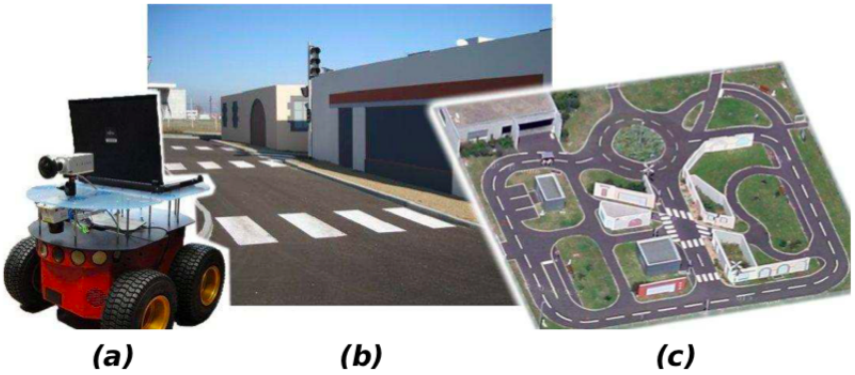


Figure 13: (a) a pioneer robot, (b) the experimental area and (c) the Google aerial view.

We first studied the quality of the partition into regions built by the greedy decomposition algorithm. In highly structured environments (where obstacles or walls separate the different parts of the environment), we obtained effective partitions. Nonetheless, in free from obstacles environments, we observed some overlaps between the regions.

We then studied the efficiency of the policies computed by our hierarchical approach combined with greedy partitioning of the road-map. In fact, we applied the greedy partitioning algorithm and we recorded the sum of the expected gains of the policies computed by the high-level MDPs (one high-level MDP per agent). We then compared the performance of our hierarchical decomposition approach with the optimal sum of expected gains obtained from an optimal allocation and an optimal solving of the individual MDPs. In this case, we considered all possible allocations of points of interest among the agents. For each possible allocation, we computed the optimal expected gain of each agent and we selected the allocation yielding to the maximum sum of expected gains. As a worst case baseline, we also registered the worst sum of expected gains (referred to as the worst allocation). For each experiment, the sum of expected gains have been normalized. Hence, a score of 1 corresponds to the optimal whereas a score of 0 corresponds to the expected gain of the worst allocation.

Figure 14 presents the distribution of the scores obtained by the hierarchical approach combined with greedy decomposition, for different sizes of problems. We were not able to optimally solve problems involving more than 12 points of interest because of the complexity of finding an optimal allocation of the points of interest among the robots. For each size of problem, we randomly generated 200 multi-robot exploration problems. The average score obtained by the hierarchical approach is around 0.77 for problems involving 6 to 12 points of interest. It is an encouraging average considering that the computation is instantaneous. Although the optimal expected gain is rarely obtained (with 8 points of interest, the hierarchical approach yielded to the maximum expected gain for only 7% of the scenarios), solutions are often close to the optimal (with 8 points of interest, 52% of the solutions yield to scores between 0.8 and 0.9). Furthermore, our average scores are negatively affected by a few numbers of poor quality instances induced by unsuitable partitions. We noticed that the scores and the number of poor quality instances are inversely proportional to the number of points of interest.

Finally, we studied the scalability and the running time of our approach. Indeed, one of our main concern was to design an efficient on-line approach that could deal with large number of points of interest. We considered problems involving up to 120 points of interest and 3 robots. For problems involving up to 100 points of interest, the hierarchical approach was able to partition the set of tasks, allocate the tasks and compute the high-level strategy in less than one second[8]. For problems involving 120 points of interest, the approach took less than 5 seconds. These results comply with our

[8] Experiments were performed on a computer equipped with Intel Core2 Quad CPU Q9650 at 3.00GHz

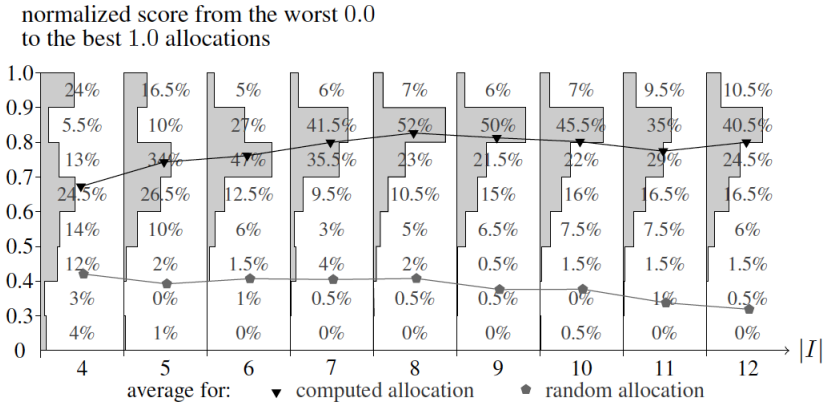


Figure 14: Distribution of the scores obtained for randomly generated sets of points of interest

initial requirements related to on-line planning of exploration missions. Experiments showed that the approach allows the robots to deal with tens of tasks during the mission, by controlling the number of tasks in each region and the number of regions.

3.5.3 *Discussion*

In the previous sections, we introduced the constrained Dec-MDP model that extends the original Dec-POMDPs framework to formalize different types of constraints usually encountered in real-world domains. This model also allows for formalizing temporally extended actions and thus handles problems where the decisions of the agents may not be synchronized. Although constraints allow for reducing the set of possible actions at each time-step, the state description must be augmented with information related to constraint satisfaction. The formalization of real-world problems thus results in large state and action spaces. In general, solving a constrained Dec-MDP is as difficult as solving a standard Dec-MDP.

In order to efficiently solve constrained Dec-MDPs, we investigate decomposition methods which constitute an attractive approach to cope with the high dimensionality of decision problems. The main idea is to divide the initial problem into a set of loosely-coupled sub-problems. We first considered decomposing the problem among the agents. We also exploited the spatial configuration of the environment to decompose the decision problem of an agent. It is important to note that the computational overhead incurred by the decomposition of the problem must be compensated by the savings obtained during policy computation (Boutilier et al., 1999). When identifying the different sub-problems is not straightforward, efficient decomposition methods have to be developed. For effective decomposition, we proposed a greedy algorithm that aims at minimizing dependencies between the sub-problems while balancing the size of the sub-problems.

In the best case, the problem decomposition results in a set of independent problems, that can be solved separately. It is the case, for instance, when decomposing transition, observation and reward independent Dec-POMDPs. Nonetheless, most of the time, the value of a sub-problem depends on the other sub-problems (or at least on a subset of these sub-problems). In this context, we investigated different distributed methods to coordinate the policies of the sub-problems: opportunity cost communication, iterative policy improvement, hierarchical computation of the global policy. Our approaches have been successfully used to solve large size problems while respecting different kinds of constraints on action execution. It may nevertheless be regretted that these approaches do not provide guarantees

on the quality of the solution. Although, experimental results show that high quality strategies are often computed, we could not estimate the distance to the optimal solution.

Several works investigated settings where the interactions of the agents are limited and proposed distributed algorithms where each agent separately computes her strategy (Nair et al., 2005; Oliehoek et al., 2008; Witwicki and Durfee, 2010). Closely related to our work, these algorithms exploit the fact that a Dec-POMDP can be decomposed as set of loosely-coupled POMDPs. Witwicki and Durfee (2010) proposed a joint planning method based on abstraction of policies. Abstraction of actions are quite similar to the tasks we considered though, constraints on action execution are not explicitly formalized in this framework. Oliehoek et al. (2008) considered settings where the neighborhood of an agent can change at each time step, i.e. an agent can interact with different sets of agents at each time-step. Dependencies between the agents are formalized by influence variables in the POMDP states. Their optimal solving approach builds the solution by searching through the space of influences. For each influence, optimal individual policies are computed for each neighboring agent. Although this approach provided encouraging results, it is expected that it would poorly scale as soon as the agents get more tightly connected.

Hierarchical approaches applied to Dec-POMDP problems have received little attention so far. Oliehoek and Visser (2006) described a hierarchical model based on Dec-POMDPs in order to formalize rescue missions. Nonetheless, solutions for solving this model have not been studied. Amato et al. (2014) introduced *macro-actions* to formalize high-level actions which may require different amounts of time to execute. A macro-action is formalized using the *option* framework developed in the single-agent context. Coordination between the agents is limited to a high-level and only handles interactions between macro-actions. The authors proposed to extend standard Dec-POMDP algorithms to plan the execution of macro-actions instead of considering primitive actions. Although this framework is able to consider larger environments and longer horizon than standard Dec-POMDP approaches, scaling to large set of agents remain challenging.

3.6 Non-stationary frameworks

Markovian models run under the assumption that the environment is stationary, i.e. the transition function and the reward function do not evolve over time. Unfortunately, the stationary hypothesis does not hold in many real-world problems because uncontrolled or unforeseen events may change the environment dynamics. For instance, weather conditions may change action outcomes of outdoor robots. Mobile robots in smart cities may have to face with different driving contexts along the day according to traffic conditions. In network routing problems, the topology of the network may change over time and link costs may evolve because of congestion phenomenons. Moreover, in multiagent systems, the environment may appear as non-stationary from the point of view of an agent if the behaviors of the other agents change over time. In particular, it may be the case in adversarial domains and in multiagent learning settings.

In this section, we first focus on *single-agent decision problems in non-stationary environments*. More specifically, we are interested in settings where the non-stationary environment can be viewed as a set of stationary contexts. We introduce a new model for formalizing such decision problems and propose methods for computing near-optimal strategies. We also propose a learning approach to handle settings where the set of contexts is not known *a priori*. Finally, we turn to *multiagent* sequential decision making problems where the non-stationarity arises from evolving adversarial behaviors. This latter work has been motivated by multiagent patrolling problems where a team of defenders aims at detecting illegal actions performed by adversaries.

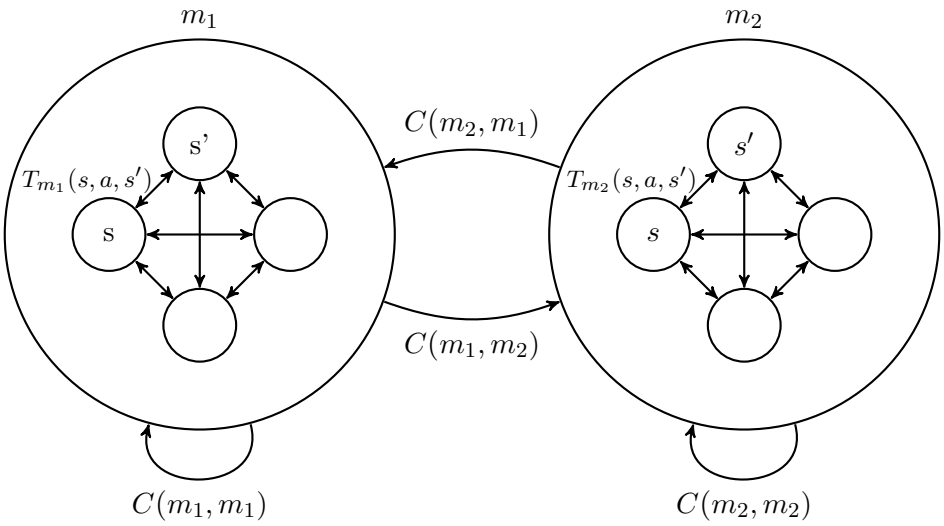


Figure 15: HM-MDP representation with 2 modes and 4 states

3.6.1 *Single-agent sequential decision problems*

In non-stationary environments, the decision-maker has to adapt online her strategy to the changes in the environment dynamics. When the dynamics remain stationary over several time-steps and then abruptly change, the non-stationary dynamics can be viewed as a set of stationary contexts between which the environment can switch (Choi et al., 2000).

Hidden-Mode Markov Decision Processes

In the context of non-stationary MDPs, Choi et al. (2001) proposed the *Hidden-Mode Markov Decision Process* (HM-MDP) model to formalize this subclass of non-stationary problems. The non-stationarity is limited to a number of stationary settings, called modes or contexts. Each mode represents a possible stationary environment, formalized as an MDP. Transitions between modes represent environmental changes.

Definition 31. ***Hidden-Mode Markov Decision Processes***
An Hidden-Mode Markov Decision Process (HM-MDP) *is defined by a tuple* $\langle M, C \rangle$ *with:*

- $M = \{m_1, \ldots, m_p\}$, *a finite set of modes where* $m_i = \langle S, \mathcal{A}, T_i, R_i \rangle$, *i.e. an MDP,*
- $C : M \to Pr(M)$, *a transition function over modes.*

Note that S and $\mathcal{A}$ are shared by all m_i's and that an HM-MDP with $p = 1$ is a standard MDP. In HM-MDPs, the only observable information is the current state $s \in S$. The current mode $m_i \in M$ is not observable. As for the probabilistic functions, one can imagine that the set of the states and the set of actions could evolve as well. In such a case, it is sufficient to define the global set of states as the union of the set of states of each mode. It is identical for the set of actions.

Figure 15 illustrates an HM-MDP with 2 modes and 4 states per mode.

Hidden Semi-Markov Mode MDP

The HM-MDP framework is not always the most suitable model for representing sequential decision-making in non-stationary environments as it assumes that the environment may change at every time-step. We argue that this assumption is not always realistic. Indeed, in driving problems for

instance, traffic conditions do not change at each time-step. A traffic jam, for instance, lasts over several time-steps. In adversarial problems, the adversaries often keep the same strategy over several time-steps. When this assumption does not hold, the usual modeling trick is to set a low probability of transition between modes. However, from a theoretical viewpoint, this is more than questionable when mode transitions are not geometrically distributed.

In (Hadoux et al., 2014b), we proposed a natural extension of HM-MDPs, called *Hidden Semi-Markov-Mode Markov Decision Processes* (HS3MDPs), where the non-stationary environment evolves according to a semi-Markov chain. In HS3MDPs, when the environment stochastically changes to a new mode, it stays in that mode during a stochastically drawn duration.

Definition 32. ***Hidden Semi-Markov-Mode MDP***

An Hidden Semi-Markov-Mode MDP (HS3MDP) *is defined by a tuple* $\langle M, C, H \rangle$ *where:*

- M *and* C *are defined as for HM-MDPs,*
- $H : M \times M \to Pr(\mathbb{N})$ *is a* mode duration function.

The transition function $C(m_i, m_j)$ represents the probability of moving to new mode m_j from current mode m_i knowing that the *duration* in m_i (i.e. the number of remaining time-steps to stay in m_i) is null. $H(m_i, m_j, h)$ gives the probability of staying h time-steps in the new mode m_j when the current mode is m_i. The mode and the duration are both not observable. Note that, it is not always relevant for the duration function H to take into account the previous mode. For this purpose, the duration function may be specified as $H(m_j, h)$.

At each time-step, after a state transition in current mode m_i, the next mode m_j and its duration h' are determined as follows:

$$\begin{cases} \text{if } h > 0 & m_j = m_i, \\ & h' = h - 1, \\ \text{if } h = 0 & m_j \sim C(m_i, \cdot), \\ & h' = k - 1 \text{ where } k \sim H(m_i, m_j, \cdot) \end{cases} \tag{5}$$

where h is the duration of current mode m_i. If h is positive, the environment dynamics do not change. But, if h is null, the environment moves to a new mode according to the transition function C and the number of steps to stay in this new mode is drawn following the conditional probability H.

Like HM-MDPs, HS3MDPs form a sub-class of POMDPs. An HS3MDP can thus be reformulated as a POMDP and solved using standard POMDP algorithms. Moreover, we proved that HM-MDPs and HS3MDPs are equivalent (Hadoux et al., 2014b). In fact, a problem represented as an HS3MDP can also be exactly represented as an HM-MDP by augmenting the modes. Nonetheless, representing HS3MDPs in such a way feels unnatural and leads to a higher number of modes, which would have a negative impact on the solving time.

Planning with non-stationary models

As for POMDPs, solving a problem modeled as an HS3MDP is a difficult task to address.

Proposition 13. *Optimally solving an HS3MDP is a PSPACE-complete problem.*

Proof. In their work, Chadès et al. (2012) proposed the *hidden-model MDP* model or hmMDP (note the lower case) and proved that finding an optimal policy in an hmMDP is a PSPACE-complete problem. Independently discovered, hmMDPs turn out to be a subclass of HM-MDPs where there the mode, once selected, cannot be changed. As finding an optimal policy for a POMDP is also a PSPACE-complete problem (Papadimitriou and Tsitsiklis, 1987), both HM-MDPs and HS3MDPs are PSPACE-complete to solve. □

In order to be able to tackle large instances of problems, we therefore focused on an approximate solving algorithm and we considered the POMCP algorithm[9] (see Section 3.2.1). A first naive approach is to apply POMCP to directly solve the POMDP derived from an HS3MDP. In that case, a particle in POMCP represents a mode m, a state s and a duration h of the HS3MDP. We proposed two possible improvements to this naive approach. Notice that, as a subclass of HS3MDPs, these solving methods can also be applied to HM-MDPs.

POMCP VARIANTS TO EXPLOIT THE STRUCTURE In large instances, POMCP can suffer from a lack of particles to approximate the belief state, especially if the number of states in the POMDP and/or the horizon are large. To tackle this issue, a particle reinvigoration technique is usually used but it is often insufficient. If POMCP then runs out of particles, it samples the action set according to a uniform distribution, which obviously leads to suboptimal decisions.

We proposed a first adaptation of POMCP that exploits the structure of HS3MDPs to delay the lack of particles. In fact, in the derived POMDP, as the agent observes a part of the state, a particle needs only to represent non-observable information, that is the mode m and the duration h. This adaptation allows us to initially distribute the same amount of particles over a set whose cardinality is much smaller.

Nonetheless, the above adaptation of POMCP still suffers from lack of particles when solving large-sized problems. We thus proposed a second adaptation where we replaced the particle set used in POMCP by an exact representation of the belief state. This representation consists in a probability distribution μ over $M \times \mathbb{N}$ (modes and duration in the current mode). Particles are then drawn from this probability distribution which is updated after each new observation.

The spatial complexity of this second adaptation of POMCP does not depend on the number of simulations. Indeed, μ is a probability distribution over $M \times \mathbb{N}$. Assuming a finite maximum duration $h_{\max}$, which is often the case in practice, there always exists a number of simulations $\mathbb{S}$ for which the size of the particle set is greater than the length of the probability distribution. In such a case, our second adaptation will be more interesting to consider. The time complexity of updating the exact representation is $\mathcal{O}(|M| \times h_{\max})$. It has to be compared to the particle invigoration of the original POMCP combined with the first adaption which is $\mathcal{O}(\mathbb{S})$ with $\mathbb{S}$ being the number of simulations.

EXPERIMENTAL RESULTS We ran experiments on four non-stationary problems: the traffic light problem, the sailboat problem, the elevator problem and randomly generated environments[10]. The first three environments are problems from the literature (Choi et al., 2001). We solved an extended version of each problem modeled as an HS3MDP[11]. We compared the performance obtained by the original POMCP and by our adaptations of POMCP: the Structure Adapted (SA) and Structure Adapted combined with the Exact Representation (SAER) of belief states. We also compared the performances with the optimal policy when it could be computed, using Cassandra's POMDP toolbox[12] and *MO-IP* (Araya-López et al., 2010a). Finally, we used *MO-SARSOP* (Ong et al., 2010) with one hour of policy computation time when the model could be generated for offline computing.

Experimental results show that (for a given number of simulations) our adapted version leads to at least as good performances as the original POMCP. Indeed, on small problems (like the traffic

[9] Interestingly, HM-MDPs and HS3MDPs are particular instances of Mixed-Observable MDPs (MOMDPs) (Ong et al., 2010; Araya-López et al., 2010a), a subclass of POMDPs. Indeed, with the state being observable and the mode (as well as the duration for HS3MDPs) being not, both models can be translated into an equivalent MOMDP. Therefore, MOMDPs algorithms could be used for solving HS3MDPs. However, we chose to base our solving method on POMCP, because it tends to be more efficient than specialized algorithms on MO-MDPs and more generally on factored POMDPs, even when POMCP is run using non-factored representations (Silver and Veness, 2010).

[10] We refer the interested reader to (Hadoux, 2015) for more details about the experiments.

[11] Recall that those adapted versions of the problems cannot be represented as efficiently with HM-MDPs.

[12] http://www.pomdp.org/code/index.html

Simulations	Original	SA	SAER	Optimal
1	-3,42	0.0%	0.0%	38.5%
2	-2,86	3.0%	**4.0%**	26.5%
4	-2,80	8.1%	**8.8%**	25.0%
8	-2,68	6.0%	**9.4%**	21.7%
16	-2,60	**8.0%**	**8.0%**	19.2%
32	-2,45	5.3%	**6.9%**	14.3%
64	-2,47	**10.0%**	9.1%	14.9%
128	-2,34	**4.3%**	3.4%	10.4%
256	-2,41	8.5%	**10.5%**	12.7%
512	-2,32	**5.6%**	4.7%	9.3%
1024	-2,31	5.1%	**7.0%**	9.3%
2048	-2,38	9.0%	**10.5%**	11.8%

Table 1: Results for the traffic light problem

Simulations	Original	SA	SAER
1	0.39	0.1%	**8.9%**
2	0.39	21.0%	**57.5%**
4	0.40	9.9%	**149.0%**
8	0.41	24.0%	**224.6%**
16	0.43	33.0%	**261.3%**
32	0.48	58.2%	**275.8%**
64	0.60	76.2%	**248.7%**
128	0.83	75.4%	**184.5%**
256	1.16	64.1%	**115.9%**
512	1.61	41.5%	**61.5%**
1024	2.05	2.2%	**28.8%**

Table 2: Results for random environments with $n_s = 50$, $n_a = 5$ and $n_m = 10$

light problem), both adaptations of POMCP are roughly similar. In fact, the size of the problem is quite small so, the original POMCP and the structured adapted POMCP do not run out of particles. Moreover, there are enough particles to draw a high quality estimation of the belief state. However, our adaptations of POMCP outperform the original version since exploiting the structure of the HS3MDP leads to more accurate belief states.

Table 1 describes the results obtained for the traffic light problem (8 states, 2 actions and 2 modes), using different algorithms: original POMCP, Structure Adapted (SA), Structure Adapted combined with Exact Representation of belief states (SAER), Finite Grid, MO-IP and MO-SARSOP. The last three algorithms yield the same results, which are presented in column "Optimal" to give an idea of the optimal value. We give the raw results for the original POMCP and percentages for the others. Reported percentages correspond to the percentages of improvement brought by our modified versions over the original POMCP. For each number of simulations we averaged the cumulative discounted rewards over 1000 runs.

The performances of the original POMCP almost strictly increase with the number of simulations used in the algorithm. They therefore get closer to the optimal value, which translates into decreasing percentages in Column "Optimal" of Table 1. Experimental results show that our modified versions of POMCP perform better than the original one (positive percentages for columns "SA" and "SAER"). They also get closer to the optimal. For instance, with 512 simulations, 4.7% of improvement for SAER compared to 9.3% for Column "Optimal" means that the performances of SAER are half-way between those of the original POMCP and the optimal value. Note that a decreasing percentage does not mean a raw decrease in the performances. It means that the increase of the performances of the original POMCP is higher than those of the other algorithms. Nonetheless, the percentages being positive, the latter still perform better.

On larger problems, our methods significantly outperform the original POMCP method. Table 2 depicts the performance of our algorithms on randomly generated problems involving 50 states, 5 actions and 10 modes. In fact, the exact representation of belief states always outperforms POMCP versions based on particles filter on sufficiently large environments. Indeed, these methods quickly run out of particles to accurately represent the belief state.

The computation time of our adaptations are promising for application to large-sized real-life problems. For instance, in the random environment with 20 modes, one run of 1024 simulations took 1.15 seconds[13] for solving the HS3MDP with structured adapted POMCP and 1.48 seconds for solving the HS3MDP with POMCP and exact representation of the belief state.

[13] Results obtained on a computer equipped with an Intel XeonX5690 4.47 Ghz core and 16G of RAM.

Learning the model

Mode-based non-stationary environments have already been actively investigated in the *Reinforcement Learning* (RL) setting (Choi et al., 2001; Doya et al., 2002; da Silva et al., 2006). Choi et al. learned the HM-MDP in a RL setting using the Baum-Welch algorithm (2000). The drawback of this approach is the assumption of an *a priori* known number of modes. Doya et al. (2002) applied ideas from adaptive control (Narendra et al., 1995) to RL, which consists in learning multiple models, computing a "responsibility signal" to evaluate the goodness of each model and averaging the models using this signal. Here, again, the number of models is *a priori* fixed and known.

da Silva et al. (2006) developed the *Reinforcement Learning with Context Detection* algorithm (RLCD) to simultaneously learn and act in a non-stationary environment. At each time-step, a quality score of each already learned model (i.e. mode) is calculated, depending on the last seen transition and reward. The model maximizing the measure is chosen as the next current model and is updated. However, when no model has a quality above a minimum threshold, a new model is added to the list of known models, uniformly initialized and selected as the next current model. With this method, RLCD is able to tackle problems without the prior knowledge of the number of models to learn. Unfortunately, RLCD requires a set of parameters to be tuned accordingly to the problem. Moreover, this quality measure seems to be ad-hoc and also depends on a hand-tuned threshold.

In (Hadoux et al., 2014a), we proposed a new approach to learn the models allowing us to solve the sequential decision-making problem under non-stationary environments. Our main idea is to adapt tools developed in statistics and more precisely in sequential analysis (Ghosh and Sen, 1991) for detecting an environmental change (Basseville and Nikiforov, 1993). In doing so, our approach is more theoretically founded and necessitates less parameters. Parameters are thus easier to interpret and easier to set *a priori* for solving new problems.

Let $m_i = (S, \mathcal{A}, T_i, R_i)$ and $m_j = (S, \mathcal{A}, T_j, R_j)$ be two modes or MDPs that are both assumed to be known. We consider that the environment is currently represented by m_i and at some unknown timestep, the environment changes from mode m_i to mode m_j. The problem we want to tackle here is that of detecting as soon as possible this environmental change. To that aim, a natural idea is to use statistical hypothesis tests for such detections, i.e. given an observed history, a null hypothesis "the current mode is m_i" is tested against an alternative hypothesis "the current mode is m_j". When performing such tests, one wants to minimize the probabilities of two contradictory errors:

- type I error: reject the null hypothesis when it is true,
- type II error: accept the null hypothesis when it is false.

In online settings, sequential statistical tests are preferred: they perform repeated tests as observations become available and permit detection with smaller size samples in expectation (Wald, 1945) compared to standard statistical tests. Viewing detection as statistical tests highlights the contradiction between fast detection (type I error) and false detection (type II error).

A simple approach to implement those sequential statistical tests for change point detection is to recourse to cumulative sums (CUSUM) (Basseville and Nikiforov, 1993). In our setting, CUSUM can be specified as follows for detecting a change in the transition distributions. Let $(s_0, a_1, s_1, a_2, s_2, \ldots, s_{t-1}, a_t, s_t, \ldots)$ denotes the observed history and define $V_0^T = 0$. At each timestep $t \geq 1$, compute:

$$V_t^T = \max(0, V_{t-1}^T + \ln(\frac{T_j(s_t, a_t, s_{t+1})}{T_i(s_t, a_t, s_{t+1})})) \tag{6}$$

and compare V_t^T to a threshold $c^T > 0$. If $V_t^T \geq c^T$, then a change in the transition function is detected. The intuitive idea of CUSUM is quite simple: If m_j is more likely than m_i to have generated the recent history, then decide that the environment has changed.

To detect a change in the reward function, the same procedure as for the transitions can then be applied. Let $(r_1, r_2, \ldots, r_t, \ldots)$ be the sequence of obtained rewards and $V_0^R = 0$. At each timestep $t \geq 1$, compute:

$$V_t^R = \max(0, V_{t-1}^R + \ln(\frac{R_j(s_t, a_t, r_t)}{R_i(s_t, a_t, r_t)})) \tag{7}$$

If V_t^R is greater than a threshold $c^R > 0$, then a change of the reward function is detected.

The two previous sums can be combined by computing at each timestep $t \geq 1$:

$$V_t^{TR} = \max(0, V_{t-1}^{TR} + \ln(\frac{T_j(s_t, a_t, s_{t+1})R_j(s_t, a_t, r_t)}{T_i(s_t, a_t, s_{t+1})R_i(s_t, a_t, r_t)})) \tag{8}$$

with $V_0^{TR} = 0$. Sum V_t^{TR} is to be compared with a threshold $c > 0$ to detect a change of mode.

Computing V_t^T and V_t^R separately can be advantageous in some situations as this makes it possible to detect a change in the transition function or in the reward function alone. Indeed, in some domains, the non-stationarity is only limited to one of the two functions and/or they can evolve in an asynchronous way. The advantage of using V_t^{TR} is that it may allow for faster detection of the environmental change because of the combined effects of the simultaneous change of the transition function and the reward function.

In (Hadoux et al., 2014a), we proposed an adaptation of RLCD, replacing the quality measure by the one presented in Equation 8. While results show that our modification of the RLCD algorithm is more efficient than the original RLCD, it also requires less parameters which are more understandable. Therefore, our method can be used in a wider field of problems where the parameters of the original method may prevent us from tuning them efficiently.

However, our method is not complete enough to learn HS3MDPs. RLCD and our extension are reactive algorithms, meaning that they adapt when detecting a change in the dynamics of the environment but they fail to anticipate changes. Indeed, waiting for the detection introduces a lag in the learning of the different modes. Moreover, approaches based on modes are especially well suited when the environment remains stationary over a long period and then abruptly changes to a very different context. In highly flickering environments, where the mode changes at each decision step, we cannot apply those methods. In fact, in this particular case, the methods will learn a mean model over all modes of the environment.

3.6.2 *Non-stationary multi-agent decision problems*

The work presented in the previous section addressed problems involving a single decision maker. We now turn to multiagent non-stationary decision problems and investigate how our approach can be extended to such contexts. In particular, we study how non-stationarity can be handled in the Dec-POMDP framework.

In multiagent settings, the non-stationarity may not only come from the dynamics of the environment but may also arise from the behaviors of the other agents. Each agent anticipates the joint action based on her knowledge about the other agents' strategies. If an agent i deviates from her behaviors and the others agents $j \neq i$ are not aware of it, action outcomes will differ from the expected outcomes based on the (incorrect) knowledge about i. The environment will thus appear as non-stationary from the point of view of the agents $j \neq i$.

In this section, we take the point of view of a team of cooperative agents facing some adversaries. The adversaries' behaviors influence the action outcomes and the rewards of our team of cooperative agents. We consider contexts where the adversaries can adapt and change their behavior online thus leading to non-stationary transition functions from the point of view of our team of cooperative agents.

Multiagent patrolling

Multiagent patrolling is the problem faced by a set of agents that have to periodically visit a set of targets $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$. In adversarial domains, the agents have to prevent some threats or illegal actions performed by some adversaries on these targets. Obviously, the number of defending resources is not enough to cover all targets at all time. Defenders have thus to coordinate their actions in order to detect as much adversary actions as possible.

A large amount of recent work has been dedicated to Security Games where a defender has to deploy a set of resources to prevent an attack on a subset of the targets (Jain et al., 2012; Nguyen et al., 2016; Sinha et al., 2018). A solution for a Security Game is a mixed strategy for the defender, i.e. a probabilistic distribution over all pure strategies where a pure strategy is an allocation of defending resources among targets. Security Games have been successfully deployed in real-world settings and are currently used to conduct surveillance at the Los Angeles International Airport (Jain et al., 2010) or in Boston harbor (Shieh et al., 2012). Nonetheless, the original model of Security Games makes strong assumptions on the domain. Indeed, the game is assumed to be played only once, i.e. the opponent is assumed to play a one-shot attack. This assumption is well suited when preventing a terrorist attack but it does not hold when the defenders and the adversaries frequently interact.

Some works consider the problem of several defenders that have to face multiple adversaries performing frequently and repeatedly illegal actions (Agmon et al., 2008; Zhang et al., 2015; Fang et al., 2015, 2016). The decision problem is formalized as a repeated game where the strategy at stage t consists in a probabilistic assignment of defending resources to targets. These works have been applied to domains such as preventing crime in urban areas (Zhang et al., 2015), avoiding intrusions on frontiers (Agmon et al., 2008), or detecting illegal fishing or poaching (Fang et al., 2015).

Nonetheless, multiagent patrolling is often much more than a repeated assignment of defending resources to targets. The spatial dimension of the environment to patrol has to be considered and raises constraints limiting the actions of the defenders at each time step. In fact, most of the time, a defender cannot be reassigned on every target given her current position. Moreover, targets are usually loosely connected and moving from one target to another takes time.

Because uncertainty is unavoidable when acting in real-world security domains (Nguyen et al., 2016), there is also a need to handle non-deterministic action outcomes. In patrolling scenarios, moves between different targets to patrol are inevitably stochastic and take an uncertain amount of time. Moreover, the agents (the defenders and the adversaries) usually have partial observability of the environment. This also raises uncertainty on action outcomes. Furthermore, uncertainty may arise from partial knowledge about the adversaries. In fact, standard security games usually assume full observability and full rationality of the adversary. In such settings, defenders are able to anticipate a best-response from the adversaries. Obviously, this may not be the case and thus makes the defender uncertain about the response of the adversary even under full observability. Most existing approaches only handle a specific type of uncertainty: uncertainty on adversary's payoffs, uncertainty on defender's strategy or uncertainty on the rationality of the adversary. Nguyen et al. (2014a) proposed a unified framework handling the different forms of uncertainty. Again, this approach computes a one-shot assignment of defending resources. (Nguyen et al., 2013; Kar et al., 2015; Zhang et al., 2015) have been interested in modeling and learning adversary bounded rational behaviors. These approaches aim at computing the probability that the adversaries attack a target τ_i at time t. Closely related to our work, Kar et al. (2015) consider adaptive adversaries. Nonetheless, this later approach assumes that the defenders have full observability of all successful attacks and know the payoffs of the adversaries.

Effective coordination of the defenders in uncertain environments has been little investigated so far. In the context of a one-shot attack, Shieh et al. (2016) have shown that defender effective teamwork significantly improves security. In order to enable effective cooperation between several patrollers, Shieh et al. combine security games and Dec-MDPs. Nonetheless, the approach considers a single fully rational adversary which is assumed to perform a prior extensive surveillance phase and to attack the target with the lowest coverage. In fact, the issue of effective cooperation between patrolling resources acting in uncertain and partially observable environments over several decision-steps, is still challenging.

When the model of the adversaries is known, the defenders can anticipate the response of the adversary and thus deduce the outcomes of patrolling actions. If the action outcomes are probabilistically known, the multiagent cooperative patrolling problems can then be represented as a constrained Dec-POMDP as introduced in the previous sections. Nonetheless, because of limited observability and bounded rationality, we argue that it is not realistic to assume that the patrollers are able to build a full model of the adversaries (and thus to include it in their model of the environment). In fact, patrolling agents cannot observe all the actions of the adversaries over the whole environment. Patrollers then make decisions based on limited knowledge about the adversaries' behaviors. Moreover, adversaries may have bounded rationality and not always commit to an optimal policy. They can also keep on adapting their strategy online from their past observations about the patrollers. In such settings, assuming a strong rational adversary and anticipating a stationary best response of the adversaries is no more optimal for the patrollers. The defenders have therefore to handle *partially observable* and *non-stationary* adversary strategies.

Hidden Modes Dec-POMDPs

Following our previous works on Hidden-Mode Markovian models, a multiagent non-stationary decision problem can be envisioned as a series of stationary decision problems. In adversarial domain, a mode corresponds in fact to a stationary adversary behavior and the system will move to another mode as the adversaries change their strategy. It has to be noticed that such an approach is well suited when the adversaries keep the same strategy over a long enough period. One of the advantage of mode-based approaches is that the strategies computed for a context can be reused if the adversaries return to a previous context (da Silva et al., 2006; Hernandez-Leal et al., 2017).

HM-MDPs (and similarly HS3MDPs) can be extended to define *Hidden-Mode Decentralized Partially Observable Markov Decision Processes* (HM-Dec-POMDPs).

Definition 33. ***Hidden-Mode Decentralized Partially Observable Markov Decision Processes***

An HM-Dec-POMDP *is defined by a tuple* $\langle M, C \rangle$ *as follows:*

- $M = \{m_1, \ldots, m_p\}$, *a finite set of modes where* $m_i = \langle \mathcal{N}, S, \mathcal{A}, T_i, O, \Omega, R_i \rangle$, i.e. *a Dec-POMDP,*
- $C : M \to Pr(M)$, *a transition function over modes.*

As previously discussed in single-agent contexts, formalizing the problem as an HM-Dec-POMDP requires to *a priori* know the set of possible modes and the probabilistic transition function C. In patrolling domain, it means that the defenders have to know (before the execution) how the opponent will behave and how the behavior of the adversaries could evolve over time. Obviously, this assumption about the adversaries rarely holds.

Online learning of the current context

When the set of adversary behaviors is not *a priori known*, the agents may try to learn the current context online from partial observations about the adversaries. Several questions have thus to be considered among which: How to exploit local observations and deduce the current context? How to exploit new observations as the agents act in a specific context? How to detect changes of contexts and restart from a new context?

In (Beynier, 2016, 2017), we investigated settings where the defenders have limited observability of the adversaries and do not know the payoffs of the adversaries. Based on their local observations (detected actions of the adversaries), the agents try to build a probabilistic model of the adversaries. In fact, for each target τ_i, we proposed to maintain a probability $PI_{\tau_i}(t)$ that the adversaries would initiate an illegal action on target τ_i at time t. These probabilities allow for deducing transition probabilities inside the current mode. Unfortunately, the defenders have not enough information to build an accurate model of the adversaries and thus have accurate knowledge of these probabilities. Probabilities PI_{τ_i} are thus estimated from the history of observations made by the patrollers.

Nonetheless, as the strategies of the adversaries may change over time, old observations may become obsolete. We thus restrict the size of the history that we consider, to the κ latest observations. Intuitively, latest observations are more likely to reflect the actual behaviors of the adversaries. Let $NI_{t_i}(t-\kappa, t)$ be the number of detected adversaries on target τ_i (defined for all τ_i in κ) between $t-\kappa$ and t. We define the following estimate:

$$PI_{\tau_i}(t) = \frac{NI_{\tau_i}(t-\kappa, t)}{\sum_{\tau_k \in \mathcal{N}} NI_{\tau_k}(t-\kappa, t)} \tag{9}$$

Based on the current estimates of PI values, transition probabilities related to the current mode (i.e. Dec-POMDP) can then be deduced (see (Beynier, 2016, 2017) for more details about the computation of transition probabilities).

As the patrollers get more and more observations, they update probabilities values consequently. As the agents refine these probabilities, they should also update the transition function of the current Dec-POMDP. Since updating the Dec-POMDP model and the patrolling strategies at each time-step would be too costly (in terms of time and computational resources), we introduced a context horizon h that fixes the period of validity of the current context. The Dec-POMDP model is then updated every h steps based on the probabilities PI computed from the observations made by the patrolling agents over the last κ steps.

However, adversaries may change their strategy over these h time-steps and the strategies are likely to become inefficient until the next Dec-POMDP update. In order to avoid a loss of performance, patrollers should be able to detect such changes and adapt their strategies consequently.

Online detection of context changes

In (Beynier, 2016, 2017), we proposed a mathematical method allowing the agents to detect adversary policy changes. Our method consists in monitoring the variations in the number of detected adversaries. In fact, empirical studies showed that the number of detected illegal actions significantly decreases when adversaries change their strategy. We thus aim at efficiently detecting such decreases to update the context as soon as possible. Our method relies on monitoring the variations of a finite moving average over the number of detected adversaries. The variations of this moving average are compared to a threshold value. If a variation exceeds the threshold, it is assumed that the adversaries have changed their strategy. As soon as an adversarial policy change is detected, the current context is updated based on the latest observations made by the agents. This method is quite similar to the

CUSUM method described in Section 3.6.1. Nonetheless, in the patrolling domain, our approach does not require to compute the transition functions at each time step.

Note that the threshold of the procedure is a parameter of the method and has to be tuned considering the Dec-POMDP formalization of the problem. Low thresholds provide sensitive detection but could lead to "false" detection of strategy changes. On the other hand, high thresholds might miss some strategy changes.

Multiagent planning with non-stationary models

Given a mode, solving the corresponding Dec-POMDP returns a joint policy $\pi = \{\pi_1, ..., \pi_n\}$ maximizing the global expected reward of the agents. Existing algorithms to solve Dec-POMDPs (see Section 3.2.2) could be used to solve the Dec-POMDP related to a mode. However, in our settings, existing algorithms suffer from several limitations. In non-stationary environments, it is necessary to update online the strategies to the changes of dynamics. In fact, each time a new mode is considered, a new joint policy must be computed. If a centralized algorithm is used, a central entity would have to collect all the observations made by the patrolling agents to deduce the new mode, to update the Dec-POMDP model and to compute a new strategy. This strategy will then be communicated to the patrolling agents. Such an approach obviously creates a bottleneck in the system and would result in high communication cost. Furthermore, most Dec-POMDP algorithms would take too much time to solve the current Dec-POMDP and would not be suited to our context where the strategies of the agents have to be updated online frequently. More specifically, existing algorithms usually compute the joint policy *from scratch*. They have not been designed to update joint strategies during the execution. They cannot re-use previously computed strategies to speed up the computation of a new joint strategy.

We developed a *distributed evolutionary algorithm* to compute online the individual strategies of each stationary Dec-POMDP. It has to be noticed that other works have also considered solving Dec-POMDPs using evolutionary methods (Mazurowski and Zurada, 2007; Eker and Akın, 2013) and showed significant improvement in the size of the horizon that can be handled. We adapted the (1+1) evolutionary algorithm (Droste et al., 2002) to optimize the patrolling strategy over horizon h. The evolutionary algorithm selects an initial solution (called *champion*) and then iterates to improve the champion until a computation deadline is reached. At each iteration, a mutation operator is applied to the current champion thus obtaining a *challenger*. This new solution is evaluated and becomes the new champion if its value is higher than the one of the current champion. The process is iterated until the deadline of the algorithm is reached or no more improvement of the current champion is possible.

Similarly to the work presented in Section 3.3, the Dec-POMDP formalizing patrolling context must handle temporal and spatial constraints on actions. The population of individuals thus consists in the set of joint policies that comply with these constraints.

For each new mode, the initial solution of the evolutionary algorithm is built from a probability distribution over the nodes reflecting the likelihood of an illegal action on each site deduced from observations. In fact, the higher the probability of an illegal action on a target τ_k, the higher the probability of selecting τ_k. Probabilities are also weighted by the visit frequency of the target in the previous mode. The mutation of the current champion strengthens the weakest targets: the targets with the lowest probabilities of threats are replaced by the targets with the highest probabilities of threats.

Our evolutionary algorithm has the advantage of being anytime. Moreover, thanks to its low complexity, it scales well to large numbers of agents and long planning horizon h. However, no guarantee on the quality of the solution could be given. In fact, the algorithm does not provide any bound regarding the distance to the optimal solution and can be trapped in local optima. It has to

be pointed out that our approach allows patrolling agents to adapt their strategies online every h steps. From the point of view of the attackers, even if they had full observability of the patrollers, the patrolling strategy will then not seem deterministic all along the execution.

Experiments with non-stationary adversarial strategies

We performed some experiments[14] on randomly generated patrolling missions to evaluate the performances of the defenders facing non-stationary adversaries. We first studied the detection ratio (percentage of illegal actions that are detected) along the patrolling mission. We compared the results obtained by our evolutionary algorithm with the optimal solution computed using the MADP toolbox[15]. We observed that our evolutionary algorithm leads to high detection ratios (Figure 16). Moreover, the detection ratio is decreased by only 3% over the optimal solution for 5 targets and by 6% for 7 targets. It has to be noticed that even the optimal approach does not provide full detection of illegal actions since agents cannot cover all targets at all time.

We experimented our approach on several sizes of graphs considering a fixed number of 4 agents. The larger the number of targets, the lower the detection ratio. In fact, as the number of targets increases, it becomes more and more difficult for the 4 agents to cover all the targets and to reach high detection ratios. In order to guarantee good performances of patrolling agents, a minimum number of agents is required. This number is closely related to the number of targets to patrol and to move durations. As shown in Figure 17 for a scenario with 16 targets, a detection ratio of 100% is almost reached with 12 agents and considering 6 agents leads to a detection ratio greater than 90%. More generally, our approach provides good detection ratios for $m \ll n$.

We then studied the scalability of our approach. Although we were not able to optimally solve problems larger than 2 agents and 7 targets, our evolutionary algorithm successfully solved problems up to 50 targets and 7 agents (with a deadline of 10 seconds). Larger problems can even be solved by enlarging the deadline of the evolutionary algorithm.

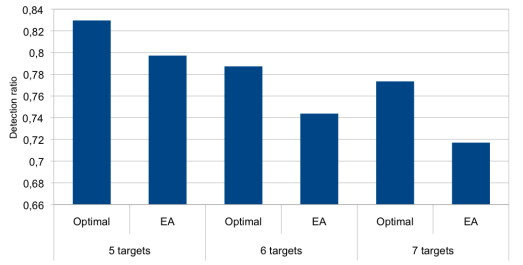


Figure 16: Detection ratios of the executed strategies

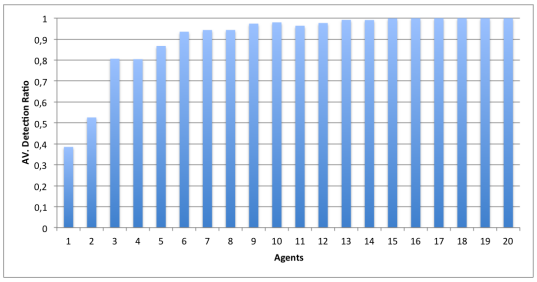


Figure 17: Influence of the number of agents

We also studied how the detection ratio evolves over time during the execution (Figure 18). We observed that the detection ratio remains stable over the execution except when the adversaries change their strategy (this change occurs at 270 on Figure 18). Nonetheless, the change of strategy is quickly detected by our approach and the agents immediately adapt their strategy. The detection ratio thus quickly returns to its previous level.

Finally, we varied the number of times the adversaries change their strategy and we studied the impact on the detection ratio (Figure 19). The less the adversaries change their strategies, the higher the detection ratio. In fact, when the adversaries often change their strategy it becomes more and

[14] Experiments were performed on a computer equipped with an Intel(R) Core(TM)2 Duo processor, 2000 MHz, 8Gb.
[15] http://www.fransoliehoek.net/index.php?fuseaction=software.madp

more difficult to deduce the current context. In highly dynamic problems, the percentage of detected illegal actions falls below 60%.

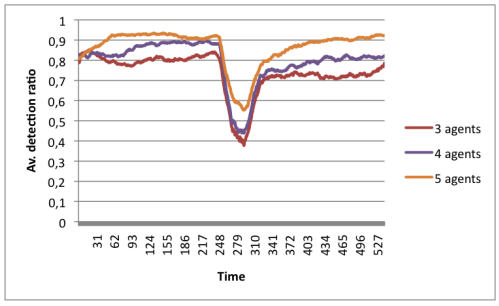


Figure 18: Detection ratio over time

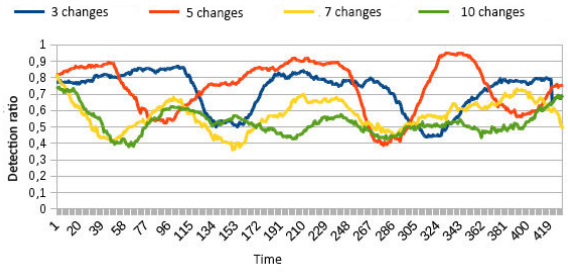


Figure 19: Influence of the number of strategy changes

3.6.3 *Discussion*

Our approach proposes to exploit the observations of the defenders in order to learn a model of the adversaries and then to compute cooperative strategies based on the learned model. We consider defenders with limited observability about the system. In particular, our model does not make the assumption of full-rationality from the defenders and does not require the defenders to know the reward function of the adversaries. We defined a simple model of the adversaries where the probabilities of attacks among the targets are derived from the limited knowledge that the defenders can obtain about the system. This model benefits from little assumption about the degree of observability of the system by the defenders and by the adversaries as well.

Recently, approaches dealing with adversary bounded rational behaviors have focused on Subjective Utility Quantual Response (SUQR) models (Nguyen et al., 2013; Kar et al., 2015). These works are derived from the theory of Subjective Utility introduced in behavioral economics (Fischhoff et al., 1983) and define the subjective utility of the adversary as a linear combination of values (rewards and penalties) and probabilities. The subjective utility is then introduced in a Quantal Response (QR) model to predict a probability distribution of the adversary strategy. QR models suppose that the greater the expected value of a target, the more likely the adversary will attack that target (Nguyen et al., 2013). However, these models require to know the payoffs of the adversary and to be able to tune the parameters of the subjective utility functions. Finally, the high computational complexity of these models prevent from scaling up to large set of targets.

An alternative approach consists in addressing adversary bounded rationality using robust optimization (Pita et al., 2012; Haskell et al., 2014). These approaches have been developed to solve problems were the adversary can attack only one target. Instead of building a model of the adversary, the solution is computed so as to mitigate the defender's maximum utility assuming an adversarial optimal strategy on the one hand; and the expected utility of the defender when the adversary deviates from her rational behavior on the other hand. This approach bounds the loss induced by a potential deviation of the adversary.

3.7 PERSPECTIVES

Over the last decade, Dec-POMDPs have become a well-recognized model to deal with cooperative decentralized control problem. Our work raises various open issues that we would like to investigate in the future.

DIVERSIFICATION OF CONSTRAINTS AND APPLICATIONS Up to now, we have been more specifically interested in application domains related to multi-robot cooperative exploration. Our model and algorithms are not restricted to such contexts and we plan to investigate other applications such as distributed sensor networks, server networks, smart cities or smart homes.

The set of constraints that we studied is not exhaustive and our model could be enriched with other kinds of constraints. For instance, it would be possible to extend temporal constraints to a set of convex time intervals. Diversification of constraints can also be guided by the set of relations introduced in TAEMS formalism (Decker and Lesser, 1993). Hence, we could relax precedence constraints to define facilitation relations such as "the execution of task τ_i facilitates the execution of the task τ_j".

We also plan to work on the generalization of the hierarchical approach to a wider set of domains. Although, hierarchical decomposition of Dec-POMDPs has been little investigated, we believe that it is a promising avenue to deal with large set of states and actions.

APPROXIMATE APPROACHES WITH QUALITY BOUNDS Our work has mainly focused on designing *distributed* solving methods that could efficiently compute high quality solution. We thus developed several approaches that fulfill our initial requirements. Under some restricted assumptions, we were able to guarantee the optimality of the solution. However, in the general case, we could not provide theoretical guarantees on the quality of the solutions. Rabinovich et al. (2002; 2003) proved that deciding whether there exists an ϵ-approximate joint policy for a Dec-POMDP is NEXP-hard and finding an ϵ-approximate solution remains as hard as optimally solving a Dec-POMDP. Instead of proving quality guarantee on the solution, a less demanding approach consists in estimating the quality of the approximate solution. There is then a need to estimate upper bounds on the value of the optimal solution. Oliehoek et al. (2015) proposed some techniques to provide upper bounds on the performance of factored Dec-POMDPs. The idea is to decompose the problem into a set of sub-problems and to make optimistic assumptions about how one sub-problem is influenced by the other sub-problems. We would like to investigate whether these methods could be used in our setting to estimate the distance of ours solutions to the optimal.

DISTRIBUTED ONLINE LEARNING Although recent advances in learning methods to solve Dec-POMDPs allow for handling larger state and action spaces, challenges remain to be met. The first issue we would like to consider deals with handling constraints about task execution when learning policies. Excepting the work of Liu et al. (2016), learning approaches have been designed to solve generic Dec-POMDPs. Therefore, there is no guarantee that existing approaches would be able to deal with temporally extended actions and constraints such as temporal, precedence spatial or resource constraints. It would thus be necessary to investigate deeper whether existing learning approaches would be suited to solve Constrained Dec-MDPs. It is likely that more effective learning methods could be developed by adequately handling constraints on actions.

Another perspective is to exploit the structure of the problems in order to enhance the efficiency of learning approaches. Building on our previous works on Dec-POMDP decomposition, the idea is to develop learning approaches that could learn the policy of each sub-problem in a distributed way. In hierarchical models, we would like to investigate how learning algorithms could be used at the different levels of the model to efficiently compute coordinated strategies among the sub-problems and the different levels of the hierarchy.

In our work, we specially focused on distributed solving methods that could be used online by the agents while most other solving approaches use to compute policies off-line in a centralized way. Similarly, most existing learning approaches for Dec-POMDPs consist in centralized off-line learning for distributed control. We would like to pursue our research direction by developing *distributed online learning* methods. To our knowledge, the only notable approach adopting such a perspective is the deep decentralized Multi-Task Multi-Agent learning approach proposed by Omidshafiei et al. (2017).

online distributed learning is challenging in the Dec-POMDP domain and raises well-known issues pointed out by (Claus and Boutilier, 1998; Matignon et al., 2012b) in multiagent reinforcement learning. Indeed, distributed online learning involves cooperative independent learners acting in stochastic and partially observable environments. As the agents simultaneously learn their strategy and have partially observability of the system, they are not aware of the policy changes of the other agents. The environment thus becomes non-stationary from the point of view of each agent.

UTILITY ELICITATION When the reward function of a decision problem is unknown or partially specified, utility elicitation can be envisioned in order to incrementally acquire more knowledge about the reward functions from queries to an expert (Chajewska et al., 2000; Pigozzi et al., 2016). Indeed, designing a complete and precise reward function is challenging as it is prohibitively expensive in many cases and potentially error-prone. When a partially specified reward function can still be modeled, one can take advantage of questioning a human expert during the course of interaction in order to acquire relevant utility information.

Let consider for instance multi-agent exploration problems (multi-robot exploration or multiagent patrolling). The vast majority of the approaches assume that the agents have prior full knowledge on the rewards of the planning problem, i.e. the value of each target is assumed to be exactly known. Nonetheless, in unknown environments or adversarial domains, the designer of the system may not have enough knowledge about the utility of each target. In adversarial domain, the agents may be uncertain about the payoffs of the adversaries (Nguyen et al., 2014b). Interactive elicitation protocols could therefore be investigated in order to assess, from expert knowledge, the relative importance of the different locations or to give insights about possible adversarial actions. Note that the objective is not to obtain a precise specification of the utilities but to allow the agents to make good decisions even with partial utility information.

When the agents have partial prior knowledge on the payoffs of the problem, utility elicitation can be used to refine this information. In the multi-robot exploration problem that we previously considered, such prior utility information may be acquired by the UAV flying over the area to explore. In security domains, utility information may be extracted from previously observed adversarial behaviors such as poaching or illegal fishing traces. One popular approach is to adopt a Bayesian representation of the uncertainty over the possible utility functions. The distribution is updated using the Bayes rule according to the feedback received from the expert (Viappiani and Boutilier, 2010). The reward elicitation could then be modeled as a planning problem where we aim at optimizing sequentially the next question to ask by measuring the expected value of information, i.e. the long-term effect of the new information. Another approach consists in using minimax regret to decide for the next question to ask and the next action to undertake (Boutilier et al., 2006).

We plan to investigate interactive elicitation protocols in order to assess, from expert knowledge, the relative importance of different tasks, different locations or to give insights about possible illegal actions in patrolling domains. Answers to elicitation queries will then be used to optimize multiagent cooperative strategies. Since computing optimal strategies under uncertainty on utilities is a hard problem, we expect that interactions with the expert could allow to restrict the set of strategies to consider and limit computational efforts. We would like to address the issues related to the non-stationarity of the preferences and interleave elicitation and optimization to update multiagent strategies over time.

OPPONENT MODELING Our work on multiagent patrolling with adversaries led us to consider opponent modeling and learning of the opponent model. Opponent modeling has been widely studied in game-theory but also in learning and planning (Hernandez-Leal et al., 2017). In our context, we considered the problem of modeling partially observable (and non-stationary) opponents while acting in stochastic environments. We proposed a probabilistic model of the opponents defined as a set of

intrusion probabilities. In addition to being compact, this model can be efficiently updated based on the observations of the defenders about detected illegal actions of the opponents.

We plan to investigate other models of the opponent and to study whether more efficient patrolling strategies could be computed using these new models. Indeed, opponent modeling raises the question of selecting the most appropriated model. A wide range of models can be considered such as Bayesian models, decision trees, Markov Decision Processes (Hernandez-Leal et al., 2013), logic-based model, neural networks (He et al., 2016), etc. The relevance of a model is influenced by the amount of data that is available to build the model and by the decision-theoretic approach that will exploit the model.

It important to keep in mind that the objective is to exploit the model of the opponent in order to plan more efficient actions for a team of cooperative agents. Hence, the model of the opponent must be easily embedded in the model of the multiagent planning problem. Probabilistic and MDP-based models of the adversary are thus particularly well suited when the planning problem is modeled as a Dec-POMDP. However, combining logic-based and Markov models could also be envisioned as proposed by Saffidine et al. (2018) who combined epistemic logic and Dec-POMDPs.

In our work, we assumed that the defenders only observe detected illegal actions. The agents thus had little information to exploit and were not able to build a sophisticated model of the opponents. A similar problem has been encountered in distributed resource allocation (see Chapter 2) where we assumed that an agent only observes the resources held by another agent upon encounters. As the agents did not know the other agents' preferences, they were not able to model their possible behaviors. One way to overcome this lack of information would be to relax assumptions about the observability of the other agents and of the opponents. More sophisticated opponent models could then be designed. Nonetheless, a wider observability range may not be possible in real-world domains. Another direction is to introduce tactical actions allowing the agents to seek for relevant information about the opponent. These actions would be included in the set of possible actions and would be part of the agent's strategies. The agents would have to choose between executing information seeking actions for improving opponent modeling or executing rewarding actions with a possibly unaccurate model.

4

STRATEGIC ARGUMENTATION

When agents have different views of the system, they may engage in a conversation and exchange arguments in order to share some information, persuade each other, deliberate or negotiate (Walton and Krabbe, 1995). *Argumentation theory* is an interdisciplinary field studying how conclusions can be reached through logical reasoning (van Eemeren et al., 1996). In multiagent systems, argumentation theory provides tools for designing and analyzing interaction protocols to resolve conflicts between agents that arise from inconsistent knowledge or objectives (Maudet et al., 2007). Multiagent argumentation is mainly concerned with structuring interactions between the agents, i.e. designing interaction protocols that fulfill established principles. Multiagent argumentation also provides tools to analyze and evaluate the arguments of the conversation and come up with a conclusion.

In this chapter we consider a general setting where the agents are engaged in a dialogue where they try to persuade the others to modify their beliefs. The negotiation dialogue can be viewed as an *argumentation game* where, at her turn, an agent puts a new argument in the debate or attacks an existing argument. Dialectics have been investigated in formal argumentation, but mostly as a mean to provide a proof-theoretical counterpart to argumentation semantics (Modgil and Caminada, 2009), leaving no room for proper strategies. Hence, agents typically fail to have winning strategies, to act fully rationally, to act strategically (such as sometimes hiding arguments that they know to be true), etc.

However, in multiagent systems, an autonomous rational agent is expected to exploit her knowledge about the argumentation problem in order to decide, at her turn, what is the best argument to put forward in the course of the dialogue in order to influence the outcome of the debate. Such argumentation dialogues can be viewed as argumentation games (Thimm, 2014) where self-interested agents strategically select their arguments in order to reach a desirable state of the dialogue according to their individual preferences. It has to be noticed that strategic argumentation is even more crucial when the horizon of the dialogue is limited. Indeed, if the agents may not have enough time to proceed all arguments, they have to select the most relevant arguments to put in the dialogue before the deadline.

As described in the classification proposed by Thimm and Garcia (2010), a key element to consider in strategic argumentation is the *awareness* of agents. Two extremes of the spectrum are when agents are *fully ignorant*, i.e. they just know their own arguments; or *omniscient*, i.e. they know the arguments (and strategies) that the opponents have at their disposal. In the former case the agent will typically have to rely on heuristic approaches (*e.g.* (Kontarinis et al., 2014)). While this may be efficient in practice, it is usually very difficult to offer any guarantee on the outcome. In the case of omniscient agents, one can use game-theoretic approaches, like backward induction. However, the strong required assumptions are often problematic. In fact, the degree of awareness corresponds to the level of knowledge studied in the previous chapters. An omniscient agent is assumed to have full observability of the system which is not realistic in many situations.

Following our work on planning under partial observability, we consider a more realistic intermediate level of awareness where each agent has partial knowledge of the other agents' possible moves. In fact, each agent may have partial knowledge about the arguments and the attacks that could be played by the opponents. While an agent may have full knowledge of the arguments of the domain,

she may not know whose arguments are actually endorsed by the other agents. The agent is thus uncertain about the moves that the opponent could play.

In this chapter, we show that strategic argumentation is tightly related to the planning issues that we previously studied in the context of sequential decision making under uncertainty.

We first investigate the problem, for an agent, of *optimizing a sequence of moves* to be put forward in a debate. We show that the planning problem of a strategic argumentative agent can be formalized as a Mixed Observability Markov Decision Problem (MOMDP). Since argumentation problems are highly structured, we design optimization techniques to reduce the size of the problems and improve the efficiency of strategy computation.

We then turn to the problem of *strategic mediation in argumentation debates* where conflicting agents that may be organized as teams exchange arguments to persuade each other. Assuming that the argumentation strategies of the debating agents are stochastically known, we proposed to formalize and solve the planning problem of the mediator as an HS3MDP.

4.1 Research context

The work presented in this chapter has been initiated in 2014 during the co-supervision (with Nicolas Maudet and Paul Weng) of the PhD of Emmanuel Hadoux who has then joined UCL as a post-doc. This work has continued until now and has led to collaborations with UCL and more specifically with Emmanuel Hadoux and Anthony Hunter.

4.2 Formal argumentation

We begin with an introduction of the necessary background about computational models in argumentation. In particular, we introduce *abstract argumentation frameworks* as defined by Dung (1995) and we define the *grounded semantics* that will be used in the rest of the chapter.

Definition 34. ***Argumentation framework***
An argumentation framework $\mathcal{AF}$ is a pair $(\mathcal{A}, \mathcal{E})$ where:

- *$\mathcal{A}$ is a finite set of arguments,*
- *$\mathcal{E}$ is a binary relation between the arguments called* attack relation*, such that $(a, b) \in \mathcal{E}$ if $a \in \mathcal{A}$, $b \in \mathcal{A}$ and a attacks b.*

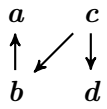
An argument can represent any piece of information such as a belief, a statement, an action... In fact, Dung's framework does not pay attention to the internal structure of the arguments but focuses on the attack relationship between the arguments.

An argumentation framework can be represented by a directed graph where each edge represents an argument and directed arcs represent the attacks among arguments.

Example 12. *Let us consider the argumentation framework $\mathcal{AF} = (\mathcal{A}, \mathcal{E})$ defined as:*

- $\mathcal{A} = \{a, b, c, d\}$
- $\mathcal{E} = \{(b, a), (c, b), (c, d)\}$

This argumentation framework can be represented by the following graph:



Given an argumentation framework, a rational agent would decide which arguments are acceptable. Dung formalized the notion of conflict-freeness, acceptability and admissibility.

Definition 35. ***Conflict-freeness***
A set of arguments $\mathcal{B} \subseteq \mathcal{A}$ is conflict-free *if $\forall a \in \mathcal{B}$, $\forall b \in \mathcal{B}, (a,b) \notin \mathcal{E}$ and $(b,a) \notin \mathcal{E}$.*
This means that there are no attacks between arguments belonging to $\mathcal{B}$.

Example 13. ***Example 12 continued***
In Example 12, the conflict-free sets are: $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{a,c\}$, $\{a,d\}$, $\{b,d\}$.

Definition 36. ***Acceptability***
An argument $a \in \mathcal{A}$ is acceptable *with respect to a set $\mathcal{B} \subseteq \mathcal{A}$ iff $\forall b \in \mathcal{A}$ such that $(b,a) \in \mathcal{E}, \exists c \in \mathcal{B}$ such that $(c,b) \in \mathcal{E}$.*
In other words, $\mathcal{B}$ defends argument a from every possible attacks.

Example 14. ***Example 12 continued***
In Example 12, arguments a and c are acceptable. with respect to $\mathcal{A}$.

Definition 37. ***Admissibility***
A set of arguments $\mathcal{B} \subseteq \mathcal{A}$ is admissible *if it is conflict-free and all of its arguments are acceptable with respect to $\mathcal{B}$.*
In other words, a set of arguments $\mathcal{B}$ is admissible if it is conflict-free and it defends itself against any possible attack in $\mathcal{A}$

Example 15. ***Example 12 continued***
In Example 12, the admissible sets are $\{c\}$ and $\{a,c\}$.

An *argumentation semantic* defines the method ruling the argument evaluation process. Two main categories of semantics are identified: extension-based semantics and labeling-based semantics[1].

EXTENSION-BASED SEMANTICS An extension-based semantic specifies the rules to derive extensions, i.e. to decide which subsets of arguments that should be regarded as acceptable. Dung (1995) introduced four semantics (complete, grounded, stable and preferred semantics) refining the admissibility principle by requiring other properties to hold.

Definition 38. ***Complete semantics***
A set $\mathcal{B} \subseteq \mathcal{A}$ is a complete extension *of an argumentation framework $\mathcal{AF} = (\mathcal{A}, \mathcal{E})$ if $\mathcal{B}$ is* admissible *and all acceptable arguments of $\mathcal{A}$ with respect to $\mathcal{B}$ belong to $\mathcal{B}$.*
In other words, all arguments defended by $\mathcal{B}$ are also in $\mathcal{B}$.

Example 16. ***Example 12 continued***
In Example 12, the only complete extension is $\{a,c\}$.

It has to be noticed that the complete semantics may return several extensions. The grounded semantics refines the notion of complete extension in order to return an unique set of arguments.

Definition 39. ***Grounded semantics***
A set $\mathcal{B} \subseteq A$ is a grounded extension *of an argumentation framework $\mathcal{AF} = (\mathcal{A}, \mathcal{E})$ if it is the minimal (with respect to set inclusion) complete extension of $\mathcal{AF}$.*

[1] We refer the interested reader to (Rahwan and Simari, 2009) for more details about semantics and more specifically to Chapter 2 of this book.

Example 17. ***Example 12 continued***
The grounded extension is $\{a, c\}$.

A grounded extension contains all the arguments of an argumentation framework that are non-attacked and all arguments that are directly or indirectly defended by non-attacked arguments. The unicity of the grounded extension makes the grounded semantics a widely used approach to determine which arguments are accepted and which are not. Moreover, the grounded extension $\mathcal{B}$ can be easily computed incrementally by initializing $\mathcal{B}$ with the set of unattacked arguments. The extension is then increased iteratively. At each iteration, the set of arguments attacked by one of the argument already present in $\mathcal{B}$ are removed and the new unattacked arguments are added to $\mathcal{B}$.

LABELLING-BASED SEMANTICS Labelling-based semantics aim at characterizing each argument of an argumentation framework $\mathcal{AF}$ with a label taken from a predefined set. A common set of labels is $\{in, out, undec\}$ to distinguish between *in* (respectively *out*) arguments for which all defenders are defended (respectively attacked) and *undec* arguments where only part of the defenders are defended or attacked (Caminada, 2006). More formally, a label l is a value from $\{in,\ out,\ undec\}$ associated to an argument a such that:

- $L(a) = in$ iff $\forall b \in \mathcal{A}$ such that $(b, a) \in \mathcal{E}, L(b) = out$,
- $L(a) = out$ iff $\exists b \in \mathcal{A}$ such that $(b, a) \in \mathcal{E}$ and $L(b) = in$,
- $L(a) = undec$ iff $L(a) \neq in$ and $L(a) \neq out$.

4.2.1 *Argumentation games and opponent modeling*

In order for a strategic agent to optimize her sequence of moves to be put forward in a dialogue, she must be able to anticipate how her opponent will react. Opponent modeling is thus a crucial issue to develop efficient argumentation strategies. An increasing interest has recently been dedicated in building and exploiting models of the opponent to enhance argumentation strategies.

A first possible approach is to consider that the agents are fully rational and informed. More specifically, the strategic agent is assumed to have full knowledge about the opponent strategy. A game theoretical approach can then be used where the strategic agent computes a best-response strategy (Oren and Norman, 2010; Black and Atkinson, 2011). However, empirical studies of human argumentative behaviors have shown that people does not act as a fully rational decision-maker and often do not play optimally (Rosenfeld and Kraus, 2014). In order to compute argumentative strategies, it is thus crucial to take into account the bounded rationality of the opponent. Moreover, discussion partners are usually not fully informed about the strategy of the others. In fact, agents do not endorse equally the arguments of the argumentative framework and each agent has individual private beliefs in the arguments of the dialogue. Because of partial observability over the beliefs of the others, an agent may be unable to anticipate the strategy of her opponent and to deduce an optimal best-response.

Black et al. (2017) proposed a robust approach that aims at finding a strategy for a persuader that guarantees a certain probability of success no matter which arguments the opponent asserts.

When it is not possible to assume full knowledge of all agents' strategies or when they act non-deterministically, one can use probabilities to reflect the likelihood that an agent plays a given argument or attack.

Rienstra et al. (2013) extended the recursive opponent model initially proposed in (Oren and Norman, 2010) by capturing uncertainty in the opponent model as a belief state that enumerates all possible utility functions of the opponent. Similarly to what is done in game-theory, the opponent is assumed to play a best-response strategy.

In (Hadjinikolis et al., 2013), the opponent modeling is updated through the information exchanged during the dialogue. Given an history of dialogues, a relationship graph formalizing support between the opponent arguments, is built. As the opponent puts new arguments in the dialogue, the opponent model is updated and augmented based on the relationship graph. The opponent model associates a confidence value to the arguments of the framework. This value formalizes the probability that a given argument is part of the beliefs of an agent.

In this chapter, we take a similar approach to model the knowledge about the opponent. Following the work of Hunter (2014), we adopt a probabilistic modeling of the opponent strategies. The behavior of the opponent is modeled by a set of rules mapping possible moves of the opponent to a probability distribution.

4.2.2 *Probabilistic argumentation*

The *Argumentation problem with Probabilistic Strategies* (APS) framework has been proposed to model argumentation problems using probabilistic executable logic (Hunter, 2014). Generalizing it to any number of agents, an APS is characterized by a tuple $\langle \mathcal{N}, \mathcal{A}, \mathcal{E}, (\mathcal{S}_i)_{i\in\mathcal{D}}, (g_i)_{i\in\mathcal{D}}, (\mathcal{R}_i)_{i\in\mathcal{D}}, \mathcal{P} \rangle$ with:

- $\mathcal{N}$, a set of agents;
- $\mathcal{A}$, a set of arguments;
- $\mathcal{E}$, a set of attacks $e(x, y)$, meaning that argument x attacks argument y;
- $\mathcal{S}_i$, a set of all possible private states of agent i;
- g_i, the argumentative goal of agent i;
- $\mathcal{R}_i$, a set of rules (defined below) specifying the possible moves of agent i;
- $\mathcal{P}$, a set of all possible *public* states.

The set of arguments $\mathcal{A}$ and the set of attacks $\mathcal{E}$ are assumed to be known by all agents. However, this does not mean that agents endorse equally all arguments. The private state accounts for this, by representing as a conjunction of predicates $h_i(x)$ (or their negation) the fact that agent i endorses (i.e. is willing to use in the debate) argument x. In the public state a conjunction of predicates $a(x)$ and $e(x, y)$ (or their negation) captures the fact that some arguments and attacks have been made public. Predicate $a(x)$ means that argument x is put forward in the public state.

In a debate, agents have *argumentative goals* that characterize their desired argumentation outcomes. Such goals typically refer to the evaluation of the current state of the debate thanks to argumentation theory, allowing in particular to assess which arguments are acceptable or not. In order to model realistic argumentation games, the goal of an agent is assumed to be private information and cannot be observed by the other agents. To characterize the possible desired argumentation outcomes, each agent i has a goal state g_i which is a conjunction of $g(x)$ or $g(\neg x)$ where each x is an argument and $g(x)$ (respectively $g(\neg x)$) means that x is (respectively is not) accepted in the public state. In our work, we shall always refer to the *grounded semantics* as defined by Dung (1995). We say that an argumentative goal is fully satisfied when all the predicates of the goal are true, and partially satisfied when only some of the predicates are. Although the agents are considered as selfish, individual goals might not be antagonistic. Indeed, in some cases, the public state may satisfy both goals. In those situations, both agents are then considered as winners.

In an APS, the agents' behaviors are governed by probabilistic rules. A *rule* $r \in \mathcal{R}_i$ is of the form $r : prem \Rightarrow Pr(Acts_i)$ where premise *prem* is a conjunction of predicates $a(\cdot), h_i(\cdot)$ and $e(\cdot,\cdot)$ (or their negations) applied to one or more arguments. $Pr(Acts_i) = [p_1/\alpha_1, p_2/\alpha_2, \ldots, p_k/\alpha_k]$ is a probability distribution over a set $Acts_i$ of possible acts. An act $\alpha \in Acts_i$ is a set of modifications on predicates of the public space and private state of agent i: $\boxplus(p)$ (resp. $\boxminus(p)$) stands for adding (respectively removing) p to the public space, where p is either $a(x)$ or $e(x,y)$. $\boxminus(p)$ stands for removing p from the public space. $\oplus(p)$ corresponds to adding predicate p to the private state of agent i. $\ominus(p)$ corresponds to removing predicate p from the private state of agent i. A rule can only be fired by an agent i if its premise is fulfilled. Premises formalize the conditions (i.e. arguments and/or attacks) that must hold in order to play a rule. Rules are assumed to be coherent and thus two rules cannot have the same premises with different conclusions (different acts or different probabilities on acts). Note that following the assumption of the original APS framework, at most one rule has its premises satisfied at any step (Hunter, 2014).

We denote as r^i_j the j-th rule of agent i and $r^i_{j,k}$ the k-th act of rule r^i_j, i.e. $r^i_{j,k} = \alpha_k$ if $r^i_j : \text{prem}_j \Rightarrow [p_1/\alpha_1, p_2/\alpha_2, \ldots, p_n/\alpha_n]$.

Example 18. *Consider a concrete dialogical argumentation problem involving 2 agents (agent* 1 *and agent* 2*). A famous debate in the gamer community is whether e-sport is a sport or not. The arguments are as follows:*

(*a*) *e-sport is a sport,*

(*b*) *e-sport requires focusing and generates tiredness,*

(*c*) *not all sports are physical,*

(*d*) *sports not referenced by International Olympic Committee (IOC) exist,*

(*e*) *chess is a sport,*

(*f*) *e-sport is not a physical activity,*

(*g*) *e-sport is not referenced by IOC,*

(*h*) *working requires focusing and generates tiredness but is not a sport.*

Assume that agent 1 wants to persuade that e-sport is a sport.
This example can be formalized by an APS, from the viewpoint of agent 1, as follows:

- $\mathcal{A} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}\}$
- $\mathcal{E} = \{\, e(\boldsymbol{f}, \boldsymbol{a}), e(\boldsymbol{g}, \boldsymbol{a}), e(\boldsymbol{b}, \boldsymbol{f}), e(\boldsymbol{c}, \boldsymbol{f}), e(\boldsymbol{h}, \boldsymbol{b}), e(\boldsymbol{g}, \boldsymbol{c}),$
  $e(\boldsymbol{d}, \boldsymbol{g}), e(\boldsymbol{e}, \boldsymbol{g})\}$
- $g_1 = g(\boldsymbol{a})$

Assume that the following rules formalize the agents' behaviors:

- $\mathcal{R}_1 = \{h_1(\boldsymbol{a}) \Rightarrow [1.0/\boxplus a(\boldsymbol{a})],$
  $h_1(\boldsymbol{b}) \wedge a(\boldsymbol{f}) \wedge h_1(\boldsymbol{c}) \wedge e(\boldsymbol{b}, \boldsymbol{f}) \wedge e(\boldsymbol{c}, \boldsymbol{f}) \Rightarrow$
  $[0.5/\boxplus a(\boldsymbol{b}) \wedge \boxplus e(\boldsymbol{b}, \boldsymbol{f}) \vee 0.5/\boxplus a(\boldsymbol{c}) \wedge \boxplus e(\boldsymbol{c}, \boldsymbol{f})],$
  $h_1(\boldsymbol{d}) \wedge a(\boldsymbol{g}) \wedge h_1(\boldsymbol{e}) \wedge e(\boldsymbol{d}, \boldsymbol{g}) \wedge e(\boldsymbol{e}, \boldsymbol{g}) \Rightarrow$
  $[0.8/\boxplus a(\boldsymbol{e}) \wedge \boxplus e(\boldsymbol{e}, \boldsymbol{g}) \vee 0.2/\boxplus a(\boldsymbol{d}) \wedge \boxplus e(\boldsymbol{d}, \boldsymbol{g})]\}$

- $\mathcal{R}_2 = \{h_2(\boldsymbol{h}) \wedge a(\boldsymbol{b}) \wedge e(\boldsymbol{h}, \boldsymbol{b}) \Rightarrow [1.0/ \boxplus a(\boldsymbol{h}) \wedge \boxplus e(\boldsymbol{h}, \boldsymbol{b})],$
  $h_2(\boldsymbol{g}) \wedge a(\boldsymbol{c}) \wedge e(\boldsymbol{g}, \boldsymbol{c}) \Rightarrow [1.0/ \boxplus a(\boldsymbol{g}) \wedge \boxplus e(\boldsymbol{g}, \boldsymbol{c})],$
  $a(\boldsymbol{a}) \wedge h_2(\boldsymbol{f}) \wedge h_2(\boldsymbol{g}) \wedge e(\boldsymbol{f}, \boldsymbol{a}) \Rightarrow$
  $[0.8/ \boxplus a(\boldsymbol{f}) \wedge \boxplus e(\boldsymbol{f}, \boldsymbol{a}) \vee 0.2/ \boxplus a(\boldsymbol{g}) \wedge \boxplus e(\boldsymbol{g}, \boldsymbol{a})]\}$

g_2 is unknown to agent 1.
There are $3^{|\mathcal{A}|} = 6561$ *possible goal states. The sizes of the state spaces are:* $|\mathcal{S}_1| = |\mathcal{S}_2| = 256, |\mathcal{P}| = 65536$.

The initial state $(s_1, p, s_2) \in \mathcal{S}_1 \times \mathcal{P} \times \mathcal{S}_2$ *of this problem is assumed to be:* $(\{h_1(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e})\}, \{\}, \{h_2(\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h})\})$.

From Example 18, we can build the graph of arguments and attacks of Figure 20. Bold face arguments are used by agent 1 while the others are used by the opponent.

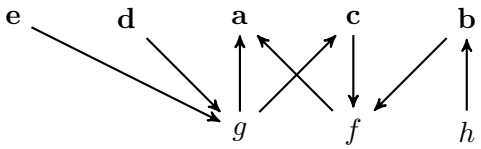


Figure 20: Graph of arguments of Example 18

While the APS framework is descriptive, it does not tackle the issue of optimizing the sequence of moves of the agents. All the possible sequences of states can be represented as a *Probabilistic Finite State Machine* (PFSM) (Hunter, 2014). For instance, starting from the initial state given in Example 18, the sequence of rules $(r^1_{1,1}, r^2_{3,2}, r^1_{3,1})$, alternatively for agent 1 and agent 2, leads the environment to the state $\{a(\mathbf{a}), a(\mathbf{g}), e(\mathbf{g}, \mathbf{a}), a(\mathbf{e}), e(\mathbf{e}, \mathbf{g})\}$. This is a winning state for agent 1 as $a(\mathbf{a})$ is true, is attacked but also defended. $a(\mathbf{a})$ is therefore accepted.

In order to compute an optimal policy for agent 1, one can use dynamic programming methods on the PFSM in order to backtrack the policy from the winning state, but this requires to know the internal state of the opponent. Indeed, in order to know which rules the opponent may fire, we need to either know the internal state or build a PFSM for each possible internal state.

4.3 Strategic behavior in argumentative debates

We now investigate the problem, for an agent, of optimizing a sequence of moves to be put forward in a debate, against an opponent assumed to behave stochastically, and equipped with an unknown initial belief state (Hadoux et al., 2015). In the following, the strategic agent that we consider will be arbitrarily assumed to be agent 1. For clarity reason, we will limit our discussion to dialogues where two agents exchanges some arguments. The opponent of our strategic agent is thus referred to as agent 2. Nonetheless, our work can be extended to settings where our strategic agent faces several opponents.

At each decision-step, the agent 1 has thus to decide which alternative of a rule to apply based on probabilistic knowledge about the argumentative behavior of her opponent. We first showed that one can take advantage of the fact that arguments are exchanged through a public space, making Mixed Observable Markov Decision Processes (MOMDPs) a suitable model to compute the strategy of a debating agent. Next, we exploited the fact that the domain of argumentation is highly structured: different schemes can be designed to minimize the obtained model, while preserving the optimality of the policy.

4.3.1 *Sequential decision problem under uncertainty*

In various sequential decision-making problems, some components of the state are fully observable while the rest of the state is partially observable. *Mixed Observability Markov Decision Processes* (Ong et al., 2010) have been proposed to account for such problems. MOMDPs form a sub-class of the POMDPs (see Section 3.2.1). MOMDPs exploit the mixed-observability property thus leading to a higher computational efficiency.

An MOMDP is characterized by a tuple $\langle S_v, S_h, A, O_v, O_h, T, \Omega, R\rangle$ with:

- S_v the observable part of the state,
- S_h the hidden part of the state,
- A the set of actions,
- O_v and O_h, the sets of observations on the visible and hidden parts of the state respectively,
- $T : S_v \times S_h \times A \to \Pr(S_v \times S_h)$ the transition function,
- $\Omega : S_v \times S_h \times A \to \Pr(O_v \times O_h)$ the observation function,
- $R : S_v \times S_h \times A \to \mathbb{R}$, the reward function.

Recall that our strategic agent observes the state of the debate but does not know the internal state of her opponent. The assumption on the knowledge of the agent complies with the definition of states and observations in MOMDPs. Indeed, the states of a MOMDP contain a directly observable part and a partially observable part. The directly observable part is the public state of the problem and the private part of agent 1. On the other hand, the non-observable part is the combination of the private states of all the other agents (in our case the private state of agent 2). This makes MOMDPs more suitable than other Markov models to represent such decision problems.

In order to optimize the argumentation strategy of agent 1, we transform the APS into an MOMDP defined as follows:

- $S_v = \mathcal{S}_1 \times \mathcal{P}$,
- $S_h = \mathcal{S}_2$,
- $A = \{\text{prem}(r) \Rightarrow \alpha | r \in \mathcal{R}_1 \text{ and } \alpha \in \text{Acts}(r)\}$. This set is obtained by decomposing each act α with a positive probability of each probabilistic rule r in $\mathcal{R}_1$.
- $\Omega(\langle s_v, s_h\rangle, a, \langle s'_v\rangle) = 1$ if $s_v = s'_v$, otherwise 0,
- T, O_v, O_h and R are defined as below.

When generalizing this transformation to more than two agents, the only modified part above is S_h, being the Cartesian product of the private states of the agents except agent 1.

ACTION SET A The possible (deterministic) actions of agent 1 are defined by splitting each act of the rules of agent 1 defined in the general APS, into separate actions. For instance, the fist rule of agent 1 in Example 18:

$$h_1(\mathbf{b}) \wedge a(\mathbf{f}) \wedge h_1(\mathbf{c}) \wedge e(\mathbf{b},\mathbf{f}) \wedge e(\mathbf{c},\mathbf{f}) \Rightarrow$$
$$[0.5 / \boxplus\, a(\mathbf{b}) \wedge \boxplus e(\mathbf{b},\mathbf{f}) \vee 0.5 / \boxplus\, a(\mathbf{c}) \wedge \boxplus e(\mathbf{c},\mathbf{f})]$$

This rule is split into two actions:

$$r_2^{1'} : h_1(\mathbf{b}) \wedge a(\mathbf{f}) \wedge h_1(\mathbf{c}) \wedge e(\mathbf{b},\mathbf{f}) \wedge e(\mathbf{c},\mathbf{f}) \Rightarrow \boxplus a(\mathbf{b}) \wedge \boxplus e(\mathbf{b},\mathbf{f})$$

$$r_2^{1''} : h_1(\mathbf{b}) \wedge a(\mathbf{f}) \wedge h_1(\mathbf{c}) \wedge e(\mathbf{b},\mathbf{f}) \wedge e(\mathbf{c},\mathbf{f}) \Rightarrow \boxplus a(\mathbf{c}) \wedge \boxplus e(\mathbf{c},\mathbf{f})$$

TRANSITION FUNCTION T From a state s, the agent 1 will first play an action and the other agent will then reply with another action leading to a state s'. Since we focus on optimizing the decision of agent 1, we aim at computing the probability that the public state of the debate moves from s to s' when the agent 1 takes a given action. To specify the transition function T on states, we first need to introduce the notions of *compatible rules* and *application set*.

Definition 40. ***Compatible rule.*** *A rule is* compatible *with a state s if it can be fired in state s. We denote $C_s(\mathcal{R}_i)$ the set of rules of $\mathcal{R}_i$ compatible with state s.*

Definition 41. ***Application set.*** *The* application set *$F_r(\alpha, s)$ is the set of predicates resulting from the application of act α of a rule r on s. If r cannot be fired in s or if act α does not modify s, $F_r(\alpha, s) = s$.*

Example 19. ***Example 18 continued.*** *Let $s = \{a(\boldsymbol{b}), h_2(\boldsymbol{h}), h_2(\boldsymbol{g})\}$, therefore, $C_s(\mathcal{R}_2) = \{r_1^2\}$ with r_1^2 being the first rule of $\mathcal{R}_2$.*

Let α_1 and α_2 be respectively the acts of r_1^2 and r_2^2 drawn to be executed (with r_1^2 and $r_2^2 \in \mathcal{R}_2$). The application sets are defined such that $F_{r_1^2}(\alpha_1, s) = \{a(\boldsymbol{b}),\ a(\boldsymbol{h}),\ e(\boldsymbol{h},\boldsymbol{b}),\ h_2(\boldsymbol{h}),\ h_2(\boldsymbol{g})\}$ as $r_1^2 \in C_s(\mathcal{R}_2)$ and $F_{r_2^2}(\alpha_2, s) = s$ as $r_2^2 \notin C_s(\mathcal{R}_2)$.

From a state s, agent 1 will select one of her compatible deterministic actions. In fact, the stochasticity in the transitions between two states s and s' arise from the probabilistic knowledge about the actions played by the other agent. Indeed, the probabilistic transition function models the uncertainty about the opponent's actions, i.e. the actions of agent 2.

Let $r : p \Rightarrow \alpha$ be a rule/action in A, with α the only act. The state $s' = F_r(\alpha, s)$ is the application set resulting from the application of α on state s. The rule $r' \in C_{s'}(\mathcal{R}_2)$ is a rule of agent 2 compatible with s' such that $r' : p' \Rightarrow [p_1 / \alpha 1, \ldots, p_n / \alpha_n]$ and $F_{r'}(\alpha, s') = s_i''$. Assuming that r' is the only rule of agent 2 compatible with state s', the function T can then be defined as $T(s, r, s_i'') = p_i$. With more rules compatible with s' involving several acts leading to the same s_i'', it is necessary to sum the probability of each act multiplied to a uniform probability across all fireable rules.

OBSERVATION SETS O_v AND O_h In the MOMDP, there is no observation on the hidden part of the state that is not already in the visible part. What is left is never observable. Therefore, $O_v = S_v$ and $O_h = \emptyset$.

REWARD FUNCTION R The reward function is defined as follows: each action that does not reach a goal state needs to return a strictly negative reward (i.e. a positive cost). If the goal is reached, the

reward needs to be positive. That way, the policy favors shorter argument sequences reaching the goal. However, the notion of goal can be extended to account for partially reached goals. For instance, if the goal of the agent is to have $g(\mathbf{a})$ and $g(\mathbf{b})$ but, only $g(\mathbf{a})$ is reached, a part of the reward could be obtained. More generally, the reward can be modulated depending on the value of the accepted arguments in the goal provided the semantics used indeed allows such gradual valuation. For instance, using the *General gradual valuation* (Cayrol and Lagasquie-Schiex, 2005), the reward function can be defined as the sum of the current valuation of each argument composing the goal. Besides considering attack and support relations, the valuation of the arguments can also involve positive and negative votes on the arguments (Evripidou and Toni, 2012).

Example 20. ***Example 18 continued.*** *After conversion, Example 18 yields an MOMDP whose sets have the following sizes:*

- $|S_v| = 256 * 65536 = 16\,777\,216 = |O_v|$,
- $|S_h| = 256$,
- $|A| = 5$.

Note that in the corresponding POMDP, the size of the set of states would be $|S| = |S_v| \times |S_h| = 4\,294\,967\,296$. Of course, such a large number of states is very limiting for POMDP solving methods.

4.3.2 *Model optimization*

In order to improve the scalability of argumentation problems that can be formalized and solved, we proposed several optimization schemes reducing the size of the generated MOMDP. A subtlety occurs because these optimizations may depend upon each other, and it may be useful to apply them several times. We say that we reach a *minimal* model when no further reduction of the model is possible by application of these techniques. Now this raises an obvious question: as optimizations may influence each other, we may well reach *different* minimal models, depending on the sequence of application chosen. In (Hadoux et al., 2015), we provided several guarantees in this respect: (i) we show uniqueness of the minimal model under the iterated application of three schemes, (ii) we show that for the last scheme, uniqueness of the model requires some mild conditions to hold.

[**irr.**] PRUNING IRRELEVANT ARGUMENTS. The first optimization consists in removing the arguments of each agent that are neither modified and never used as premises ("Irrelevant arguments"). This optimization is applied separately on the public and private states. An argument can thus be irrelevant in the description of the private state but can be relevant in the public state. We refer to an internal (respectively public) argument to denote the argument in the private (respectively public) state.

Example 21. *In Example 18, we can, for instance, remove the internal argument* ***f*** *from the private state of agent 1. Applying this optimization on the example removes respectively 3 and 5 arguments from the private state of agent 1 and 2.*

Note that, if part of the goal turns out to be an irrelevant argument, this optimization could modify the goal. But this is a degenerate case: when the irrelevant argument is not compatible with the goal state, the outcome of the debate is known a priori (the agent loses the debate anyway), thus we do not consider these cases. Otherwise, the argument is removed from the goal state.

[enth.] INFERRING ATTACKS The second optimization considers the set of attacks [2]. Let y be a public argument ($a(y)$), if $e(x,y)$ exists and $\boxplus e(x,y) \Rightarrow \boxplus a(x)$ (i.e. each time $e(x,y)$ is added, $a(x)$ also is), as the set of attacks is fully observable, we can infer attacks from the sequence of arguments put forward in the public space and thus remove the attacks from the rules and the states. In fact $e(x,y)$ is no longer used and the semantic of $\boxplus a(x)$ becomes "add argument a and attack y if it is present".

Example 22. *In Example 18, this optimization removes the 8 attacks.*

[irr(s_0).] PRUNING ARGUMENTS WRT. INITIAL STATE. For this optimization, we exploit the knowledge about the initial state s_0. As a result, this optimization requires to regenerate the MOMDP if the initial state changes. This optimization consists of two steps:

1. for each predicate $p \in s_0$ that is not later modified
   (a) update the set of rules by removing all the rules that are not compatible with p,
   (b) remove p from the premises of the remaining rules.
2. remove all rules of the opponent that can never be fired after any action of agent 1.

This procedure can be formalized as follows:

1. $\forall i, \forall p \in s_0$ s.t. $\exists r \in \mathcal{R}_i$ s.t. $p \in \text{prem}(r)$ and $\nexists r' \in \mathcal{R}_i$ s.t. $p \in \text{acts}(r')$:
   (a) $\mathcal{R}_i \leftarrow \{r \in \mathcal{R}_i | \neg p \notin \text{prem}(r)\}$
   (b) $\forall r \in \mathcal{R}_i, \text{prem}(r) \leftarrow \text{prem}(r) \backslash p$
2. Let S' be the set of states resulting from the execution of an action of agent 1, i.e. states $s' = F_r(\alpha, s)$, $\forall s \in \mathcal{S}_1 \times \mathcal{P} \times \mathcal{S}_2$, $\forall r \in C_s(\mathcal{R}_1)$, $\forall \alpha \in \text{acts}(r)$. $\forall r' \in \mathcal{R}_2$ if $r' \notin C_{s'}(\mathcal{R}_2)$ $\forall s' \in S'$ then, $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \setminus \{r'\}$

Note that this optimization is an extension of the optimization on irrelevant arguments. Indeed, after being replaced by their initial value in premises, the arguments become unused and are thus removed.

Example 23. *In Example 18, this optimization removes the 5 internal arguments of agent 1.*

Note that this optimization cannot be performed for the opponent side since her initial internal state is unknown.

The optimization procedures presented above deeply modify the representation of the problem. We need to ensure that the problem solved before the application of those procedures is the same after the application. In other words, the optimal policy computed after reduction of the problem needs to be applicable in the original problem as well as to remain optimal. The proofs of the following propositions are detailed in (Hadoux et al., 2015).

Proposition 14. ***(a)*** *Applying* ***Irr.***, ***Enth.***, *and* ***Irr(s_0).*** *does not affect the optimal policy and* ***(b)*** *the optimized model is unique, minimal for those three optimization schemes and independent of the order in which they are applied (as long as they are applied until reaching a stable model).*

Optimizations can be pushed further by using the graph of attacks.

[2] Enth. is the abbreviation of Enthymeme expressing the fact that some premises are omitted because they are obvious.

[DOM.] PRUNING DOMINATED ARGUMENTS. We start by defining the notion of *dominance* for an argument. Note that unattacked arguments are leaves of the graph.

Definition 42. ***Dominance.*** *If an argument is attacked by some unattacked argument, it is* dominated.

Hence, dominated arguments cannot belong to an optimal strategy. As we want the minimal sequence of arguments, the optimization scheme consists in pruning dominated arguments of agent 1. Recall that no assumption is made on agent 2, in particular we do not assume that she plays rationally and tries to avoid dominated arguments.

Example 24. *In our example, we can see that argument* ***b*** *is dominated by argument* ***h****.*

This optimization scheme assumes that agent 2 will necessarily fire a rule consisting in adding an argument defeating the dominated argument.

Note that this is irrespective of the opponent being an optimal player or not. However, this does not hold if:

1. the opponent does not know all her rules,
2. the debate length is limited (in which case it may make sense to put forward an argument even though it is easily defeated because the attacking argument may lie outside of the debate),
3. the opponent cannot play all her arguments.

Proposition 15. *If* ***(a)*** *the opponent knows all her rules,* ***(b)*** *can play all her arguments and* ***(c)*** *the debate length is infinite then, applying* ***Dom.*** *does not affect the optimal policy.*

Nonetheless, applying **Irr.** or **Irr(s_0).** may modify the graph of attacks: some unattacked arguments of the opponent can be removed and dominated arguments may appear to be non-dominated. In Example 18, if the opponent cannot play argument **h**, **b** is no longer dominated and must not be pruned.

We can now define the notion of true dominance with respect to the optimization procedures.

Definition 43. ***True dominance.*** *An argument is* truly dominated *is it remains dominated after the application of* ***Irr.*** *and/or* ***Irr(s_0).***

Proposition 16. *If all dominated arguments are truly dominated, the optimized model is unique, minimal and independent of the order in which the optimization schemes are applied (as long as they are applied until reaching a stable model).*

Otherwise, **Irr.** and **Irr(s_0).** must be applied before **Dom.** in order to keep only truly dominated arguments.

4.3.3 *Experiments*

Even if the transformation of an argumentation problem to an MOMDP exploits observable information to reduce the high dimensionality of the problem, it can still lead to a huge state space. It may thus be impossible to use exact solving methods. We ran experiments to test the scalability of our approach and optimization methods. Since the exact algorithm MO-IP (Araya-López et al., 2010b) was unable to compute a solution in a reasonable amount of time (a few tens of hours), we used MO-SARSOP (Ong et al., 2010), with the implementation of the APPL library (NUS, 2014).

After solving the MOMDP built from Example 18, we obtained the following policy graph for agent 1:

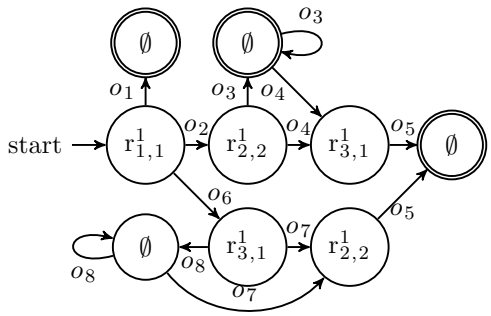


Figure 21: Policy graph for Example 18

The observations of agent 1 are:
$o_1 = \{a(\mathbf{a})\}, o_2 = \{a(\mathbf{a}), a(\mathbf{f})\}, o_3 = \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{f})\}, o_4 = \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{f}), a(\mathbf{g})\}$
$o_5 = \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{e}), a(\mathbf{f}), a(\mathbf{g})\}, o_6 = \{a(\mathbf{a}), a(\mathbf{g})\}, o_7 = \{a(\mathbf{a}), a(\mathbf{e}), a(\mathbf{f}), a(\mathbf{g})\},$
$o_8 = \{a(\mathbf{a}), a(\mathbf{e}), a(\mathbf{g})\}$

To follow this policy, start on the first node, apply the rule and move in the graph depending on the observation received. From the point of view of agent 1, accepting states (double circled) are final states of the debate. The agent has no more actions to execute unless the other agent adds or removes a predicate that changes the state. Note that the second node of the top row is an accepting state from which the agent can transition. Indeed, receiving observation o_3 can have two meanings: either the opponent has not played $a(\mathbf{g})$ yet or she will never be able to. From that, the decision-maker can consider waiting for the opponent to play or not. Of course, this policy takes into account the ability for the opponent to apply a rule she has already applied before.

We investigated another example (Example 25) where some predicates can be removed from the state. The purpose of this example is to show that the solving algorithm gives an optimal policy, even if a cycle can be created by the agents when adding and removing arguments.

Example 25. *This example contains three arguments* ***a, b, c*** *and a special argument* ***s*** *meaning agent 1 surrenders and thus loses the debate immediately. Rules are:*

- $\mathcal{R}_1 = \{h_1(\boldsymbol{a}) \wedge a(\boldsymbol{b}) \Rightarrow [1.0 / \boxplus a(\boldsymbol{a}) \wedge \boxplus e(\boldsymbol{a}, \boldsymbol{b}) \wedge \boxminus e(\boldsymbol{b}, \boldsymbol{a})]$
  $a(\boldsymbol{c}) \Rightarrow [1.0 / \boxplus a(\boldsymbol{s})]\}$
- $\mathcal{R}_2 = \{h_2(\boldsymbol{b}) \wedge h_2(\boldsymbol{c}) \Rightarrow [0.9 / \boxplus e(\boldsymbol{b}, \boldsymbol{a}) \wedge \boxminus e(\boldsymbol{a}, \boldsymbol{b}),$
  $0.1 / \boxplus a(\boldsymbol{c}) \wedge \boxplus e(\boldsymbol{c}, \boldsymbol{a})]\}$

The initial state is $(\{h_1(\boldsymbol{a})\}, \{a(\boldsymbol{b})\}, \{h_2(\boldsymbol{b}), h_2(\boldsymbol{c})\})$, $g_1 = g(a)$.

Figure 22 shows the optimal policy graph for Example 25.

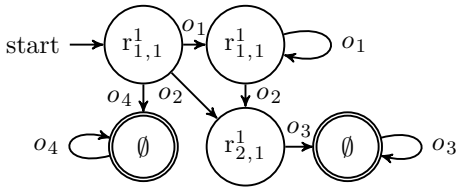


Figure 22: Policy graph for Example 25

The observations of agent 1 are as follows:
$o_1 = \{a(\mathbf{a}), e(\mathbf{b}, \mathbf{a})\}, o_2 = \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b}), a(\mathbf{c}), e(\mathbf{c}, \mathbf{a})\}$
$o_3 = \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b}), a(\mathbf{c}), e(\mathbf{c}, \mathbf{a}), a(\mathbf{s})\}, o_4 = \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b})\}$

Finally, we investigated the influence of each optimization on the computation time[3]. Table 3 reports computation times required to solve the problems while applying different sets of optimizations before solving the problem with MO-SARSOP. We considered Example 18, Example 25 and a slightly modified version (in order to fit it in our framework) of Dvorak (Dv.) problem taken from DBAI group (2013). A dash in the table means that the computation of the optimal policy took more than 30 min. and 0 means that the time is less than 0.01 secs.

	None	Irr.	Enth.	Dom.	Irr(s_0).	All
Ex 18	—	—	—	—	—	0.56
Ex 25	3.3	0.3	0.3	0.4	0	0
Dv.	—	—	—	—	—	32
6	1313	22	43	7	2.4	0.9
7	—	180	392	16	20	6.7
8	—	—	—	—	319	45
9	—	—	—	—	—	—

Table 3: Computation time (in seconds)

We can see that for Example 18 only the fully optimized problem can be solved in a reasonable amount of time. In order to study how the method scales, we also generated instances built on bipartite argumentative graphs (but not necessarily trees) with an increasing number of arguments evenly split among the two agents. In Table 3, line n (where $n = 6, \ldots, 9$) shows the time needed to solve problems with n arguments.

Our experiments show the effectiveness of these optimization schemes, which make several examples solvable in practice. Nonetheless the optimal resolution remains extremely costly, and the algorithms considered in the experiments seem very unlikely to handle instances involving more than a dozen of arguments. In order to improve the efficiency of solving methods, we could use the POMCP algorithm presented in Section 3.6.1. Indeed, without using this algorithm, the optimization procedures help to tackle problems of higher dimension but POMCP is not as limited as the other algorithms by the size of the problems. Hence, in this context, using the procedures would allow POMCP to reach a better quality solution.

4.4 Debate mediation

Argumentation debates involve different conflicting agents, or teams of agents, exchanging arguments to persuade each other. Although such debates may take place without a mediator, in some situations (*e.g.*, large number of agents), it is necessary to call on a mediator to preside the debate. When a mediator is introduced, she acts as a referee among debating parties (single agents or teams). Her role is essentially to allocate turn-taking, but she could also decide on issues being discussed, that is, set the *agenda* of the discussion.

The problem of mediation has recently emerged as an important challenge for formal argumentation. Quoting Janier and Reed (2017): *"the development of argumentation theories linked to computational applications opens promising new horizons since computational tools could support mediators, making sessions quicker and more efficient"*. In (Prakken, 2008) a persuasion dialogue game for two players is

[3] The experiments have been performed on a machine equipped with an Intel XeonX5690 4.47 Ghz core and 16G of RAM.

extended to consider a neutral *adjudicator*. In (Janier et al., 2016), a dialectical system designed for mediation is proposed. Such dialogue games look in details at the types of moves that can be played, and prescribe what agents can play, but not *how* the mediator should play.

In our setting, we suppose that agents are split into several teams, exchanging arguments to persuade each other. The role of the mediator is to decide which agent of which team will speak next. To solve this decision problem, our mediator exploits her knowledge about the debating agents and more specifically about their argumentative strategies. A first issue is the amount of information that the mediator has at her disposal. While it is conceivable that the mediator knows which team each agent belongs to, it is difficult to assume that she could assign a deterministic strategy to each agent, or that agents play optimally. Instead, agents will be viewed as reasoning with probabilistic strategies as investigated in the previous section of this chapter. This represents both the fact that an agent can act non-deterministically and that the mediator does not know perfectly the strategy of each agent. However, assuming that those strategies are stationary may also be too strong.

4.4.1 *Dynamic Mediation Problems*

Inspired by the APS formalization introduced in the previous section, we proposed the *Dynamic Mediation Problem* (DMP) framework to consider a strategic mediator managing turn-taking between debating agents with non-stationary strategies (Hadoux et al., 2018a). Although our framework extends APS, it tackles different issues: an APS formalizes the decision problem of a strategic debating agent without mediation whereas a DMP considers the decision problem of a mediator facing non-stationary debating agents. Our objective is to allow an active mediator to decide for the best turn-taking sequence by adapting to the changes of argumentative behaviors.

A DMP is defined by a tuple $\langle \mathcal{N}, \mathcal{T}, \mathcal{A}, \mathcal{E}, \mathcal{P}, (\mathcal{M}_i)_{i\in\mathcal{N}}, ((\mathcal{R}_i^{\mu})_{\mu\in\mathcal{M}_i})_{i\in\mathcal{N}}, (\mathcal{B}_i)_{i\in\mathcal{N}}, (g_j)_{j=0\ldots|\mathcal{T}|}, (\mathcal{F}_i)_{i\in\mathcal{N}}\rangle$ with:

- $\mathcal{N}$, a set of agents,
- $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{|\mathcal{T}|}\}$, a set of teams (i.e. subset of agents) where $\mathcal{T}$ forms a partition of $\mathcal{N}$,
- $\mathcal{A}, \mathcal{E}$ and $\mathcal{P}$ as in APS,
- $\mathcal{M}_i = \{\mu_i^1 \cdots \mu_i^l\}$, the set of argumentative behaviors for agent i,
- $\mathcal{R}_i^{\mu_i^j}$, the set of rules of agent i in the argumentative behavior $\mu_i^j \in \mathcal{M}_i$,
- $\mathcal{B}_i : \mathcal{M}_i \times \mathcal{M}_i \to [0, 1]$ models the probability of agent i to change from one behavior to another,
- g_j, the goal of team $\mathcal{T}_j$ and g_0, the mediator's goal,
- $\mathcal{F}_i : \mathcal{M}_i \times \mathcal{M}_i \times \mathbb{N} \to [0, 1]$ models the probability of agent i to move from one behavior to another after a given number of steps in the first behavior.

Debating agents are split into several teams so that all members of a same team share the same common argumentative goal. A goal (for a team or for the mediator) consists in having some arguments present or absent from the grounded extension of the arguments played in the common public debate space.

We use the labeling $\{in, out, undec\}$ to characterize which arguments are *in* and which arguments are *out* at the end of the debate. More specifically, this labeling allows us to determine which agent is the winner of the debate and whether the goals of the mediator are fulfilled or not.

We consider a general context where the mediator does not observe the private states of the debating agents. Although private states of the debating agents are not explicitly represented in a DMP, probabilities on acts in the rules indirectly formalize how private knowledge influences the moves of the debating agents.

We exemplify our application context and framework with the following example:

Example 26. *A government is discussing a bill to legalize communication surveillance. Two teams debate at the legislative assembly: the pro- and the anti-bill. The modeling contains 9 arguments (4 pros and 5 cons):*

*(**a**) anonymization software should not be seen as suspicious,*
*(**b**) innocents have nothing to hide,*
*(**c**) whistleblowers are not protected,*
*(**d**) sensitive jobs are protected (journalists/lawyers),*
*(**e**) no judge is required to monitor a user,*
*(**f**) the system is controlled by an independent committee,*
*(**g**) the government can possibly abuse control,*
*(**h**) the bill should allow any form of control to be bypassed in case of "absolute emergency",*
*(**i**) no possible control on the hidden algorithm.*

Figure 23 describes the attack graph between arguments.

Assume that $g_1 = \{in(\boldsymbol{c}), in(\boldsymbol{i})\}, g_2 = \{in(\boldsymbol{d}), in(\boldsymbol{h})\}$.

*Under the grounded semantics, **a**, **c**, **h**, and **f** are acceptable, thus* g_2 *is not fully satisfied.*

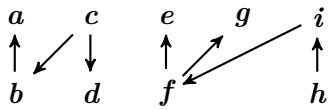


Figure 23: Graph of arguments and attacks

For the sake of clarity, we only give below examples of rules in one of the modes for two agents i *and* j *from two opposite teams:* i *(resp.* j*) belongs to team* $\mathcal{T}_1$ *(resp.* $\mathcal{T}_2$*). For conciseness, we remove the attacks from the rules, though they are still used to determine which arguments are attacked or defended.*

$\mathcal{R}_i : \{\emptyset \Rightarrow [0.7/\boxplus a(\boldsymbol{a}) \vee 0.3/\boxplus a(\boldsymbol{e})]$
$a(\boldsymbol{b}) \Rightarrow [0.55/\boxplus a(\boldsymbol{g}) \vee 0.45/\boxplus a(\boldsymbol{c})]$,
$a(\boldsymbol{d}) \Rightarrow [0.5/\boxplus a(\boldsymbol{i}) \vee 0.5/\boxplus a(\boldsymbol{c})]$,
$a(\boldsymbol{f}) \Rightarrow [0.9/\boxplus a(\boldsymbol{c}) \vee 0.1/\boxplus a(\boldsymbol{i})]$,
$a(\boldsymbol{e}) \wedge a(\boldsymbol{f}) \Rightarrow [1.0/\boxplus a(\boldsymbol{i})]\}$

$\mathcal{R}_j : \{\emptyset \Rightarrow [0.6/\boxplus a(\boldsymbol{d}) \vee 0.4/\boxplus a(\boldsymbol{h})]$,
$a(\boldsymbol{a}) \Rightarrow [0.7/\boxplus a(\boldsymbol{d}) \vee 0.3/\boxplus a(\boldsymbol{b})]$,
$a(\boldsymbol{e}) \Rightarrow [0.8/\boxplus a(\boldsymbol{f}) \vee 0.2/\boxplus a(\boldsymbol{f})]$,
$a(\boldsymbol{g}) \Rightarrow [0.5/\boxplus a(\boldsymbol{f}) \vee 0.5/\boxplus a(\boldsymbol{b})]$,
$a(\boldsymbol{i}) \Rightarrow [1.0/\boxplus a(\boldsymbol{h})]\}$

Argumentative modes

Each possible stationary strategy μ_i^j of an agent i is referred to as an argumentative *behavior*. Transitions between behaviors formalize the non-stationarity of the agents' argumentative behaviors (each time an agent changes her strategy, she will move to another argumentative behavior). Of course, the agents' current behavior cannot be directly observed by the mediator.

In order to define the possible argumentative behaviors, we proposed to exploit the typology of *constructive* versus *destructive* behavior, due to Moore (1993) and consider two argumentative behavior for each agent. This constitutes a basic case that has the advantage of being grounded in

argumentation theory, and which can be easily extended with mixtures of these extreme behaviors. In the *constructive behavior*, the agent favor acts that build her goals, while in the *destructive behavior* she seeks to destroy the arguments that are potential goals of the other team. As the debate progresses, the agent may become less constructive towards her goal and more destructive towards the goal of the adversary. For instance, an agent can feel to be rushed if the time has almost run out, she can also feel angry or bored if the debate lasts for too long and thus becomes more aggressive. Probabilities over acts in the rules thus vary from one mode to another in order to reflect these changes in the argumentative behavior.

Example 27. ***Example 26 continued.*** *Consider the following modes, capturing different attitudes for an agent of team $\mathcal{T}_2$ when argument $\boldsymbol{a}$ holds on the public state (either to put forward argument $\boldsymbol{d}$, which is one of the goal of team $\mathcal{T}_2$, or instead attack argument $\boldsymbol{a}$ by playing argument $\boldsymbol{b}$).*

constructive:	$a(\boldsymbol{a}) \Rightarrow [0.7/\boxplus a(\boldsymbol{d}) \vee 0.3/\boxplus a(\boldsymbol{b})]$
destructive:	$a(\boldsymbol{a}) \Rightarrow [0.2/\boxplus a(\boldsymbol{d}) \vee 0.8/\boxplus a(\boldsymbol{b})]$
mean:	$a(\boldsymbol{a}) \Rightarrow [0.45/\boxplus a(\boldsymbol{d}) \vee 0.55/\boxplus a(\boldsymbol{b})]$

The mediation problem

A DMP models a multi-team debate mediated by a strategic agent. A *sequence of speak-turns* is a sequence of agents organized by the mediator. Note that a *sequence of acts* σ is the sequence of acts effectively performed by the agents involved in a speak-turns sequence.

Example 28. ***Example 26 continued.*** *Let agents 1, 2 and 3 be in team $\mathcal{T}_1$ and agents 4 to 7 be in team $\mathcal{T}_2$. We assume that agents in a same team have the same rules. The sequence of speak-turns $(1, 4, 1, 5)$ starting from state $s = \emptyset$ can yield the sequence of acts $(r^1_{1,1}, r^4_{2,1}, r^1_{3,1}, r^5_{5,1})$ with $r^k_{i,j}$ the j-th act of rule i of agent k. After the application of this sequence of acts, the public state is $s' = a(\boldsymbol{a}) \wedge a(\boldsymbol{d}) \wedge a(\boldsymbol{h}) \wedge a(\boldsymbol{i})$.*

The objectives of the mediator can be of different nature and will be captured through an appropriate setting of the reward function of the decision problem. These objectives are usually defined so as to ensure that the debate will follow some desired rules. Mediating debates is a longstanding issue in democracies. As early as 1876, Henry Martin Robert designed a set of rules, Robert's Rule of Order (Robert, 2011), which prescribe how assembly discussions should be conducted. For instance, a general guideline of the rules is that "no member can speak twice on the same issue until everyone else wishing to speak has spoken once". But it also goes in much deeper details regarding the agenda of a meeting, the amendments and the motions, the votes, and how the "floor" (the right to speak) can be allocated in assembly discussions. Prakken and Gordon formalized (some of) those rules and argue that they may be used in electronic debates, showing by example how this could be done in the ZENO's discussion forum (Prakken and Gordon, 1999).

We distinguish different types of objectives that the mediator should pursue, and classify these principles as belonging either to the *efficiency* or the *fairness* of the debate.

DEBATE EFFICIENCY. An important feature to define the debate efficiency is the *goal of the debate*. Strict neutrality would imply that the mediator holds an empty goal. However, it is legitimate for the mediator to have an impartial goal which is slightly different from an empty goal. By impartial we mean that the mediator does not favor any team a priori. This could typically correspond to the goal of the interaction itself, which depends on the type of interaction considered (Walton and Krabbe, 1995; Prakken, 2006). Indeed, the main objective of the mediator is to lead the debate to its expected outcome. This is expressed by a goal (g_0) that the mediator pursues.

Example 29. ***Example 26 continued.*** *Suppose the mediator has identified the pro- and anti- bill teams. A possible impartial goal could be that both teams manage to reach a consensus at least on argument* $\boldsymbol{g}$*, i.e.* $g_0 = \{in(\boldsymbol{g})\}$*. Another type of impartial goal would be* $g_0 = \{in(\boldsymbol{c}) \vee in(\boldsymbol{h})\}$*. These goals are impartial in the sense that this is a disjunction which does not discriminate between goals of team* $\mathcal{T}_1$ *and goals of team* $\mathcal{T}_2$*.*

• *Impact on audience (Imp)*— At each turn, the debate yields a public state where the goal of the mediator can be evaluated. Sometimes it makes sense to do so *at each step* of the debate, for instance, for debates broadcasted on radio, where an audience might be convinced depending on how long they were exposed to convincing arguments. Sometimes only the state *at the end* of the debate is relevant, as in a trial, where only the ultimate state is considered by the judges.

• *Progress of the debate (Prog)*— Making regular progress in the debate should be favored, i.e. circular arguments (Mackenzie, 1979) or empty moves should be discouraged, and the mediator is legitimate to intervene to avoid this.

• *Length of the debate (Len)*— Short debates are preferred.

DEBATE FAIRNESS. The following properties are crucial to ensure that the debate is conducted in a way that is fair to all the participating agents.

• *Alternation between teams (Alt)*— This is one of the main guidelines of Robert's Rules of Order (Robert, 2011), and it has a very intuitive appeal. Janier et al. (2016) also note that fairness is achieved in their system by balancing the agents' positions. We reformulate this rule in the context of several teams: "the turn should not be given to the same team again, as long as all the other teams did not have the opportunity to speak".

• *Fair opportunity to respond (Resp)*— Priority should be given to agents who have (supposedly) a move directly connected to the most recent argument made in the debate. This captures a notion of relevance, but not as stringent as the one used in (Prakken, 1998; Bonzon and Maudet, 2011), which enforces that the status of the issue of the dialogue is directly impacted by the move.

• *Full participation (Part)*— As long as agents have something relevant to say, they must in principle be allowed to do so. Robert's Rules of Order prescribe that: "under no circumstances should "undue strictness" be allowed to intimidate members or limit full participation". This means that no team should have the power to decide upon the termination of the debate on their own, and that the mediator should pay attention to leave it open as long as required.

Clearly, all of these principles may not be satisfied simultaneously: they are even sometimes contradictory. In the next section, we are going to see how all these principles can be captured through some appropriate setting of states, goals and reward functions.

4.4.2 *Dealing with Non-stationary Behaviors.*

Since the moves of the debating agents are uncertain, the decision problem of the mediator can be viewed as a sequential decision-making problem under uncertainty. The objective of the mediator is then to maximize a value function ensuring that the computed policy, i.e. the sequence of speak-turns, yields the highest expected discounted sum of rewards. An additional difficulty comes from the non-stationarity of the decision problem. The following two observations suggest that HS3MDPs can handle DMP problems: (1) there is a fixed and known number of possible environment dynamics, which corresponds to all combinations of the debating agents' behaviors; (2) an environment dynamics mode prevails for several time steps since agents engage in a consistent behavior and keep the same behavior over several time steps.

Formally, the decision problem of the mediator in a DMP can be modeled as an HS3MDP (see Section 3.6.1) with the following components:

- $\mathbf{M} = \prod_{i \in \mathcal{D}} \mathcal{M}_i$ the set of all possible combinations $(\mu_1, \cdots, \mu_{|\mathcal{D}|})$ of argumentative behaviors such as μ_i is a behavior of agent i in the DMP. The elements of $\mathbf{M}$ are numbered and denoted m_k. Each mode m_k corresponds to an MDP $\langle S, A, T_k, R_k \rangle$ with:
  - $S = \mathcal{P} \times \{1, \ldots, |\mathcal{T}|\}$, all possible combinations of public states, plus the team of the agent who has just spoken,
  - $A = \mathcal{N}$, as an action consists of allowing one agent to speak, i.e. to fire one rule,
  - T_k and R_k (for each mode $m_k \in \mathbf{M}$), as specified below.
- $C : \mathbf{M} \times \mathbf{M} \to [0, 1]$ the transition function over modes induced by $\mathcal{B}_i$ of the DMP, assuming independence between the changes of the agents' behavior.
- $H : \mathbf{M} \times \mathbf{M} \times \mathbb{N} \to [0, 1]$ the mode duration function derived from $\mathcal{F}_i$ of the DMP with the independence assumption. There is a (HS3MDP) mode change if the duration of at least one agent's behavior is equal to zero.

Capturing Efficiency and Fairness Principles.

The reward function R_k formalizes the objectives of the mediator and has to be defined in compliance with the semantics of the problem. The different objectives of the mediator can generally be captured with a specifically designed reward function, and simultaneous objectives can be handled by combining (*e.g.*, additively) several reward signals. We are now going to see how the principles of mediation presented above can be captured. As we shall see, some of them would require to augment the state space.

For *efficiency*:

- Progress of the debate *(Prog)* is captured by assigning negative rewards to *vacuous acts*, i.e. acts that do not change the state of the debate. A less obvious situation is to avoid *circular arguments*. Such moves can be penalized with a negative reward, but one would need to augment the state space to represent the *history* of the moves.
- For the impact on audience*(Imp)*, recall that the mediator may have some impartial goals (*e.g.*, consensus) specific to the debate. This is simply handled by giving a positive reward when (parts of) those goals hold. We distinguish *final* vs. *step-wise* reward depending on whether the reward is given in the final step, or at each decision step. In the step-wise case, the reward for a fulfilled goal is given at each step where the goal holds. In the final approach, a reward is only given at the end of the debate if the goal of the mediator holds.
- Regarding the length of the debate *(Len)*, it suffices to penalize every time-step with a small negative reward value, or alternatively to use an adequate discount factor (the smaller the factor, the shorter the sequences of moves).

For *fairness*:

- To favor alternation *(Alt)* between two teams, a negative reward can be given to the mediator if she lets the same team speak twice consecutively (by choosing this *alternation penalty* high enough, strict alternation can be enforced). When considering more than two teams, the state has to be augmented with the history of the $|\mathcal{N} - 1|$ last teams' turns.
- Regarding fair opportunity to respond *(Resp)*, the mediator may be rewarded if a move is relevant, i.e. related to the last (or some recent) moves. In this case, the state space would need to be augmented to keep track of a few last moves.

- Finally, full participation *(Part)* is guaranteed in our model since the debate only ends when each team has triggered the skip act.

To summarize, the designer only has to specify *goal* and *relevance rewards*, along with *progress*, *length* and *alternation penalties* to model the mediator of her choice. As already mentioned, while some objectives are synergistic (*e.g.*, progress of the debate and full participation), others are contradictory (*e.g.*, length of the debate and full participation). In the latter case, different rewards may cancel out. While modeling a DMP, the relative importance of the objectives needs to be tuned by setting appropriately the values of the different reward signals.

4.4.3 *Mode detection and strategy computation*

To account for the high dimensionality of the HS3MDP obtained by converting a DMP, we used our adaptation of POMCP developed to solve HS3MDPs (see Section 3.6.1).

We ran experiments to test the relevance of formalizing the possible behaviors of the agents in the decision process. We compared the performance of the mediator while making decisions using an HS3MDP policy against a policy issued from a mean model over all behaviors. Indeed, exploiting a mean model is a common method (see, *e.g.*, Doya et al. (2002); da Silva et al. (2006)) that approximates the non-stationarity to solve the problem while allowing for the use of standard algorithms. It can perform well if the additional information brought by the non-stationary model is not significant enough. In the experiments, given an instance of debate mediation, the mean model is defined by averaging over the behaviors, rule by rule, the probability distributions over possible acts. We obtain a "mean" MDP with stationary state transition and reward functions (see Example 27). The HS3MDP and the "mean" MDP are then solved using POMCP. We report the performances of both approaches.

Each part of Table 4 corresponds to debates involving respectively 3 agents in one team vs. 4 in the other, 12 vs. 12, 25 vs. 25 and finally 50 vs. 50. For each team size, we generate 100 instances of the problem described in Example 26 with different probabilities on acts in the rules. The problems were defined randomly for each agent with respect to the behaviors, i.e. in the constructive behavior, the probability of the act moving the debate towards the goal is higher than the probability of trying to defeat the opponent. The mediator's goal is randomized for each instance. We recorded the mediator's performance (i.e. discounted sum of rewards) for each instance and average over the 100 instances. We also increased the numbers of simulations done by POMCP while averaging over 1000 runs with the given number of simulations. The number of simulations is the number of Monte-Carlo executions done in the simulator before executing in the real environment the best action found. It starts with eight simulations and doubles the number of simulations until it takes more than one hour for 1000 runs (for at least one of the 100 averaging instances). Recall that POMCP is theoretically guaranteed to tend towards the optimal solution when increasing the number of simulations (Silver and Veness, 2010). Note that, in a real context, the decision-maker chooses a number of simulations suitable to the application and running-time requirements.

In these experiments, we used a goal reward of 10 for each part of the goal accepted, -100 for the alternation penalty and a discount factor of 0.9 accounting for both the length and the progress penalties. Note that the fact that the final reward is positive or negative has no specific meaning. Reported performances correspond to the results obtained by the mediator using *step-wise* and *final* reward functions. The left value of each column is obtained using the "mean" model and the right value is obtained using the HS3MDP model. Bold face values mean that the relative improvement is at least 1% and that the difference is statistically significant. Both values are in bold face in the opposite case. For all sizes of instances, with a sufficient number of simulations, one can see that HS3MDP always outperforms the "mean" model for both reward functions. However, as the size of the instances increases, more simulations are needed to outperform the "mean" model. In fact, without

Teams	# Sim.	Step-wise	Final
3-4	8	-93.66 / **-86.28**	-116.97 / **-108.90**
	16	-52.26 / **-39.40**	-79.27 / **-64.99**
	32	-10.29 / **-4.99**	-35.49 / **-30.40**
	64	3.12 / **4.46**	-21.27 / **-19.73**
	128	4.57 / **5.73**	-19.89 / **-18.82**
	256	4.36 / **5.97**	-19.90 / **-18.51**
12-12	64	-8.70 / **-5.82**	-36.20 / **-32.87**
	128	16.15 / **16.75**	-9.81 / **-9.46**
	256	20.58 / **20.87**	**-4.94** / **-4.97**
25-25	128	-5.08 / **-3.56**	-31.93 / **-30.68**
	256	-15.59 / **16.74**	-10.56 / **-10.28**
50-50	256	-1.50 / **-0.31**	-27.84 / **-27.32**

Table 4: Performances for Teams 3-4, 12-12, 25-25 and 50-50

# Sim.	Step-wise
8	-73.71/**-64.58**
16	-29.62/**-16.91**
32	13.94/**16.41**

Table 5: Performances for Teams 3-4 with disjunctive goals

enough simulations, the additional information brought by the HS3MDP model is not used and leads to wrong choices of actions when the model believes to be in a wrong mode. Nonetheless, it has to be noticed that, even for large-sized instances, the number of simulations required to outperform the "mean" model remains small. Furthermore, HS3MDPs can lead to significant improvements since the relative improvements are up to 79%. Apart from the results for Teams 12-12 and Teams 50-50 at 256 simulations for the "final" reward function, all results are statistically significant with $p < 0.05$ and most of them with $p < 0.001$ under a Student t-test.

DISCUSSION As a general model with minimal assumptions, DMPs can represent various mediation problems. Although we consider an active mediator, she does not take actions to directly modify the state of the debate. Yet, the mediator may be able to put forward arguments in the public space in order to make the debate evolve and escape from a dead end (*e.g.*, (Chalamish and Kraus, 2012; Janier and Reed, 2017)). In our framework, handling such mediators is straightforward: a fictitious team of only one agent, embodying the mediator, is added to the DMP. The rules of the fictitious player consists in the possible arguments the mediator may want to play. Putting forward an argument for the mediator consists in fact in letting this fictitious player speak. Finally, dialogue games that include a mediator (Janier et al., 2016; Prakken, 2008) suggest that other types of argumentative moves are useful (questioning, asking for resolution, etc.).

4.5 PERSPECTIVES

Strategic argumentation has been only recently investigated in formal argumentation. The work presented in this chapter raises many issues that have been little studied in the domain.

LEARNING OF THE OPPONENT MODEL In the previous sections, we assumed that the behavior of the opponent is probabilistically known. Specifically, we considered that, given a certain state of the debate, it is known probabilistically how the opponent may react. These probabilities may have been obtained by expert knowledge, or by observation of previous interactions with the same agent (or at least, type of agent), e.g., a vendor may be able to predict from past interactions the possible counter-arguments that could be put forward by a skeptical consumer. In (Franz et al., 2017), we focused on persuasion dialogue and we started investigating how the model of the opponent could be learnt from the interactions along the dialogue.

A persuasion dialogue involves a *persuader* trying to convince a *persuadee* (also called *opponent*) to believe in a combination of arguments. For instance, a doctor may try to convince a patient to stop smoking or to start eating healthier food. A persuasion dialogue is then a sequence of moves where the persuader and the persuadee alternatively take turn to put forward an argument or an attack in the dialogue. The number of steps of the dialogue is assumed to be limited to an horizon h. Hence, it gives more chance for the persuadee to keep engaged in the debate until the end.

Hadoux and Hunter (2017) investigated strategic argumentation in persuasion dialogues but they assumed that the opponent follows a specific decision rule that translates her predisposition towards the goal of the dialogue or the overall subject, and this decision rule is never updated. If the decision rule assumed to be used by the opponent does not correspond to the rule actually played, this may lead to bad performance for the persuader. In order to improve the performance of the persuader, we plan to develop frameworks where the persuader maintains a belief model of the opponent and could update this model during the dialogue.

The first key challenge to address is to *identify relevant models of the opponent*. While our works considered that the opponent behaviors is modeled by a set of probabilistic rules, other models could be more efficient. Rosenfeld and Kraus (2016) proposed to predict the opponent's model formalized as a Weighted Bipolar Argumentation Framework (WBAF). Nonetheless, WBAF includes an argument belief function and an interaction belief function that both return continuous values. The number of possible models of the opponent to consider is therefore infinite. An important issue is thus to specify adequate models summarizing the relevant information about the opponent and that can be efficiently exploited by the persuader to make decisions.

A second important issue will be to design efficient methods that compute an argumentation strategy allowing the persuader to *learn the opponent model* and that *exploit this knowledge to optimize her goal* as well. In the work presented in this chapter, we exploited a known model of the opponent to optimize an argumentative goal. Given an initial model of the opponent, we would like to let the persuader update this model as the dialogues goes on, i.e. as she obtains new observations from the opponent. The updated opponent model would then be exploited by the persuader in the next steps to make more efficient decisions.

Besides making decisions to reach her goal by updating and exploiting the model of the opponent, the persuader could also make decisions in the hope of improving her opponent model. Indeed, some moves in the dialogue may reveal more relevant information about the opponent behavior than others. It would then be interesting to allow the persuader to use some arguments in order to confirm whether her current model is valid or not and to refine the model. Nonetheless, such actions may incur a cost for the persuader. For instance, if the persuader selects the argument yielding to the highest expected information gain, she may open the door to counter-arguments that would decrease the overall value of the dialogue even if the opponent model is improved. For instance, the counter-argument may prevent the persuader from reaching the persuasion goal. On the other hand, if the opponent model is inaccurate, the persuader may take non-relevant decisions that hurt the overall value of the dialogue. The decision process has thus to take into account the long-term effects of persuader's actions and the opponent responses.

This decision problem could be formalized and solved as a POMDP where the state of the decision-problem would combine the model of the opponent and the state of the dialogue (Rosenfeld and Kraus, 2016). The persuader would only observe the state of the debate but observations about the moves of the opponent could provide some information about the model of the opponent. Belief states would then formalize the probability distribution over the possible opponent models. However, such a POMDP formalization requires to have prior probabilistic knowledge about the possible moves of the opponent, i.e. to *a priori* be able to fully define the observation function. In order to alleviate this assumption, we intend to investigate machine learning methods.

Moreover, a special attention has to be paid to the size of the POMDP formalization. This issue is closely related to opponent modeling since the number of possible opponent models would influence the state space size of the model. Abstract models of the opponent could also be investigated in order to merge similar models of the opponent. In order to solve large instances, we also plan to consider exploiting the structure of the argumentation framework.

EXPERIMENTATION WITH PEOPLE Until now, we investigated debating systems where the goal of the agents is to reach a desired outcome of the dialogue. The performances of our argumentation strategies have then been tested using virtual agents, by looking at the final state of the dialogue. However, when considering persuasion systems, it is important to evaluate whether the persuader managed to change the attitude of the persuadee. For instance, if the objective of the persuader (e.g. a doctor) is to make her patient eat healthier food, it is required to test whether the patient is indeed more willing to change her feeding habits.

Such an evaluation requires to *conduct experiments with human subjects* and comparative studies. Recently, some works presented some experiments on strategic persuasion involving human subjects (Rosenfeld and Kraus, 2016; Hadoux et al., 2018b; Polberg and Hunter, 2018). Undoubtedly, we will have to develop such experiments to test the strategic persuasion models that we envisioned above. Such experiments will contribute to bridge the gap between abstract argumentation and mixed systems involving human and virtual agents that interact by exchanging arguments.

IMPACT OF THE ARGUMENTS ON THE OPPONENT STRATEGY Although we investigated settings where the argumentative behaviors of the agents may evolve over time, these changes are assumed to be independent of the arguments put in the dialogue. However, some arguments may have an impact on the interlocutors and may change their private states and behaviors.

Such changes can be related to an *emotional reaction from the discussion partner* (Hadoux et al., 2018b). It would be interesting to investigate how these emotional effects could be handled and anticipated in argumentation strategies. One avenue to consider would be to relate the arguments to some identified topics or emotions that could influence the behavior of the opponent. The opponent model should then include some information about the topics that are willing to impact the opponent and how the emotions evoked by the arguments could change her behavior. Hadoux et al. (2018b) investigated the emotions invoked by the words used in persuasion dialogues. The strategy of the persuader is based on a multi-criteria decision process taking into account the belief in arguments and the emotional response evoked. In this work, the persuadee is assumed not to play strategically nor stochastically and instead always select the argument according to her believes regarding the emotional effect evoked by the persuader. Under probabilistic modeling of the opponent, it would be interesting to investigate settings where the arguments put forward in the dialogue change the distribution probability of the rules during the dialogue.

STRATEGIC OPPONENT Finally, we did not consider that the opponent plays strategically, i.e. adapts her strategy from the observations made along the dialogue. While it is a reasonable assumptions when the discussion partners are not competitive, this may not be the case in debates where the agents have conflicting goals. In fact, the opponent may also model her interlocutor and strategically adapt her decisions from the information obtained along the dialogue. *Game theory* and computational *theory of mind* offer interesting prospects to optimize decisions in such contexts.

5 CONCLUSION

The work presented in this document considered issues dealing with multiagent distributed planning and decision-making under uncertainty and partial observability.

In the first chapter, we studied the multiagent resource allocation problem from a distributed point of view. This work highlighted the relevance of bilateral deals in order to solve resource allocation problems in a distributed way. In this context, uncertainty on the system state arises from the limited observability of the agents about the current allocation. Moreover, when individual preferences are private information, each agent is uncertain about the acceptation of her proposals. We investigated fairness issues in distributed settings where each agent has partial and dynamic observations of the whole allocation. In particular, our work introduced new notions of envy-freeness accounting for incomplete and incorrect knowledge when only the number of resources held is assumed to be known initially. We also developed a distributed protocol and informed heuristics allowing the agents to exploit their knowledge about the system in order to decide for the next agent to contact.

In the second chapter, we considered multiagent planning for distributed control in uncertain and partial observable environments. We were more specifically interested in Markovian decision processes. We proposed to enrich the Dec-POMDP model to formalize multitask planning problems with constraints. Because of the high complexity of multiagent planning under uncertainty and partial observability, we investigated approximate solving approaches. While most existing approaches consist in centralized planning for distributed control, we paid particular attention to distributed planning algorithms allowing each agent to compute her strategy. We proposed different approaches based on agent and spatial decomposition to split the initial multiagent planning problem. We also described different methods to coordinate the resulting sub-problems. We introduced the notion of opportunity cost values. We also proposed to define a high-level MDP dedicated to coordination among sub-problems. In the last part of the chapter, we looked at non-stationary environments and investigated issues dealing with the detection of context changes. While our previous contributions make the assumption of a known probabilistic model of the uncertainty, we considered learning of the new dynamics of the environment. In adversarial domains, this is highly related to learning the model of the adversary.

In the third chapter, we were interested in the contribution of decision theory and planning to abstract argumentation. We highlighted the relevance of Markov models in two types of problems: optimizing the sequence of arguments of an agent in a debate and strategically organizing speak-turns allocated by a mediator in non-stationary mediation problems. In this work, we assumed that the uncertainty about the discussion partners can be modeled as a set of probabilistic rules. We showed that these planning problems can be formalized as an HS3MDP. Because of the high number of arguments and agents that may be involved in a debate, the size of the models quickly become too large. We thus proposed optimization procedures to reduce the size of the problems without impacting the optimality of the solutions.

The conclusion and the perspectives of this work have been highlighted at the end of each chapter. We end this final conclusion with general remarks about the issues discussed all along the document.

As soon as an agent does not have full observability of an uncertain environment, she has to make efficient decisions from her sequence of partial observations about the system. Designing a compact and sufficient representation of this sequence is not an easy task. When the uncertainty on the action

outcomes and on the observations is known and can be represented by probabilistic functions (as in POMDPs), the sequence of observations can be summarized as a belief state. Nonetheless, in multiagent systems, even a statistic on the possible states of the system is not sufficient to make optimal decisions. In fact, if an agent is not aware of the observations made by the other agents and of their policies, she is uncertain about the actions taken by the other agents. When the agents are cooperative and share a common reward function (as assumed in Dec-POMDPs), the recourse to a central planner is helpful since it allows for computing a distribution over the state and agent histories while computing joint strategies of control. However, when planning is distributed (as we considered in Chapter 3) or when the agents do not know the other agents' preferences (as it is the case in Chapter 2), anticipating the actions of the others is even more challenging. We described different contributions in order to provide practical solutions to these problems. In MARA context, for instance, we maintained knowledge on the global allocation based on local observations of the bundles and gave heuristics to estimate the uncertainty. In Chapter 3, we developed a distributed planning method where the agents can communicate some opportunity cost values to coordinate their strategies. This work provides approximate solutions since decisions do not rely on exact information about the other agents' strategies. Now, a major step forward would be to design solutions with upper bounds on the performances.

In adversarial settings (as studied in Chapters 3 and 4), it is also desirable to anticipate the actions of the adversaries. To do so, we envisioned learning a model of the adversary. Nonetheless, this requires the agent to collect enough observations about the adversary in order to build an accurate and correct model. Because of partial observability, the agent may miss some actions of the adversary and may not be able to build such a model. Under highly limited observability, it is likely that other kinds of approaches, such as robust planning, would perform better. Knowing the degree of observability of the agents, a relevant question is to decide whether it is worthwhile to try to learn a model of the adversary.

BIBLIOGRAPHY

Abdallah, S. and Lesser, V. (2005). Modeling Task Allocation Using a Decision Theoretic Model. In *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 719–726, Utrecht, Netherlands. ACM Press.

Abebe, R., Kleinberg, J., and Parkes, D. C. (2017). Fair Division via Social Comparison. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-17)*, pages 281–289, São Paulo, Brazil.

Abraham, D. J., Blum, A., and Sandholm, T. (2007a). Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 295–304, New York, NY, USA. ACM.

Abraham, D. J., Cechlárová, K., Manlove, D. F., and Mehlhorn, K. (2005). *Pareto Optimality in House Allocation Problems*, pages 3–15. Springer Berlin Heidelberg, Berlin, Heidelberg.

Abraham, D. J., Irving, R. W., and Manlove, D. F. (2007b). Two algorithms for the student-project allocation problem. *J. of Discrete Algorithms*, 5(1):73–90.

Agmon, N., Sadov, V., Kaminka, G. A., and Kraus, S. (2008). The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS '08)*, volume 1, pages 55–62.

Akin, H. L., Ito, N., Jacoff, A., Kleiner, A., Pellenz, J., and Visser, A. (2013). Robocup rescue robot and simulation leagues. *AI Magazine*, 34:78–87.

Aknine, S., Pinson, S., and Shakun, M. F. (2004). An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems*, 8(1):5–45.

Amanatidis, G., Birmpas, G., and Markakis, V. (2018). Comparing approximate relaxations of envy-freeness. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 42–48.

Amato, C. (2018). Decision-making under uncertainty in multi-agent and multi-robot systems: Planning and learning. In *In the Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.

Amato, C., Chowdhary, G., Geramifard, A., Ure, N. K., and Kochenderfer, M. J. (2013). Decentralized control of partially observable markov decision processes. In *Proceedings of the 2013 IEEE Conference on Decision and Control*, pages 2398–2405.

Amato, C., D.S., B., and Zilberstein, S. (2007). Optimizing memory-bounded controllers for decentralized pomdps. In *Proceedings of the Twenty Third Conference on Uncertainty in Artificial Intelligence*.

Amato, C., Konidaris, G., Anders, A., Cruz, G., How, J. P., and Kaelbling, L. P. (2016). Policy search for multi-robot coordination under uncertainty. *The International Journal of Robotics Research*, 35(14):1760–1778.

Amato, C., Konidaris, G., Cruz, G., Maynor, C., How, J., and Kaelbling, L. (2015). Planning for decentralized control of multiple robots under uncertainty. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*.

Amato, C., Konidaris, G. D., and Kaelbling, L. P. (2014). Planning with macro-actions in decentralized pomdps. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1273–1280, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Annibal B. M. da Silva, Luis G. Nardin, J. S. S. (2000). *RoboCup Rescue Simulator Tutorial*.

Araya-López, M., Thomas, V., Buffet, O., and Charpillet, F. (2010a). A closer look at MOMDPs. In *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.

Araya-López, M., Thomas, V., Buffet, O., and Charpillet, F. (2010b). A closer look at MOMDPs. In *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*.

Arrow, K., Sen, A., and Suzumura, K. (2002). *Handbook of social choice and welfare*, volume 1. North-Holland.

Arrow, K. J. (1951). *Social choice and individual values.(Cowles Commission Mongr. No. 12.)*. Wiley.

Artificielle, F. I. (2017). Rapport de syntheèse des groupes de travail. Technical report.

Asadpour, A. and Saberi, A. (2010). An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989.

Åström, K. J. (1965). Optimal control of Markov Decision Processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205.

Aziz, H., Biró, P., Lang, J., Lesca, J., and Monnot, J. (2016a). Optimal reallocation under additive and ordinal preferences. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, pages 402–410.

Aziz, H., Bouveret, S., Caragiannis, I., Giagkousi, I., and Lang, J. (2018). Knowledge, fairness, and social constraints. In *Proceedings of the 32nd AAAI conference on Artificial Intelligence (AAAI'18)*, New Orleans, Louisiana, USA. AAAI Press. (to appear).

Aziz, H., Kalinowski, T., Walsh, T., and Xia, L. (2016b). Welfare of sequential allocation mechanisms for indivisible goods. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 787–794.

Bansal, N. and Sviridenko, M. (2006). The santa claus problem. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 31–40, New York, NY, USA. ACM.

Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall.

Baumeister, D., Bouveret, S., Lang, J., Nguyen, N.-T., Nguyen, T. T., and Rothe, J. (2014). Scoring rules for the allocation of indivisible goods. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, ECAI'14, pages 75–80, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Becker, R., Zilberstein, S., Lesser, V., and Goldman, C. V. (2004). Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22(1):423–455.

Bei, X., Qiao, Y., and Zhang, S. (2017). Networked Fairness in Cake Cutting. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 3632–3638, Melbourne, Australia.

Bellman, R. (1957). A Markovian decision process. *Indiana University Mathematical Journal*, 6.

Bellman, R. E. (2003). *Dynamic Programming*. Dover Publications, Incorporated.

Bernoulli, D. (1954). *Exposition of a New Theory on the Measurement of Risk*. University of Chicago Press.

Bernstein, D., Zilberstein, S., and Immerman, N. (2002a). The complexity of decentralized control of mdps. In *Mathematics of Operations Research*, pages 27(4):819–840.

Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. (2002b). The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840.

Bernstein, D. S., Zilberstein, S., Washington, R., and Bresina., J. L. (2001). Planetary rover control as a Markov decision process. In *Proceedings of the The Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Montreal, Canada.

Beviá, C. (1998). Fair allocation in a general model with indivisible goods. *Review of Economic Design*, 3(3):195–213.

Beynier, A. (2016). Cooperative Multiagent Patrolling for Detecting Multiple Illegal Actions Under Uncertainty. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, San José, United States. Best Paper.

Beynier, A. (2017). A multiagent planning approach for cooperative patrolling with non-stationary adversaries. *International Journal on Artificial Intelligence Tools*, 26(05):1760018.

Beynier, A., Bouveret, S., Lemaître, M., Maudet, N., and Rey, S. (2018). Efficiency, Sequenceability and Deal-Optimality in Fair Division of Indivisible Goods. Technical report.

Beynier, A., Chevaleyre, Y., Gourvès, L., Lesca, J., Maudet, N., and Wilczynski, A. (2018a). Local envy-freeness in house allocation problems. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 292–300, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Beynier, A. and Estivie, S. (2013). Optimizing distributed resource exchanges in multiagent systems under uncertainty. pages 8–16, Roma, Italy.

Beynier, A., Maudet, N., and Damamme, A. (2018b). Fairness in multiagent resource allocation with dynamic and partial observations. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 1868–1870, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Beynier, A. and Mouaddib, A.-I. (2009). Decentralized Decision Making Process for Document Server Networks. In Huang, J. and Srikant, R., editors, *IEEE International Conference on Game Theory for Networks (GAMENETS 2009)*, pages 26–32, Istanbul, Turkey. IEEE.

Beynier, A. and Mouaddib, A.-I. (2011a). Applications of DEC-MDPs in multi-robot systems. In Enrique Sucar, Eduardo Morales, J. H., editor, *Decision Theory Models for Applications in Artificial Intelligence Concepts and Solutions*, pages 361–384. IGI Global.

Beynier, A. and Mouaddib, A.-I. (2011b). Solving efficiently Decentralized MDPs with temporal and resource constraints. *Autonomous Agents and Multi-Agent Systems*, 23(3):486 – 539.

Black, D. (1948). On the rationale of group decision-making. *The Journal of Political Economy*, 56(1):23.

Black, D., Newing, R. A., McLean, I., McMillan, A., and Monroe, B. L. (1958). *The theory of committees and elections*. Springer.

Black, E. and Atkinson, K. (2011). Choosing persuasive arguments for action. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*, pages 905–912.

Black, E., Coles, A. J., and Hampson, C. (2017). Planning for persuasion. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil*, pages 933–942.

Bonzon, E. and Maudet, N. (2011). On the outcomes of multiparty persuasion. In *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 47–54.

Bordini, R. H., Dastani, M., Dix, J., and Seghrouchni, A. E. F. (2014). *Multi-Agent Programming: Languages, Tools and Applications*. Springer Publishing Company, Incorporated.

Bourgne, G., Seghrouchni, A. E. F., and Maudet, N. (2009). Towards refinement of abductive or inductive hypotheses through propagation. *Journal of Applied Logic*, 7(3):289 – 306. Special Issue: Abduction and Induction in Artificial Intelligence.

Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Int. Res.*, 11(1):1–94.

Boutilier, C., Patrascu, R., Poupart, P., and Schuurmans, D. (2006). Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artif. Intell.*, 170(8-9):686–713.

Bouveret, S., Endriss, U., and Lang, J. (2010). Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 387–392.

Bouveret, S. and Lang, J. (2008). Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *J. Artif. Int. Res.*, 32(1):525–564.

Bouveret, S. and Lang, J. (2011). A general elicitation-free protocol for allocating indivisible goods. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 73–78. AAAI Press.

Bouveret, S. and Lemaître, M. (2014). Characterizing conflicts in fair division of indivisible goods using a scale of criteria. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 1321–1328.

Bouveret, S. and Lemaître, M. (2016). Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems*, 30(2):259–290.

Bouveret, S. and Lemaître, M. (2016). Efficiency and sequenceability in fair division of indivisible goods with additive preferences. In *Proceedings of the Sixth International Workshop on Computational Social Choice (COMSOC'16)*, Toulouse, France.

Brams, S., Kilgour, D., and Klamler, C. (2014). Two-person fair division of indivisible items: An efficient, envy-free algorithm. *Notices of the American Mathematical Society*, 61(2):130–141.

Brams, S. and Taylor, A. (1996). *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.

Brams, S. J. and Taylor, A. D. (2000). *The Win-win Solution. Guaranteeing Fair Shares to Everybody*. W. W. Norton & Company.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D. (2016). *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA, 1st edition.

Bredereck, R., Kaczmarczyk, A., and Niedermeier, R. (2018). Envy-free allocations respecting social networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 283–291. International Foundation for Autonomous Agents and Multiagent Systems.

Bruner, M. and Lackner, M. (2015). On the likelihood of single-peaked preferences. *CoRR*, abs/1505.05852.

Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061 – 1103.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21:376–386.

Burt, R. S. (1982). *Toward a Structural Theory of Action: Network Models of Social Structure, Perception and Action*. Quantitative studies in social relations. Elsevier Science.

Caminada, M. (2006). On the issue of reinstatement in argumentation. In *Logics in artificial intelligence*, pages 111–123. Springer.

Caragiannis, I., Kaklamanis, C., Kanellopoulos, P., and Kyropoulou, M. (2009). On low-envy truthful allocations. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT-2009)*, pages 111–119, Venice, Italy.

Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A. D., Shah, N., and Wang, J. (2016). The unreasonable fairness of maximum nash welfare. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 305–322, New York, NY, USA. ACM.

Cassandra, A. R., Littman, M. L., and Zhang, N. L. (1997). Incremental Pruning: A simple, fast, exact method for Partially Observable Markov Decision Processes. In *Proceedings of the 13th Conference on Uncertainties in Artificial Intelligence (UAI)*, pages 54–61.

Cavallo, R. (2012). Fairness and welfare through redistribution when utility is transferable. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Cayrol, C. and Lagasquie-Schiex, M.-C. (2005). Graduality in argumentation. *Journal of Artificial Intelligence Research (JAIR)*, 23:245–297.

Chadès, I., Carwardine, J., Martin, T., Nicol, S., Sabbadin, R., and Buffet, O. (2012). MOMDPs: A solution for modelling adaptive management problems. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*.

Chadès, I., Scherrer, B., and Charpillet, F. (2002). A heuristic approach for solving decentralized-POMDP: Assessment on the pursuit problem. In *Proceedings of the Sixteenth ACM Symposium on Applied Computing*.

Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 363–369.

Chalamish, M. and Kraus, S. (2012). Automed: an automated mediator for multi-issue bilateral negotiations. *Autonomous Agents and Multi-Agent Systems*, 24(3):536–564.

Chen, Y., Lai, J. K., Parkes, D. C., and Procaccia, A. D. (2013). Truth, justice, and cake cutting. *Games and Economic Behavior*, 77(1):284 – 297.

Chen, Y. and Shah, N. (2017). Ignorance is often bliss: Envy with incomplete information. Working paper, Harvard University.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodriguez-Aguilar, J. A., and Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica (03505596)*, 30(1).

Chevaleyre, Y., Endriss, U., Estivie, S., and Maudet, N. (2007a). Reaching Envy-free States in Distributed Negotiation Settings. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1239–1244, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Chevaleyre, Y., Endriss, U., Estivie, S., and Maudet, N. (2008). Multiagent resource allocation in k-additive domains: preference representation and complexity. *Annals of Operations Research*, 163(1):49–62.

Chevaleyre, Y., Endriss, U., Lang, J., and Sabatier, U. P. (2007b). Expressive power of weighted propositional formulas for cardinal preference modeling. In *In Proceedings of KR07*, pages 145–152.

Chevaleyre, Y., Endriss, U., and Maudet, N. (2007c). Allocating goods on a graph to eliminate envy. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, pages 700–705. AAAI Press.

Chevaleyre, Y., Endriss, U., and Maudet, N. (2017). Distributed Fair Allocation of Indivisible Goods. *Artificial Intelligence*, 242:1–22.

Choi, S., Yeung, D., and Zhang, N. (2000). An environment model for nonstationary reinforcement learning. In *NIPS*, pages 981–993.

Choi, S. P.-M., Zhang, N. L., and Yeung, D.-Y. (2001). Solving Hidden-Mode Markov Decision Problems. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 19–26.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 746–752, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *J. Artif. Int. Res.*, 35(1):161–191.

da Silva, B., Basso, E., Bazzan, A., and Engel, P. (2006). Dealing with non-stationary environments using context detection. In *ICML*.

Damamme, A., Beynier, A., Chevaleyre, Y., and Maudet, N. (2015). The Power of Swap Deals in Distributed Resource Allocation. In *The 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 625–633, Istanbul, Turkey.

Damamme, J. A. (2016). *Approche multi-agent pour les problèmes de partage*. PhD thesis, Université Pierre et Marie Curie.

David, E. and Jon, K. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA.

DBAI group (2013). dbai.tuwien.ac.at /research/project/argumentation/dynpartix/.

de Keijzer, B., Bouveret, S., Klos, T., and Zhang, Y. (2009). On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In Rossi, F. and Tsoukias, A., editors, *Algorithmic Decision Theory*, pages 98–110, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dean, T. and Lin, S.-H. (1995). Decomposition techniques for planning in stochastic domains. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1121–1127, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Decker, K. and Lesser, V. (1993). Quantitative modeling of complex environments. *International Journal of Intelligent Systems in Accounting Finance and Management*, 2(4):215–234.

Dibangoye, J. S., Amato, C., Buffet, O., and Charpillet, F. (2016). Optimally solving dec-pomdps as continuous-state mdps. *Journal of Artificial Intelligence Research*, 55:443–497.

Dickerson, J. P., Goldman, J., Karp, J., Procaccia, A. D., and Sandholm, T. (2014). The computational rise and fall of fairness. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1405–1411. AAAI Press.

Doya, K., Samejima, K., Katagiri, K., and Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Computing*, 14(6):1347–1369.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358.

Dunne, P. E., Wooldridge, M., and Laurence, M. (2005). The complexity of contract negotiation. *Artificial Intelligence*, 164(1):23 – 46.

Eker, B. and Akın, H. L. (2013). Solving decentralized pomdp problems using genetic algorithms. *Autonomous Agents and Multi-Agent Systems*, 27(1):161–196.

Endriss, U. (2017). *Trends in Computational Social Choice*. LULU Press.

Endriss, U. and Maudet, N. (2004). Welfare engineering in multiagent systems. In Omicini, A., Petta, P., and Pitt, J., editors, *Engineering Societies in the Agents World IV*, pages 93–106, Berlin, Heidelberg. Springer Berlin Heidelberg.

Endriss, U., Maudet, N., Sadri, F., and Toni, F. (2003). On optimal outcomes of negotiations over resources. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 177–184, New York, NY, USA. ACM.

Endriss, U., Maudet, N., Sadri, F., and Toni, F. (2006). Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348.

Esben, H. O., Maja, J. M., and Gaurav, S. S. (2002). Multi-robot task allocation in the light of uncertainty. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3002–3007.

Escoffier, B., Lang, J., and Öztürk, M. (2008). Single-peaked consistency and its complexity. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 366–370, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Estivie, S., Chevaleyre, Y., Endriss, U., and Maudet, N. (2006). How equitable is rational negotiation? In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 866–873, New York, NY, USA. ACM.

Evripidou, V. and Toni, F. (2012). Argumentation and voting for an intelligent user empowering business directory on the web. In Krötzsch, M. and Straccia, U., editors, *Web Reasoning and Rule Systems*, pages 209–212, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fang, F., Nguyen, T. H., Pickles, R., Lam, W. Y., Clements, G. R., An, B., Singh, A., and Tambe, M. (2016). Deploying paws to combat poaching: Game-theoretic patrolling in areas with complex terrains. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Fang, F., Stone, P., and Tambe, M. (2015). When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '15)*.

Feldman, A. M. and Kirman, A. (1974). Fairness and envy. *American Economic Review*, 64(6):995–1005.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Ferreira, P. R., dos Santos, F., Bazzan, A. L. C., Epstein, D., and Waskow, S. J. (2010). Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. *Autonomous Agents and Multi-Agent Systems*, 20(3):421–443.

Fischhoff, B., Goitein, B., and Shapira, Z. (1983). Subjective expected utility: A model of decision-making. In Scholz, R. W., editor, *Decision Making Under Uncertainty*, volume 16 of *Advances in Psychology*, pages 183 – 207. North-Holland.

Fishburn, P. (1970). *Utility Theory for Decision Making*. Wiley, New York.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. (2016). Learning to communicate to solve riddles with deep distributed recurrent q-networks. *CoRR*, abs/1602.02672.

Foerster, J. N., Nardelli, N., Farquhar, G., Torr, P. H. S., Kohli, P., and Whiteson, S. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. *CoRR*, abs/1702.08887.

Foley, D. K. (1967). Resource allocation and the public sector. *YALE ECON ESSAYS*, 7(1):45–98.

Franz, R., Beynier, A., Hadoux, E., Hunter, A., and Maudet, N. (2017). Using bayesian inference to guess the opponent model in persuasive argumentation. Technical report.

Galstyan, A., Czajkowski, K., and Lerman, K. (2005). Resource allocation in the grid with learning agents. *Journal of Grid Computing*, 3(1):91–100.

Gehrlein, W. V. and Fishburn, P. C. (1976). The probability of the paradox of voting: A computable solution. *Journal of Economic Theory*, 13(1):14 – 25.

Gerkey, B. P. and Matarić, M. J. (2002). Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768.

Ghosh, B. K. and Sen, P. K. (1991). *Handbook of Sequential Analysis*. CRC Press.

Goldman, C. V. and Zilberstein, S. (2004). Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174.

Gourvès, L., Lesca, J., and Wilczynski, A. (2017). Object allocation via swaps along a social network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 213–219, Melbourne, Australia.

Grabisch, M. (1997). k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(2):167 – 189. Fuzzy Measures and Integrals.

Gupta, J. K., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In Sukthankar, G. and Rodriguez-Aguilar, J. A., editors, *Autonomous Agents and Multiagent Systems*, pages 66–83. Springer International Publishing.

Hadjinikolis, C., Siantos, Y., Modgil, S., Black, E., and McBurney, P. (2013). Opponent modelling in persuasion dialogues. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 164–170.

Hadoux, E. (2015). *Markovian sequential decision-making in non-stationary environments: application to argumentative debates*. Theses, UPMC, Sorbonne Universités CNRS.

Hadoux, E., Beynier, A., Maudet, N., and Weng, P. (2018a). Mediation of debates with dynamic argumentative behaviors. In *COMMA 18, to appear*.

Hadoux, E., Beynier, A., Maudet, N., Weng, P., and Hunter, A. (2015). Optimization of Probabilistic Argumentation With Markov Decision Models. In *International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina.

Hadoux, E., Beynier, A., and Weng, P. (2014a). Sequential Decision-Making under Non-stationary Environments via Sequential Change-point Detection. In *Learning over Multiple Contexts (LMCE)*, Nancy, France.

Hadoux, E., Beynier, A., and Weng, P. (2014b). Solving Hidden-Semi-Markov-Mode Markov Decision Problems. In *Scalable Uncertainty Management*, volume 8720 of *Lecture Notes in Computer Science*, pages 176–189, Oxford, United Kingdom. Springer International Publishing.

Hadoux, E. and Hunter, A. (2017). Strategic sequences of arguments for persuasion using decision trees. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1128–1134.

Hadoux, E., Hunter, A., and Corrégé, J. (2018b). Strategic dialogical argumentation using multi-criteria decision making with application to epistemic and emotional aspects of arguments. In *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings*, pages 207–224.

Halpern, J. Y. (2003). *Reasoning About Uncertainty*. MIT Press, Cambridge, MA, USA.

Hanna, H. and Mouaddib, A. (2002). Task selection as decision making in multiagent system. In *International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 616–623.

Hansen, E. A., Bernstein, D. S., and Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 709–715. AAAI Press.

Haskell, W. B., Kar, D., Fang, F., Tamb, M., Cheung, S., and Denicola, L. E. (2014). Robust protection of fisheries with compass. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 2978–2983. AAAI Press.

He, H., Boyd-Graber, J., Kwok, K., and Daumé, III, H. (2016). Opponent modeling in deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1804–1813.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. (2017). A survey of learning in multiagent environments: Dealing with non-stationarity. *CoRR*, abs/1707.09183.

Hernandez-Leal, P., Munoz de Cote, E., and Sucar, L. E. (2013). Modeling non-stationary opponents. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 1135–1136, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Howard, R. A. (1970). *Dynamic programming and Markov processes*. MIT Press.

Hunter, A. (2014). Probabilistic strategies in dialogical argumentation. In *International Conference on Scalable Uncertainty Management (SUM'14) LNCS volume 8720*.

INRIA (2016). Intelligence artificielle, livre blanc numéro 01, les défis actuels et l?action d?inria. Technical report.

Jackson, M. O. (2008). *Social and Economic Networks*. Princeton University Press, Princeton, NJ, USA.

Jain, M., An, B., and Tambe, M. (2012). An overview of recent application trends at the AAMAS conference: Security, sustainability and safety. *AI Magazine*, 33(3):14–28.

Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., and Ordóòez, F. (2010). Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces*, 40(4):267–290.

Janier, M. and Reed, C. (2017). Towards a theory of close analysis for dispute mediation discourse. *Argumentation*, 31(1):45–82.

Janier, M., Snaith, M., Budzynska, K., Lawrence, J., and Reed, C. (2016). A system for dispute mediation: The mediation dialogue game. In *Computational Models of Argument*.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence Journal*, 101(1–2):99–134.

Kar, D., Fang, F., Delle Fave, F., Sintov, N., and Tambe, M. (2015). "a game of thrones": When human behavior models compete in repeated stackelberg security games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1381–1390, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Kavraki, L., Svestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. 12:566 – 580.

Kitano, H. and Tadokoro, S. (2001). Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52.

Kochenderfer, M. J., Amato, C., Chowdhary, G., How, J. P., Reynolds, H. J. D., Thornton, J. R., Torres-Carrasquillo, P. A., Üre, N. K., and Vian, J. (2015). *Decision Making Under Uncertainty: Theory and Application*. The MIT Press, 1st edition.

Kohler, D. A. and Chandrasekaran, R. (1971). A class of sequential games. *Operations Research*, 19(2):270–277.

Kontarinis, D., Bonzon, E., Maudet, N., and Moraitis, P. (2014). Empirical evaluation of strategies for multiparty argumentative debates. In *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings*, pages 105–122.

Koutsoupias, E. and Papadimitriou, C. (1999). Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS'99, pages 404–413, Berlin, Heidelberg. Springer-Verlag.

Kurniawati, H., Hsu, D., and Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*.

Kurokawa, D., Lai, J. K., and Procaccia, A. D. (2013). How to cut a cake before the party ends. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 555–561. AAAI Press.

Lerman, K., Jones, C., Galstyan, A., and Matarić, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *The International Journal of Robotics Research*, 25(3):225–241.

Lesca, J. and Perny, P. (2010). LP Solvable Models for Multiagent Fair Allocation problems. In *European Conference on Artificial Intelligence*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 393–398, Lisbon, Portugal. IOS Press.

Lipton, R. J., Markakis, E., Mossel, E., and Saberi, A. (2004). On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, pages 125–131.

Liu, M., Amato, C., Anesta, E. P., Griffith, J. D., and How, J. P. (2016). Learning for decentralized control of multiagent systems in large, partially-observable stochastic environments. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2523–2529.

Lozenguez, G. (2012). *Stratégie coopérative pour la mise en œuvre d'une flotte de robots mobiles dans un milieu ouvert et encombré. (Cooperative strategies for a fleet of mobile robots moving in open clustered environment)*. PhD thesis, University of Caen Normandy, France.

Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A.-I., and Martinet, P. (2011). Map Partitioning to Approximate an Exploration Strategy in Mobile Robotics. In *9th International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS'11)*, pages 63–72, Salamanca, Spain.

Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A.-I., and Martinet, P. (2012). Map Partitioning to Approximate an Exploration Strategy in Mobile Robotics. *Multiagent and Grid Systems - An International Journal of Cloud Computing* , 8(3):275–288.

Lozenguez, G., Adouane, L., Beynier, A., Mouaddib, A.-I., and Martinet, P. (2016). Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation. *Autonomous Robots*, 40(4):599–613.

Lozenguez, G., Mouaddib, A.-I., Beynier, A., Adouane, L., and Martinet, P. (2013). Simultaneous Auctions for "Rendez-Vous" Coordination Phases in Multi-robot Multi-task Mission. In *Intelligent Agent Technology*, pages 67–74, Atlanta, United States. IEEE.

Ma, J. (1994). Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23:75–83.

Mackenzie, J. D. (1979). Question-begging in non-cumulative systems. *Journal of philosophical logic*, 8(1):117–133.

Madani, O., Hanks, S., and Condon, A. (2003). On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1):5 – 34. Planning with Uncertainty and Incomplete Information.

Matignon, L., Jeanpierre, L., and Mouaddib, A.-I. (2012a). Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Matignon, L., Laurent, G. J., and Fort-Piat, N. L. (2012b). Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *Knowledge Eng. Review*, 27(1):1–31.

Mattei, N. (2011). *Empirical Evaluation of Voting Rules with Strictly Ordered Preference Data*, pages 165–177. Springer Berlin Heidelberg, Piscataway, NJ, USA.

Mattei, N. and Walsh, T. (2013). Preflib: A library of preference data. In *Proceedings of Third International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer.

Maudet, N., Parsons, S., and Rahwan, I. (2007). Argumentation in multi-agent systems: Context and recent developments. In Maudet, N., Parsons, S., and Rahwan, I., editors, *Argumentation in Multi-Agent Systems*, pages 1–16, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mazurowski, M. and Zurada, J. (2007). Solving decentralized multi-agent control problems with genetic algorithms. In *IEEE Congress on Evolutionary Computation*.

Modgil, S. and Caminada, M. (2009). Proof theories and algorithms for abstract argumentation frameworks. In Rahwan, I. and Simari, G., editors, *Argumentation in Artificial Intelligence*, pages 105–132. Springer.

Moore, D. J. (1993). *Dialogue game theory for intelligent tutoring systems*. PhD thesis, Leeds Metropolitan University.

Moulin, H. (1988). *Axioms of cooperative decision making*. Number 15. Cambridge University Press.

Moulin, H. (2003). *Fair division and collective welfare*. MIT Press.

Nair, R., Pradeep, V., Milind, T., and Makoto, Y. (2005). Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*.

Nair, R., Tambe, M., Yokoo, M., Marsella, S., and Pynadath, D.V. (2003). Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 705–711.

Narendra, K. S., Balakrishnan, J., and Ciliz, M. K. (1995). Adaptation and learning using multiple models, switching, and tuning. *Control Systems, IEEE*, 15(3):37–51.

Nash, J. (1950). The bargaining problem. *Econometrica*, 18(2):155–162.

Netzer, A., Meisels, A., and Zivan, R. (2016). Distributed envy minimization for resource allocation. *Autonomous Agents and Multi-Agent Systems*, 30(2):364–402.

Neumann, J. and Morgenstern, O. (1953). *Theory of games and economic behavior*. Princeton Univ. Press, 3. ed. edition.

Nguyen, T. H., Jiang, A. X., and Tambe, M. (2014a). Stop the compartmentalization: unified robust algorithms for handling uncertainties in security games. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 317–324.

Nguyen, T. H., Kar, D., Brown, M., Sinha, A., Jiang, A. X., and Tambe, M. (2016). Towards a science of security games. In Toni, B., editor, *Mathematical Sciences with Multidisciplinary Applications*, pages 347–381, Cham. Springer International Publishing.

Nguyen, T. H., Yadav, A., An, B., Tambe, M., and Boutilier, C. (2014b). Regret-based optimization and preference elicitation for stackelberg security games with uncertainty. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 756–762. AAAI Press.

Nguyen, T. H., Yang, R., Azaria, A., Kraus, S., and Tambe, M. (2013). Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 718–724. AAAI Press.

Nguyen, T. T. and Rothe, J. (2014). Minimizing envy and maximizing average Nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics*, 179:54–68.

NUS, N. (2014). http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl.

Oliehoek, F. A. (2012). *Decentralized POMDPs*, pages 471–503. Springer Berlin Heidelberg, Berlin, Heidelberg.

Oliehoek, F. A. and Amato, C. (2016). *A Concise Introduction to Decentralized POMDPs*. Springer Publishing Company, Incorporated, 1st edition.

Oliehoek, F. A., Spaan, M. T. J., Whiteson, S., and Vlassis, N. (2008). Exploiting locality of interaction in factored dec-pomdps. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 517–524, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Oliehoek, F. A., Spaan, M. T. J., and Witwicki, S. J. (2015). Factored upper bounds for multiagent planning problems under uncertainty with non-factored value functions. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1645–1651. AAAI Press.

Oliehoek, F. A. and Visser, A. (2006). A hierarchical model for decentralized fighting of large scale urban fires. In *Proceedings of the AAMAS Workshop on Hierarchical Autonomous Agents and Multi-Agent Systems*, pages 14–21.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P., and Vian, J. (2017). Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2681–2690.

Ong, S. C., Png, S. W., Hsu, D., and Lee, W. S. (2010). Planning under uncertainty for robotic tasks with mixed observability. In *The International Journal of Robotics Research*.

Ooi, J. M. and Wornell, G. W. (1996). Decentralized control of a multiple access broadcast channel: performance bounds. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 1, pages 293–298 vol.1.

Oren, N. and Norman, T. J. (2010). Arguing using opponent models. In McBurney, P., Rahwan, I., Parsons, S., and Maudet, N., editors, *Argumentation in Multi-Agent Systems*, pages 160–174, Berlin, Heidelberg. Springer Berlin Heidelberg.

Othman, A., Sandholm, T., and Budish, E. (2010). Finding approximate competitive equilibria: Efficient and fair course allocation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 873–880.

Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3):441–450.

Parsons, S. and Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Peshkin, L., Kim, K., Meuleu, N., and Kaelbling, L. (2000). Learning to cooperate via policy search. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 307–314.

Pigozzi, G., Tsoukiàs, A., and Viappiani, P. (2016). Preferences in artificial intelligence. *Annals of Mathematics and Artificial Intelligence*, 77(3):361–401.

Pineau, J., Gordon, G., and Thrun, S. (2003). Point-based value iteration: An anytime algorithm for pomdps. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 1025–1032.

Pita, J., John, R., Maheswaran, R., Tambe, M., and Kraus, S. (2012). A robust approach to addressing human adversaries in security games. In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, pages 660–665, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Polberg, S. and Hunter, A. (2018). Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *International Journal of Approximate Reasoning*, 93:487–543.

Prakken, H. (1998). Formalizing robert's rules of order. an experiment in automating mediation of group decision making. Technical Report GMD Report 12.

Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21:163–188.

Prakken, H. (2008). A formal model of adjudication dialogues. *Artificial Intelligence and Law*, 16:305–328.

Prakken, H. and Gordon, T. F. (1999). Rules of order for electronic group decision making - A formalization methodology. In *Collaboration between Human and Artificial Societies, Coordination and Agent-Based Distributed Computing.*

Procaccia, A. D. (2009). Thou shalt covet thy neighbor's cake. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 239–244, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Procaccia, A. D. and Wang, J. (2014). Fair enough: Guaranteeing approximate maximin shares. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 675–692, New York, NY, USA. ACM.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete dynamic stochastic programming.* John Wiley Chichester.

Pynadath, D. V. and Tambe, M. (2011). The communicative multiagent team decision problem: Analyzing teamwork theories and models. *CoRR*, abs/1106.4569.

Rabinovich, Z., Campus, S., Ram, G., Goldman, C. V., and Rosenschein, J. S. (2002). Non-approximability of decentralized control. Technical report.

Rabinovich, Z., Goldman, C., and Rosenschein, J. (2003). The complexity of multiagent systems: The price of silence. In *Proceedings of the International Conference on Autonomous Agents (AAMAS'03)*, volume 2.

Rahwan, I. and Simari, G. R. (2009). *Argumentation in Artificial Intelligence.* Springer Publishing Company, Incorporated, 1st edition.

Rawls, J. (1971). *A theory of justice.* Oxford university press.

Rienstra, T., Thimm, M., and Oren, N. (2013). Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13).*

Robert, H. M. (2011). *Robert's Rules of Order Newly Revised, 11th ed.* Da Capo Press.

Rosenfeld, A. and Kraus, S. (2014). Argumentation theory in the field: An empirical study of fundamental notions. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014.*

Rosenfeld, A. and Kraus, S. (2016). Strategical argumentative agent for human persuasion. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 320–328.

Roth, A. and Sotomayor, M. (1992). Two-sided matching: A study in game-theoretic modeling and analysis. *Games and Economic Behavior*, 4(1):161–165.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence : A Modern Approach*. Prentice Hall Series.

Saffidine, A., Schwarzentruber, F., and Zanuttini, B. (2018). Knowledge-Based Policies for Qualitative Decentralized POMDPs. In *32nd AAAI Conference on Artificial Intelligence*, New Orleans, United States.

Sandholm, T. and Lesser, V. (2001). Leveled Commitment Contracts and Strategic Breach. *Games and Economic Behavior*, 35:212–270.

Sandholm, T. W. (1998). Contract types for satisficing task allocation: I Theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*.

Scerri, P., Farinelli, A., Okamoto, S., and Tambe, M. (2005). Allocating tasks in extreme teams. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 727–734.

Segal-Halevi, E., Aziz, H., and Hassidim, A. (2017). Fair allocation based on diminishing differences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1254–1261. AAAI Press.

Sen, A. K. (1970). *Collective choice and social welfare*. North-Holland Publishing Co.

Seuken, S. and Zilberstein, S. (2007). Memory-bounded dynamic programming for dec-pomdps. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2009–2015.

Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of mathematical economics*, 1(1):23–37.

Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., and Meyer, G. (2012). Protect: A deployed game theoretic system to protect the ports of the united states. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 13–20, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Shieh, E. A., Jiang, A. X., Yadav, A., Varakantham, P., and Tambe, M. (2016). An extended study on addressing defender teamwork while accounting for uncertainty in attacker defender games using iterative dec-mdps. *Multiagent and Grid Systems*, 11:189–226.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, New York, NY, USA.

Shor, M. (2009). Procedural justice in simple bargaining games. Working papers 2012-25.

Silver, D. and Veness, J. (2010). Monte-Carlo planning in large POMDPs. In *Proceedings of the 24th Conference on Neural Information Processing Systems (NIPS)*, pages 2164–2172.

Sinha, A., Fang, F., An, B., Kiekintveld, C., and Tambe, M. (2018). Stackelberg security games: Looking beyond a decade of success. In *IJCAI Survey Track, to appear*.

Skinner, C. and Ramchurn, S. (2010). The robocup rescue simulation platform. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1647–1648.

Smith, T. and Simmons, R. (2004). Heuristic search value iteration for pomdps. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States. AUAI Press.

Sousa, P., Ramos, C., and Neves, J. (2000). Manufacturing entities with incomplete information. *Studies in Informatics and Control*, 9(2):79–88.

Steinhaus, H. (1948). The problem of fair division. *Econometrica*, 16:101–104.

Stone, P., Brooks, R., Brynjolfsson, E., Calo, R., Etzioni, O., Hager, G., Hirschberg, J., Kalyanakrishnan, S., Kamar, E., Kraus, S., Leyton-Brown, K., Parkes, D., Press, W., Saxenian, A., Shah, J., Tambe, M., and Teller, A. (2016). Artificial intelligence and life in 2030. Technical report, One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University, Stanford, CA.

Sycara, K. P. (1998). Multiagent systems. *AI Magazine*, 19:79–92.

Szer, D., Charpillet, F., and Zilberstein, S. (2012). Maa*: A heuristic search algorithm for solving decentralized pomdps. *CoRR*, abs/1207.1359.

Thibaut, J., Walker, L., LaTour, S., and Houlden, P. (1974). Procedural justice as fairness. *Stanford Law Review*, 26(6):1271–1289.

Thimm, M. (2014). Strategic argumentation in multi-agent systems. *Künstliche Intelligenz, Special Issue on Multi-Agent Decision Making*, 28(3):159–168.

Thimm, M. and Garcia, A. J. (2010). Classification and Strategical Issues of Argumentation Games on Structured Argumentation Frameworks. In van der Hoek, W., Kaminka, G. A., Lespérance, Y., Luck, M., and Sen, S., editors, *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems 2010 (AAMAS'10)*.

van Eemeren, F. H., Grootendorst, R. F., and Henkemans, F. S. (1996). *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications*. Lawrence Erlbaum Associates, Hillsdale NJ, USA.

Varakantham, P., Marecki, J., Yabu, y., Milind, T., and Makoto, Y. (2007). Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In *Proceedings of the International Joint Conference on Agents and Multiagent Systems (AAMAS-07)*.

Varian, H. R. (1974). Equity, Envy and Efficiency. *Journal of Economic Theory*, 9:63–91.

Viappiani, P. and Boutilier, C. (2010). Optimal bayesian recommendation sets and myopically optimal choice query sets. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, pages 2352–2360, USA. Curran Associates Inc.

Vidal, J. M. (2009). *Fundamentals of Multiagent Systems with NetLogo Examples*.

Vlassis, N. (2007). *A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence*. Morgan and Claypool Publishers, 1st edition.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186.

Walsh, T. (2015). Generating Single Peaked Votes. *CoRR*, abs/1503.02766.

Walton, D. and Krabbe, E. (1995). *Commitment in dialogue: basic concepts of interpersonal reasoning.* State University of New York Press.

Weiss, G. (1999). *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence.* MIT Press.

Wieser, F. (1889). *Valeur naturelle (Der natürliche Wert).*

Witwicki, S. J. and Durfee, E. H. (2010). Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS-2010)*, pages 185–192, Toronto, Canada.

Woolridge, M. (2001). *Introduction to Multiagent Systems.* John Wiley & Sons, Inc., New York, NY, USA.

Wu, F., Zilberstein, S., and Jennings, N. R. (2013). Monte-carlo expectation maximization for decentralized pomdps. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 397–403, Beijing, China.

Zhang, C., Sinha, A., and Tambe, M. (2015). Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*, pages 1351–1359.