



Proceedings of the First International Workshop on Argumentation and Applications (Arg&App 2023)

Oana Cocarascu, Sylvie Doutre, Jean-Guy Mailly, Antonio Rago

► To cite this version:

Oana Cocarascu, Sylvie Doutre, Jean-Guy Mailly, Antonio Rago. Proceedings of the First International Workshop on Argumentation and Applications (Arg&App 2023). First International Workshop on Argumentation and Applications, Sep 2023, Rhodes, Greece, France. 3472, [ceur-ws.org](https://www.ceur-ws.org), 2023, CEUR Workshop Proceedings. <hal-04935082>

HAL Id: hal-04935082

<https://hal.science/hal-04935082v1>

Submitted on 14 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

PROCEEDINGS OF THE FIRST INTERNATIONAL
WORKSHOP ON ARGUMENTATION AND
APPLICATIONS
(ARG&APP 2023)

CO-LOCATED WITH THE TWENTIETH INTERNATIONAL CONFERENCE ON
PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR
2023)

EDITED BY

OANA COCARASCU
SYLVIE DOUTRE
JEAN-GUY MAILLY
ANTONIO RAGO

King's College London
Université Toulouse Capitole
Université Paris Cité
Imperial College London

SEPTEMBER 2023
CEUR-WS

Preface

In recent years, the increasing availability of data and computational power has driven a remarkable proliferation of Artificial Intelligence (AI) systems in everyday users' lives. However, as data-driven AI systems have become commonplace, it has become increasingly clear that fields from symbolic AI have important roles to play in the future development of these systems, with one such candidate being Computational Argumentation. Formal models of argumentation have received a significant amount of attention in recent years, both within the Knowledge Representation and Reasoning community and from AI researchers in general. Given that argumentation is a mature discipline, it provides not only a wealth of theoretical formalisms suitable for a wide range of tasks, but a whole host of software instantiating these formalisms for real world settings. These strengths mean that argumentation is particularly adaptable to various application domains, e.g. cyber-democracy, explainable AI, law, medicine, multi-agent systems, public policy making, sustainable development, etc. The goal of this workshop was to emphasise the efforts of the community in this spirit and strengthen the links between formal works on argumentation, their implementations and these domains of application.

The workshop received 11 submissions, and we accepted 8 papers on diverse applications of argumentation. They cover a range of topics from the formal foundations of argumentation when deployed in a particular context, to demonstrations of application-driven, argumentative systems. The proceedings also include an invited paper describing the ICCMA 2023 competition. We hope that the works presented in the proceedings appeal not only to the growing argumentation community, but also to researchers in general who intend to use computational argumentation in their own applications.

We thank all the authors, the invited speakers Antonis Kakas, Tuomo Lehtonen and Andreas Niskanen, as well as the program committee members (listed below), for their valued contributions to the workshop.

September 2023

Oana Cocarascu
Sylvie Doutre
Jean-Guy Mailly
Antonio Rago

Program Committee

- Leila Amgoud (IRIT, CNRS, Toulouse)
- Katie Atkinson (University of Liverpool)
- Floris Bex (Utrecht University)
- Elena Cabrio (University Côte d'Azur)
- Madalina Croitoru (University of Montpellier, LIRMM)
- Anthony Hunter (University College London)
- Antonis Kakas (University of Cyprus)
- Santiago Marro (University Côte d'Azur)
- Nir Oren (University of Aberdeen)
- Fabio Paglieri (ISTC-CNR, Rome)
- Simon Parsons (University of Lincoln)
- Guilherme Paulino-Passos (Imperial College London)
- Nico Potyka (Imperial College London)
- Matthias Thimm (FernUniversität Hagen)
- Rallou Thomopolous (INRAE, Montpellier)
- Francesca Toni (Imperial College London)
- Srdjan Vesic (CRIL, CNRS, Lens)

Contents

1	Design of ICCMA 2023 - 5th International Competition on Computational Models of Argumentation: A Preliminary Report (M. Järvisalo and T. Lehtonen and A. Niskanen)	4
2	Stable Semantics for Epistemic Abstract Argumentation Framework (G. Alfano and S. Greco and F. Parisi and I. Trubitsyna)	11
3	Argumentative information improves automatic generation of counter-narratives against Hate Speech (D. A. Furman and P. Torres and J. A. Rodríguez and D. Letzen and M. V. Martínez and L. A. Alemany)	26
4	ArguCast: A System for Online Multi-Forecasting with Gradual Argumentation (D. Gorur and A. Rago and F. Toni)	40
5	A New Evolutive Generator for Graphs with Communities and its Application to Abstract Argumentation (J.-M. Lagniez and E. Lonca and J.-G. Mailly and J. Rossit)	52
6	ADP : An Argumentation-based Decision Process Framework Applied to the Modal Shift Problem (C. Leturc and F. Balbo)	65
7	A Discussion of Challenges in Benchmark Generation for Abstract Argumentation (I. Kuhlmann and M. Thimm)	78
8	Abstract Argumentation Applied to Fair Resources Allocation: A Preliminary Study (J.-G. Mailly)	85
9	Accessible Algorithms for Applied Argumentation (D. Odekerken and A. Borg and M. Berthold)	92

Design of ICCMA 2023, 5th International Competition on Computational Models of Argumentation: A Preliminary Report

Matti Järvisalo, Tuomo Lehtonen and Andreas Niskanen

HIIT, Department of Computer Science, University of Helsinki, Finland

Abstract

ICCMA 2023 constitutes the 5th instantiation of International Competitions on Computational Models of Argumentation, the main series of international competitions for evaluating the state of the art in practical system implementations for argumentative reasoning. In this short preliminary report, we provide an overview of the design of ICCMA 2023.

1. Introduction

The series of International Competitions on Computational Models of Argumentation (ICCMA, <http://argumentationcompetition.org>) aims at nurturing research and development of implementations for computational models of argumentation. The year 2023 marks the 5th instantiation of the biennial ICCMA competitions. ICCMA 2023 (<https://iccma2023.github.io>) welcomed contributions from the community at large in the forms of new argumentation reasoning problem benchmarks, and implementations of argumentation reasoners (for abstract and assumption-based argumentation) to be evaluated within ICCMA 2023 on a heterogeneous collection of benchmarks. The community at large was invited to submit argumentation reasoning system implementations (solvers) for participation in the competition as well as interesting and/or challenging benchmark instances for evaluating solvers competing in any of the ICCMA 2023 competition tracks. We provide a short preliminary overview of the design of ICCMA 2023. The results of the competition will be presented in conjunction with the KR 2023 conference after the writing of this overview and made available through the competition webpages.

2. Competition Tracks

ICCMA 2023 consists of four tracks: the *main track* and the special *approximate*, *dynamic*, and *ABA* tracks. Each track is composed of multiple subtracks, defined by a combination of a reasoning problem and an argumentation semantics. We use the following shorthands for semantics and reasoning tasks: CO, ST, PR, SST, STG, ID for complete, stable, preferred,

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ matti.jarvisalo@helsinki.fi (M. Järvisalo); tuomo.lehtonen@helsinki.fi (T. Lehtonen); andreas.niskanen@helsinki.fi (A. Niskanen)

ORCID  0000-0003-2572-063X (M. Järvisalo); 0000-0003-3197-2075 (A. Niskanen)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

semi-stable, stage and ideal semantics, respectively, and DC, DS and SE for credulous and skeptical acceptance, and finding a single extension, respectively. Argumentation systems could be submitted for evaluation into any choice of subtracks, i.e., no requirement to support e.g. all semantics for a specific reasoning problem, or all reasoning problems for a specific semantics were enforced.

Main Track concerns solvers for reasoning in abstract argumentation [1]. The focus of the Main track is to evaluate sequential core argumentation reasoning engines available in open source. Systems combining different core reasoning engines e.g. via portfolio-style techniques, systems employing parallel computations via the use of multiple processor cores, as well as systems which will not be made available in open source were invited to the special No-Limits track which consists of the same subtracks as the Main track. The ranking is otherwise the same as for the Main track, but wall-clock time is used instead of CPU time. The following combinations of the semantics and reasoning modes constitute the Main and No-Limits subtracks: DC- $\{CO|ST|SST|STG\}$, DS- $\{PR|ST|SST|STG\}$, SE- $\{PR|ST|SST|STG|ID\}$.

Approximate Track concerns in-exact solvers developed for abstract argumentation, i.e., solvers which may not in all cases provide correct YES/NO answers to credulous/skeptical queries. Correctness requirements and ranking are different than other tracks: incorrect solutions are simply discarded and only the number of correct solutions is taken into account. The subtracks in the Approximate track are DC- $\{CO|ST|SST|STG|ID\}$, DS- $\{PR|ST|SST|STG\}$.

Dynamic Track invites solvers built especially for answering credulous/skeptical queries over sequences of related AFs. Dynamic changes to an initial AF and acceptance queries are issued by different applications via IPAFAIR, an API for incremental reasoning in abstract argumentation specified for the first time for ICCMA 2023. Similarly to the Main track, an instance of the Dynamic track is an AF and a query argument. An instance is given as input to a program which modifies the initial AF by iteratively adding and deleting arguments and attacks, and checks whether the query argument is accepted in the resulting modified AFs. Resource limits are applied to this program as a whole, and an instance is solved exactly when this program terminates correctly. The subtracks in the Dynamic track are DC-CO, DS-PR, DC-ST, and DS-ST.

ABA Track concerns solvers developed for reasoning in the structured argumentation formalism of Assumption-based Argumentation (ABA) [2], specifically focusing on so-called flat ABA frameworks in the commonly studied logic programming fragment of ABA. In this fragment, atoms are derived from assumptions using rules with a list of atoms in the body and a non-assumption atom as the head. Assumptions have contraries, the derivation of which produces an attack on this assumption. The subtracks for the ABA track are DC- $\{CO|ST\}$, DS- $\{PR|ST\}$, SE- $\{PR|ST\}$.

Ranking Scheme. For the Main, Dynamic, Approximate and ABA tracks, the score of a solver on a subtrack is the sum of PAR-2 scores (CPU time if instance solved within resource

limits, $2 \times$ the per-instance time limit otherwise) of the solver over all instances of a subtrack. The No-limits ranking is otherwise the same but wall-clock time is used instead of CPU time. The winner of a subtrack is the solver with the lowest score. For the Approximate track, the solver with the largest number of correctly solved instances wins. If needed, cumulative CPU running time over solved instances is used as a tie-breaker.

Input-Output Interface. In short (see the website for details), a specific compact numerical input format for AFs was enforced for the Main, Dynamic and Approximate tracks. The format was also extended for use in the ABA track by beginning-of-line identifiers for distinguishing between assumptions, rules and contraries. In the Dynamic track, I/O is implemented using IPAFair, an incremental API for reasoning in AFs, with functionality for initializing a solver with an input AF and semantics, adding and deleting arguments and attacks, and performing credulous and skeptical acceptance queries. For details on IPAFair, see <https://bitbucket.org/coreo-group/ipafair>.

3. Rules and Execution

The rules are available in full on the ICCMA 2023 webpages. As a new development for 2023, the requirement of witnessing certificates was enforced in the main track as follows. For DS- σ , if the query argument is credulously accepted, solvers should output “YES” *along with a certificate*, i.e., a σ -extension containing the query. Analogously, for DS- σ , if the query argument is not skeptically accepted, solvers should output “NO” along with a certificate, i.e., a σ -extension not containing the query. The certificates were checked as follows (with the subtrack specification, an AF, a query argument, and an output produced by a solver participating in the Main track as input). First, we verified that a certificate is contained in the output in the required cases (SE apart from “NO” answers on SE-ST, “YES” answers for DC, and “NO” answers for DS), and for DC and DS, that it contains (DC) or does not contain (DS) the query. For subtracks involving CO and ST semantics, we constructed a standard SAT encoding [3] and verified that the certificate extends to a satisfiable assignment. For subtracks involving PR, SST, and STG semantics, we built the standard SAT encoding of CO (for PR and SST) or conflict-free (for STG) semantics, and in addition to verifying that the certificate yields a satisfiable assignment, verified the absence of a counterexample (a superset or a range-superset) via a SAT solver call. All solver calls were performed using the SAT solver Glucose [4] (v4.1) invoked via PySAT [5]. The UNSAT proofs produced by Glucose were recorded. For the SE-ID track, we instead verified that all solvers reported the same ideal extension.

The organizers used fuzz testing to check for potential buggy behavior exhibited by submitted solvers before the execution of the competition. When bugs were detected, the authors of the solvers concerned were contacted and bug fixes were allowed to the extent feasible in order to execute the competition on time. As for further solver requirements, solver descriptions were mandatory. Furthermore, for all tracks apart from No-Limits, solver source code originating from the authors (including modifications to third-party source code such as SAT solvers as part of a solver) must be submitted together with a corresponding solver binary. In practice, ICCMA 2023 was executed on a computing cluster of University of Helsinki, Finland, with

2.60-GHz Intel Xeon E5-2670 CPUs and 57GB RAM under AlmaLinux 8.4, including GCC 12.2.0, Clang 12.0.1, Boost 1.76.0, GLib 2.68.2, Rust 1.70.0, Java 17.0.4, and Python 3.9.5. A per-instance memory limit of 16 GB was enforced on all subtracks. A 1200-second per-instance time limit was enforced on the Main, Dynamic, and ABA tracks; for the Approximate track we set a 60-s per-instance time limit.

4. Benchmarks

Abstract Argumentation: Main, Approximate, and Dynamic Tracks. To sample benchmarks for the Main, Approximate, and Dynamic tracks, we collected all benchmark AFs submitted to ICCMA 2017 [6] (11 domains) and ICCMA 2019 [7] (2 domains). For the so-called GroundedGenerator, SccGenerator, and StableGenerator domains, new AFs (100 per domain) with similar parameters were generated by Matthias Thimm. In addition to these, a benchmark generator **crusti_g2io** (by Jean-Marie Lagniez, Emmanuel Lonca, Jean-Guy Mailly, Julien Rossit) submitted to ICCMA 2023 was used with suggested parameters to generate a new set of 450 AFs. This procedure resulted in 14 benchmark domains. From each of these, we sampled 25 AFs for the final benchmark set, with the exception of the **crusti_g2io** domain, from which 32 AFs were sampled. Finally, query arguments were sampled from the set of arguments with a non-zero number of attackers which are not self-attackers, to avoid trivial acceptance queries.

ABA Track. The benchmarks for the ABA track were generated with a simple random instance generator. The varying parameters are the number of atoms (25, 100, 500, 2000 or 5000), the proportion of atoms that are assumptions (10% or 30%), the maximum number of rules deriving each sentence (5 or 10), and the maximum size of each rule body (5 or 10). Ten instances with each combination of these parameters were generated for a total of 400 instances. For acceptance problems, the query for each instance was selected at random from the atoms for which there is at least one derivation in the given instance.

5. Participants

Number of solvers submitted for each track: three for the Main track, one for the No-Limits track, five for the Approximate track, and five for the ABA track.

Crustabri (by Jean-Marie Lagniez, Emmanuel Lonca and Jean-Guy Mailly) is a SAT-based solver—a rewritten version of CoQuiAAS [8]—supporting all subtracks in the Main track and ABA track, as well as DC-CO, DC-ST, and DS-ST in the Dynamic track.

Fudge [9] (by Matthias Thimm, Federico Cerutti and Mauro Vallati) is a SAT-based solver with support for all subtracks in the Main track.

μ -toksia [10, 11] (by Andreas Niskanen and Matti Järvisalo) is a SAT-based solver with support for all subtracks of the Main and Dynamic tracks.

PORTSAT (by Sylvain Declercq, Quentin Januel Capellini, Christophe Yang, Jérôme Delobelle and Jean-Guy Mailly) is a solver based on a portfolio of SAT solvers with support for DC-CO, DC-ST, DS-PR, DS-ST, SE-PR, and SE-ST subtracks of the No-Limits track.

κ -solutions (by Christian Pasero and Johannes P. Wallner) is a SAT-based solver with support for DC-CO, DC-ST, and DS-ST in the Dynamic track.

AFGCN v2 [12] (by Lars Malmqvist) is based on employing graph convolutional neural networks, and supports all subtracks of the Approximate track.

ARIPOTER-Degrees (by Jérôme Delobelle, Jean-Guy Mailly and Julien Rossit) is based on computing the grounded extension and comparing the in-degree and out-degree of the query argument, and supports all subtracks of the Approximate track.

ARIPOTER-HCAT (by Jérôme Delobelle, Jean-Guy Mailly and Julien Rossit) is based on the grounded and h-Categorizer gradual semantics, and supports all subtracks of the Approximate track.

fargo-limited (by Matthias Thimm) is based on an exact DPLL-style search algorithm for admissible sets, and supports all subtracks of the Approximate track.

harper++ (by Matthias Thimm) is based on approximating all acceptance tasks by using grounded semantics, and supports all subtracks of the Approximate track.

AcbAr [13] (by Tuomo Lehtonen, Anna Rapberger, Markus Ulbricht and Johannes P. Wallner) is based on translating ABA frameworks to AFs with support for all reasoning tasks in the ABA track.

ASPforABA [14, 15] (by Tuomo Lehtonen, Matti Järvisalo and Johannes P. Wallner) is an answer set programming approach for ABA with support for all reasoning tasks in the ABA track.

ASTRA (Andrei Popescu and Johannes P. Wallner) employs dynamic programming and supports DC-CO, DC-ST, DS-ST and SE-ST in the ABA track.

flexABLE [16, 17] (Martin Diller, Sarah Alice Gaggl, Piotr Gorczynca) implements specialized ABA algorithms, flexible dispute derivations and supports DC-CO and DC-ST in the ABA track.

As agreed with the ICCMA steering committee, for transparency all solver submissions involving any of the organizers of ICCMA 2023 were made known to the ICCMA steering committee before the submission deadline. In addition, benchmark selection was done using a random seed—811543731122527—concatenated from numbers sent separately to the organizers by each ICCMA steering committee member. The seed and benchmark selection scripts are available on the ICCMA 2023 website.

Acknowledgments

Work financially supported by Academy of Finland (grants 322869, 347588, 356046) and University of Helsinki Doctoral Programme in Computer Science DoCS. The organizers wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995) 321–358.
- [2] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, *Artificial Intelligence* 93 (1997) 63–101.
- [3] P. Besnard, S. Doutre, Checking the acceptability of a set of arguments, in: J. P. Delgrande, T. Schaub (Eds.), 10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, Canada, June 6-8, 2004, Proceedings, 2004, pp. 59–64.
- [4] G. Audemard, L. Simon, On the glucose SAT solver, *Int. J. Artif. Intell. Tools* 27 (2018) 1840001:1–1840001:25.
- [5] A. Ignatiev, A. Morgado, J. Marques-Silva, Pysat: A python toolkit for prototyping with SAT oracles, in: O. Beyersdorff, C. M. Wintersteiger (Eds.), Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings, volume 10929 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 428–437.
- [6] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the Second International Competition on Computational Models of Argumentation, *Artif. Intell.* 279 (2020).
- [7] S. Bistarelli, L. Kotthoff, F. Santini, C. Taticchi, Summary report for the Third International Competition on Computational Models of Argumentation, *AI Mag.* 42 (2021) 70–73.
- [8] J. Lagniez, E. Lonca, J. Mailly, CoQuiAAS: A constraint-based quick abstract argumentation solver, in: 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015, IEEE Computer Society, 2015, pp. 928–935.
- [9] M. Thimm, F. Cerutti, M. Vallati, Skeptical reasoning with preferred semantics in abstract argumentation without computing preferred extensions, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 2069–2075.
- [10] A. Niskanen, M. Järvisalo, Algorithms for dynamic argumentation frameworks: An incremental SAT-based approach, in: G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 849–856.
- [11] A. Niskanen, M. Järvisalo, μ -toksia: An efficient abstract argumentation reasoner, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020, 2020, pp. 800–804.
- [12] L. Malmqvist, Approximate Solutions to Abstract Argumentation Problems Using Graph Neural Networks., Ph.D. thesis, University of York, 2022.
- [13] T. Lehtonen, A. Rapberger, M. Ulbricht, J. P. Wallner, Argumentation Frameworks Induced by Assumption-based Argumentation: Relating Size and Complexity, in: Proceedings

- of the 20th International Conference on Principles of Knowledge Representation and Reasoning, 2023, pp. 440–450.
- [14] T. Lehtonen, J. P. Wallner, M. Järvisalo, Declarative algorithms and complexity results for assumption-based argumentation, *Journal of Artificial Intelligence Research* 71 (2021) 265–318.
 - [15] T. Lehtonen, J. P. Wallner, M. Järvisalo, Harnessing incremental answer set solving for reasoning in assumption-based argumentation, *Theory Pract. Log. Program.* 21 (2021) 717–734.
 - [16] M. Diller, S. A. Gaggl, P. Gorczyca, Flexible dispute derivations with forward and backward arguments for assumption-based argumentation, in: P. Baroni, C. Benz Müller, Y. N. Wáng (Eds.), *Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, October 20-22, 2021, Proceedings*, volume 13040 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 147–168.
 - [17] M. Diller, S. A. Gaggl, P. Gorczyca, Strategies in flexible dispute derivations for assumption-based argumentation, in: S. A. Gaggl, J. Mailly, M. Thimm, J. P. Wallner (Eds.), *Proceedings of the Fourth International Workshop on Systems and Algorithms for Formal Argumentation co-located with the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 3236 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 59–72.

Stable Semantics for Epistemic Abstract Argumentation Framework

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi and Irina Trubitsyna

Department of Informatics, Modeling, Electronics and System Engineering (DIMES),
University of Calabria, Rende, Italy

Abstract

Dung’s Abstract Argumentation Framework (AAF) has emerged as a central formalism in AI for modeling disputes among agents. A recent extension of the Dung’s framework is the so-called Epistemic Abstract Argumentation Framework (EAAF), which enhances AAF by allowing the representation of some pieces of epistemic knowledge [1]. EAAF generalizes the concept of attack in AAF, introducing *strong* and *weak epistemic attacks*, whose intuitive meaning is that an attacked argument is epistemically accepted only if the attacking argument is possibly or certainly rejected, respectively. The semantics of EAAF has been defined and studied for several argumentation semantics but not for the stable one, which is arguably one of the most investigated semantics in argumentation. Motivated by this, in this paper, we propose an intuitive stable semantics for EAAF that naturally extends that for AAF and coincides with the preferred semantics in the case of odd-cycle free EAAFs (analogously to what happens in the case of AAF). We analyze the complexity of two argumentation problems: *existence*, i.e. checking whether there is at least one epistemic extension; and *acceptance*, i.e. checking whether an argument is epistemically accepted.

1. Introduction

In the last decades, Argumentation [2, 3, 4] has become an important research field in the area of autonomous agents and multi-agent systems [5]. Argumentation has applications in several contexts, including modeling dialogues, negotiation [6, 7], and persuasion [8]. It has been widely used to model agents’ interactions [9, 10, 11, 12], especially in the context of debates [13, 14, 15].

Dung’s Abstract Argumentation Framework (AAF) is a simple yet powerful formalism for modeling disputes between two or more agents [16]. An AAF consists of a set of *arguments* and a binary *attack* relation over the set of arguments that specifies the interactions between arguments: intuitively, if argument a attacks argument b , then b is acceptable only if a is not. Hence, arguments are abstract entities whose status is entirely determined by the attack relation. An AAF can be seen as a directed graph, whose nodes represent arguments and edges represent attacks. Several argumentation semantics—e.g. *grounded* (gr), *complete* (co), *preferred* (pr), and *stable* (st) [16]—have been defined for AAF, leading to the characterization of σ -extensions, that intuitively consist of the sets of arguments that can be collectively accepted under semantics $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$.

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ g.alfano@dimes.unical.it (G. Alfano); greco@dimes.unical.it (S. Greco); fparisi@dimes.unical.it (F. Parisi); i.trubitsyna@dimes.unical.it (I. Trubitsyna)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)



Figure 1: AAF Λ of Example 1 (left) and EAAF Δ of Example 2 (right).

Example 1. Consider an AAF $\Lambda = \langle \{a, b\}, \{(a, b), (b, a)\} \rangle$ whose corresponding graph is shown in Figure 1(left). Λ describes the following scenario. A party planner invites Alice (a) and Bob (b) to join a party. Due to their old rivalry (i) Alice replies that she will not join the party if Bob does, and (ii) Bob replies that he will not join the party if Alice does. This situation can be modeled by AAF Λ , where an argument x states that “(the person whose initial is) x joins the party”. Under the stable semantics, there are two extensions $E_1 = \{a\}$ and $E_2 = \{b\}$ stating that only Alice or only Bob will attend the party, respectively. $\square$

Thus, as prescribed by E_1 and E_2 , in the previous example we have that the participation of Alice and Bob to the party is uncertain. To deal with uncertain information represented by the presence of multiple extensions, credulous and skeptical reasoning has been introduced. Specifically, an argument is credulously true (or accepted) if there exists an extension containing the argument, whereas an argument is skeptically true if it occurs in all extensions. However, uncertain information in AAF under multiple-status semantics proposed so far cannot be exploited to determine the status of arguments (which in turn influences the status of other arguments) by taking into account the information given by the whole set of extensions, as in the case of credulous and skeptical acceptance. To overcome such a situation, and thus provide a natural and compact way for expressing such kind of conditions, the use of *epistemic* arguments and attacks has been recently proposed in [1], leading to the definition of the so-called Epistemic Abstract Argumentation Framework (EAAF) which enhances AAF by allowing the representation of some pieces of epistemic knowledge. Informally, epistemic attacks allow considering all extensions and not only the current one. Thus, an epistemic attack from a to b is such that a defeats b if a occurs in at least one extension (*strong epistemic attack*) or in all extensions and at least one (*weak epistemic attack*), as illustrated in the following example.

Example 2. Consider the AAF Λ of Example 1 and assume that there are two more people: Carol (c) and David (d). Carol’s answer is that she will not attend the party if it is sure (i.e. it is skeptically true) that Alice will, whereas David answers that he will not attend the party if the participation of Bob is possible (i.e. it is credulously true). Intuitively, the party planner should conclude that, as the participation of both Alice and Bob is uncertain, Carol will attend the party, whereas David will not.

This situation can be modeled by means of the Epistemic AAF (EAAF) shown in Figure 1(right) where a defeats c with a weak epistemic attack, whereas b defeats d with a strong epistemic attack (we use the two kinds of edges represented in the figure to denote weak and strong epistemic attacks). Under the stable semantics, there are two extensions: $E_1 = \{a, c\}$ modeling the fact that Alice and Carol will attend the party, whereas Bob and David will not; and $E_2 = \{b, c\}$ modeling the fact that Bob and Carol will attend the party, whereas Alice and David will not. Observe that the epistemic arguments c and d (i.e. the arguments defeated by an epistemic attack) are *deterministic* [17], that is, they have the same acceptance status in all extensions (true for c and false for d). $\square$

Contributions. We introduce the stable semantics for Epistemic Abstract Argumentation Frameworks (EAAFs) and investigate the complexity of two fundamental problems (see below). The proposed EAAF semantics aims to let epistemic arguments be deterministic [17], that is, they have the same acceptance status in all extensions; the status of an argument depends on the credulous or skeptical acceptance of its attackers. Considering the dependence of the status of an argument on its attackers only is inspired by the well-known directionality property proposed for AAF [18, 19], which, if satisfied, then guarantees that the status of each argument depends only on that of its attackers. Specifically, our main contributions are as follows.

- We formally present EAAF stable semantics; it extends that of AAF and coincides with EAAF preferred semantics in case of odd-cycle free EAAFs (as it happens for the case of AAF).
- We investigate the complexity of the *acceptance* and *existence* problems under stable semantics. Our complexity results are summarized in Table 2 (in Section 4).

2. Preliminaries

We first review the Dung’s framework and then discuss an extension of AAF with epistemic constraints.

2.1. Abstract Argumentation Framework

An *Abstract Argumentation Framework* (AAF) is a pair $\langle A, \Omega \rangle$, where A is a (finite) set of *arguments* and $\Omega \subseteq A \times A$ is a set of *attacks* (also called *defeats*). Different argumentation semantics have been proposed for AAF, leading to the characterization of collectively acceptable sets of arguments called *extensions* [16].

Given an AAF $\Lambda = \langle A, \Omega \rangle$ and a set $S \subseteq A$ of arguments, an argument $a \in A$ is said to be *i) defeated* w.r.t. S iff $\exists b \in S$ such that $(b, a) \in \Omega$; *ii) acceptable* w.r.t. S iff $\forall b \in A$ with $(b, a) \in \Omega$, $\exists c \in S$ such that $(c, b) \in \Omega$. The sets of defeated and acceptable arguments w.r.t. S are defined as follows (where Λ is understood):

- $Def(S) = \{a \in A \mid \exists b \in S. (b, a) \in \Omega\}$;
- $Acc(S) = \{a \in A \mid \forall b \in A. (b, a) \in \Omega \text{ implies } b \in Def(S)\}$.

To simplify the notation, we will often use S^+ to denote $Def(S)$.

Given an AAF $\langle A, \Omega \rangle$, a set $S \subseteq A$ of arguments is said to be:

- *conflict-free* iff $S \cap S^+ = \emptyset$;
- *admissible* iff it is conflict-free and $S \subseteq Acc(S)$.

Given an AAF $\langle A, \Omega \rangle$, a set $S \subseteq A$ is an *extension* called:

- *complete* (co) iff it is conflict-free and $S = Acc(S)$;
- *preferred* (pr) iff it is a $\subseteq$ -maximal complete extension;
- *stable* (st) iff it is a total complete extension, i.e. a complete extension such that $S \cup S^+ = A$;
- *grounded* (gr) iff it is the $\subseteq$ -smallest complete extension.



Figure 2: AAF Λ of Example 3 (left) and AAF Λ' of Example 4 (right).

The set of complete (resp. preferred, stable, grounded) extensions of an AAF Λ will be denoted by $\text{co}(\Lambda)$ (resp. $\text{pr}(\Lambda)$, $\text{st}(\Lambda)$, $\text{gr}(\Lambda)$). It is well-known that the set of complete extensions forms a complete semilattice w.r.t. $\subseteq$, where $\text{gr}(\Lambda)$ is the meet element, whereas the greatest elements are the preferred extensions. All the above-mentioned semantics except the stable admit at least one extension. The grounded semantics, that admits exactly one extension, is said to be a *unique-status* semantics, while the others are said to be *multiple-status* semantics. With a little abuse of notation, in the following we also use $\text{gr}(\Lambda)$ to denote the grounded extension. For any AAF Λ , $\text{st}(\Lambda) \subseteq \text{pr}(\Lambda) \subseteq \text{co}(\Lambda)$ and $\text{gr}(\Lambda) \in \text{co}(\Lambda)$.

Example 3. Let $\Lambda = \langle A, \Omega \rangle$ be an AAF where $A = \{a, b, c\}$ and $\Omega = \{(a, b), (b, a), (b, c)\}$, whose graph is show in Figure 2 (left). The set of complete extensions of Λ is $\text{co}(\Lambda) = \{E_0 = \emptyset, E_1 = \{a, c\}, E_2 = \{b\}\}$. E_0 is the grounded extension, while E_1 and E_2 are preferred and stable extensions. $\square$

Given an AAF $\Lambda = \langle A, \Omega \rangle$ and a semantics $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$, for $g \in A$, the *credulous* (resp. *skeptical*) *acceptance* problem, denoted as CA_σ (resp. SA_σ) is deciding whether g is credulously (resp. skeptically) accepted, that is deciding whether g belongs to *any* (resp. *every*) σ -extension of Λ . Clearly, CA_{gr} and SA_{gr} coincide.

Recently, a satisfaction problem for AAF called *determinism* (DS_σ) has been introduced [17]. Given a σ -extension E , an argument $g \in A$ is said to be: *accepted* if $g \in E$; *rejected* if $g \in E^-$; *undecided* otherwise ($g \notin E \cup E^-$). For a semantic σ , an argument is said to be deterministic if all σ -extensions assign the same status (either accepted, rejected, or undecided) to it.

Finally, the *existence* (resp. *non-empty existence*) problem denoted as Ex_σ (resp. $Ex_\sigma^{-\emptyset}$) is deciding whether there exists at least one (resp. at least one non-empty) σ -extension for AAF Λ .

For AAFs, the complexity of the existence and acceptance problems has been investigated (see [20] for an overview). The complexity of the determinism problem is investigated in [17]. The complexity results of these problems are summarized in the left-hand side part of Table 2.

Example 4. Consider the AAF Λ of Example 3. Under preferred and stable semantics, both arguments a and b are credulously accepted. None of them is skeptically accepted, nor deterministic.

Considering the AAF Λ' obtained from Λ by adding the self-attack (c, c) (see Figure 2 (right)), there are three complete extensions $E'_0 = \emptyset$, $E'_1 = \{a\}$ and $E'_2 = \{b\}$. Both E'_1 and E'_2 are preferred extensions, but only E'_2 is stable. $\square$

2.2. AAF with Epistemic Constraints

An Epistemic Argumentation Framework (EAF) has been proposed in [21]. An EAF is a triple $\langle A, \Omega, C \rangle$, where $\langle A, \Omega \rangle$ is an AAF and C is an epistemic constraint, that is, a propositional

formula extended with the modal operators **K** and **M**. Here, the constraint is the belief of an agent which must be satisfied. Intuitively, $\mathbf{K}\phi$ (resp. $\mathbf{M}\phi$) states that the considered agent believes that ϕ is always (resp. possibly) true. EAF semantics is given by sets of feasible extensions of the underlying AAF, called ω -extension sets (ω -labeling sets in [21, 1]), consisting of maximal sets of arguments that satisfies the constraint. There could be different ω -extension sets (ω -sets) for the same epistemic formula, as shown in the following example.

Example 5. Consider the AAF $\Lambda = \langle A = \{a, b, c, d\}, \Omega = \{(a, b), (b, a), (c, d), (d, c), (b, c)\}$ having 5 complete extensions $E_0 = \emptyset$, $E_1 = \{a\}$, $E_2 = \{a, c\}$, $E_3 = \{a, d\}$ and $E_4 = \{b, d\}$. E_0 is the grounded extension, while E_2 , E_3 and E_4 are preferred and stable extensions. Under the preferred semantics, considering the epistemic constraint $C_1 = \mathbf{K}c$, there exists a unique ω -set $\{E_2\}$ for EAF $\langle A, \Omega, C_1 \rangle$, whereas considering $C_2 = \mathbf{K}c \vee \mathbf{K}d$ there are the two alternative ω -sets $\{E_2\}$ and $\{E_3, E_4\}$ for EAF $\langle A, \Omega, C_2 \rangle$. $\square$

We point out that despite the name Epistemic Argumentation Framework is used, the role of epistemic formulae is only that of introducing constraints over the set of feasible extensions, that is it is similar to that of constraints or preferences in AAF [22, 23, 24, 25, 26].

3. Epistemic Abstract Argumentation Framework

We augment AAF with epistemic attacks, leading to the concept of Epistemic Abstract Argumentation Framework (EAAF).

3.1. Syntax

We start by recalling the syntax of EAAF [1].

Definition 1 (Epistemic AAF). *An Epistemic AAF is a quadruple $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ where A is a set of arguments, $\Omega \subseteq A \times A$ is a set of (standard) attacks, $\Psi \subseteq A \times A$ is a set of weak (epistemic) attacks, and $\Phi \subseteq A \times A$ is a set of strong (epistemic) attacks such that $\Omega \cap \Psi = \Omega \cap \Phi = \Psi \cap \Phi = \emptyset$.*

In the following, we represent attacks $(a, b) \in \Omega$ by $a \rightarrow b$, $(a, b) \in \Psi$ by $a \Rightarrow b$, $(a, b) \in \Phi$ by $a \Rrightarrow b$. An EAAF $\langle A, \Omega, \Psi, \Phi \rangle$ can be seen as a directed graph, where A denotes the set of nodes and Ω , Ψ , and Φ denotes three different kinds of edges. Arguments defeated through epistemic attacks are called *epistemic arguments*.

We say that there is a path from an argument $a \in A$ to argument $b \in A$ if either (i) there exists an attack (a, b) in Δ or (ii) there exists an argument $c \in A$ and two paths, from a to c and from c to b . We say that an argument $b \in A$ *depends* on an argument $a \in A$ if b is *reachable* from a in Δ , that is, if there exists a path from a to b in Δ . Moreover, an argument a depends on attack $\gamma \in (\Omega \cup \Psi \cup \Phi)$ if there exists a path in Δ that contains γ and reaches a .

We now introduce well-formed and plain EAAFs.

Definition 2. *An EAAF Δ is said to be:*

- *well-formed* if there are no cycles in Δ with epistemic edges.
- *in plain form* if every epistemic argument is attacked by a single (epistemic) attack.

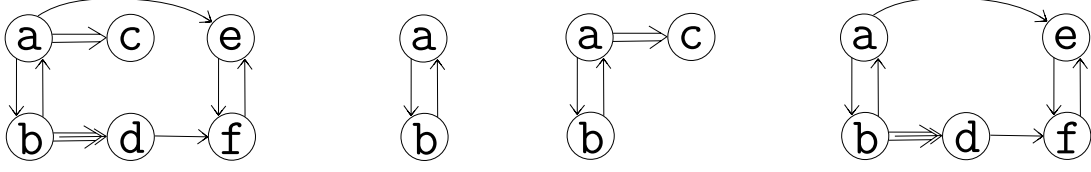


Figure 3: (From left to right) EAFs Δ , Δ' , Δ'' and Δ''' of Example 9.

In the following we assume that our EAFs are well-formed. The reason for such a restriction is to guarantee that there exists at most one *world view* (c.f. Theorem 1). In the following we also assume that our EAFs are in *plain form*. As it will be clear after introducing EAAF semantics, for well-formed EAFs in plain form, epistemic arguments are deterministic (c.f. Proposition 2).

Example 6. The EAAF $\Delta = \langle A = \{a, b, c, d\}, \Omega = \{(a, b), (b, a)\}, \Psi = \{(a, c)\}, \Phi = \{(b, d)\}$ of Example 2, whose graph is shown in Figure 1 (right), is well-formed and in plain form. $\square$

The semantics of EAAF is given by relying on the concept of sub-framework (sub-EAAF), which is defined as follows.

Definition 3. Given two EAFs Δ and Δ' , we say that Δ' is a sub-EAAF of Δ (denoted as $\Delta' \sqsubseteq \Delta$) if Δ' is obtained from Δ by deleting a subset S of the set of epistemic arguments of Δ and all the arguments depending on an argument in S w.r.t. Δ . Moreover, we write $\Delta' \sqsubset \Delta$ if $\Delta' \sqsubseteq \Delta$ and $\Delta' \neq \Delta$.

Clearly, in Definition 3 by deleting arguments we also delete attacks having as a source or target element a deleted argument.

Example 7. Consider the EAAF $\Delta = \langle \{a, b, c, d, e, f\}, \{(a, b), (b, a), (a, e), (d, f), (e, f), (f, e)\}, \{(a, c)\}, \{(b, d)\}$ shown in Figure 3 (left). We have four sub-EAFs $\Delta^* \sqsubseteq \Delta$, as shown in the figure: the first one (from left to right) coincides with Δ , the others are obtained by deleting all arguments depending on: (i) both arguments c and d , (ii) only d , and (iii) only c , respectively. $\square$

3.2. Semantics

We first introduce the stable semantics of EAAF and then present some results concerning properties of the proposed framework.

For any EAAF $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$, a set W of sets of arguments in A is called *world view* of Δ . Informally, a world view can be seen as a set of extensions that are to be used to compute the status of epistemic arguments. Given EAAF $\Delta' = \langle A', \Omega', \Psi', \Phi' \rangle \sqsubseteq \Delta$, we denote by $W_{\downarrow \Delta'} = \{S \cap A' \mid S \in W\}$ the projection of W over A' .

With the aim of providing EAAF semantics by extending AF semantics, we first extend the definitions of defeated and acceptable arguments for EAAF by taking into account the additional concept of world view, that is a candidate set of extensions, which is used to decide if an argument is epistemically defeated/acceptable. Given an EAAF Δ , a world view W of Δ ,

and a set $S \in W$, the sets of arguments defeated (resp. accepted) w.r.t. S and W are defined as follows:

- $Def(W, S) = \{b \in A \mid (\exists a \in S. a \rightarrow b) \vee (\exists T \in W. \exists a \in T. a \Rightarrow b) \vee (\forall T \in W. \exists a \in T. a \Rightarrow b)\}.$
- $Acc(W, S) = \{b \in A \mid \forall a \in A. ((a \rightarrow b) \text{ implies } a \in Def(W, S)) \wedge ((a \Rightarrow b) \text{ implies } \forall T \in W. a \in Def(W, T)) \wedge ((a \Rightarrow b) \text{ implies } \exists T \in W. a \in Def(W, T))\}.$

Example 8. Considering the EAAF Δ of Example 7 and the world view $W = \{S_1 = \{c\}, S_2 = \{a, c\}, S_3 = \{b, c\}\}$, we have that:

- $Def(W, S_1) = \{d\}$ and $Acc(W, S_1) = \{c\};$
- $Def(W, S_2) = \{b, d\}$ and $Acc(W, S_2) = \{a, c\};$ and
- $Def(W, S_3) = \{a, d\}$ and $Acc(W, S_3) = \{b, c\}.$

□

Given an EAAF $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ and a world view W of Δ , a set $S \in W$ is:

- W -conflict-free if $S \cap Def(W, S) = \emptyset;$
- W -admissible if it is W -conflict-free and $S \subseteq Acc(W, S);$
- W -complete (W -co) if it is W -conflict-free and $S = Acc(W, S).$

Moreover, a W -complete set S is said to be :

- W -preferred (W -pr) if S is $\subseteq$ -maximal;
- W -stable (W -st) if $S \cup Def(W, S) = A;$
- W -grounded (W -gr) if S is $\subseteq$ -minimal.

We are now ready to define EAAF semantics. The meaning of EAAF under the grounded, complete and preferred semantics has been introduced in [1]. For the sake of completeness and to easy readability we include those semantics in the next definition, where the meaning of EAAF under stable semantics is defined by generalizing the definition in [1].

Definition 4 (EAAF Semantics). *Let $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$ be a semantics and W a world view of EAAF Δ . Then, W is a σ -world view for Δ if $\forall \Delta' \sqsubseteq \Delta$ the following conditions hold:*

- every $S \in W_{\downarrow \Delta'}$ is a $W_{\downarrow \Delta'}$ - σ set, and*
- there is no world view W^* for Δ' such that $W_{\downarrow \Delta'} \subset W^*$ and every $S^* \in W^*$ is W^* - σ for Δ' .*

Table 1 σ -world view for each EAAF $\Delta^* \sqsubseteq \Delta$ in Figure 3.

Δ^*	$\text{gr}(\Delta^*)$	$\text{co}(\Delta^*)$	$\text{pr}(\Delta^*) = \text{st}(\Delta^*)$
Δ'	$\{\emptyset\}$	$\{\emptyset, \{a\}, \{b\}\}$	$\{\{a\}, \{b\}\}$
Δ''	$\{\emptyset\}$	$\{\{c\}, \{a, c\}, \{b, c\}\}$	$\{\{a, c\}, \{b, c\}\}$
Δ'''	$\{\emptyset\}$	$\{\emptyset, \{f\}, \{a, f\}, \{b\}, \{b, e\}, \{b, f\}\}$	$\{\{a, f\}, \{b, e\}, \{b, f\}\}$
Δ	$\{\emptyset\}$	$\{\{c\}, \{c, f\}, \{a, c, f\}, \{b, c\}, \{b, c, e\}, \{b, c, f\}\}$	$\{\{a, c, f\}, \{b, c, e\}, \{b, c, f\}\}$

We now explain Definition 4. Given a semantics σ , a W - σ set intuitively represents a candidate set of σ -extensions for an EAAF. Then, such a set turns out to actually be a set of extensions if the conditions in Definition 4 hold, whose rationale is as follows. Given a world view W of an EAAF Δ , we check that for all sub-frameworks Δ' , every element $S \in W' = W_{\downarrow \Delta'}$ is a W' - σ set (condition *i*) and W' is maximal (condition *ii*). Intuitively, the first condition ensures that the status of an argument is confirmed in all sub-frameworks considered. The second condition of Definition 4 ensures that, if there is a larger σ -world view for which condition *i* holds, then we prefer to take it. That is, intuitively, we aim at having the whole set of extensions. In [1], it is shown that this set is unique under grounded, complete and preferred semantics. Finally, as shown below in Example 9, checking that the above-mentioned conditions hold for all sub-frameworks is important to avoid returning wrong conclusions (i.e., world views that contradict our intuition).

It is worth noting that whenever $\Psi = \Phi = \emptyset$, we have that the definitions of defeated and acceptable arguments coincide with the ones defined for AAF, that is $\text{Def}(\{S\}, S) = \text{Def}(S)$ and $\text{Acc}(\{S\}, S) = \text{Acc}(S)$. This lead to the following result that states that EAAF semantics extends that of AAF.

Proposition 1. *Let $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ be a well-formed EAAF with $\Psi = \Phi = \emptyset$, and $\Lambda = \langle A, \Omega \rangle$ the AAF corresponding to Δ . Then, if $\text{st}(\Lambda) \neq \emptyset$ then $\text{st}(\Lambda)$ is the only stable-world view of Δ .*

Clearly, as stable semantics is not guaranteed to exist in AAF, the same holds in EAAF. Indeed, as stated next, any well-formed EAAF has at most one stable world view.

Theorem 1. *Any well-formed EAAF admits at most one st -world view.*

For any (well-formed) EAAF Δ and semantics $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$ we use $\sigma(\Delta)$ to denote the σ -world view of Δ , and will often call its elements σ -extensions.

Example 9. Continuing with Example 7, Table 1 reports the σ -world view for each EAAF $\Delta^* \sqsubseteq \Delta$ in Figure 3 and $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$.

Now, consider the EAAF Δ'' (shown in Figure 3), the world view $W = \{S = \{a\}\}$, and the stable semantics. If in Definition 4 we had only focused on the given EAAF Δ'' without looking at its sub-frameworks, as S is a W -stable set and W is maximal (i.e., both conditions *i* and *ii* of Definition 4 are satisfied if focusing on Δ'' only), we would have concluded that c is defeated. However, we had expected that c would have been accepted. Indeed, according to Definition 4, the only stable-world view of Δ'' is $W'' = \{\{a, c\}, \{b, c\}\}$ (cf. Table 1). In fact, considering the sub-framework Δ' (cf. Figure 3) obtained from Δ'' by deleting the epistemic argument c ,

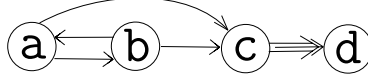


Figure 4: EAAF Δ of Example 10.

the only stable-world view of Δ' is $W' = W''_{\Delta'} = \{\{a\}, \{b\}\}$, which using Definition 4 allows us to discard $W = \{\{a\}\}$ from being a stable-world view of Δ'' . $\square$

According to the proposed EAAF semantics, epistemic arguments are deterministic, that is, they have the same “truth assignment” in a world view, that in turn depends on either the credulous or skeptical acceptance of its attackers.

Proposition 2. *Let $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ be an EAAF, and W the st-world view of Δ . Then, any epistemic argument $x \in A$ is deterministic, that is, one of the following three conditions hold:*

- i) $\forall S \in W. x \in \text{Acc}(W, S);$
- ii) $\forall S \in W. x \in \text{Def}(W, S);$
- iii) $\forall S \in W. x \notin (\text{Acc}(W, S) \cup \text{Def}(W, S)).$

An alternative way to define stable extensions for EAAF could be that of choosing among complete extensions those that are total, as it is done for AAF. More in detail, given an EAAF Δ and its complete-world view $W = \text{co}(\Delta)$, we could have defined the stable-world view for Δ as $\text{st}(\Delta) = \{S \in W \mid S \cup \text{Def}(W, S) = A\}$. This is different from what is done in Definition 4 where to define a st-world view we start with a world view W that is not necessarily $\text{co}(\Delta)$. However, the above-mentioned alternative way to define stable extensions for EAAF may lead to counter-intuitive solutions, as shown in the following example.

Example 10. Consider the EAAF $\Delta = \langle \{a, b, c, d\}, \{(a, b), (b, a), (a, c), (b, c)\}, \{(c, d)\}, \emptyset \rangle$, shown in Figure 4, and the stable semantics. Intuitively, the strong epistemic attack states that d is accepted if c is skeptically rejected. The stable extensions of Δ , that is, the elements in its st-world view are $\{a, d\}$ and $\{b, d\}$. Thus, we obtain that c is skeptically defeated and, consequently, d is accepted.

However, if we start with the complete-world view $\text{co}(\Delta)$, we have that there are three complete extensions $S_1 = \emptyset$, $S_2 = \{a\}$ (with b and c defeated and d undecided) and $S_3 = \{b\}$ (with a and c defeated and d undecided). As there are no total sets in $\text{co}(\Delta)$, we conclude that under the above-mentioned “alternative” stable semantics there is no stable status for d and c , contradicting our intuition. $\square$

As stated next, differently from AAF, stable extensions are not guaranteed to be complete extensions of EAAF. Related to this, even in AAF credulous and skeptical acceptance may give different results under different semantics.

Proposition 3. *There exists an EAAF Δ such that $S \in \text{st}(\Delta)$ and $S \notin \text{co}(\Delta)$.*

Particularly, consider the EAAF $\Delta = \langle \{a, b, c, d, e, f\}, \{(a, b), (b, a), (a, c), (b, c), (c, d)\}, \{(d, e), (e, f)\}, \emptyset \rangle$. With a little effort, it can be checked that $\text{st}(\Delta) = \{S_1 = \{a, d, f\}, S_2 = \{b, d, f\}\}$ and $\text{co}(\Delta) = \{\emptyset, \{a, d\}, \{b, d\}\}$, and thus neither $S_1 \in \text{co}(\Delta)$ nor $S_2 \in \text{co}(\Delta)$.

Finally, stable semantics coincides with preferred semantics in case of odd-cycle free EAFs.

Proposition 4. *Let Δ be a well-formed, odd-cycle free EAAF. Then, it holds that $\text{st}(\Delta) = \text{pr}(\Delta)$.*

4. Complexity

We investigate the complexity of two fundamental reasoning problems for EAAF under stable semantics. In particular, we study the *existence* and *credulous/skeptical* acceptance problems, that are often considered for analyzing the complexity of argumentation frameworks.

We recall the main complexity classes used in this section and, in particular, the definition of the classes P , Σ_h^p , Π_h^p and Δ_h^p , with $h \geq 0$ (see e.g. [27]). For $h > 0$: $\Sigma_0^p = \Pi_0^p = \Delta_0^p = P$; $\Sigma_1^p = NP$ and $\Pi_1^p = coNP$; $\Delta_h^p = P^{\Sigma_{h-1}^p}$; $\Sigma_h^p = NP^{\Sigma_{h-1}^p}$, and $\Pi_h^p = co\Sigma_h^p$. Herein, P^C (resp. NP^C) denotes the class of problems that can be solved in polynomial time using an oracle in the class C by a deterministic (resp. non-deterministic) Turing machine. The class $\Theta_h^p = \Delta_h^p[\log n]$ denotes the subclass of Δ_h^p consisting of the problems that can be solved in polynomial time by a deterministic Turing machine performing $O(\log n)$ calls to an oracle in the class Σ_{h-1}^p . Under the standard complexity-theoretic assumptions, we have that $\Sigma_h^p \subset \Theta_{h+1}^p \subset \Delta_{h+1}^p \subset \Sigma_{h+1}^p \subseteq PSPACE$ and $\Pi_h^p \subset \Theta_{h+1}^p \subset \Delta_{h+1}^p \subset \Pi_{h+1}^p \subseteq PSPACE$. A decision problem is in D_h^p iff it is the conjunction of a decision problem in Σ_h^p and a decision problem in Π_h^p . Hence, D_1^p (or simply DP) denotes the class of the problems that are a conjunction of a problem in NP and one in $coNP$. Under the standard complexity-theoretic assumptions, we have that $NP \subset DP$, $coNP \subset DP$, and $DP \subset \Theta_2^p$.

Given an EAAF $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ and a semantics $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}\}$:

- the *existence* (resp. *non-empty existence*) problem for EAAF, denoted as Ex_σ (resp. $Ex_\sigma^{-\emptyset}$) consists in deciding whether there exists at least one (resp. at least one non-empty) σ -extension S for Δ ;
- the *credulous* (resp. *skeptical*) acceptance problem, denoted as CA_σ (resp. SA_σ), consists in deciding whether a given goal argument $g \in A$ belongs to any (resp. every) σ -extension of Δ .

Observe that if argument g is epistemic, credulous and skeptical acceptance problems coincide (cf. Proposition 2). Therefore, we call this problem *epistemic acceptance* and denote it as EA_σ .

The following fact states that the epistemic acceptance problem captures the credulous and skeptical acceptance problems also for non-epistemic arguments under stable semantics.

Fact 1. *Let $\Delta = \langle A, \Omega, \Psi, \Phi \rangle$ be an EAAF, $g \in A$ any of its non-epistemic arguments. Then:*

- $CA_{\text{st}}(\Delta, g) = EA_{\text{st}}(\Delta', g'')$ with $\Delta' = \langle A \cup \{g', g''\}, \Omega \cup \{(g, g')\}, \Psi \cup \{(g', g'')\}, \Phi \rangle$
- $SA_{\text{st}}(\Delta, g) = EA_{\text{st}}(\Delta', g'')$ with $\Delta' = \langle A \cup \{g', g''\}, \Omega \cup \{(g, g')\}, \Psi, \Phi \cup \{(g', g'')\} \rangle$.

Thus, asking for the credulous and skeptical acceptance of an argument g w.r.t. an EAAF Δ is equivalent to asking for the epistemic acceptance of a fresh epistemic argument g'' w.r.t. an EAAF Δ' , that is obtained from Δ by adding only a pair of attacks.

For this reason and for the fact that epistemic arguments are deterministic (Proposition 2), w.l.o.g. we study the complexity of existence and epistemic acceptance problems in EAAFs (without considering credulous and skeptical acceptance that, as shown above, can be immediately reduced to epistemic acceptance).

Table 2

Complexity of the credulous acceptance (CA_σ), skeptical acceptance (SA_σ), existence (Ex_σ), non-empty existence ($Ex_\sigma^{-\emptyset}$), and determinism problems for AAF, and of the epistemic acceptance (EA_σ), existence (Ex_σ), and non-empty existence ($Ex_\sigma^{-\emptyset}$) problems for EAAF. For any complexity class C , C -c (resp. C -h) means C -complete (resp. C -hard); an interval C -h, C' means C -hard and in C' . The results for $\sigma \in \{\text{gr}, \text{co}, \text{pr}\}$ have been presented in [1], while those for st are new.

σ	AAF					EAAF		
	CA_σ	SA_σ	Ex_σ	$Ex_\sigma^{-\emptyset}$	DS_σ	EA_σ	Ex_σ	$Ex_\sigma^{-\emptyset}$
gr	P	P	trivial	P	trivial	P	trivial	P
co	NP-c	P	trivial	NP-c	coNP-c	Θ_2^P -h, Δ_2^P	trivial	NP-c
st	NP-c	coNP-c	NP-c	NP-c	DP-c	DP-h	NP-h	NP-h
pr	NP-c	Π_2^P -c	trivial	NP-c	Π_2^P -c	Π_2^P -h, Δ_3^P	trivial	NP-c

The next theorem states the complexity of epistemic acceptance under stable semantics.

Theorem 2. EA_{st} is DP-hard.

The following corollary states that for EAAF the existence of at least one extension is not always guaranteed, as for the case of AAF.

Corollary 1. Ex_{st} coincides with $Ex_{\text{st}}^{-\emptyset}$ and it is NP-hard.

The results of this section, along with some related complexity results for AAF, are summarized in Table 2. We have also reported the results for $\sigma \in \{\text{gr}, \text{co}, \text{pr}\}$ which are from [1]; those for st are new. We found that the complexity generally increases w.r.t. that of AAF for the acceptance problems under stable semantics. This particularly holds if we compare the complexity of EA_{st} for EAAF with that of CA_{st} and SA_{st} for AAF. Finally, deciding acceptance (resp. existence) in EAAF under stable semantics is at least hard as checking determinism (resp. existence) in AAF. For future work we plan to close the complexity gap related to the complexity of acceptance problems in EAAF under the different semantics.

5. Conclusion and Future Work

Several proposals have been made to extend Dung's framework with the aim of better modeling the knowledge to be represented. The extensions include Bipolar AAF [28, 29], AAF with recursive attacks and supports [30, 31, 32], Dialectical framework [33], Abstract Reasoning Framework [34], AAF with preferences [35, 36] and constraints [22, 23], as well extensions for representing uncertain information, e.g. incomplete AAF [37] and probabilistic AAF [38, 39, 40, 41, 42, 43, 44].

We have presented the stable semantics for Epistemic Abstract Argumentation Framework, a generalization of Dung's framework where epistemic attacks and arguments can be expressed. We also provided complexity bounds for the existence and acceptance problems in EAAF under the well-known stable argumentation semantics. Our complexity analysis shows that the

epistemic elements (i.e., epistemic attacks/arguments) impact on the complexity of some of the problems considered. In general, it turns out that EAAF is more expressive than AAF.

The idea of extending logic with epistemic constructs has been investigated also in the field of Answer Set Programming (ASP) [45, 46, 47]. Epistemic logic programs, firstly proposed in [45], extend disjunctive logic programs under the stable model semantics with modal constructs called subjective literals [46, 48, 49, 47]. The introduction of this extension was originally motivated to correctly represent incomplete information in programs that have several stable models. Using subjective literals, it is possible to check whether a literal is true in every or some stable model of the program. These models in this context are also called belief sets, being collected in a set called world view. The main idea was to expand the syntax and semantics of Answer Set Programming by modal operators $\mathbf{K}$ and $\mathbf{M}$ where $\mathbf{K}\varphi$ holds if φ is true in all answer sets of a program and $\mathbf{M}\varphi$ holds if φ is true in at least one answer set. Using this notation, $\text{not } \mathbf{K}p \wedge \text{not } \mathbf{K}\neg p$ would correspond to “the truth value of p is unknown” even in the presence of multiple answer sets. In such a context, several problems are still open and they regard the support required by stable models, as well as splitting properties that are satisfied by classical ASP semantics, but not satisfied by epistemic ASP-based semantics [50, 49, 51].

Although our focus is on argumentation, we believe that our results could be of interest to the logic community. In fact, by exploiting the correspondence between AF and Logic Programming [52], the proposed EAAF semantics could be seen as an alternative semantics for a special class of epistemic logic programs whose complexity and computation can be characterized by using our results.

Future work will be devoted to considering other argumentation semantics such as the semi-stable semantics. Another interesting direction for future work is exploring EAF in a dynamic setting [53, 54, 55, 56, 57, 58, 59], where objective evidence (underlying AF) and subjective beliefs (epistemic formulae) may change over time.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU. This work was also funded by the Next Generation EU - Italian NRRP, Mission 4, Component 2, Investment 1.5, call for the creation and strengthening of ‘Innovation Ecosystems’, building ‘Territorial R&D Leaders’ (Directorial Decree n. 2021/3277) - project Tech4You - Technologies for climate change adaptation and quality of life improvement, n. ECS0000009. This work reflects only the authors’ views and opinions, neither the Ministry for University and Research nor the European Commission can be considered responsible for them.

References

- [1] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Epistemic abstract argumentation framework: Formal foundations, computation and complexity, in: Proc. of AAMAS, 2023, pp. 409–417.
- [2] T. Bench-Capon, P. E. Dunne, Argumentation in artificial intelligence, Artif. Intell. 171 (2007) 619 – 641.

- [3] G. R. Simari, I. Rahwan (Eds.), *Argumentation in Artificial Intelligence*, Springer, 2009.
- [4] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, S. Villata, Towards artificial argumentation, *Artificial Intelligence Magazine* 38 (2017) 25–36.
- [5] Y. Shoham, K. Leyton-Brown, *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2009.
- [6] L. Amgoud, Y. Dimopoulos, P. Moraitis, A unified and general framework for argumentation-based negotiation, in: *Proc. of AAMAS*, 2007, p. 158.
- [7] Y. Dimopoulos, J. Mailly, P. Moraitis, Argumentation-based negotiation with incomplete opponent profiles, in: *Proc. AAMAS*, 2019, pp. 1252–1260.
- [8] H. Prakken, Models of persuasion dialogue, in: *Argumentation in Artificial Intelligence*, 2009, pp. 281–300.
- [9] L. Amgoud, M. Serrurier, Agents that argue and explain classifications, *Auton. Agents Multi Agent Syst.* 16 (2008) 187–209.
- [10] Á. Carrera, C. A. Iglesias, A systematic review of argumentation techniques for multi-agent systems research, *Artif. Intell. Rev.* 44 (2015) 509–535.
- [11] S. Ontañón, E. Plaza, An argumentation-based framework for deliberation in multi-agent systems, in: *Int. Workshop, ArgMAS*, volume 4946, 2007, pp. 178–196.
- [12] S. Parsons, C. Sierra, N. R. Jennings, Agents that reason and negotiate by arguing, *J. Log. Comput.* 8 (1998) 261–292.
- [13] X. Fan, F. Toni, Agent strategies for aba-based information-seeking and inquiry dialogues, in: *Proc. of ECAI*, volume 242, 2012, pp. 324–329.
- [14] P. McBurney, S. Parsons, Dialogue games for agent argumentation, in: *Argumentation in Artificial Intelligence*, Springer, 2009, pp. 261–280.
- [15] H. Prakken, G. Sartor, Modelling reasoning with precedents in a formal dialogue game, *Artif. Intell. Law* 6 (1998) 231–287.
- [16] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.* 77 (1995) 321–358.
- [17] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Incomplete argumentation frameworks: Properties and complexity, in: *Proc. of AAI*, 2022, pp. 5451–5460.
- [18] P. Baroni, M. Giacomin, G. Guida, Scc-recursiveness: a general schema for argumentation semantics, *Artificial Intelligence* 168 (2005) 162–210.
- [19] P. Baroni, M. Giacomin, On principle-based evaluation of extension-based argumentation semantics, *Artif. Intell.* 171 (2007) 675–700.
- [20] W. Dvorák, P. E. Dunne, Computational problems in formal argumentation and their complexity, *FLAP* 4 (2017).
- [21] C. Sakama, T. C. Son, Epistemic argumentation framework: Theory and computation, *J. Artif. Intell. Res.* 69 (2020) 1103–1126.
- [22] S. Coste-Marquis, C. Devred, P. Marquis, Constrained argumentation frameworks, in: *Proc. of (KR)*, 2006, pp. 112–122.
- [23] O. Arieli, Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks, *J. Appl. Log.* 13 (2015) 582–604.
- [24] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Abstract argumentation framework with conditional preferences, in: *Proc. of AAI*, 2023, pp. 6218–6227.

- [25] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Preferences and constraints in abstract argumentation, in: *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [26] M. Bernreiter, W. Dvorák, S. Woltran, Abstract argumentation with conditional preferences, in: *Proc. of COMMA*, 2022, pp. 92–103.
- [27] C. H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.
- [28] F. Nouioua, Afs with necessities: Further semantics and labelling characterization, in: *Proc. of SUM*, 2013, pp. 120–133.
- [29] S. Villata, G. Boella, D. M. Gabbay, L. W. N. van der Torre, Modelling defeasible and prioritized support in bipolar argumentation, *Ann. Math. Artif. Intell.* 66 (2012).
- [30] A. Cohen, S. Gottifredi, A. J. Garcia, G. R. Simari, An approach to abstract argumentation with recursive attack and support, *J. Appl. Log.* 13 (2015) 509–533.
- [31] S. Gottifredi, A. Cohen, A. J. Garcia, G. R. Simari, Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations, *Artif. Intell.* 262 (2018) 336–368.
- [32] C. Cayrol, J. Fandinno, L. F. del Cerro, M. Lagasquie-Schiex, Structure-based semantics of argumentation frameworks with higher-order attacks and supports, in: *Proc. of COMMA*, 2018, pp. 29–36.
- [33] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, S. Woltran, Abstract dialectical frameworks revisited, in: *Proc. of IJCAI*, 2013, pp. 803–809.
- [34] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On acceptance conditions in abstract argumentation frameworks, *Information Sciences* 625 (2023) 757–779.
- [35] L. Amgoud, C. Cayrol, Inferring from inconsistency in preference-based argumentation frameworks, *J. Autom. Reason.* 29 (2002) 125–169.
- [36] S. Modgil, H. Prakken, A general account of argumentation with preferences, *Artif. Intell.* 195 (2013) 361–397.
- [37] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, J. Rothe, Acceptance in incomplete argumentation frameworks, *Artif. Intell.* (2021) 103470.
- [38] P. M. Dung, P. M. Thang, Towards (probabilistic) argumentation for jury-based dispute resolution, in: *Proc. of COMMA*, 2010, pp. 171–182.
- [39] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: *Proc. of TAFA*, 2011, pp. 1–16.
- [40] A. Hunter, Some foundations for probabilistic abstract argumentation, in: *Proc. of COMMA*, 2012, pp. 117–128.
- [41] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic abstract argumentation: Complexity and approximation, in: *Proc. of KR*, 2020, pp. 33–43.
- [42] B. Fazzinga, S. Flesca, F. Furfaro, Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence (extended abstract), in: *Proc. of IJCAI*, 2019, pp. 6362–6366.
- [43] B. Fazzinga, S. Flesca, F. Furfaro, Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence, *Artif. Intell.* 268 (2019) 1–29.
- [44] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic and incomplete abstract argumentation frameworks, *Artificial Intelligence* (2023)

103967.

- [45] M. Gelfond, Strong introspection, in: T. L. Dean, K. R. McKeown (Eds.), Proc AAAI Conf., 1991, pp. 386–391.
- [46] M. Gelfond, New semantics for epistemic specifications, in: Proc. of LPNMR Conf., volume 6645, 2011, pp. 260–265.
- [47] J. Fandinno, W. Faber, M. Gelfond, Thirty years of epistemic specifications, Theory Pract. Log. Program. 22 (2022) 1043–1083.
- [48] P. Cabalar, J. Fandinno, L. F. del Cerro, Autoepistemic answer set programming, Artif. Intell. 289 (2020) 103382.
- [49] P. Cabalar, J. Fandinno, L. F. del Cerro, Splitting epistemic logic programs, Theory Pract. Log. Program. 21 (2021) 296–316.
- [50] Y. Shen, T. Eiter, Considering constraint monotonicity and foundedness in answer set programming, in: L. D. Raedt (Ed.), Proc. of IJCAI, 2022, pp. 2741–2747.
- [51] A. Herzig, A. Yuste-Ginel, On the epistemic logic of incomplete argumentation frameworks, in: Proc. of KR, 2021, pp. 681–685.
- [52] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On the semantics of abstract argumentation frameworks: A logic programming approach, Theory Pract. Log. Program. 20 (2020) 703–718.
- [53] G. Alfano, S. Greco, F. Parisi, Computing stable and preferred extensions of dynamic bipolar argumentation frameworks, in: Proc. of the 1st Workshop on Advances In Argumentation In Artificial Intelligence AI³, 2017, pp. 28–42.
- [54] G. Alfano, S. Greco, F. Parisi, G. I. Simari, G. R. Simari, Incremental computation for structured argumentation over dynamic delp knowledge bases, Artif. Intell. 300 (2021) 103553.
- [55] G. Alfano, S. Greco, F. Parisi, Computing extensions of dynamic abstract argumentation frameworks with second-order attacks, in: Proc. of the 22nd International Database Engineering & Applications Symposium (IDEAS), 2018, pp. 183–192.
- [56] G. Alfano, S. Greco, F. Parisi, On scaling the enumeration of the preferred extensions of abstract argumentation frameworks, in: Proceedings of ACM/SIGAPP Symposium on Applied Computing (SAC), 2019, pp. 1147–1153.
- [57] G. Alfano, S. Greco, F. Parisi, Incremental computation in dynamic argumentation frameworks, IEEE Intell. Syst. 36 (2021) 80–86.
- [58] A. Niskanen, M. Järvisalo, μ -toksia: An efficient abstract argumentation reasoner, in: Proc. of KR, 2020, pp. 800–804.
- [59] G. Alfano, S. Greco, Incremental skeptical preferred acceptance in dynamic argumentation frameworks, IEEE Intell. Syst. 36 (2021) 6–12.

An Initial Exploration of How Argumentative Information Impacts Automatic Generation of Counter-Narratives Against Hate Speech

Damián Ariel Furman^{1,2}, Pablo Torres³, José A. Rodríguez³, Diego Letzen³,
Vanina Martínez⁴ and Laura Alonso Alemany³

¹University of Buenos Aires (UBA)), Intendente Güiraldes 2160 - Ciudad Universitaria, Buenos Aires, Argentina

²CONICET, Godoy Cruz 2290, Buenos Aires, Argentina

³Universidad Nacional de Córdoba, Argentina

⁴Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain

Abstract

Fighting hate speech through automatic counter-narrative generation is gaining interest because of the increasing capabilities of Large Language Models. However, counter-narrative generation is a challenging task that can benefit from insightful analyses of text. In this work, we present an approach to improve the generation of counter-narratives by providing Large Language Models with high-quality examples. In addition, we show that enhancing the original hate speech with an argumentative analysis, identifying justifications and conclusions, together with collectives and the properties associated to them, seems to produce some improvements, specially with smaller training datasets, helping to orient the generation towards a particular response strategy. The dataset of counter-narratives with argumentative information is made publicly available.

Warning: This work contains offensive and hateful text that may be distressing. It does not represent the views of the authors.

Keywords

Counter-narrative generation, Hate speech, Argument mining, Large Language Models

1. Introduction

In social media platforms, hate speech is amplified beyond human scale, spreading faster and increasing their reach, with negative impacts in societies, like polarization or an increase in violent episodes against targeted communities or individuals. It is because of these known consequences that many legal systems typify it as a crime, at least in some of its forms.

The predominant strategy adopted so far to counter hate speech in social media is to recognize, block and delete these messages and/or the users that generated it. This strategy has two main disadvantages. The first one is that blocking and deleting may prevent a hate message from spreading, but does not counter its consequences on those who were already reached by it.

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ damian.a.furman@gmail.com (D. A. Furman)

🌐 <https://damifur.github.io/> (D. A. Furman)

🆔 0000-0002-0877-7063 (D. A. Furman)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

📄 CEUR Workshop Proceedings (CEUR-WS.org)

The second one is that there is no place for subtleties or shades while defining hate speech: it must be done as a binary classification because the consequence of that classification is binary. This can generate accusations of overblocking or censorship, and not just because of errors in automated systems, which have been shown to be highly biased [1], but because blocking seems to be an overly simplistic approach to deal with the inherent complexity of hate speech.

An alternative to blocking that has been gaining attention in the last years, is to "*oppose hate content with counter-narratives (i.e. informed textual responses)*" [2, 3]¹. This way, the consequences of errors in the hate classification are minimized, overblocking is avoided, and it helps to spread a message against hate that can reach people that are not necessarily convinced, or even not involved in the conversation.

However, the huge volume of online hate messages makes the manual generation of counter-narratives an impossible task. In this scenario, automating the generation of counter-narratives is an appealing avenue, but the task poses a great challenge due to the complex linguistic and communicative patterns involved in argumentation.

Traditional machine learning approaches have typically produced less than satisfactory results for argumentation mining and generation. However, the recent availability of Large Language Models (LLMs) provides a promising approach to address the task of counter-narrative generation. Indeed, LLMs seem capable of generating satisfactory text for many tasks. Thorburn and Kruger [4] showed that a version of ChatGPT can tackle 6 argumentative reasoning tasks with some degree of success. They also find that finetuning the LLM parameters outperforms prompt-only based approaches.

However, as Hinton and Wagemans [5] show in their in-depth analysis of the argumentative capabilities of GPT-3, the argumentative text generated by LLMs tends to show some weaknesses. Although the language they use is clearly argumentative, as is the structure of arguments they create, most of them are not considered acceptable by humans, falling in fallacies like 'begging the question' and providing mostly irrelevant information.

In this paper we present an initial exploration of the impact of argumentative information in improving the quality of arguments generated by LLMs, more concretely, in improving the quality of automatically generated counter-narratives against hate speech. We compare different scenarios: LLMs without any specific adaptation to the task or domain, with fine-tuning using a dataset of counter-narratives, in a few-shot approach, and providing additional information about some of the argumentative aspects of the hate speech.

To assess the quality of the counter-narratives generated in the different scenarios, we carry out a preliminary evaluation with human judges, who achieved moderate agreement between each other. Based on those judgements, we can say that argumentative information by itself does not produce an improvement in the counter-narratives, but high-quality, specifically targeted fine-tuning seems to have a positive impact. Argumentative information does produce improvements in scenarios with very small training data and very specific fine-tuning, which seems promising to produce highly tailored counter-narratives, as in Gupta et al. [6].

The rest of the paper is organized as follows. In the next section, we review relevant work related to automated counter-narrative generation and argumentative analysis of hate speech. Then in Section 3 we describe our dataset of counter-narratives, with which we carry out the

¹No Hate Speech Movement Campaign: <http://www.nohatespeechmovement.org/>

comparison of scenarios described in Section 4, where we also describe extensively our approach to the evaluation of generated counter-narratives, based on human judgements, and the prompts used to obtain the counter-narratives. Results analyzed in Section 5 show how fine-tuned LLMs and argumentative information provide better results, which we illustrate with some examples.

2. Related work

Automated counter-narrative generation has been recently tackled by leveraging the rapid advances in neural natural language generation. As with most natural language generation tasks in recent years, the basic machine learning approach has been to train or fine-tune a generative neural network with examples specific to the target task.

The CONAN dataset [3] is, to our knowledge, the first dataset with counter-narratives. It has 4078 Hate Speech – Counter Narrative original pairs manually written by NGO operators, translated to three languages: English, French and Italian. Data was augmented using automatic paraphrasing and translations between languages to obtain 15024 final pairs of hate speech – counter-narrative. Unfortunately, this dataset is not representative of the language in social media.

Similar approaches were carried out by Qian et al. [7] and Ziems et al. [8]. Qian et al. [7]’s dataset consists of reddit and Gab conversations where Mechanical Turkers identified hate speech and wrote responses. Ziems et al. [8] did not produce new text, but labeled COVID-19 related tweets as hate, counter-speech or neutral based on their hatefulness towards Asians.

In follow-up work to the seminal CONAN work, Tekiroğlu et al. [9] applied LLMs to assist experts in creating the corpus, with GPT-2 generating a set of counter-narratives for a given hate speech and experts editing and filtering them. Fanton et al. [10] iteratively refined a LLM where the automatically generated counter-narratives were filtered and post-edited by experts and then fed them to the LLM as further training examples to fine-tune it, in a number of iterations. Bonaldi et al. [11] apply this same approach to obtain a machine-generated dataset of dialogues between people producing hate speech and experts in hate countering. As a further enhancement in the LLM-based methodology, Chung et al. [12] enhanced the LLM assistance with a knowledge-based retrieval architecture to enrich counter-narrative generation.

Ashida and Komachi [13] use LLMs for generation with a *prompting* approach, instead of fine-tuning them with manually created or curated examples. They also propose a methodology to evaluate the generated output, based on human evaluation of some samples. This same approach is applied by Vallecillo-Rodríguez et al. [14] to create a dataset of counter-narratives for Spanish. Both these approaches are targeted to user-generated text, closely related to social media.

However, none of the aforementioned datasets or approaches to counter-narrative generation includes or integrates any additional annotated information apart from the hate message, possibly its context, and its response. That is why we consider an alternative approach that aims to reach generalization not by the sheer number of examples, but by providing a richer analysis of such examples that guides the model in finding adequate generalizations. We believe that information about the argumentative structure of hate speech, may be used as constraints for automatic counter-narrative generation.

Chung et al. [15] address an argumentative aspect of hate speech countering. They classify counter-narratives by type, using a LLM, and showing that knowledge about the type of counter-narratives can be successfully transferred across languages, but they do not use this information to generate counter-narratives.

To our knowledge, ours is the only corpus where tweets of hate speech have been annotated with argumentative information: ASOHMO [16], based on the Hateval corpus [17]. This dataset enriches the argumentative tweets of Hateval [17] with a manual analysis of their argumentative aspects, following an adaptation of the proposal of Wagemans [18], an analytic approach to represent the semantics of the core schemes proposed by Walton et al. [19], with fewer categories based on a limited set of general argument features. The following argumentative aspects are manually identified in tweets:

- **Justifications and Conclusions.**
- **Type** of Justification and Conclusion: Fact, Policy or Value.
- A **Pivot** signalling the argumentative relation between Justification and Premise.
- Two domain-specific components: the **Collective** which is the target of hate, and the **Property** that is assigned to such Collective.

In this work, we present counter-narratives manually associated to the hate tweets in ASOHMO and present an initial exploration of the impact of different kinds of information (counter-narratives, counter-narratives by subtype and information about argumentative components) in improving automatic generation of counter-narratives.

3. Creating counter-narratives associated to argumentative aspects of hate speech

Here we present CONEAS (Counter-Narratives Exploiting Argumentative Structure), a dataset of counter-narratives defined according to the argumentative information labeled on tweets from ASOHMO [16]. Each argumentative tweet is paired with counter-narratives of three different types defined by applying systematic transformations over argumentative components of the tweet, and a fourth type consisting of any counter-narrative that does not fall under any of the other three.

All counter-narratives, regardless of their type, also follow the guidelines of the Get The Trolls Out project²: *don't be aggressive or abusive, don't spread hate yourself, try to de-escalate the conversation, respond thinking on a wider audience than the person posting the original tweet and try to build a narrative*. Annotators were suggested to try to write at least one counter-narrative of each type but only if they came naturally, otherwise they could leave it blank.

The instructions to generate each type of counter-narrative are as follows:

Negate Relation Between Justification And Conclusion (Type A) Negate the implied relation between the justification and the conclusion.

²<https://getthetrollsout.org/stoppinghate>

HATE TWEET: user must deport all illegal migrants india already reeling under constant threat of muslim radicals curb population Justification: india already reeling under constant threat of muslim radicals curb population (fact) Conclusion: must deport all illegal migrants (policy) Collective: illegal migrants Property: muslim radicals COUNTER NARRATIVE A (Negate relation between justification and conclusion) <i>Deporting illegal migrants will not mitigate the problems with muslim radicals.</i> COUNTER NARRATIVE B (Negate relation between collective and property) <i>Illegal migrants are not necessarily muslim radicals.</i> COUNTER NARRATIVE C (Negate justification based on type) <i>It is not true that India is reeling under threat of muslim radicals.</i> FREE COUNTER NARRATIVE (Free) <i>Deporting illegal migrants without consideration to their circumstances is an inhumane move.</i>
--

Figure 1: Examples of each type of counter narratives.

Negate association between Collective and Property (type B) Attack the relation between the property, action or consequence that is being assigned to the targeted group and the targeted group itself.

Attack Justification based on it is type (Type C) If the justification is a fact, then the fact must be put into question or sources must be asked to prove that fact. If it is of type “value”, it must be highlighted that the premise is actually an opinion, possibly relativizing it as a xenophobic opinion. If it is a “policy”, a counter policy must be provided.

Free Counter-Narrative (type D) All counter-narratives that the annotator comes up with and do not fall within any of the other three types.

An example of each type of counter-narrative can be seen in Figure 1. Our dataset³ consists of a total of 1722 counter-narratives for 725 argumentative tweets in English and 355 counter-narratives for 144 tweets in Spanish (an average of 2.38 and 2.47 per tweet respectively). Table 1 shows the percentage of tweets that has a counter-narrative of each type.

4. Experiments

We designed a series of experiments to assess the impact of high-quality examples and argumentative information in the automatic generation of counter-narratives via prompting LLMs. We want to explore the following approaches:

Fine-tuned vs Few-shot Use a LLM that has been trained for general purposes to generate counter-narratives by prompting the LLM with some examples of the desired input-output,

³<https://github.com/ConeasDataset/CONEAS/>

as shown in the left column of Figure 3, or take a general LLM and fine-tune it with the examples of hate tweets associated to manually generated counter-narratives.

With or without argumentative information We want to assess the impact of different combinations of argumentative information provided within the input of the model: Collective and Property; Justification, Conclusion and Pivot; and all types.

With specific kinds of counter-narratives We pretrained two models for each type of counter-narrative using only that type: one without extra information and another adding argumentative information relevant for the correspondent type (Justification and Conclusion for type A, Collective and Property for type B and Justification for type C).

Small or Big size of the same kind of LLM We want to compare performance of a larger model with higher hardware requirements against a smaller one, fine-tuned, cheaper to run but requiring a specific annotated dataset. After testing behavior of similar alternatives (Bloom, GPT-J and GPT2), we chose Flan-T5 [12], an open model with base (250M parameters) and XL (3B parameters) versions that is instruction-fine-tuned.

Few-shot experiments were conducted for Flan-T5 Base (small) and XL (larger) models. fine-tuning was only conducted on Flan-T5 Base due to computational resource constraints.

We conducted some manual evaluation of prospective to find optimal parameters for generation, and we found that using Beam Search with 5 beams yielded the best results, so this is the configuration we used throughout the paper.

4.1. Fine-tuning of the LLM with counter-narratives

To fine-tune FLAN-T5 with our dataset of counter-narratives, we randomly split our dataset in training, development and test partitions, assuring that all counter-narratives for the same hate tweet are contained into the same partition. Details can be seen on Table 1.

	English						Spanish					
	#Tweets	#CNs	% corpus	A	B	C	#Tweets	#CNs	% corpus	A	B	C
Train	509	1201	69.8%	496	238	467	105	257	72.4%	101	59	97
Dev	71	173	10.0%	67	38	68	12	27	7.6%	12	8	7
Test	145	348	20.2%	138	74	136	27	71	20%	27	21	23
Proportion of tweets with counter-narrative												
				96%	47%	90%				97%	61%	89%

Table 1

Size of dataset partitions of English and Spanish datasets. Columns A, B and C show the amount of counter-narratives used for each partition when training only with counter-narratives of a given type.

All models were trained starting from Flan-T5-Base, in a multilingual setting using mixed English and Spanish examples, with a learning rate of 2e-05 for 8 epochs.

4.2. Experiments based on few-shot

For the few-shot experiments, the prompt has an instruction followed by two random examples taken from the test partition of the dataset. For each example, the hate tweet and its

corresponding counter-narrative are enclosed in special tokens defining the start and end.

4.3. Evaluation method for generated counter-narratives

Evaluation of counter-narratives is not straightforward. So far, no automatic technique has been found satisfactory for this specific purpose. Automatic metrics proposed for other NLP tasks, like BLEU [20] for automatic translation or ROUGE [21] for summarization, are not adequate for this task because they rely strongly on word or n-gram overlap with manually generated examples. These measures are disputed in the NLP community because, among other factors, they can't be adapted to cases where there can be many possible good outputs of the model, with significant differences between themselves, such as our case. We discarded these measures after comparing different counter-narratives of a same tweet from our dataset and noting that many of them scored 0 on both.

Faced with the lack of appropriate automatic metrics adequate for the task, many authors have conducted manual evaluations for automatically generated counter-narratives. Manual evaluations typically distinguish different aspects of the adequacy of a given text as a counter-narrative for another. Chung et al. [12] evaluate three aspect of the adequacy of counter-narratives: *Suitableness* (if the counter-narrative was suited as a response to the original hate message), *Informativeness* (how specific or generic the response is) and *Intra-coherence* (internal coherence of the counter-narrative regardless of the message it is responding to). Ashida and Komachi [13], on the other hand, assess these three other aspects: *Offensiveness*, *Stance* (towards the original tweet) and *Informativeness* (same as Chung et al. [12]).

Based on these previous works, we have put together a first version of criteria to manually evaluate⁴ the adequacy of counter-narratives, considering four different aspects:

- **Offensiveness:** if the tweet is offensive to either the target group, the author of the tweet or any other group or person. Possible values are: Offensive; Possibly Offensive/Not clear; Not offensive.
- **Stance:** if the tweet supports or counters the specific message of the hate tweet. Possible values are: Supports the original message; Not clear/Changes subject wrt original tweet; Counters the original message. Stance incorporates a certain notion of suitableness, since it assigns value "Changes the subject" if the counter-narrative is not responding specifically to the standpoint of the original tweet.
- **Informativeness:** Evaluates the complexity and specificity of the generated text. Only counter-narratives with a "Counters" Stance are evaluated. Possible values are:
 1. **Generic statement:** replies that don't incorporate any information mentioned on the tweet and could counter many different hate messages (e.g "I don't think so" or "That is not true").
 2. **Specific but not argumentative:** the reply is a simple statement, possibly composed of a single sentence without providing justification for the stance but referring to some specific aspect of the original tweet. Usually they comply with a formula composed of a prefix (like "I don't think that" or "Do you have proof that") and a verbatim copy of some part of the hate tweet.

⁴Results of the evaluation can be found on <https://shorturl.at/aetFZ>

3. **Specific and Argumentative:** counter-narratives with some degree of elaboration of the information contained on the hate message. We identified three common patterns that we associate with this value:

- A - replies that take more than one element from the original message and establish some relation between them (e.g. "I don't see the relation between {*element from the original message*} and {*other element from the original message*}").
 - B - A simple statement declaring stance over a single element from the original tweet but adding a second coordinated statement with personal appreciations about it (e.g. "I don't think we should {*some policy mentioned on the tweet*}. It is a bad idea").
 - C - An argumentative reply based on information not mentioned explicitly on the original tweet, but necessarily inferred, showing a comprehensive understanding of the meaning of the hate message (e.g. a reply to a tweet concluding with #BuildTheWall saying "*Building a wall would cost the taxpayers more*" or "*Building a wall won't give you more control over illegal trafficking*").
- **Felicity:** This category is related to Chung et al. [12]'s Intra-Coherence, but also considering additional dimensions like syntactical and semantic correctness. It evaluates independently of the original tweet, if the generated text sounds, by itself, fluent and correct. There are three possible values: The text is incoherent or semantic or syntactically incorrect; The text is coherent with small errors like incoordination of genre/tense/etc. or repeating parts of the original text without adapting them to the text being generated; The text is fluent and sounds correct.

Aggregating the results for these four categories, we define two extra concepts: Good and Excellent counter-narratives. Good counter-narratives will be those with optimal values on Offensiveness, Stance and Felicity. Excellent counter-narratives will be those that also have the optimal value for Informativeness. We believe Informativeness is the most valuable of the four categories, that is why it is determinant in characterizing Excellent counter-narratives. The Good indicator shows that productions are not harmful or totally random.

We are planning to improve the kind of information that is currently captured in the Informativeness category in a second version of the evaluation criteria.

4.4. Annotation environment and agreement

To properly evaluate the quality of the generated counter-narratives with the presented method, we conducted a preliminary manual evaluation. We evaluated three random subsets of 20 hate tweets in English and 10 in Spanish. One contains only tweets associated with counter-narratives of both types A and C on our dataset, and was used to evaluate models fine-tuned only with these kinds of counter-narratives. Another contains only tweets associated with counter-narratives of type B and was also used to evaluate models fine-tuned only with this type of counter-narratives. The last subset contains tweets with counter-narrative pairs of all types, and was used for all the rest of the experiments.

We generated one counter-narrative for each tweet in the corresponding evaluation subset for each combination of features to be assessed: few-shot, fine-tuned, with different kinds of

	1 vs 2	2 vs 3	1 vs 3
Offensiveness	0.47	0.40	0.41
Stance	0.63	0.58	0.63
Informativeness	0.49	0.42	0.54
Felicity	0.67	0.37	0.36

Table 2

Agreement scores between annotators 1, 2 and 3 using Cohen’s Kappa.

argumentative information, with different sizes of LLM. For the larger version of FLAN-T5 we only applied the few-shot approach, and, after assessing no improvement on the smaller version, we aborted the rest of experiments with this version of the LLM to reduce the carbon footprint of our experiments. The results for the 18 experiments can be seen in Table 3.

Then, three annotators labeled each tweet according to the four categories described above. The final value for each category was obtained by calculating the value with more votes (at least two annotators agreed on the value). In total, each annotator labeled 540 hate tweet/counter-narrative pairs. Of all these, there were 10 cases where each of the three annotators labeled a different value. In these cases, we adopted a conservative criterion and assigned the worst of the three possible values.

Table 2 shows the agreement scores between the three annotators, calculated using Cohen’s Kappa [22]. In most cases, agreement ranges from Moderate ($0.41 < \kappa < 0.60$) to Substantial ($0.61 < \kappa < 0.80$), except for the agreement achieved by annotator 3 against the other two on the category of Felicity which is just Fair ($0.21 < \kappa < 0.40$)⁵. As can be expected for such an interpretative task, agreement between annotators can be improved. However, this initial assessment served as a starting approach to assess the impact of different factors in the quality of generated counter-arguments.

We are currently working on a second version of the evaluation criteria, with more insightful categories, expanding on Informativeness and trying to capture argument acceptability, relevance and persuasiveness. We will check whether this improved criteria improve inter-annotator agreement. If so, we will engage a higher number of judges and aim to obtain a more reliable assessment of the quality of automatically generated counter-narratives.

5. Analysis of results

Results of the manual evaluation of different strategies for counter-narrative generation for English can be seen in Table 3. A summary of this table can be seen in Figure 2, which displays the aggregated proportion of Good and Excellent counter-narratives for each strategy.

We can clearly see that the larger versions of the model (XL) produce counter-narratives that are less satisfactory in general, and that argumentative information only decreases the quality of the generated text. Fine-tuned models produce better counter-narratives in general, even if smaller. A very valuable conclusion that can be obtained from these results is that a small number of high quality examples produce a much bigger improvement in performance than

⁵The interpretation of the ranges of values of the kappa coefficient is according to Landis and Koch [23].

Approaches	Offensiveness		Stance		Informative		Felicity	
	Off	NotOff	Supp	Count	Gen	Arg	Infel	Felic
Few-shot Approaches								
Base	10%	60%	15%	40%	40%	0%	15%	70%
Base All	40%	35%	45%	25%	10%	5%	10%	45%
Base Collective	5%	50%	15%	20%	15%	5%	5%	90%
Base Premises	30%	35%	30%	35%	20%	5%	5%	70%
XL	60%	25%	60%	25%	10%	0%	10%	45%
XL All	80%	10%	80%	10%	0%	10%	5%	15%
XL Collective	55%	25%	55%	15%	10%	5%	20%	0%
XL Premises	55%	10%	60%	0%	0%	0%	30%	25%
Fine-tuned Approaches								
Base	10%	65%	10%	65%	25%	35%	15%	80%
Base All	15%	45%	15%	30%	0%	10%	15%	80%
Base Collective	0%	55%	0%	60%	0%	35%	5%	85%
Base Premises	10%	40%	10%	45%	0%	30%	10%	80%
Base CNs A	10%	45%	10%	35%	5%	25%	35%	60%
Base CNs A Premises	10%	60%	10%	45%	0%	40%	25%	65%
Base CNs B	30%	20%	25%	5%	5%	0%	80%	10%
Base CNs B Collective	0%	15%	0%	15%	5%	10%	85%	15%
Base CNs C	0%	50%	5%	25%	20%	5%	65%	25%
Base CNs C Justification	10%	30%	10%	35%	5%	20%	25%	55%

Table 3

Manual evaluation of automatically generated counter-narratives for English hate tweets, using different sizes of the model (Base and XL), two learning techniques (few-shot and fine-tuning), two different training settings (all counter-narratives or only one kind: A, B or C) and different combinations of argumentative information (no information, Collective and Property, Premises and pivot and all the information available). We report the percentage of counter-narratives for the two extreme values of our four analysis categories: Offensiveness, Stance, Informativeness and Felicity.

using larger models, which are also more taxing.

If we focus on Informativeness (third dimension of evaluation in Table 3, we can see that the approaches that produce most informative counter-narratives are fine-tuned (lower half of the Table), without a detriment in any of the other dimensions of evaluation. Interestingly, when fine-tuned only with counter-narratives of a single type, providing argumentative information consistently improves the informativeness of the counter-narratives, even if only slightly. We have to take into account that such approaches use a much smaller number of counter-narratives, as can be seen in Table 1. Even in the case of type B counter-narratives, with extremely few examples to fine-tune, argumentative information produces an improvement in informativeness.

When we make a qualitative analysis of the generated counter-narratives, we can see that providing argumentative information about the hate tweet does yield counter-narratives that are more specific and informative, as can be seen in Figure 3. Models counting with this information frequently use it by negating the relation between Collective and Property or between Justification and Conclusion.

Results obtained for counter-narratives for Spanish hate tweets were much worse, as could be expected given the much smaller number of examples for fine-tuning and that base LLMs

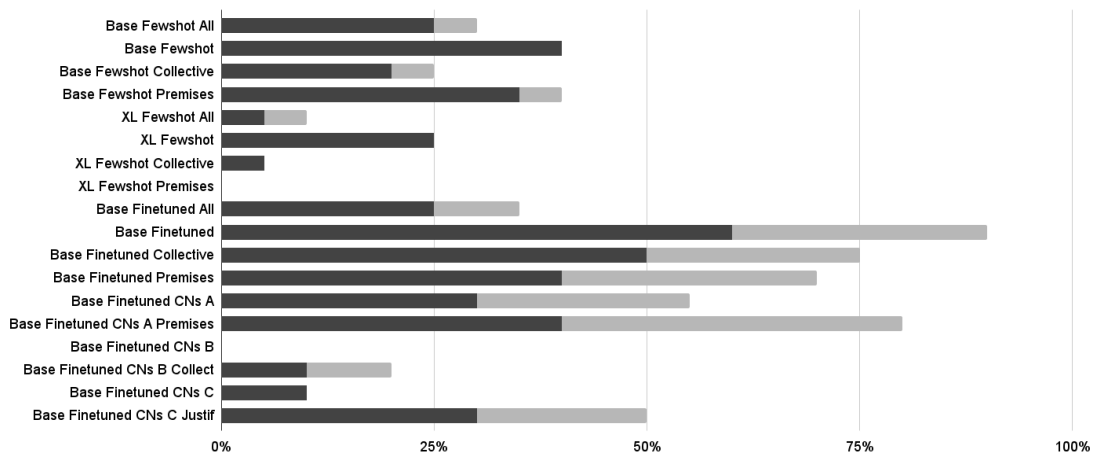


Figure 2: Proportion of Excellent (light gray) and Good (light + dark gray) counter-narratives produced by each approach.

<u>Tweet with argumentative information:</u>	<u>Tweet without argumentative information:</u>
street interview whit italians "send all migrants back to where they came from they block streets to pray " - free speech time - https://t.co/d5dqr8pg3r @user Justification: street interview whit italians "send all migrants back to where they came from they block streets to pray " (fact) Conclusion: "send all migrants back to where they came from they block streets to pray " (policy) Pivot: migrants - they - they	street interview whit italians "send all migrants back to where they came from they block streets to pray " - free speech time - https://t.co/d5dqr8pg3r @user
<u>Counter-narrative:</u>	<u>Counter-narrative:</u>
I don't think it's a good idea to send all migrants back to where they came from.	I don't think it's the right thing to do.

Figure 3: Counter-narratives obtained for the same tweet with different strategies: including argumentative information (above) and without argumentative information (below).

perform worse for tasks in Spanish in general. Indeed, values for Informativeness and Felicity almost never reach more than 10% positive, and Stance and Offensiveness are almost never beyond 30% positiveness. However, the same tendency as for English could be observed: fine-tuned models perform better than non-fine-tuned models, even if the latter are bigger. Moreover, argumentative information seems to make a bigger impact in improving the generated counter-narratives than in the case of English, with increases in the range of 30%-50% in the reduction of negative scores for Offensiveness and Stance, although a decrease in Felicity. Given these

encouraging results with such few examples, we will be increasing the number of examples with argumentative information in future work.

6. Conclusions and future work

We have presented an approach to generate counter-narratives against hate speech in social media by prompting large language models with information about some argumentative aspects of the original hate speech. We have carried out a small manual evaluation of the quality of generated counter-narratives. This evaluation is preliminary, with a small number of judgements and moderate to substantial inter-annotator agreement, but we have found promising tendencies.

We have shown that argumentative information by itself does not improve the quality of counter-narratives generated by LLMs, on the contrary, it may even be detrimental, specially in the case of bigger models. However, fine-tuning a smaller model with a small corpus of high-quality examples of pairs hate speech – counter-narrative yields some improvement in performance. This finding has a significant impact both because smaller language models are more accessible to low-budget scenarios, and because of their smaller carbon footprint.

We have also shown that some kinds of argumentative information do have some positive impact in generating more specific, more informative counter-narratives. In particular, we have found that the types of counter-narrative that negate the relation between the Justification and the Conclusion and that negate the Justification have an improvement in performance if argumentative information about the Justification and the Conclusion is provided.

Moreover, we have also found that argumentative information makes a positive impact in scenarios with very few tweets, as shown by our experiments for Spanish. Although the quality of the counter-narratives generated for Spanish is much lower than for English, the fact that argumentative information has a positive impact is encouraging, and we will continue to annotate examples for Spanish to improve the generation of counter-narratives.

We will also explore other aspects of the quality of counter-narratives, with a more insightful, more extensive human evaluation. We will also explore the interaction between argumentative information and other aspects, like vocabulary, level of formality, and culture.

Finally, the evaluation of counter-narratives is still far from being solved. We are currently considering different avenues to improve it, as it is a crucial step to advance the field. We are working on obtaining a higher number of judgements, but also on more insightful guidelines that reflect more valuable aspects of counter-narratives, more related to argument acceptability.

7. Acknowledgments

This work was funded in part by Secretaría de Investigación Científica y Tecnológica FCEN–UBA (RESCS-2020-345-E-UBA-REC), CONICET under the PIP (grant 11220200101408CO), Agencia Nacional de Promoción Científica y Tecnológica, Argentina under grants PICT-2018-0475 (PRH-2014-0007), PICT-2020- SERIEA-01481, and the NAACL Regional Americas Fund (2022). This work used computational resources from CCAD – UNC (<https://ccad.unc.edu.ar/>), which are part of SNCAD – MinCyT, Argentina. We specially want to thank two anonymous reviewers that contributed to improve this work with their thoughtful and constructive comments.

References

- [1] T. Davidson, D. Bhattacharya, I. Weber, Racial bias in hate speech and abusive language detection datasets, in: *Proceedings of Third Workshop on Abusive Language Online*, 2019.
- [2] S. Benesch, *Countering dangerous speech: New ideas for genocide prevention*, United States Holocaust Memorial Museum, 2014.
- [3] Y.-L. Chung, E. Kuzmenko, S. S. Tekiroglu, M. Guerini, CONAN - COUNTER NARRATIVES through nichesourcing: a multilingual dataset of responses to fight online hate speech, in: *ACL*, 2019.
- [4] L. Thorburn, A. Kruger, Optimizing language models for argumentative reasoning, in: *Proceedings of the 1st Workshop on Argumentation & Machine Learning co-located with 9th International Conference on Computational Models of Argument (COMMA 2022)*, 2022.
- [5] M. Hinton, J. H. M. Wagemans, How persuasive is ai-generated argumentation? an analysis of the quality of an argumentative text produced by the GPT-3 AI text generator, *Argument Comput.* 14 (2023) 59–74. URL: <https://doi.org/10.3233/AAC-210026>. doi:10.3233/AAC-210026.
- [6] R. Gupta, S. Desai, M. Goel, A. Bandhakavi, T. Chakraborty, M. S. Akhtar, Counterspeeches up my sleeve! intent distribution learning and persistent fusion for intent-conditioned counterspeech generation, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 5792–5809. URL: <https://aclanthology.org/2023.acl-long.318>.
- [7] J. Qian, A. Bethke, Y. Liu, E. M. Belding, W. Y. Wang, A benchmark dataset for learning to intervene in online hate speech, *CoRR abs/1909.04251* (2019).
- [8] C. Ziemis, B. He, S. Soni, S. Kumar, Racism is a virus: Anti-asian hate and counterhate in social media during the covid-19 crisis, 2020.
- [9] S. S. Tekiroğlu, Y.-L. Chung, M. Guerini, Generating counter narratives against online hate speech: Data and strategies, in: *ACL*, 2020.
- [10] M. Fanton, H. Bonaldi, S. S. Tekiroğlu, M. Guerini, Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech, in: *ACK*, 2021.
- [11] H. Bonaldi, S. Dellantonio, S. S. Tekiroğlu, M. Guerini, Human-machine collaboration approaches to build a dialogue dataset for hate speech countering, in: *EMNLP*, 2022.
- [12] Y.-L. Chung, S. S. Tekiroğlu, M. Guerini, Towards knowledge-grounded counter narrative generation for hate speech, in: *Findings of the ACL-IJCNLP 2021*, 2021.
- [13] M. Ashida, M. Komachi, Towards automatic generation of messages countering online hate speech and microaggressions, in: *Proceedings of Sixth Workshop on Online Abuse and Harms (WOAH)*, 2022.
- [14] M. Vallecillo-Rodríguez, A. Montejo Ráez, M. Martín-Valdivia, Automatic counter-narrative generation for hate speech in spanish, *Procesamiento del Lenguaje Natural* 71 (2023).
- [15] Y.-L. Chung, M. Guerini, R. Agerri, Multilingual counter narrative type classification, in: *Proceedings of the 8th Workshop on Argument Mining*, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 125–132. URL: <https://aclanthology.org/2021.argmining-1.12>. doi:10.18653/v1/2021.argmining-1.12.

- [16] D. Furman, P. Torres, J. Rodríguez, D. Letzen, V. Martínez, L. Alonso Alemany, Which argumentative aspects of hate speech in social media can be reliably identified?, in: Proceedings of Fourth International Workshop on Designing Meaning Representations, co-located with IWCS 2023, 2023.
- [17] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, M. Sanguinetti, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter, in: Proceedings of 13th International Workshop on Semantic Evaluation, 2019.
- [18] J. H. M. Wagemans, Constructing a periodic table of arguments, in: Proceedings of 11th International Conference of the Ontario Society for the Study of Argumentation, 2016.
- [19] D. Walton, C. Reed, F. Macagno, Argumentation Schemes, CUP, 2008.
- [20] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: ACL, 2002.
- [21] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004.
- [22] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement 20 (1960) 37.
- [23] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174.

ArguCast: A System for Online Multi-Forecasting with Gradual Argumentation

Deniz Gorur, Antonio Rago and Francesca Toni

Department of Computing, Imperial College London, UK

Abstract

Judgmental forecasting is a form of forecasting which employs (human) users to make predictions about specified future events. Judgmental forecasting has been shown to perform better than quantitative methods for forecasting, e.g. when historical data is unavailable or causal reasoning is needed. However, it has a number of limitations, arising from users' irrationality and cognitive biases. To mitigate against these phenomena, we leverage on computational argumentation, a field which excels in the representation and resolution of conflicting knowledge and human-like reasoning, and propose novel *ArguCast frameworks* (ACFs) and the novel online system *ArguCast*, integrating ACFs. ACFs and ArguCast accommodate *multi-forecasting*, by allowing multiple users to debate on multiple forecasting predictions simultaneously, each potentially admitting multiple outcomes. Finally, we propose a novel notion of *user rationality* in ACFs based on votes on arguments in ACFs, allowing the filtering out of irrational opinions before obtaining *group forecasting* predictions by means commonly used in judgmental forecasting.

Keywords

Bipolar Argumentation, Gradual Semantics, Judgmental Forecasting,

1. Introduction

Judgmental forecasting is a form of forecasting which employs (human) users to make predictions about specified future events [1]. It is advocated as a valuable alternative to conventional quantitative methods for forecasting when historical data is unavailable or causal reasoning is required [1]. However, judgmental forecasting has a number of limitations, arising from (human) users' irrationality and cognitive biases [2] arising from over-/under-confidence [3] in their judgment. To overcome these issues many solutions have been proposed. Researchers have investigated the best ways of eliciting probabilities from humans [4], how incentives and training change users' forecasting abilities [5], and the effect of scoring rules on users [1]. Another research direction has focused on employing many experts or humans and aggregating their predictions since it has been found that group judgment usually performs better as the impact of bias is reduced by cancelling random error [1]. However, when there are many humans involved in forecasting, a new problem arises: how to effectively combine all the predictions that are made. A further, orthogonal issue with existing systems, e.g. [6], is that any information provided by users, e.g. their forecasts or reasoning therefor, concern a single event, and thus must be provided separately for different events. It is easy to see that being able to consider

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ d.gorur@imperial.ac.uk (D. Gorur); a.rago@imperial.ac.uk (A. Rago); ft@imperial.ac.uk (F. Toni)

🆔 0009-0008-8976-8919 (D. Gorur); 0000-0001-5323-7739 (A. Rago); 0000-0001-8194-1459 (F. Toni)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

different events in one forecasting framework would utilise the information provided by users much more effectively.

Meanwhile, *argumentation* constitutes a major component of human intelligence since the ability to engage in arguments is essential for humans to understand new problems, perform scientific reasoning, and express, clarify, and defend their opinions in their daily lives [7]. *Computational argumentation* (see [8, 9] for overviews) has become an important topic in artificial intelligence due to its ability to conjugate representational needs with user-related cognitive models and computational models for automated reasoning [10]. These computational models are formalised as *argumentation frameworks*. Argumentation involves reasoning with uncertainty, and resolving conflicting information so we posit that it is natural to apply techniques from argumentation to the area of judgmental forecasting. However, there has been very little research in the use of argumentation in forecasting technology in the past. We are aware of only one such approach [6], which restricts users' provided information to single events and single outcomes.

In order to address the aforementioned issues, and thus make contributions to the field of judgmental forecasting, we leverage on computational argumentation. Specifically, we propose the novel ArguCast frameworks (ACFs) and a novel online system ArguCast, integrating ACFs. Like [6], ACFs (and thus ArguCast) allows for groups of users to make forecasts on events while engaging in argumentative debates supported by votes on arguments exchanged in these debates, encouraging users to consider and share their reasoning for their forecasts. However, differently from the existing approach, ACFs accommodate multiple forecasting predictions with multiple outcomes from multiple users, allowing for information to be shared across events. We also propose a novel notion of *user rationality*, comprising vote rationality and prediction rationality in ACFs, allowing the filtering out of irrational opinions before obtaining *group forecasting* predictions. In doing so, we provide a novel, argumentative method for combining forecasts, which introduces *multi-forecasting* on multiple events simultaneously and accounts for human biases and rationality.

The structure of the paper is as follows. In Section 2, we describe the most relevant existing approaches to forecasting. In Section 3, we give the necessary background on computational argumentation, which is used for defining rationality. In Section 4, we define ACFs. In Section 5, we provide an overview of the implementation of ACFs as the ArguCast system. In Section 6, we define our notions of user rationality in ACFs and demonstrate how they can be used to filter out irrational predictions before we aggregate users' predictions. Finally, Section 7 concludes and considers possible future directions of work.

2. Related Work

In this section, we will discuss the most relevant of the existing approaches to forecasting from the literature.

There are two approaches to combining forecasts: the qualitative approach (e.g. a group discussion to reach consensus) and the mechanical approach (e.g. a simple or weighted average of the forecasts). It has been shown that those which are mechanical are more likely to lead to greater accuracy than those which are qualitative [2]. Many ways of mechanical combinations

methods have been proposed before such as linear, log-linear, and democratic opinion pools. The aggregation method that we use in Section 6 is a variation of the log-linear pooling method.

Some of the systems that attempt to improve the forecasting capability of users by incentives, e.g. via monetising the users' predictions based on their accuracy, are Hypermind¹, Smarkets², PredictIt³, and Polymarket⁴. The idea behind all of these systems is based on prediction markets, where people bet on the predictions. Smarkets and PredictIt do not have any functionality for the agents to debate amongst each other, whereas Polymarket and Hypermind only have a general chat/forum.

Good Judgment Open (GJOpen)⁵ [11], Metaculus⁶, and Infer⁷ are examples of group judgment systems. They can all support binary and multiple-answer questions. In addition to these Metaculus supports numeric interval and date interval questions. They both have a comment section for users to put forward their reasoning for their forecast. GJOpen and Infer elicit both reasoning for why the forecast could be correct, and why the forecast could be wrong from users, as it has been shown that forcing users to think about why they might be wrong makes them better forecasters (see Appendix A of [12]). GJOpen also investigates what would happen if the top forecasters of their tournaments were put on teams called 'Superforecasters' [13], and they were found to outperform the simple average of the crowd.

However, all of the systems we have discussed lack any mechanisms for eliciting, representing and evaluating the argumentative reasoning which takes place in the debates amongst users. We are aware of only Irwin et al.[6] who have formalised an argumentation framework that supports forecasting. However, for all its strengths, this approach hosts a number of shortcomings, such as the fact that it can only handle questions with binary answers and does not allow the same argument to be used for multiple questions, or even in the same question. This could introduce repetition and sparsity, which could cause confusion in users. In our novel ArguCast frameworks, we address all of these issues.

3. Background

Our approach uses *Quantitative Bipolar Argumentation Frameworks (QBAFs)* [14]. A QBAF is a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{S}, \tau \rangle$ where $\langle \mathcal{X}, \mathcal{A}, \mathcal{S} \rangle$ is a *Bipolar Argumentation Framework (BAF)* [15] and τ is a *base score function*, such that:

- $\mathcal{X}$ is a set of *arguments*;
- $\mathcal{A}$ is a binary relation of *attack* on $\mathcal{X}$, $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{X}$;
- $\mathcal{S}$ is a binary relation of *support* on $\mathcal{X}$, $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{X}$; and
- $\tau : \mathcal{X} \rightarrow [0, 1]$ is a total function; $\tau(a)$ is the *base score* of $a \in \mathcal{X}$.

¹<https://predict.hypermind.com>

²<https://smarkets.com/>

³<https://www.predictit.org/>

⁴<https://polymarket.com/>

⁵<https://www.gjopen.com>

⁶<https://www.metaculus.com>

⁷<https://www.infer-pub.com>

In this paper we focus on the *Discontinuity-Free QuAD gradual semantics (DF-QuAD)* [16] for QBAFs. DF-QuAD determines the strength of arguments based on combining their base scores and the aggregated strength of their attackers and supporters, where, for $a \in \mathcal{X}$, the *attackers* of a are $\mathcal{A}(a) = \{b \mid (b, a) \in \mathcal{A}\}$ and the *supporters* of a are $\mathcal{S}(a) = \{b \mid (b, a) \in \mathcal{S}\}$. Let the *strength aggregation function* be $\delta : [0, 1]^* \rightarrow [0, 1]$ such that, for $T = (v_1, \dots, v_n) \in [0, 1]^*$:

$$\begin{aligned} \text{if } n = 0 : \delta(T) &= 0; \\ \text{if } n = 1 : \delta(T) &= v_1; \\ \text{if } n = 2 : \delta(T) &= f(v_1, v_2); \\ \text{if } n > 2 : \delta(T) &= f(\delta(v_1, \dots, v_{n-1}), v_n) \end{aligned}$$

where, for $x, y \in [0, 1]$, $f(x, y) = x + (1 - x) \cdot y = x + y - x \cdot y$. Let the *combination function* be defined as $c : [0, 1] \times [0, 1] \times [0, 1] \rightarrow [0, 1]$, where for $v_0, v_a, v_s \in [0, 1]$:

$$\begin{aligned} c(v_0, v_a, v_s) &= v_0 - v_0 \cdot |v_s - v_a| & \text{if } v_a \geq v_s; \\ c(v_0, v_a, v_s) &= v_0 + (1 - v_0) \cdot |v_s - v_a| & \text{if } v_a < v_s. \end{aligned}$$

Then, DF-QuAD computes the strength of arguments by the *score function* $\sigma : \mathcal{X} \rightarrow [0, 1]$ where, for any $a \in \mathcal{X}$, $\sigma(a) = c(\tau(a), \delta(\sigma(\mathcal{A}(a))), \delta(\sigma(\mathcal{S}(a))))$ such that $\sigma(\mathcal{A}(a)) = (\sigma(a_1), \dots, \sigma(a_n))$, where $(a_1, \dots, a_n)$ is an arbitrary permutation of the $(n \geq 0)$ attackers in $\mathcal{A}(a)$, and $\sigma(\mathcal{S}(a)) = (\sigma(s_1), \dots, \sigma(s_m))$, where $(s_1, \dots, s_m)$ is an arbitrary permutation of the $(m \geq 0)$ supporters in $\mathcal{S}(a)$.

4. ArguCast Frameworks

We introduce novel ArguCast frameworks, accommodating multi-forecasting, i.e. multiple *forecasting predictions* with multiple outcomes from multiple *users*, supported by argumentative debates and *votes* on arguments exchanged in these debates.

Definition 1. An ArguCast framework (ACF) is a tuple $\langle \mathcal{X}, \mathcal{R}, \mathcal{U}, \mathcal{V}, \mathcal{P} \rangle$ such that:

- $\mathcal{X} = \mathcal{F} \cup \mathcal{D}$ is a finite set of arguments where $\mathcal{F}$ and $\mathcal{D}$ are disjoint; elements of $\mathcal{F}$ and $\mathcal{D}$ are referred to, respectively, as *forecasting* and *non-forecasting* arguments;
- $\mathcal{R} = \mathcal{A} \cup \mathcal{S} \subseteq \mathcal{D} \times \mathcal{X}$, where $\mathcal{A}$ and $\mathcal{S}$ are disjoint relations (i.e. sets of pairs from $\mathcal{D} \times \mathcal{X}$) of attack and support, respectively;
- $\mathcal{U}$ is a finite set of users;
- $\mathcal{V} : \mathcal{U} \times \mathcal{D} \rightarrow \{-, +\}$ is a (partial) function; $\mathcal{V}(u, a)$ is the vote of user $u \in \mathcal{U}$ on (non-forecasting) argument $a \in \mathcal{D}$;
- $\mathcal{P} : \mathcal{U} \times \mathcal{F} \rightarrow [0, 1]$ is a (partial) function; $\mathcal{P}(u, b)$ is the forecasting prediction by user $u \in \mathcal{U}$ on (forecasting) argument $b \in \mathcal{F}$.

Forecasting arguments represent answers to forecasting questions. There may be any number of forecasting arguments, as the answers may be Yes/No or take any value in a discrete set (thus the forecasting predictions may have multiple outcomes). If there is a single forecasting question of

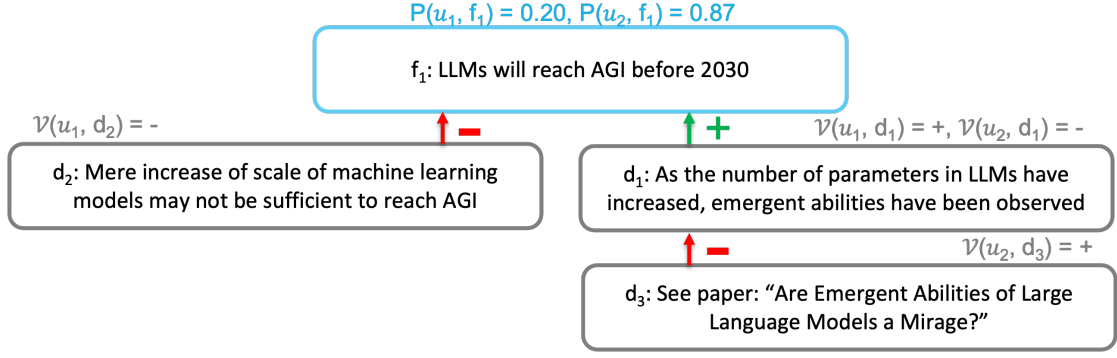


Figure 1: A visual representation of Example 1, where attacks and supports are represented by red and green, respectively, edges.

interest admitting a binary answer (e.g. ‘Will Miami Heat win the 2022-23 NBA Championship?’) then we assume that $\mathcal{F}$ is a singleton set consisting of an argument for the positive answer to the question. If there are multiple forecasting questions or a single forecasting question admitting several alternative answers (e.g. ‘Which team will win the 2022-23 NBA Championship?’) then $\mathcal{F}$ will consist of several forecasting arguments. Non-forecasting arguments can be seen as the users’ rationales/opinions around the forecasting arguments. Note that, by definition of $\mathcal{A}$ and $\mathcal{S}$, forecasting arguments can be attacked/supported but they cannot attack/support other arguments, whereas non-forecasting arguments can attack/support (or be attacked/supported by) any arguments, including potentially attacking/supporting more than one argument. The users are forecasters, who can vote (positively or negatively) on non-forecasting arguments and/or express a numerical prediction (in $[0,1]$) for forecasting arguments. The votes indicate agreement or disagreement with the non-forecasting argument, whereas prediction forecasts indicate the users’ degree of belief in the forecasting arguments. Note that, as $\mathcal{V}$ and $\mathcal{P}$ may be partial, users may refrain from voting and forecasting.

Example 1. A possible ACF for the question ‘Will large language models (LLMs) reach AGI before 2030?’ is $\mathcal{F} = \{f_1 = \text{‘LLMs will reach AGI before 2030’}\}$, $\mathcal{D} = \{d_1 = \text{‘As the number of parameters in LLMs have increased, emergent abilities have been observed’}, d_2 = \text{‘Mere increase of scale of machine learning models may not be sufficient to reach AGI’}, d_3 = \text{‘See paper: “Are Emergent Abilities of Large Language Models a Mirage?”’}\}$, $\mathcal{A} = \{(d_2, f_1), (d_3, d_1)\}$, $\mathcal{S} = \{(d_1, f_1)\}$, $\mathcal{U} = \{u_1, u_2\}$, $\mathcal{V}(u_1, d_1) = +$, $\mathcal{V}(u_1, d_2) = -$, $\mathcal{V}(u_2, d_1) = -$, $\mathcal{V}(u_2, d_3) = +$, $\mathcal{P}(u_1, f_1) = 0.87$, $\mathcal{P}(u_2, f_1) = 0.20$, as shown in Figure 1.

Finally, note that our ACFs share some features of existing argumentation frameworks, but are different therefrom as follows. Like BAFs [15], ACFs use two relations of attack/support but in addition ACFs distinguish two types of arguments (forecasting and non-forecasting arguments) and include users, users’ votes on non-forecasting arguments and users’ predictions on forecasting arguments. Like QuAD frameworks [17], ACFs single out a specific type of argument under debate (forecasting arguments in ACFs but answer arguments in [17]) but

QuAD frameworks also distinguish con/pro arguments and admit a single relation whereas we distinguish attack/support relations, allowing for arguments to potentially attack some arguments and support some others. Also, QuAD frameworks lack users, votes and predictions, but include base scores for arguments, absent in ACFs (where, however, they can be obtained using votes, see below). QuaD-V [18] is an extension of QuAD frameworks like ACFs including users and votes and excluding base scores. While the votes in QuAD-V are given by a total function into $\{-, ?, +\}$, we use a partial function into $\{-, +\}$. QuAD-V frameworks also lack support for forecasting predictions. Like the *forecasting argumentation frameworks* (FAFs) of [6], ACFs are designed to support forecasting but FAFs can only handle questions with binary answers (as they can only have one proposal argument at a time). Like FAFs, ACFs single out a specific type of argument under debate (forecasting arguments in ACFs but *proposal arguments* in [6]), and they also support users with votes and forecasts. The votes in FAFs are assigned by a total function that forces users to provide their opinion on every argument. FAFs also use *amendment arguments* (arguments proposing the forecasted probability is increased or decreased) as well as pro/con arguments as in QuAD and V-QuAD frameworks, and a single relation between arguments, where amendment arguments can only relate to proposal arguments and con/pro arguments can only relate to amendment and other con/pro arguments. A further difference between ACFs and FAFs lies in the fact that, like in QuAD and QuAD-V, FAFs distinguish con/pro arguments and admit a single relation, which could introduce repetition and sparsity, which will lead to confusion in users. ACFs avoid this issue by adopting attack/support relations rather than a single relation type.

5. ArguCast

ArguCast is an online system, available at <https://argucast.herokuapp.com>, accommodating ACFs in practice as the basis for judgemental forecasting.⁸ We focus here on the system’s functionalities. Note that, even though the formalisation of ACFs handles binary and multi-answer forecasting questions, ArguCast supports only binary questions currently. Also, whereas ACFs allow for the same non-forecasting arguments to contribute to debating several forecasting arguments, ArguCast assumes for the time being that each non-forecasting argument contributes to debating only one forecasting argument. Thus, each ACF in ArguCast can be seen as the composition of disjoint ACFs, one for each forecasting question.

ArguCast is login-protected so all users need to register before engaging in forecasting. Users can add their own forecasting questions (with accompanying forecasting arguments, amounting to a positive answer to each question, as illustrated in Figure 2), or select from currently active questions (as illustrated in Figure 3, also showing that users can search for specific questions and add new questions by clicking on the plus button).

Figure 4 shows ArguCast’s representation of the ACF for one of the forecasting questions in Figure 3 from the viewpoint of a single user (i.e. $u \in \mathcal{U}$). Note that users do not have access to other users’ votes and predictions but they can see everything else. Note that ArguCast

⁸ArguCast’s user interface is implemented with React.js (<https://react.dev>). Storage of arguments, users, and predictions were on a PostgreSQL database. The Web API that connects to the database and executes queries requested from the user interface was implemented with Python’s Flask library.

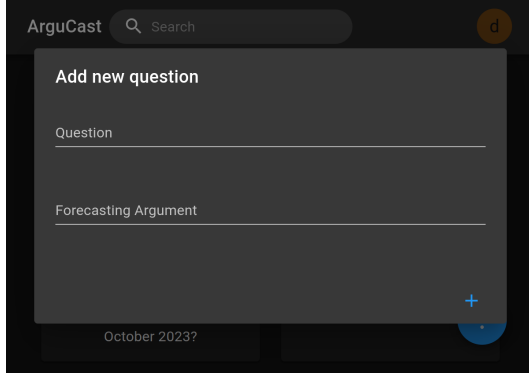


Figure 2: Dialogue to add new questions.

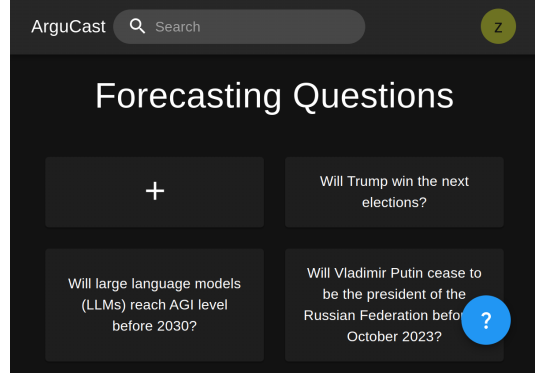


Figure 3: Overview of the active questions.

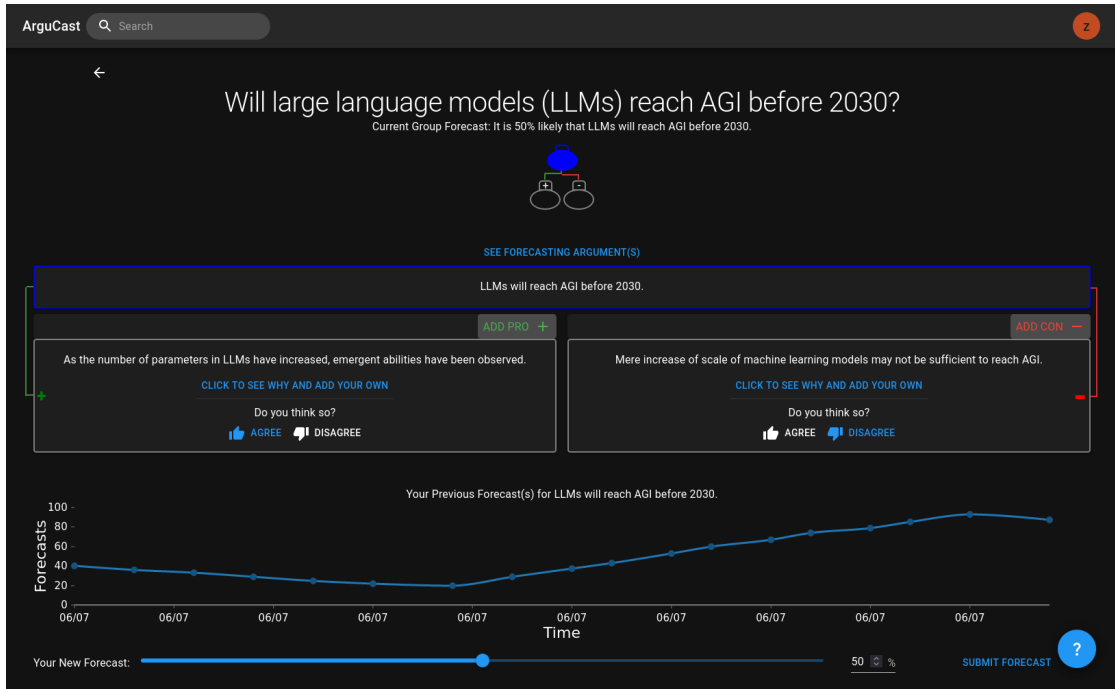


Figure 4: A fragment of Example 1 represented in ArguCast, from the viewpoint of user u_1 . Here we show arguments $f_1/d_1/d_2$, u_1 's votes on d_1/d_2 , and u_1 's forecasting predictions on f_1 over time.

supports two tree-based visualisations of each debate/ACF: a global, abstract visualisation (as shown at the top of Figure 4), focusing on the relations between the arguments in the ACF; and a localised visualisation around specific arguments (as shown in the centre of Figure 4), showing their attackers and supporters and allowing the user to vote (cf. $\mathcal{V}$). The localised visualisation supports the addition of supporting and attacking arguments. In both visualisations, forecasting arguments are outlined in blue and non-forecasting arguments are outlined in grey.

The attack/support relations (i.e. $\mathcal{A}/\mathcal{S}$) between arguments are represented as red/green edges, respectively, with a minus/plus (-/+) sign, respectively.

Below the question in Figure 4, the current group forecast is shown. The group forecast does not change in our system as of yet as the aggregation of users' predictions (which will be defined in Section 6) is not implemented in ArguCast.

Below the debate, the user has the ability to put forward their forecasting prediction (i.e. $\mathcal{P}$) for the forecasting argument using the slider. The slider ranges from $[0, 100]\%$, which maps to $[0, 1]$. The user can change their vote as they please, iteratively.

Any user can see the debates already present in the system. However, in order to add new arguments, cast opinions by voting, and put forward predictions, the user needs to be signed in to their account. If a user does not have an account they can sign up with their email or Google account.

6. Extensions of ArguCast Frameworks

The ArguCast frameworks are the base for improving forecasting systems using argumentation. One way of doing so is to define notions of rationality so we can filter out irrational users when aggregating forecasting predictions, which we will now demonstrate. Note that filtering for rationality and the aggregation of users' forecasts has not, as of yet, been implemented into ArguCast system.

In the remainder, unless specified otherwise, we will assume as given an ACF $\langle \mathcal{X}, \mathcal{R}, \mathcal{U}, \mathcal{V}, \mathcal{P} \rangle$. We will also assume that $\mathcal{R}$ in this ACF is acyclic.

An ACF captures the opinions of all users (in $\mathcal{U}$) involved in forecasting. We can filter out the opinions of individual users as *user QBAFs*, i.e. a QBAF representing a single user's votes in the ACF, and then apply gradual semantics thereto for determining *rationality* of users by comparing the strengths of arguments and votes/forecasting predictions.

Definition 2. A user QBAF for $u \in \mathcal{U}$ is a QBAF $\langle \mathcal{X}, \mathcal{A}, \mathcal{S}, \tau_u \rangle$ such that, for $a \in \mathcal{X}$:

$$\tau_u(a) = \begin{cases} 1 & \text{if } \mathcal{V}(u, a) = +, \\ 0 & \text{if } \mathcal{V}(u, a) = -, \\ 0.5 & \text{if } \mathcal{V}(u, a) \text{ is undefined.} \end{cases}$$

The attacking and supporting strengths of an argument $a \in \mathcal{X}$ in the user QBAF are defined as $\delta(\sigma(\mathcal{A}(a)))$ and $\delta(\sigma(\mathcal{S}(a)))$, denoted $\sigma_{\mathcal{A}}^u(a)$ and $\sigma_{\mathcal{S}}^u(a)$.

Example 2. A user QBAF for u_1 for the ACF given in Example 1 would be $\mathcal{X} = \{f_1, d_1, d_2, d_3\}$, $\mathcal{A} = \{(d_2, f_1), (d_3, d_1)\}$, $\mathcal{S} = \{(d_1, f_1)\}$, $\tau_{u_1}(f_1) = 0.5$, $\tau_{u_1}(d_1) = 1$, $\tau_{u_1}(d_2) = 0$, and $\tau_{u_1}(d_3) = 0.5$. Using DF-QuAD the strength of arguments is $\sigma(d_3) = 0.5$, $\sigma(d_2) = 0$, $\sigma(d_1) = c(1, 0.5, 0) = 1 - 1 \cdot |0 - 0.5| = 0.5$, and $\sigma(f_1) = c(0.5, 0, 0.5) = 0.5 + (1 - 0.5) \cdot |0.5 - 0| = 0.75$. Then attacking and supporting strengths of the non-forecasting argument d_1 is $\sigma_{\mathcal{A}}^u(d_1) = 0.5$ and $\sigma_{\mathcal{S}}^u(d_1) = 0$, respectively.

We define two notions of *user rationality* for ACFs: *vote rationality*, which compares the vote of a user on any non-forecasting argument with its strength; and *prediction rationality*, which compares the user's forecasting prediction on any forecasting argument with its strength. In the remainder, we will assume as given a user QBAF $\langle \mathcal{X}, \mathcal{A}, \mathcal{S}, \tau_u \rangle$ for a user $u \in \mathcal{U}$ in the ACF.

Definition 3. User u is vote rational iff $\forall a \in \mathcal{X}$:

$$\begin{aligned} \text{if } \mathcal{V}(u, a) = - \text{ then } \sigma_{\mathcal{A}}^u(a) &\geq \sigma_{\mathcal{S}}^u(a); \\ \text{if } \mathcal{V}(u, a) = + \text{ then } \sigma_{\mathcal{A}}^u(a) &\leq \sigma_{\mathcal{S}}^u(a). \end{aligned}$$

Example 3. Continuing Example 2, user u_1 is not vote rational. The user agreed with d_1 and the strength of the attacking arguments is bigger than the strength of the supporting arguments (i.e. $\sigma_{\mathcal{A}_u}(d_1) > \sigma_{\mathcal{S}_u}(d_1)$). In this instance, the vote rationality forces the user to add reasoning for the argument d_1 or put forward their opinion on d_3 .

Definition 4. User u is prediction rational iff $\forall a \in \mathcal{A}$:

$$\begin{aligned} \text{if } \sigma(a) < 0.5 \text{ then } \mathcal{P}(u, a) &< 0.5; \\ \text{if } \sigma(a) > 0.5 \text{ then } \mathcal{P}(u, a) &> 0.5; \\ \text{if } \sigma(a) = 0.5 \text{ then } \mathcal{P}(u, a) &= [0.5 - \epsilon, 0.5 + \epsilon] \text{ for some small } \epsilon. \end{aligned}$$

Example 4. Continuing Example 2, user u_1 is not prediction rational. User u_1 's forecasting prediction is $P(u_1, f_1) = 0.2$ and the strength of the forecasting argument is $\sigma(f_1) = 0.75$. In this instance, u_1 needs to change its forecasting prediction to be below 0.5 or update its vote(s) to decrease the strength of f_1 . This demonstrates how prediction rationality requires that a user's forecasting predictions be in line with their votes.

Definition 5. The ACFs are collectively rational iff $(\forall u \in \mathcal{U})$ are vote rational and prediction rational.

Aggregation of forecasts requires all the agents to be collectively rational. The process of aggregation thus uses only the forecasting predictions.

We use a weighted aggregation function where the weights are *Brier scores* [19] which represent the accuracy of each user in the previous questions that have an outcome. So, the users with better Brier scores will have a greater influence on the aggregated prediction. The outcome of each question is represented by $O_i \in \{0, 1\}$, where $O_i = 1$ if the outcome was true and $O_i = 0$ if the outcome was false.

Definition 6. Given all N outcomes as a set $(\{O_1, \dots, O_N\})$ and the corresponding forecasting predictions for user $u \in \mathcal{U}$ $(\{P(u)_1, \dots, P(u)_N\})$, the Brier score of u is:

$$b_u = \frac{1}{N} \sum_{t=1}^N (P_t - O_t)^2$$

Brier scores are the mean squared error of the user's forecasting accuracy. A low b_u represents higher accuracy and a high b_u represents lower accuracy.

Then, our aggregation function is an adaptation of [20] where we also use (the negation of the) Brier scores to obtain a weighted aggregation.

Definition 7. The geometric mean of odds with systematic bias $\alpha \geq 1$, $\Omega : ACF \rightarrow [0, 1]$ is:

$$\Omega(ACF) = \left[\sqrt[|\mathcal{U}|]{\prod_{u \in \mathcal{U}} \left(e^{(1-b_u)} \frac{\mathcal{P}(u)}{1 - \mathcal{P}(u)} \right)} \right]^\alpha$$

The aggregation function $\lambda : ACF \rightarrow [0, 1]$ is:

$$\begin{aligned} \text{if } |\mathcal{U}| \neq 0 : & \quad \lambda(ACF) = \frac{\Omega(ACF)}{\Omega(ACF) + 1} \\ \text{otherwise} & \quad \lambda(ACF) = 0 \end{aligned}$$

The geometric mean of odds has been shown (empirically) to outperform [20] the arithmetic mean of odds as uncertain predictions will have less influence. Note also that if the systematic bias is 1 then the geometric mean of odds is similar to the arithmetic mean of odds. We will use $\alpha = 2.5$ for simplicity, however in practice, the value of α could be estimated [20].

7. Conclusions and Future Work

We have introduced our novel ACFs, accommodating forecasting predictions from users, argumentative debates (as Bipolar AFs) amongst users, and votes on arguments exchanged in these debates. We also described ArguCast, our online platform which instantiates ACFs. Then, we defined our notions of rational users for ACFs and showed how we can filter out irrational users when we combine users' predictions. We have also shown a way to combine users' predictions using the geometric mean of odds weighted by users' Brier scores.

ACFs open up numerous avenues for future work. First, we plan to implement rationality constraints and prediction aggregation (in the forms discussed in Section 6 as well as others) in our online system and then empirically evaluate how much the accuracy of the forecasts improves, comparing with those defined in [6]. Second, we will build on the fact that ACFs provide the formal basis for further theoretical developments combining forecasting and argumentation. For example, at the moment, ACFs only allow users to vote on non-forecasting arguments so that we can apply rationality constraints to users. However, we would like to see how we can accommodate votes on attack/support relations to capture the users' beliefs on the relations between arguments, which would allow us to extend the rationality constraints we have introduced. Another possible theoretical development would be to include a mapping from users to their contributed arguments to assess how this ownership affects voting, possibly allowing us to model users' cognitive biases, such as *confirmation bias* [21], with argumentation, as in [22]. Finally, it would be interesting to generate explanations for the combined prediction, leveraging on argumentation's amenability for explanation (see [23, 24] for recent surveys on its application to explainable AI).

Acknowledgments

This research was partially funded by the ERC under the EU's Horizon 2020 research and innovation programme (grant agreement no. 101020934, ADIX), by J.P. Morgan and by the Royal Academy of Engineering, UK, under the Research Chairs and Senior Research Fellowships scheme. Any views or opinions expressed herein are solely those of the authors.

References

- [1] M. Zellner, A. E. Abbas, D. V. Budescu, A. Galstyan, A survey of human judgement and quantitative forecasting methods, *Royal Society Open Science* 8 (2021) rsos.201187, 201187. doi:10.1098/rsos.201187.
- [2] M. Lawrence, P. Goodwin, M. O'Connor, D. Önkal, Judgmental forecasting: A review of progress over the last 25 years, *International Journal of Forecasting* 22 (2006) 493–518. doi:10.1016/j.ijforecast.2006.03.007.
- [3] D. A. Moore, P. J. Healy, The trouble with overconfidence., *Psychological Review* 115 (2008) 502–517. doi:10.1037/0033-295X.115.2.502.
- [4] T. S. Wallsten, D. V. Budescu, A review of human linguistic probability processing: General principles and empirical evidence, *The Knowledge Engineering Review* 10 (1995) 43–62. doi:10.1017/S0269888900007256.
- [5] W. Chang, E. Chen, B. Mellers, P. Tetlock, Developing expert political judgment: The impact of training and practice on judgmental accuracy in geopolitical forecasting tournaments, *Judgment and Decision Making* 11 (2016) 509–526. doi:10.1017/S1930297500004599.
- [6] B. Irwin, A. Rago, F. Toni, Argumentative forecasting, in: P. Faliszewski, V. Mascardi, C. Pelachaud, M. E. Taylor (Eds.), *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, 2022, pp. 1636–1638. doi:10.5555/3535850.3536060.
- [7] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995). doi:10.1016/0004-3702(94)00041-X.
- [8] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, S. Villata, Towards artificial argumentation, *AI Magazine* 38 (2017) 25–36.
- [9] P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), *Handbook of Formal Argumentation*, College Publications, 2018.
- [10] M. Lippi, P. Torroni, Argumentation mining: State of the art and emerging trends, *ACM Transactions on Internet Technology* 16 (2016). doi:10.1145/2850417.
- [11] P. E. Tetlock, B. A. Mellers, N. Rohrbaugh, E. Chen, Forecasting tournaments: Tools for increasing transparency and improving the quality of debate, *Current Directions in Psychological Science* 23 (2014). doi:10.1177/0963721414534257.
- [12] C. W. Karvetski, C. Meinel, D. T. Maxwell, Y. Lu, B. A. Mellers, P. E. Tetlock, What do forecasting rationales reveal about thinking patterns of top geopolitical forecasters?, *International Journal of Forecasting* 38 (2022) 688–704. doi:https://doi.org/10.1016/j.ijforecast.2021.09.003.

- [13] P. E. Tetlock, D. Gardner, *Superforecasting: The art and science of prediction*, Random House, 2016.
- [14] P. Baroni, A. Rago, F. Toni, From fine-grained properties to broad principles for gradual argumentation: A principled spectrum, *International Journal of Approximate Reasoning* 105 (2019) 252–286. doi:10.1016/j.ijar.2018.11.019.
- [15] C. Cayrol, M. C. Lagasquie-Schiex, On the Acceptability of Arguments in Bipolar Argumentation Frameworks, in: D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, L. Godo (Eds.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571, Berlin, Heidelberg, 2005, pp. 378–389. doi:10.1007/11518655_33, series Title: *Lecture Notes in Computer Science*.
- [16] A. Rago, F. Toni, M. Aurisicchio, P. Baroni, Discontinuity-free decision support with quantitative argumentation debates, in: *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning, KR’16*, AAAI Press, 2016, p. 63–72.
- [17] P. Baroni, M. Romano, F. Toni, M. Aurisicchio, G. Bertanza, Automatic evaluation of design alternatives with quantitative argumentation, *Argument & Computation* 6 (2015) 24–49. doi:10.1080/19462166.2014.1001791.
- [18] A. Rago, F. Toni, Quantitative Argumentation Debates with Votes for Opinion Polling, in: B. An, A. Bazzan, J. Leite, S. Villata, L. Van Der Torre (Eds.), *PRIMA 2017: Principles and Practice of Multi-Agent Systems*, volume 10621, Cham, 2017, pp. 369–385. doi:10.1007/978-3-319-69131-2_22, series Title: *Lecture Notes in Computer Science*.
- [19] G. W. BRIER, Verification of forecasts expressed in terms of probability, *Monthly Weather Review* 78 (1950) 1 – 3. doi:https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.
- [20] V. A. Satopää, J. Baron, D. P. Foster, B. A. Mellers, P. E. Tetlock, L. H. Ungar, Combining multiple probability predictions using a simple logit model, *International Journal of Forecasting* 30 (2014) 344–356. doi:https://doi.org/10.1016/j.ijforecast.2013.09.009.
- [21] R. S. Nickerson, Confirmation bias: A ubiquitous phenomenon in many guises, *Review of General Psychology* 2 (1998) 175 – 220.
- [22] A. Rago, H. Li, F. Toni, Interactive explanations by conflict resolution via argumentative exchanges, *CoRR* abs/2303.15022 (2023). doi:10.48550/arXiv.2303.15022. arXiv:2303.15022.
- [23] A. Vassiliades, N. Bassiliades, T. Patkos, Argumentation and explainable artificial intelligence: a survey, *The Knowledge Engineering Review* 36 (2021) e5. doi:10.1017/S0269888921000011.
- [24] K. Cyraś, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative XAI: A survey, in: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 2021, pp. 4392–4399. doi:10.24963/ijcai.2021/600.

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

Jean-Marie Lagniez¹, Emmanuel Lonca¹, Jean-Guy Mailly² and Julien Rossit²

¹CRIL, Université d'Artois - CNRS

²Université Paris Cité, LIPADE

Abstract

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

Keywords

Benchmark generation, Graph generation, Abstract argumentation

1. Introduction

Graph-based models are widespread in many fields of Knowledge Representation and Reasoning, including abstract argumentation [1]. This appeals automated graphs generation approaches to provide challenging benchmarks that can put to the test practical tools developed within these various frameworks. The literature offers different methods to generate graphs, which exhibit different properties and various applicabilities to concrete problems and scenarios. In particular, one challenge consists in generating *structured* instances, i.e. random graphs which present interesting patterns that are relevant for some specific application. A well-known example of such a structured generation model is the Watts-Strogatz model [2], where the generated graphs have a *small world* property. Among the variety of graphs that have been studied, some recent works are interested in the generation of graphs with communities of nodes, i.e. parts of the graphs which are densely connected, but with fewer connections between different communities [3]. Such models include BTER [4] and Darwini [5], that propose to link nodes inside so-called affinity blocks, and then to add links between the nodes from different blocks. Being a model of choice to represent people communities [3], graphs with communities seem to be a good candidate to encode large debates, which could be the source of argumentative reasoning.

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ lagniez@cril.fr (J. Lagniez); lonca@cril.fr (E. Lonca); jean-guy.mailly@u-paris.fr (J. Mailly);

julien.rossit@u-paris.fr (J. Rossit)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

However, until recently, there was an important lack of practical approach for computing the solutions of argumentation problems. Although there were some algorithmic approaches proposed in the literature, few pieces of software were actually available for the community. This has changed (mainly) thanks to the organization of the First International Competition on Computational Models of Argumentation (ICCMA), in 2015. Since then, some solvers have been proposed, based either on original techniques dedicated to argumentation frameworks [6, 7, 8], or on translation into other frameworks which have already proven efficient computational benefits (e.g. Boolean satisfaction problem (SAT) [9, 10, 11]). The efforts of the community at the occasion of the various editions of ICCMA have seen a general increase of the quality of the computational approaches for argumentation, both with respect to the correctness of the approaches and their runtime efficiency. However, the lack of challenging and realistic benchmarks for argumentation is still an issue for the community. Using (community-based) graph generators was naturally quickly considered to fill this hole.

In this paper, we propose a new generation method for obtaining community-based graphs and we apply it to abstract argumentation. Our approach is based on three components: we first generate an *outer graph* which gives a global skeleton for the structure of the generated instance; then in each node of the outer graph, we generate an *inner graph* i.e. a community of nodes; and finally when two nodes of the outer graph are connected, we use a *linker* to add some relations between the corresponding inner graphs. We then show how our method can be applied to generate structured, challenging graphs for argumentation purpose. The added value of our approach compared to the previous ones lies in its ability to be generic and modular, since any of the three components can be easily replaced by other versions. In particular, the outer and inner graphs can be generated through classical generation models like Erdős-Rényi [12], Watts-Strogatz [2] or Barabási-Albert [13], but any other model could be plugged instead (including BTER and Darwini graphs themselves). Our contribution includes a documented, open-source graph generator following this inner/outer template. This application has been made to be easily used by any user, but also to be convenient for developers who want to add new features like graph generators, linkers or output formats.

The paper is organized as follows. We first give some background on abstract argumentation in Section 2, and we introduce the inner/outer model in Section 3. Section 4 presents some related works. Necessary and relevant features of our framework are presented in Section 5, followed by some experiments in Section 6. Finally, Section 7 draws some conclusions and highlights avenues for future work.

2. Background on Abstract Argumentation

An *abstract argumentation framework* (AF) [1] is a directed graph $\mathcal{F} = \langle A, R \rangle$ where A is a set of *arguments* and $R \subseteq A \times A$ is the *attack relation* between arguments. We say that an argument a *attacks* an argument b if $(a, b) \in R$. This is generalized to sets of arguments: S *attacks* b (resp. S') if there is some $a \in S$ which attacks b (resp. some $b \in S'$). A set S *defends* an argument a if for any b attacking a , there is a $c \in S$ attacking b .

Acceptability of arguments is usually evaluated thanks to the notion of extensions, i.e. sets of collectively acceptable arguments. Various semantics exist for defining extension [1]. Formally,

a semantics is a function $\sigma : \mathcal{F} = \langle A, R \rangle \mapsto \mathcal{E} \subseteq 2^A$. We focus on the semantics *cf*, *ad*, *co*, *pr*, *stb* and *gr*, standing respectively for *conflict-free*, *admissible*, *complete*, *preferred*, *stable* and *grounded*. Given an AF $\mathcal{F} = \langle A, R \rangle$, and a set of argument $S \subseteq A$, $S \in \text{cf}(\mathcal{F})$ iff $\forall a, b \in S$, $(a, b) \notin R$; $S \in \text{ad}(\mathcal{F})$ iff $S \in \text{cf}(\mathcal{F})$ and S defends all its elements; $S \in \text{co}(\mathcal{F})$ iff $S \in \text{ad}(\mathcal{F})$ and S does not defend any argument in $A \setminus S$; $S \in \text{pr}(\mathcal{F})$ if S is a $\subseteq$ -maximal element of $\text{ad}(\mathcal{F})$; $S \in \text{stb}(\mathcal{F})$ iff $S \in \text{cf}(\mathcal{F})$ and S attacks all the arguments in $A \setminus S$; $S \in \text{gr}(\mathcal{F})$ iff S is the $\subseteq$ -minimal element of $\text{co}(\mathcal{F})$. See e.g. [1] for more details about these semantics as well as other semantics defined in the literature. Let us illustrate the complete, preferred, stable and grounded semantics with the following example:

Example 1. The extensions for *co*, *pr*, *stb* and *gr* of $\mathcal{F} = \langle A, R \rangle$ from Figure 1 are $\text{co}(\mathcal{F}) = \{\emptyset, \{a_1\}, \{a_2, a_4\}\}$, $\text{pr}(\mathcal{F}) = \{\{a_1\}, \{a_2, a_4\}\}$, $\text{stb}(\mathcal{F}) = \{\{a_2, a_4\}\}$ and $\text{gr}(\mathcal{F}) = \{\emptyset\}$.

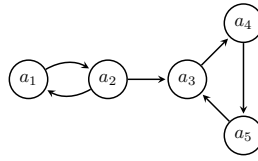


Figure 1: The AF $\mathcal{F}$

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

3. The Inner/outer Model

We propose a new approach for generating graphs that considers underlying graph structures. More precisely, an *outer graph* $G_{\mathcal{G}^O}$ that will be used as a skeleton for the instance is first constructed from a graph generator $\mathcal{G}^O$. Then, each node of this graph is associated with a fresh *inner graph* (fresh in the sense where nodes of each inner graph are disjoint) built by another generator $\mathcal{G}^I$. In order to link inner graphs together, we successively consider each inner graph G_n rooted to a node n of $G_{\mathcal{G}^O}$ and add edges between it and the inner graphs $G_{n'}$ rooted to a node n' when an edge exists in the outer graph between n and n' . The final graph is then the set of inner graphs together with the added edges. Interestingly, such generation process can handle both directed and undirected graphs (with the constraint that the inner graphs generator and the added edges involve edges of the same kind¹). However, here we focus on the directed case, since the goal is to generate argumentation frameworks. Formally, the function in charge of linking inner graphs together in the directed case is defined as follows:

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

¹Note that the outer graph may be non-directed even when the final graph is directed: the presence of directed edges may represent a “hierarchical” relation between the communities, while non-directed edges at this level mean that the communities are, in a way, equivalent.

Algorithm 1 Inner/outer graph generation

Input: an outer graph generator $\mathcal{G}^{\mathcal{O}}$, an inner graph generator $\mathcal{G}^{\mathcal{I}}$ and a linker $\mathcal{L}$

Output: an inner/outer graph

```
1:  $G_{\mathcal{G}^{\mathcal{O}}} \leftarrow \langle N, E \rangle$  a  $\mathcal{G}^{\mathcal{O}}$ -generated graph
2: for  $n \in N$  do
3:    $G_n \leftarrow \langle N_n, E_n \rangle$  a  $\mathcal{G}^{\mathcal{I}}$ -generated graph
4: end for
5:  $L = \emptyset$ 
6: for  $(n, n') \in E$  do
7:    $L \leftarrow L \cup \mathcal{L}(G_n, G_{n'})$ 
8: end for
9: return  $\langle (\bigcup_{n \in N} N_n), (\bigcup_{n \in N} E_n) \cup L \rangle$ 
```

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

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

Example 2. *Let us illustrate the generation algorithm with $\mathcal{G}^{\mathcal{O}} = \mathcal{T}$, $\mathcal{G}^{\mathcal{I}} = \mathcal{ER}$, and $\mathcal{L}$ a function which returns a random set of edges between two graphs. An example of generation process is given at Figure 2. Figure 2a shows the outer graph, which is thus a (non-directed) balanced binary tree. Then, in each node of the tree, an inner graph is generated thanks to the Erdős-Rényi model (Figure 2b). Figure 2c shows the addition of edges between the inner graphs thanks to the linker. And finally, the resulting graph is shown at Figure 2d.*

4. Related Works

The next sections presents the application we developed to generate inner/outer graphs and its application to generate AF benchmarks. There already exists tools for generating AFs from random graph generators. But, from the best of our knowledge, these tools do not modify the underlying graph generated by these models. In [15], the authors propose the C++ framework AFBenchGen. It is an AF generator based on the Erdős-Rényi model ($\mathcal{ER}$). In [16], the same authors proposed an extension of AFBenchGen, called AFBenchGen2 which is written in Java,

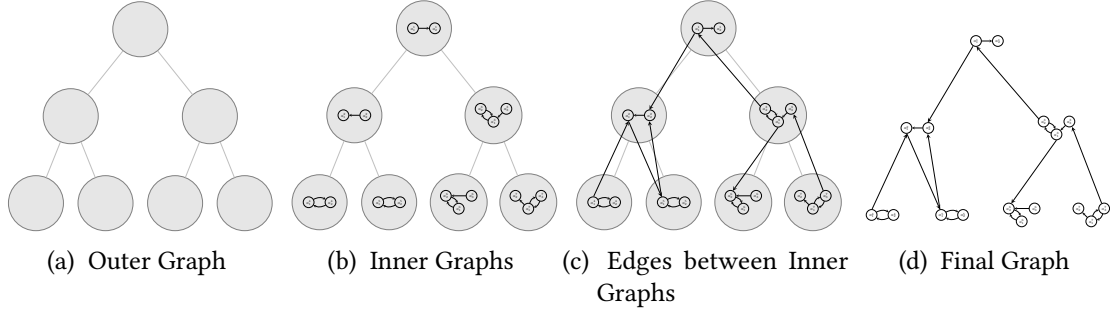


Figure 2: Generation process.

that also consider two additional random graph generator models, which are the Watts-Strogatz ($\mathcal{WS}$) and Barabási-Albert ($\mathcal{BA}$) models. For these two generators the random graphs are used as such. Our tool is much more general than the AFBenchGen family of AFs generators. Indeed, by considering the simple graph consisting in one node as outer graph, it is possible to have the exactly same behaviour.

In [17], we introduced a new method for generating challenging benchmarks for the ICCMA’21 competition. This generator is the fundamental basis of our tool. More precisely, we have proposed three variants of our generator $\langle \mathcal{G}^O, \mathcal{G}_i^I, \mathcal{L} \rangle$, with $i \in \{1, 2, 3\}$, defined as follows. In our case $\mathcal{G}^O = \mathcal{T}$, meaning that the underlying graph is actually a perfectly balanced d -tree of height h , where d and h are fixed and provided as parameters. The only difference between the three variants is the inner graphs generator: $\mathcal{G}_1^I = \mathcal{ER}$, $\mathcal{G}_2^I = \mathcal{BA}$, while $\mathcal{G}_3^I$ is a random pick of either $\mathcal{ER}$ or $\mathcal{BA}$, which means that in the first case all the local graphs are Erdős-Rényi graphs, in the second case they are all Barabási-Albert graphs, and in the last case they can be either of them with a probability 0.5.

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

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

5. The crustio_g2io graph generator

We built a command line application called crustio_g2io, dedicated to the generation of inner/outer graphs. It is made available under the terms of the GNU GPLv3 on Github account of the Centre

de Recherche en Informatique de Lens.² We took advantage of the Rust programming language to provide an efficient, memory-safe application, even in parallel context. In addition, Rust allows `crusti_g2io` to be both an application and a library (the project is mainly a Rust library with additional code to create the application). Interestingly, Rust libraries can be turned into C libraries (static or dynamic) or be linked with them. This makes `crusti_g2io` able to use any library that can be turned into a C library or to be used itself with any program that can load C libraries, allowing for example Go and Python bindings.

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

```
me@PC:~/crusti_g2io generate-directed -o tree/10 -i er/100,0.5 -l min_incoming -x out.apx -f apx
! [INFO ] [2023-03-03 10:54:39] crusti_g2io 0.1.0
[...]
! [INFO ] [2023-03-03 10:54:39] generated a graph with 1000 nodes and 24882 edges
! [INFO ] [2023-03-03 10:54:39] exiting successfully after 45.6625ms
```

Figure 3: Example on invocation of `crusti_g2io`.

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

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

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

²https://github.com/crillab/crusti_g2io

one links nodes with the least incoming edges, and the last one links the nodes with index 0 — which can have some meaning, in particular if a graph is initialized with a special value like in the Barabási-Albert model. Finally, The Graphviz DOT and GraphML formats are available, just like the abstract argumentation related format APX and DIMACS (from ICCMA 2023).

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

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

6. Using `crusti_g2io` to generate challenging abstract argumentation problems

Now, we use `crusti_g2io` to generate structured AF instances. The goal is to generate overall instances composed of multiple communities. In addition, we want to generate instances with a large amount of small communities, but also instances with less communities of a greater size. To achieve this, we aim at drawing the frontier between hard and too-hard instances for a set of community sizes, densities and counts. In order to evaluate the difficulty induced by the generated argumentation graphs, we chose to compute extensions (putting acceptance queries aside) to consider the whole graphs instead of problems that could be related to a reduced area

of the graph. We arbitrary selected a problem of the first level of the polynomial hierarchy (SE-ST: compute an extension for the stable semantics) and one of the second level (SE-PR: compute an extension for the preferred semantics). For both tracks, we used the solvers that got the best results at the ICCMA'21 competition, namely A-Folio-DPDB³ for the SE-ST track and μ -Toksia [11] for the SE-PR track. As A-Folio-DPDB delegates the SE-ST problems to the μ -Toksia solver submitted at ICCMA'19, we finally used μ -Toksia (2019) for SE-ST problems. We chose to build communities of Erdős-Rényi graphs, since those graphs were already used to generate AFs and can be naturally generated as directed graphs. Communities were linked following a tree template (like ICCMA'21 instances). The linker processes in a way inspired by the $\mathcal{ER}$ generator: each possible edge from the source graph to the target graph is added with probability 0.2.

In the first part of our experiments, we sought which sizes of communities are small enough to be part of our graphs. We used `crusti_g2io` to generate single Erdős-Rényi graphs (by asking for an outer graph composed of a single node) with different number of nodes (from 100 to 1000) and probability for each edge to appear (0.1, 0.2 and 0.5). For each setting, we generated 10 different graphs by feeding the app with random seeds from 0 to 9; the computation times are averages of these 10 values, and a timeout of at least one makes the average be also timeout. We run experiments on machines equipped with Intel Xeon E5-2637 v4 processors and 128GB of RAM, and the timeout was fixed to 600s, as in ICCMA'21. Table 1 shows some experimental results.

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

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

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

For this second experiment session, we considered Erdős-Rényi with nodes between 100 and 500 with the same three probability settings. We assumed the multiplicity of the communities

³https://github.com/gorczyca/dp_on_dbs/tree/competition

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

Table 1

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

would make the instances very hard for the 0.5 probability for more than 500 nodes per community. We considered (directed) outer tree heights from 3 to 9, making the outer graphs contain from 7 to 511 nodes. For each setting, 10 instances were generated with random seeds going from 0 to 9. We used the same machines and timeout as before. Figures 4 and 5 report the interesting parts of these new results. The plots on Figure 4 correspond to the results for the SE-ST track, while Figure 5 reports the results for SE-PR. For each figure, the three subfigures are each associated with a density setting (0.1, 0.2 and 0.5). For each subfigure, the average computation time is given on the y-axis, while the x-axis gives the number of communities; the lines give the different community sizes.

We first focus on the SE-ST results, given by the plots at Figures 4a, 4b and 4c. Concerning the results of μ -Toksia 2021 for the 0.1 probability setting (Figure 4a), we can observe that the problems are too easy when the number of nodes per community is lower than 200 (all solved in few seconds even for 511 communities) and too hard when it is above this value (such problems cannot be solved when there are more than 31 communities). Thus, this setting does not allow us to draw a clear frontier between the hard and the too-hard instances. This is also the case for the 0.5 probability setting (Figure 4c) for which the instances are surprisingly very difficult even for low values of community sizes and community counts. This is not an unexpected result since as noted below, these instances have a special structure that might prevent μ -Toksia to solve them. By the way, we discovered that μ -Toksia was not able to prove the absence of stable

extension in any community-based instance with this density. If such instances are included in our benchmarks, then μ -Toksia may suffer from this special kind of instances. Fortunately, the 0.2 case (Figure 4b) perfectly fits our needs of frontier as it shows multiple settings of community sizes and counts are solvable but difficult (hundreds of seconds required to solve) namely the sets of 511 communities of size 225, the sets of 255 communities of size 250 and the sets of 63 communities of size 275.

Now, we discuss the SE-PR results, given by the plots at Figures 5a, 5b and 5c. Just like for SE-ST, the 0.1 probability setting (Figure 5a) does not seem to be an interesting value for us since little changes in community sizes makes the difficulty a lot higher: see e.g. the difference between communities of 175 nodes — almost difficult instances when there are 511 of them — and 200 nodes — where instances are too difficult for 255 communities. Things are a little better for the 0.2 probability (Figure 5b) when considering communities of size between 225 and 300, but the real interesting setting in this case is the 0.5 probability (Figure 5c). In this case, we can find at least three cases of different community sizes for which hard instances exist: the sets of 511 communities of 175 nodes, the sets of 255 communities of 300 nodes and the sets of 127 communities of 500 nodes.

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

7. Conclusion

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

Several avenues for future work can be highlighted. Regarding the tool, a natural development

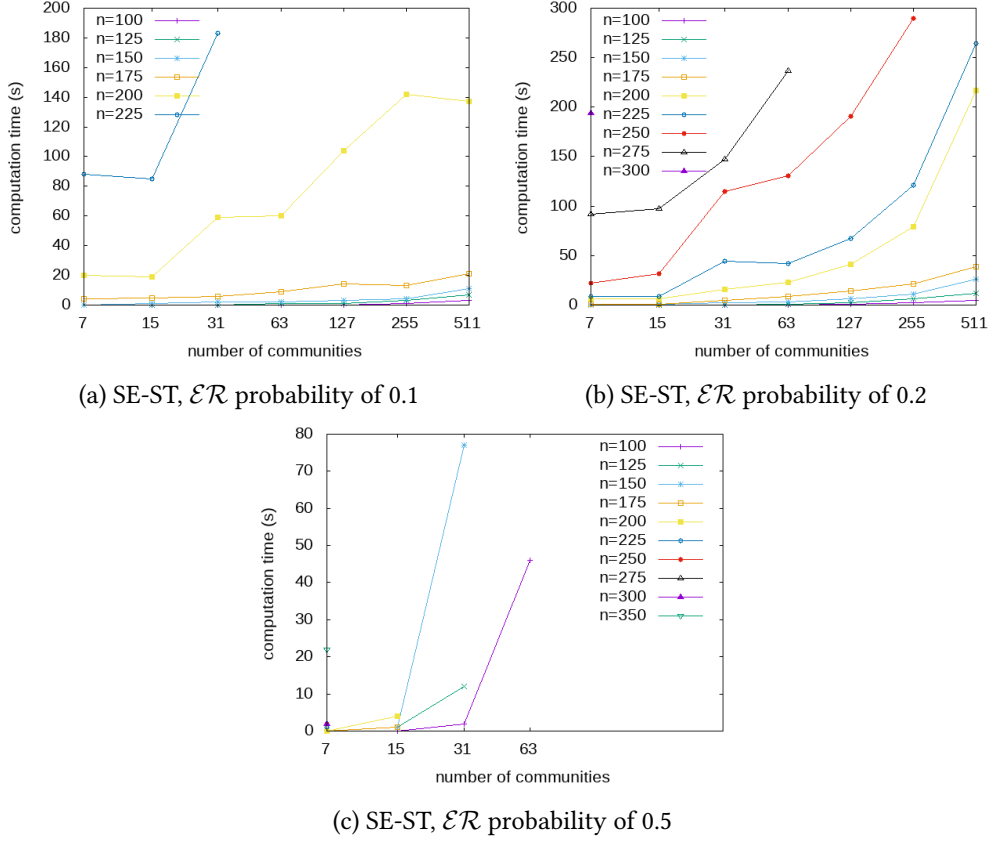
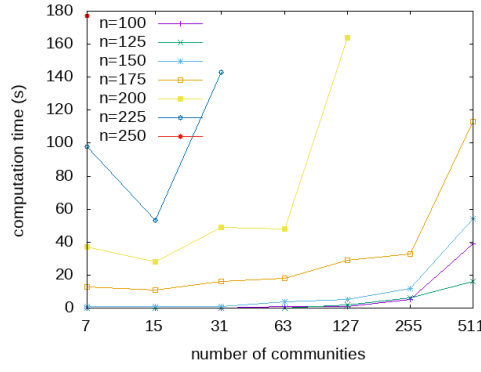


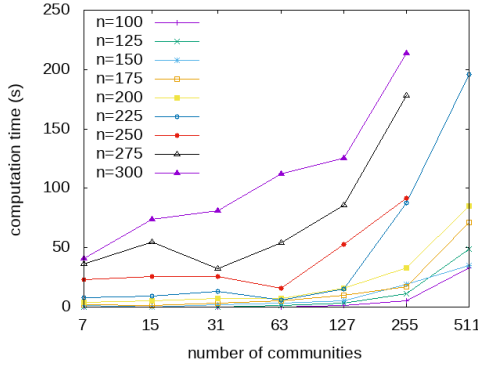
Figure 4: CPU time (in seconds) required by μ -Toksia 2019 to compute a single stable extension for community graphs of different community sizes and different community count. CPU times are an average of 10 values.

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

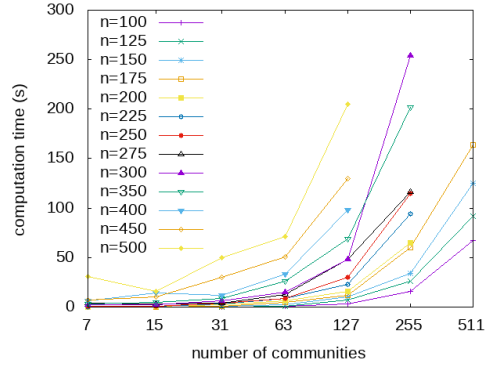
Regarding the issue of AF generation, we can improve the relevance of the tool by incorporating linkers which make sense in the context of abstract argumentation frameworks (for instance, we could add edges concerning in priority arguments which are skeptically accepted w.r.t. some given semantics). Another interesting future work consists in proposing generation models for more complex argumentation frameworks, which would require e.g. graphs with different kinds of edges or arguments (to incorporate supports [18] or incompleteness [19]) or graphs with weights associated with edges [20] or arguments [21].



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



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



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

Figure 5: CPU time (in seconds) required by μ -Toksia 2021 to compute a single preferred extension for community graphs of different community sizes and different community count. CPU times are an average of 10 values.

Acknowledgements

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

References

- [1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.* 77 (1995) 321–358.
- [2] D. Watts, S. Strogatz, Collective dynamics of “small-world” networks, *Nature* 393 (1998) 440–442.
- [3] M. Girvan, M. Newman, Community structure in social and biological networks, *Proc. of the NAS of the USA* 99 (2002) 7821–7826.

- [4] T. Kolda, A. Pinar, T. Plantenga, C. Seshadhri, A scalable generative graph model with community structure, *SIAM J. Sci. Comput.* 36 (2014).
- [5] S. Edunov, D. Logothetis, C. Wang, A. Ching, M. Kabiljo, Generating synthetic social graphs with darwini, in: *Proc. of ICDCS*, 2018, pp. 567–577.
- [6] N. Geilen, M. Thimm, Heureka: A general heuristic backtracking solver for abstract argumentation, in: *Proc. of TAFA 2017*, 2017, pp. 143–149.
- [7] M. Heinrich, The matrixx solver for argumentation frameworks, *CoRR* abs/2109.14732 (2021).
- [8] L. Kinder, M. Thimm, B. Verheij, A labeling based backtracking solver for abstract argumentation, in: *Proc. of SAFA 2022*, 2022, pp. 111–123.
- [9] W. Dvorák, M. Järvisalo, J. P. Wallner, S. Woltran, Complexity-sensitive decision procedures for abstract argumentation, *Artif. Intell.* 206 (2014) 53–78.
- [10] J.-M. Lagniez, E. Lonca, J.-G. Mailly, Coquiaas: A constraint-based quick abstract argumentation solver, in: *Proc. of ICTAI 2015*, 2015, pp. 928–935.
- [11] A. Niskanen, M. Järvisalo, μ -toksia: An efficient abstract argumentation reasoner, in: *Proc. of KR 2020*, 2020, pp. 800–804.
- [12] P. Erdős, A. Rényi, On random graphs. I., *Publicationes Mathematicae* 6 (1959) 290–297.
- [13] A. Barabási, R. Albert, Emergence of scaling in random networks, *Science* 286 (1999) 509–512.
- [14] W. Dvorák, P. E. Dunne, Computational problems in formal argumentation and their complexity, in: *Handbook of Formal Argumentation*, College Publications, 2018, pp. 631–688.
- [15] F. Cerutti, M. Giacomin, M. Vallati, Generating challenging benchmark afs, in: *Proc. of COMMA 2014*, 2014.
- [16] F. Cerutti, M. Giacomin, M. Vallati, Generating structured argumentation frameworks: AFBenchGen2, in: *Proc. of COMMA 2016*, 2016.
- [17] J.-M. Lagniez, E. Lonca, J.-G. Mailly, J. Rossit, Design and results of ICCMA 2021, *CoRR* abs/2109.08884 (2021).
- [18] C. Cayrol, M.-C. Lagasquie-Schiex, Bipolarity in argumentation graphs: Towards a better understanding, *Int. J. Approx. Reason.* 54 (2013) 876–899.
- [19] J.-G. Mailly, Yes, no, maybe, I don’t know: Complexity and application of abstract argumentation with incomplete knowledge, *Argument Comput.* 13 (2022) 291–324.
- [20] P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, M. Wooldridge, Weighted argument systems: Basic definitions, algorithms, and complexity results, *Artif. Intell.* 175 (2011) 457–486.
- [21] J. Rossit, J.-G. Mailly, Y. Dimopoulos, P. Moraitis, United we stand: Accruals in strength-based argumentation, *Argument Comput.* 12 (2021) 87–113.

ADP : An Argumentation-based Decision Process Framework Applied to the Modal Shift Problem

Christopher Leturc¹, Flavien Balbo²

¹Inria, Université Côte d’Azur, CNRS, I3S, 06902 Valbonne, France

²Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F-42023 Saint-Étienne, France

Abstract

This article introduces an argumentation-based decision process framework specifically designed to model context-based decisions and its application to the challenge of promoting responsible modal choices in transportation. Despite the growing demand for sustainable transportation options, many urban travelers continue to rely heavily on private cars. We show that our argumentation model can be used to understand how the traveler context influences the transportation modal choice decisions of the travelers. To validate the efficacy of our framework, we deploy it within a simulator of multimodal transportation networks, utilizing formal argumentation to represent various behaviors. By examining the underlying reasons behind individuals’ car usage and investigating potential avenues for influencing their modal choices, we aim to contribute to the advancement of sustainable transportation solutions.

Keywords

Argumentation, decision model, multi-agent simulation, transportation modal shift

1. Introduction

The growth of cities is accompanied by an increasing transportation demand, resulting in heightened pollution and congestion. This is primarily attributed to travelers’ preference for using private vehicles over other modes of transportation. The shift from private vehicle mode to alternative modes such as collective or non-motorized modes has become a significant concern for transportation authorities. To discourage private vehicle usage, authorities have implemented low-emission zones (LEZ) as a new measure. LEZ restricts access to certain parts of the city exclusively to vehicles with low emissions.

However, defining the boundaries of these zones is a challenge as it requires striking a balance between travelers’ mobility needs and the traffic implications. Unfortunately, when the definition is solely based on traffic flow analysis, only the traffic consequences are taken into account. This approach is unfair as it places the burden solely on excluded travelers or those who can afford low-emission vehicles.

Neglecting the impact on travelers’ needs presents two risks. Firstly, there is a high likelihood of non-compliance, resulting in additional costs to enforce the rule. Secondly, there is a limited effect on modal shift, as most travelers simply adjust their car routes to avoid the LEZ. Finally,

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ christopher.leturc@inria.fr (C. Leturc); flavien.balbo@emse.fr (F. Balbo)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

to align with the users' needs, cities may introduce exceptions that make the rule unclear¹². The LEZ definition problem emphasizes the necessity of conducting a more comprehensive analysis of travelers' decision-making processes to grasp the rule's impact on travelers and, consequently, assess its effectiveness.

Agent-based simulations focus on individual decision-making processes, making them valuable tools for analyzing the modal shift problem [1, 2]. However, these works often limit the traveler's context to their activities and locations, while the decision process of the agents is primarily guided by a single criterion, such as price impact [1] or the utilization of shared autonomous vehicles [2]. As a result, the diverse contexts of travelers are not adequately considered.

Modal shifting is determined by a whole range of factors that are interrelated to a larger or smaller extent. For instance, [3] conducted a study involving 205 Australian university students to examine the relative importance and correlation between psychological and situational factors in predicting commuter transport mode choices. The study's findings include: (1) individuals' values influence their commuting behavior through their corresponding beliefs regarding the environmental impact of cars, (2) factors such as cost, time, and accessibility contribute to individuals' choices of commuting mode, and (3) both situational and psychological factors jointly influence pro-environmental behavior. For a comprehensive review of the modal choice concept, interested readers can refer to [4].

To address this complexity, this article aims to propose a framework that captures the various contexts within which travelers make their travel decisions.

In this article, we argue that formal argumentation, such as the Dung framework [5], can be used to represent the decision-making context of travelers. Arguments and attacks pertain to specific situations for the traveler and elucidate the support or refutation of a modal choice. In this sense, argumentation seems particularly relevant to represent complex multi-criteria decisions structures, in opposition to numerical functions or simple logical rules. An additional benefit of argumentation is its similarity to how we, humans, reason, as it has been suggested by Mercier and Sperber [6], which makes it easier to understand and use for humans. Furthermore, argumentation gives us an explicit justification about the decision while it is not necessarily the case for other AI techniques, especially the numeric-based reward functions.

The contributions of this article are:

- An argumentation framework to represent context-based decisions,
- An application of argumentation to the problematic of the modal shift.

This paper is organized as followed: Firstly, Section 2 proposes a state of the art on argumentation-based decisions frameworks. Secondly, Section 3 introduces the case study dedicated to represent urban travelers behaviors within a simulator of multimodal transportation networks. Thirdly, Section 4 presents the formal framework and recalls basics notions. Finally, Section 5 presents the first results of the proof-of-concepts and Section 6 proposes conclusion and perspective of future work.

¹https://ec.europa.eu/transport/themes/urban/studies_cs

²https://ec.europa.eu/transport/sites/default/files/uvar_final_report_august_28.pdf

2. Argumentation-based decisions systems

Argumentation has been identified as an effective tool for decision-making and decision-support systems, particularly in situations where the recommended decisions need to be explained [7, 8]. Multiple studies [9, 10, 11, 12] have investigated the introduction of argumentation capabilities in decision-making and emphasized the importance of presenting arguments in favor or against possible choices to the user of a decision-support system. For instance, argumentation has been applied to justify a multiple criteria decision or represent decisions taken by a group of agents as in vote systems [13]. In a context of computer simulations, argumentation has been applied into agent-based simulations to model the opinion of agents [14], or [15] considers a case study in which argumentation is used to assess and compare cultural options available to farmers. However these approaches in agent-based models do not consider argumentation to make agent taking decisions. In [16], they use argumentation to represent knowledge of agents and their reasoning about alternatives in an automata framework, named as Action-based Alternating Transition Systems (AATS) framework. Some approaches in the literature in decision-support systems used argumentation to justify an option w.r.t. a goal. In [15], they consider a case study in which argumentation is used to assess and compare cultural options available to farmers. In their approach a system is a set of variables X and a set of states which is an instantiation of each variable of X , as e.g. $X = X_{out} \cup X_{in}$, where X_{out} is the observation, and X_{in} is human control values. An argument is a triplet $Arg = (option, goal, justification)$ which is associated with an option, a goal and a justification.

In this article, we are interested in the model proposed in [17]. They proposed an abstract argumentation model that defines an argumentation-based decision framework as tuple (A, D, R, F_f, F_c) where A is a set of arguments, D is a set of decisions, R is an attack relation, F_f is a mapping (resp. F_c) between pros (resp. cons) arguments and their associated decisions. Their model has several advantages:

- The simplicity of the model for linking arguments and decisions without having to change the abstract structure of arguments
- It provides the possibility of extending it easily to other argumentation models like e.g. Value-based Argumentation Frameworks (VAF) [18]

3. The simulated environment

The proposal presented in this paper is evaluated using a multiagent simulator available in the *Plateforme Territoire*³. This simulator enables agent travelers to access multimodal shortest itineraries between their origin and destination and simulates their movement along the chosen itinerary at a speed corresponding to the selected transportation mode. Itineraries can be evaluated using pre-trip indicators that influence the itinerary choice of the traveler agents. These indicators can be based on factors such as distance or traffic-related aspects like noise, which depends on the number of vehicles along different parts of the itinerary. Additionally, the

³<https://territoire.emse.fr>

simulator calculates global traffic indicators for each transportation mode to assess the system, such as the number of late travelers.

Application. Each agent has to decide about one alternative which corresponds to choose a particular transportation network. In this simulator, we consider the following set of alternatives $\mathcal{Alts}$ and $\mathcal{N}$ be a set of agents :

D1 $p.t. := \text{"go by public transport"}$

D2 $bike := \text{"go by bike"}$

D3 $walk := \text{"go by foot"}$

D4 $car := \text{"go by car"}$

Each agent decides based on indicators. For each agent $i \in \mathcal{N}$, we consider the following indicators $\mathcal{Inds}$. We first define the indicators based on alternatives $\mathcal{Alts}$:

- $t : \mathcal{Alts} \rightarrow \mathbb{D}^+$ for a given alternative, it returns the duration for this alternative
- $d : \mathcal{Alts} \rightarrow \mathbb{D}^+$ for a given alternative, it returns the distance for this alternative
- $pol : \mathcal{Alts} \rightarrow \mathbb{D}^+$ for a given alternative, it returns the pollution rate associated with this alternative
- $noi : \mathcal{Alts} \rightarrow \mathbb{D}^+$ for a given alternative, it returns the noise generated by this alternative
- $cos : \mathcal{Alts} \rightarrow \mathbb{D}^+$ for a given alternative, it returns the cost of this alternative

Indicators based on the agent state:

- $em : \mathcal{N} \rightarrow \{\top, \perp\}$ is a function that represents if one agent has a medical emergency
- $isOld, isFemale, isReadyToModalShift : \mathcal{N} \rightarrow \{\top, \perp\}$ are functions that represent if one agent is old, is female, or is ready to modal shift⁴
- $hasCar, hasECar, hasBike : \mathcal{N} \rightarrow \{\top, \perp\}$ are functions that represent if one agent has a car, or has an electric car, or has a bike and are s.t. for each agent $i \in \mathcal{N}$, if $hasCar(i) = \perp$ then $hasECar(i) = \perp$

Indicators based on the state of the environment:

- $isHealthCrisis, isRushHour, isTheNight \in \{\top, \perp\}$ translate if there is a health crisis, if this is the rush hour, or if it is the night

We formally define the state space (based on previously defined indicators) such as :

$$SpInds = ((\mathbb{D}^+)^{\mathcal{Alts} \times \mathcal{N}})^4 \times (\{\top, \perp\})^{\mathcal{Alts} \times \mathcal{N}} \times (\{\top, \perp\})^{\mathcal{N}}^7 \times \{\top, \perp\}^3$$

In the sequel we consider the following notation :

$$\forall s \in SpInds, \forall ind \in \mathcal{Inds}, s_{[ind]} = ind$$

This last notation translates for all $s \in SpInds$, $s_{[t]} = t$ i.e. we return the part of the value of the component of vector that assigns the function which evaluates the duration of each alternative.

⁴For a sake of simplicity, we reduce to a small set of characteristics of the agent.

4. Argumentation-based framework for Decision Making

In this section we give the model of Amgoud and Prade [19] and their definitions of extensions w.r.t. their model. Secondly, we present our framework called ADP (Argumentation-based Decision Process) which extends their model. The main advantages of our framework are:

- The decision of an agent is contextualized w.r.t. the state thanks to its argumentation graph.
- The notion of arguments is abstract so that it can be easily extended to approaches that consider and explicit goals, or other argumentation approaches as e.g. logic-based argumentation [20].

4.1. Argumentation Framework for Decision Making

In order to map arguments to decisions, [17] extends the standard argumentation framework [5] to decisions. Arguments are mapped to supported decisions (i.e. pro arguments) and unsupported decisions (i.e. con arguments).

Definition 1. *An Argumentation Framework for Decision Making (AFDM) is a tuple $AFDM = (\mathcal{A}, \mathcal{R}, \mathcal{D}, \mathcal{F}_f, \mathcal{F}_c)$ such that:*

- $\mathcal{A}$ is a set of arguments
- $\mathcal{R}$ is a binary relation called attack relation
- $\mathcal{D}$ is a set of decisions (or actions)
- $\mathcal{F}_f : \mathcal{D} \rightarrow 2^{\mathcal{A}}$ is a function that assigns from $\mathcal{D}$ the set of pro arguments
- $\mathcal{F}_c : \mathcal{D} \rightarrow 2^{\mathcal{A}}$ is a function that assigns from $\mathcal{D}$ the set of con arguments

We note $ADF(\mathcal{A}, \mathcal{D}) = 2^{\mathcal{A}} \times 2^{\mathcal{A} \times \mathcal{A}} \times 2^{\mathcal{D}} \times (2^{\mathcal{A}})^{\mathcal{D}} \times (2^{\mathcal{A}})^{\mathcal{D}}$ the set of Argumentation-based Decision Framework based on a set of arguments $\mathcal{A}$ and a set of decisions $\mathcal{D}$ and consider:

$$\forall a \in \mathcal{A}, \mathcal{F}_f^{-1}(a) := \{d \in \mathcal{D} : a \in \mathcal{F}_f(d)\}, \mathcal{F}_c^{-1}(a) := \{d \in \mathcal{D} : a \in \mathcal{F}_c(d)\}$$

A semantics of argumentation frameworks is given by the notion of extensions [5]. Extensions characterize which arguments are considered as admissible in regard to the argumentation graph.

Definition 2 (Extensions). *Let $(\mathcal{A}, \mathcal{R}, \mathcal{D}, \mathcal{F}_f, \mathcal{F}_c)$ be an AFDM, $S \subseteq \mathcal{A}$ be a set of arguments.*

- S is an admissible extension iff S is conflict-free and all arguments $A \in S$ are acceptable w.r.t. S .
- S is a complete extension iff S is admissible and contains all acceptable arguments wrt S .
- S is a grounded extension iff S is a minimal complete extension wrt $\subseteq$ i.e. $\nexists S' \subseteq \mathcal{A}$ s. t. $S' \subsetneq S$, S' is a complete extension.
- S is a preferred extension iff S is a maximal admissible extension wrt $\subseteq$.
- S is a stable extension iff S is conflict-free, and $\forall A \in \mathcal{A} \setminus S, SRA$.
- S is an ideal extension iff S is a maximal admissible extension wrt $\subseteq$ that is included in all preferred extensions.

Let us notice that in the general case there is no consensus about which extension semantics to use. However as suggested in [21] some extensions can be considered as more preferable due to their uniqueness, e. g. the grounded or the ideal extension.

4.2. Argumentation-based Decision Process

The Argumentation-based Decision Process (ADP) framework incorporates argumentation theory to model the decision-making process, enabling the selection of decisions based on argument extensions derived from the argumentation graph.

Thus, the system is fully described by a set of agents $\mathcal{N}$, a set of states $\mathcal{S}$, a set of arguments $\mathcal{A}$ and a set of decisions $\mathcal{D}$ or actions that agents can do. In the sequel, we define an argumentation-based decision process for one agent $i \in \mathcal{N}$. A function σ defines for each state $s \in \mathcal{S}$, an instance of Amgoud and Prade's model i.e. a subset of possible decisions $\mathcal{D}' \subseteq \mathcal{D}$ that in s the agent can do, a subset of arguments $\mathcal{A}' \subseteq \mathcal{A}$ which are verified in s , a set of (un)supported decisions $\mathcal{F}'_f$ ($\mathcal{F}'_c$), and a set of attacks $\mathcal{R}'$ that are generated by the semantics of each argument in $\mathcal{A}'$. Then, a function ϵ defines the extension semantics which is used by the agent to compute her stationary politics π . Since there is no concensus about how to compute the "winning" set of arguments based on a particular extension semantics (and so the decision), we let abstract and consider rather an heuristic function h which defines the computation method to choose a decision based on an extension semantics. Thus, we assume that the politic of the agent (which is stationary) is fully defined by this heuristic function i.e. $\pi = h$. The stationary policy function ensures that a decision is chosen from the available options for each state in a consistent manner w.r.t. the set of admissible arguments.

Definition 3. An Argumentation-based Decision Process (ADP) is a tuple $ADP = (\mathcal{S}, \mathcal{A}, \mathcal{D}, \sigma, \epsilon, \pi)$ such that :

- $\mathcal{S}$ is a no-empty set of states
- $\mathcal{A}$ is a no-empty set of arguments
- $\mathcal{D}$ is a no-empty set of decisions (or actions)
- $\sigma : \mathcal{S} \rightarrow ADF(\mathcal{A}, \mathcal{D})$ is a function s.t.
 $\forall s \in \mathcal{S}, \sigma(s) = (\mathcal{A}', \mathcal{R}', \mathcal{D}', \mathcal{F}'_f, \mathcal{F}'_c)$ where :
 - $\mathcal{A}' \subseteq \mathcal{A}$ is a subset of arguments associated with state s and we note $\sigma_{[\mathcal{A}]}(s) = \mathcal{A}'$
 - $\mathcal{R}' \subseteq \mathcal{A}' \times \mathcal{A}'$ represents the set of attacks between arguments in $\mathcal{A}'$ and we note $\sigma_{[\mathcal{R}]}(s) = \mathcal{R}'$
 - $\mathcal{D}' \subseteq \mathcal{D}$ is a subset of decisions associated in a state s and we note $\sigma_{[\mathcal{D}]}(s) = \mathcal{D}'$
 - $\mathcal{F}'_f : \mathcal{D}' \rightarrow 2^{\mathcal{A}'}$ and we note $\sigma_{[\mathcal{F}_f]}(s) = \mathcal{F}'_f$ and $\sigma_{[\mathcal{F}_f^{-1}]}(s) = \mathcal{F}'_f{}^{-1}$ is a function that assigns a set of pro arguments for each decision in $\mathcal{D}'$ in a state s
 - $\mathcal{F}'_c : \mathcal{D}' \rightarrow 2^{\mathcal{A}'}$ and we note $\sigma_{[\mathcal{F}_c]}(s) = \mathcal{F}'_c$ and $\sigma_{[\mathcal{F}_c^{-1}]}(s) = \mathcal{F}'_c{}^{-1}$ is a function that assigns a set of con arguments for each decision in $\mathcal{D}'$
- $\epsilon : 2^{\mathcal{A} \times \mathcal{A}} \rightarrow 2^{2^{\mathcal{A}}}$ is a function that, from an argumentation graph, returns the set of extensions

- $\pi : \mathcal{S} \rightarrow \mathcal{D}$ is a stationary politic s.t. $\forall s \in \mathcal{S}, \pi(s) = h_s(\epsilon(\sigma_{[\mathcal{R}]}(s)))$ where $h_s : 2^{2^{\mathcal{A}}} \rightarrow \mathcal{D}$ is a function s.t. each chosen decision belongs to the set of extensions given by the AFDM:

$$\forall \mathcal{E} \subseteq 2^{\mathcal{A}}, h_s(\mathcal{E}) \in \{d \in \mathcal{D} : d \in \sigma_{[\mathcal{D}]}(s)\}$$

We now present how this model is applied in our proof-of-concept. The implemented model has 22 arguments, and for a sake of readability, we do not present all of these arguments in this article but only 4 of them.

Application. Let consider the following ADP = $(\mathcal{S}, \mathcal{A}, \mathcal{D}, \sigma, \epsilon, \pi)$ where $\mathcal{S} = \mathcal{SpInds}$ and $\mathcal{D} = \mathcal{Alts}$. We consider a set of arguments $\mathcal{A}$ and we assume in our study case that σ is such that for all $s \in \mathcal{S}$ and for each agent $i \in \mathcal{N}$:

- $A. hC := \text{"agent } i \text{ has no car"}; C. hB := \text{"agent } i \text{ has no bike"}$
Activation : $hC \in \sigma_{[\mathcal{A}]}(s)$ iff $s_{[hasCar]}(i) = \perp$ and $hB \in \sigma_{[\mathcal{A}]}(s)$ iff $s_{[hasBike]}(i) = \perp$
Pros : $\sigma_{[\mathcal{F}_f^{-1}]}(s)(hC) = \sigma_{[\mathcal{F}_f^{-1}]}(s)(hB) = \emptyset$
Cons : $\sigma_{[\mathcal{F}_c^{-1}]}(s)(hC) = \sigma_{[\mathcal{F}_c^{-1}]}(s)(hB) = \emptyset$
Attacks : $\forall (a, d) \in \{(hB, bike), (hC, car)\}, \{(a, x) : x \in \sigma_{[\mathcal{F}_f]}(s)(d)\} \subseteq \sigma_{[\mathcal{R}]}(s)$
Meaning : If she has no car (resp. no bike), then hC (resp. hB) is verified. It is not in favor, or against any alternative and attacks all verified arguments that supports one of these alternatives.
- $iAE := \text{"it is a medical emergency"}$
Activation : $iAE \in \sigma_{[\mathcal{A}]}(s)$ iff $s_{[em]}(i) = \top$
Pros : $\sigma_{[\mathcal{F}_f^{-1}]}(s)(iAE) = \{d \in \sigma_{[\mathcal{D}]}(s) : \neg \exists x \in \sigma_{[\mathcal{D}]}(s), s_{[t]}(i)(x) > s_{[t]}(i)(d)\}$
Cons : $\sigma_{[\mathcal{F}_c^{-1}]}(s)(iAE) = \sigma_{[\mathcal{D}]}(s) \setminus \sigma_{[\mathcal{F}_f^{-1}]}(s)(iAE)$
Attacks : $\{(iAE, x) : x \in \sigma_{[\mathcal{A}]}(s), \sigma_{[\mathcal{F}_f^{-1}]}(s)(x) \cap \sigma_{[\mathcal{F}_c^{-1}]}(s)(iAE) \neq \emptyset\} \subseteq \sigma_{[\mathcal{R}]}(s)$
Meaning : If there is a medical emergency, then iAE is verified. It is in favor of all alternatives that are the quickest, against the others and attacks all verified arguments that are in favor of at least one alternative that is not the quickest.
- $cRA := \text{"the car alternative crosses the regulated area"}$
Activation : $cRA \in \sigma_{[\mathcal{A}]}(s)$ iff $s_{[reg]}(i)(car) = \top, s_{[hasCar]}(i) = \top$ and $car \in \sigma_{[\mathcal{D}]}(s)$
Pros : $\sigma_{[\mathcal{F}_f^{-1}]}(s)(cRA) = \emptyset$
Cons : $\sigma_{[\mathcal{F}_c^{-1}]}(s)(cRA) = \{car\}$
Attacks : $\{(cRA, x) : x \in \sigma_{[\mathcal{F}_f]}(s)(car)\} \subseteq \sigma_{[\mathcal{R}]}(s)$
Meaning : cRA is verified when the alternative car crosses a regulated area while it should be forbidden. It attacks all arguments in favor of car.
- $iEx := \text{"it is too much expensive for agent } i \text{ when } s_{[cos]}(i)(d) > \theta_i^c(s)(d) \text{ with } d \in \mathcal{D} \text{ and } \theta_i^c(s) : \mathcal{D} \rightarrow \mathbb{D}^+ \text{ a function to set a threshold for what the agent considers as too much expensive"}$
Activation : $iEx \in \sigma_{[\mathcal{A}]}(s)$ iff $\exists d \in \sigma_{[\mathcal{D}]}(s)$ s.t. $cos(i)(s)(d) > \theta_i^c(s)(d)$
Pros : $\sigma_{[\mathcal{F}_f^{-1}]}(s)(iEx) = \{d \in \sigma_{[\mathcal{D}]}(s) : \theta_i^c(s)(d) > s_{[cos]}(i)(d)\}$
Cons : $\sigma_{[\mathcal{F}_c^{-1}]}(s)(iEx) = \{d \in \sigma_{[\mathcal{D}]}(s) : s_{[cos]}(i)(d) > \theta_i^c(s)(d)\}$
Attacks : $\{(iEx, x) : \exists d \in \sigma_{[\mathcal{F}_c^{-1}]}(s)(iEx), x \in \sigma_{[\mathcal{F}_f]}(s)(d)\} \subseteq \sigma_{[\mathcal{R}]}(s)$

Meaning : iEx is verified when at least one alternative is above the threshold of acceptability of agent i in regard to her budget. It attacks all arguments that supports one alternative which is not in the threshold of acceptability.

It is worth noting that attacks hold more weight in the decision-making process compared to simply considering a list of pros and cons. This is because a single attack can invalidate an argument and subsequently remove it from the extensions, thus impacting the agent's deliberation process for making choices.

Furthermore, this model could be easily extended to get a more realistic models by considering other arguments as e.g. "I'm relocating", "I'm the police", "I prefer biking", "There is no bicycle network", "There is no bus at this hour", etc.

5. Experimental results

In this section, we provide an application of the framework to simulate the modal shifting. We present the results obtained from our experiments, starting with an explanation of the various scenarios tested. Then, we provide an example of an argumentation graph generated for one agent. Finally, we demonstrate how decision-making based on argumentation can be utilized to evaluate the evolution of the overall transportation network.

5.1. Implemented scenarios

We aim to enhance the simulation of regulating a multimodal transportation networks, consisting of a car, bus, walk, and bicycle network, each with distinct characteristics such as average speed, environmental impact, financial cost, and noise level. Specifically, we focus on regulating the car network, which involves determining certain areas where cars are either allowed or prohibited from accessing. We present the parameters considered in two network configurations: namely, a configuration without regulated areas and another with regulated areas (LEZ). In both, we assume there are 5000 traveler agents, no health crisis, and it is rush hour. The objective of a traveler agent is to choose the most appropriate transportation mode. We also establish some thresholds based on these assumptions e.g. 90% have a bike and 100% a car, 30% of them have an electric car, 40% are ready to modal shift, 20% are senior, 10% are emergencies. The threshold for the acceptability w.r.t. time i.e. $\theta_i^t(s)(d)$ is set arbitrary as the following : if the travel time is greater than $1.2 \times t_{min}$ then it is unacceptable. The noise threshold $\theta_i^n(s)(d)$ should not exceed 1.0. The quantity of pollution, θ_i^p should not exceed 9.0, the cost becomes unacceptable when the cost is greater than $\theta_i^c(s)(d) = 0.2$ and the distance is unacceptable when it is greater than $\theta_i^l = 200.0$. We consider 6 scenarios:

- Without LEZ: the objective is to evaluate the impact of taking into account the context in the distribution of the travelers per networks considering a classical multimodal network.
 - s_1 : the traveler agents decide by considering the quickest alternative, i.e. a monocriteria decision process.
 - s_2 : the traveler agents decide with the proposed ADP.

- With LEZ: the objective is to evaluate if our model is efficient to understand the traveler decision process for the transportation modal shift.
 - s_3 : the traveler agents choose the quickest alternative. The comparison with s_1 will give information about the quality of the decision criteria to understand the consequences of the LEZ definition.
 - s_4 : the agents decide with an ADP. The comparison with s_2 will give information about the quality of the decision criteria to understand the consequences of the LEZ definition.
 - s_5 : the traveler agents decide with the ADP defined in s_4 but the distance acceptance has been reduced. The comparison with s_4 should show an increase of the modes with the shortest distance.
 - s_6 : the traveler agents decide with the ADP defined in s_5 but the pollution acceptance has been reduced. The comparison with s_5 should show an increase of the least polluting modes.

5.2. From the point of view of one agent

After running the scenario s_4 (ADP+LEZ), we illustrate an argumentation graph of one agent. Its characteristics are given by : $s_{[em]}(i) = \perp$, $s_{[isOld]}(i) = \perp$, $s_{[isRTMS]}(i) = \perp$, $s_{[hasCar]}(i) = \top$, $s_{[hasECar]}(i) = \perp$, $s_{[hasBike]}(i) = \top$.

Application. If $\mathcal{G}rd(\sigma_{[\mathcal{R}]}(s))$ is the computed grounded extension from the argumentation graph $\sigma_{[\mathcal{R}]}(s)$ where s represents the current state, then, we define the scoring function $scr_1 : \sigma_{[\mathcal{D}]}(s) \rightarrow \mathbb{R}$ such that for all decisions $d \in \sigma_{[\mathcal{D}]}(s)$:

$$\begin{aligned}
 pros(d) &= |\mathcal{G}rd(\sigma_{[\mathcal{R}]}(s)) \cap \sigma_{[\mathcal{F}_f]}(s)(d)| \\
 cons(d) &= |\mathcal{G}rd(\sigma_{[\mathcal{R}]}(s)) \cap \sigma_{[\mathcal{F}_c]}(s)(d)| \\
 scr(d) &= \begin{cases} \frac{pros(d)}{pros(d)+cons(d)} & \text{if } pros(d) + cons(d) \neq 0 \\ 0 & \text{otherwise} \end{cases}
 \end{aligned}$$

We compute the scores by considering the argumentation model given in Application 4.2. The result of the computed argumentation graph is depicted in Table 2. Then, by computing the grounded extension, we get: $\mathcal{G}rd(\sigma_{[\mathcal{R}]}(s)) = \{\{5, 17\}\}$. To deal with equalities, we assume the following order for agent i : $Bus > Bike > Walk > Car$. For a sake of simplicity we decided to set this order arbitrary. In this setting, the agent chooses the car alternative by avoiding the regulated area since $scr(Car) = 1$ while for other alternatives X , we have $scr(X) = 0$.

5.3. From a global perspective: how individual explicit decision processes may be manipulated to influence the system

We analyze the impact of a LEZ on agents considering scenarios s_1 , s_2 (ADP), s_3 (LEZ), s_4 (ADP+LEZ), s_5 (ADP+LEZ), s_6 (ADP+LEZ). The Table 1 presents the distribution of the travelers

between the transportation modes according to several scenarios. Here we compare the scenarios to illustrate three advantages of our approach. In each of them, travelers have to choose the transportation modes corresponding to their preferences with or without considering LEZ.

Scenario	Car	Bike	Walk	Bus
s_1	1	0	0	0
s_2	0,4944	0,304	0,163	0,0386
s_3	0,9502	0,005	0	0,0448
s_4	0,4408	0,297	0,163	0,0992
s_5	0,48	0,2678	0,1632	0,089
s_6	0,4454	0,2348	0,1742	0,1456

Table 1

Distribution of the travelers per transportation mode

Is our approach adapted to reproduce the diversity of travelers' modal choices? The scenario s_1 considers that travelers choose the quickest trip without other parameters while the travelers in s_2 decide according to the argumentation model presented in section 5.1. Here, the result show that with the only decision criteria based on time it is not possible in this example to have a real multimodal system, the car alternative is always the fastest transportation mode. The argumentation model is closer to the reality.

Is our approach efficient to understand the consequences of the network regulation on multimodal travelers' modal choices? The scenarios s_3 and s_4 are respectively similar to s_1 and s_2 expected that a LEZ is deployed. We can observe for both the same evolution with the transfer of around 5% of travelers from the car mode to the bus mode. The advantage of our proposal is that this transfer being based on a more realistic initial distribution we observe a multimodal traffic that is more balanced between modes.

Is our approach adapted to understand the consequences of traveler behaviors on the transportation system? The scenario s_5 is based on s_4 (LEZ deployed and ADP) with the modification of the distance constraint argument (θ_i^d) which is reduced to 0. It means that travelers prefer the shortest trips. We observe that there is a shift of travelers towards the car mode to reach a value close to that of s_2 . This illustrates that the decision is the result of a compromise, as the majority of travelers do not shift.

The scenario s_6 is based on s_5 with the reduction of the tolerance of the pollution (θ_i^p). This argument counter balances partially the one of the distance. The result is a percentage of travelers choosing the car mode that is similar to s_4 and an increase of the traveler choosing the bus mode. This last mode is a good compromise between the distance and pollution arguments.

6. Conclusion

In this article, we have presented an argumentation-based decision process framework for modeling the decision-making process of urban travelers. To demonstrate the feasibility of our framework, we have provided a proof-of-concept by instantiating the model with arguments that could be considered in a modal shifting.

It is important to highlight that our current model lacks of realistic data on real behavior of urban travelers, and we acknowledge that it represents a preliminary effort in this direction. In future research, we plan to enhance our framework by incorporating e.g. web-based data and considering more statistical data. We believe that it will provide more accurate insights into the decision-making process of agents and will provide more realistic results.

By combining our framework with comprehensive and up-to-date web data, we anticipate that our model will offer a deeper understanding of urban travelers' decision processes, ultimately leading to more effective strategies for promoting responsible modal choices in transportation.

7. Acknowledgement

We thank the 3IA Côte d'Azur ANR-19-P3IA-0002, the HyperAgents project ANR-19-CE23-0030 and the Acceler-AI project Projet-ANR-22-CE23-0028 for their support.

References

- [1] I. Kaddoura, G. Leich, K. Nagel, The impact of pricing and service area design on the modal shift towards demand responsive transit, *Procedia Computer Science* 170 (2020) 807–812.
- [2] J. Kamel, R. Vosooghi, J. Puchinger, F. Ksontini, G. Sirin, Exploring the impact of user preferences on shared autonomous vehicle modal split: A multi-agent simulation approach, *Transportation Research Procedia* 37 (2019) 115–122.
- [3] C. M. Collins, S. M. Chambers, Psychological and situational influences on commuter-transport-mode choice, *Environment and behavior* 37 (2005) 640–661.
- [4] A. De Witte, J. Hollevoet, F. Dobruszkes, M. Hubert, C. Macharis, Linking modal choice to motility: A comprehensive review, *Transportation Research Part A: Policy and Practice* 49 (2013) 329–341.
- [5] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial intelligence* 77 (1995) 321–357.
- [6] H. Mercier, D. Sperber, Why do humans reason? arguments for an argumentative theory., *Behavioral and brain sciences* 34 (2011) 57–74.
- [7] J. Fox, D. Glasspool, V. Patkar, M. Austin, L. Black, M. South, D. Robertson, C. Vincent, Delivering clinical decision support services: there is nothing as practical as a good theory, *Journal of biomedical informatics* 43 (2010) 831–843.
- [8] F. S. Nawwab, T. Bench-Capon, P. E. Dunne, Emotions in rational decision making, in: *International Workshop on Argumentation in Multi-Agent Systems*, 2009, pp. 273–291.

ArgID	Name	Attacks	F_f^{-1}	F_c^{-1}
1	AvoidAnyContactWhenThereIsHealthCrisis			
5	TimeDurationAcceptability	$\{\}$	$\{D4, D2\}$	$\{D1\}$
6	CanAlwaysWalk	$\{16, 9, 7, 6, 10, 12, 11\}$	$\{D4\}$	$\{D1, D3, D2\}$
7	CannotTravelLongDistance	$\{\}$	$\{D3\}$	$\{\}$
9	BikelsAnEfficientWayForTransportation	$\{15, 16, 9, 13, 6, 12, 11\}$	$\{D4, D1\}$	$\{D3, D2\}$
10	NoiseAcceptability	$\{\}$	$\{D2\}$	$\{\}$
11	PollutionThresholdIsNotAcceptable	$\{5, 1, 20, 16, 7, 10\}$	$\{D1, D3, D2\}$	$\{D4\}$
12	TooMuchExpensive	$\{5, 1, 20, 16, 7, 10\}$	$\{D3, D2\}$	$\{D4, D1\}$
13	IsRushHour	$\{\}$	$\{D3, D2\}$	$\{D4, D1\}$
15	BikingAndWalkingIsGoodForYourHealth	$\{\}$	$\{D3, D2\}$	$\{D4\}$
16	ItIsAResponsibleWayOfTransportation	$\{\}$	$\{D3, D2\}$	$\{\}$
17	NotReadyToModalShift	$\{1, 15, 20, 16, 9, 7, 13, 6, 10, 12, 11\}$	$\{D1, D3, D2\}$	$\{D4\}$
20	TheWeatherIsNotGood	$\{\}$	$\{\}$	$\{\}$
			$\{D4, D1\}$	$\{D3, D2\}$

Table 2
Argumentation graph for the agent i's decision

- [9] W. Ouerdane, N. Maudet, A. Tsoukias, Argumentation theory and decision aiding, *Trends in multiple criteria decision analysis* (2010) 177–208.
- [10] T. L. van der Weide, Arguing to motivate decisions, Ph.D. thesis, Utrecht University, 2011.
- [11] J. Müller, A. Hunter, An argumentation-based approach for decision making, in: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, volume 1, IEEE, 2012, pp. 564–571.
- [12] E. Ferretti, L. H. Tamargo, A. J. García, M. L. Errecalde, G. R. Simari, An approach to decision making based on dynamic argumentation systems, *Artificial Intelligence* 242 (2017) 107–131.
- [13] G. Marreiros, P. Novais, J. Machado, C. Ramos, J. Neves, An agent-based approach to group decision simulation using argumentation, in: *International MultiConference on Computer Science and Information Tecnology, Workshop Agent-Based Computing III (ABC 2006)*, Wisla, Poland, 2006, pp. 225–232.
- [14] P. Taillandier, N. Salliou, R. Thomopoulos, Coupling agent-based models and argumentation framework to simulate opinion dynamics: application to vegetarian diet diffusion, in: *Advances in Social Simulation: Proceedings of the 15th Social Simulation Conference: 23–27 September 2019*, Springer, 2021, pp. 341–353.
- [15] R. Thomopoulos, B. Moulin, L. Bedoussac, Combined argumentation and simulation to support decision, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2017, pp. 275–281.
- [16] K. Atkinson, T. Bench-Capon, States, goals and values: Revisiting practical reasoning, *Argument & Computation* 7 (2016) 135–154.
- [17] L. Amgoud, H. Prade, Using arguments for making and explaining decisions, *Artificial Intelligence* 173 (2009) 413–436.
- [18] T. J. M. Bench-Capon, Persuasion in practical argument using value-based argumentation frameworks, *Journal of Logic and Computation* 13(3) (2003) 429–448.
- [19] L. Amgoud, J.-F. Bonnefon, H. Prade, An argumentation-based approach to multiple criteria decision, in: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 2005, pp. 269–280.
- [20] P. Besnard, A. Hunter, Argumentation based on classical logic, in: *Argumentation in Artificial Intelligence*, 2009, pp. 133–152.
- [21] M. Caminada, Comparing two unique extension semantics for formal argumentation: ideal and eager, in: *Proceedings of the 19th Belgian-Dutch conference on artificial intelligence (BNAIC 2007)*, Utrecht University Press, 2007, pp. 81–87.

A Discussion of Challenges in Benchmark Generation for Abstract Argumentation

Isabelle Kuhlmann, Matthias Thimm

University of Hagen, Germany

Abstract

Abstract argumentation provides a formal framework for modeling and analyzing argumentative reasoning processes. As the field progresses, the need for benchmarks to evaluate and compare different algorithmic approaches becomes increasingly important. However, the process of generating suitable benchmarks for abstract argumentation is not without its challenges. This paper aims to explore the key challenges encountered in benchmark generation for abstract argumentation. In particular, we address the task of skeptical acceptability w.r.t. preferred semantics and describe a benchmark generator designed for this specific problem.

Keywords

Abstract argumentation, benchmarking, graph generator

1. Introduction

As a central aspect of human communication, the concept of argumentation has been adopted in the area of Artificial Intelligence in various forms. The principle of *abstract argumentation* [1], which focuses on the interplay between arguments in order to gain insights and reach conclusions, has become an established mechanism of non-monotonic reasoning. Naturally, an important issue in advancing the research in this field—in particular with regard to algorithmic solutions and applications—is the availability of benchmark data. However, generating suitable benchmarks for abstract argumentation presents several challenges that require careful consideration.

One notable initiative in advancing the evaluation of argumentation systems is the International Competition on Computational Models of Argumentation (ICCMA)¹. ICCMA serves as a platform for researchers and practitioners to showcase their systems and compare their performance on a common set of benchmarks. While ICCMA has significantly contributed to the evaluation of argumentation systems, the process of generating benchmarks for abstract argumentation remains a perpetual task. For instance, w.r.t. a set of ICCMA'17 benchmarks, it was recently pointed out that a majority of arguments that are skeptically accepted under preferred semantics (a task which is Π_2^P -complete [2]) is also accepted under grounded semantics

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ isabelle.kuhlmann@fernuni-hagen.de (I. Kuhlmann); matthias.thimm@fernuni-hagen.de (M. Thimm)

🌐 <http://mthimm.de/> (M. Thimm)

🆔 0000-0001-9636-122X (I. Kuhlmann); 0000-0002-8157-1053 (M. Thimm)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

📄 CEUR Workshop Proceedings (CEUR-WS.org)

¹<http://argumentationcompetition.org/index.html>

(which can be decided in polynomial time) [3]. Thus, in the majority of cases, it is sufficient to solve a less complex problem, which may distort the interpretation of experimental results. In this paper, we revisit this issue to highlight the impact of such properties on practical results. Additionally, we introduce in more detail the *KWT Benchmark Generator* [3], which is designed to circumvent the aforementioned problem. Further challenges arise in numerous respects, including the need for diversity in benchmark scenarios, scalability concerns, or appropriate evaluation metrics. Addressing these challenges is crucial to developing comprehensive benchmarking methodologies that accurately reflect the performance and capabilities of different argumentation systems.

2. Preliminaries

An (abstract) *argumentation framework* (AF) [1] is a pair $F = (\text{Arg}, R)$, with Arg being a set of arguments and $R \subseteq \text{Arg} \times \text{Arg}$ a relation between those arguments. An argument $a \in \text{Arg}$ *attacks* an argument $b \in \text{Arg}$ if $(a, b) \in R$. Moreover, we define the set of arguments attacking a given argument a as $a_F^- = \{b \mid (b, a) \in R\}$, and the set of arguments being attacked by a as $a_F^+ = \{b \mid (a, b) \in R\}$. In the same fashion we define E_F^- and E_F^+ for a set $E \subseteq \text{Arg}$. We call an argument $a \in \text{Arg}$ *defended* by a set of arguments $E \subseteq \text{Arg}$ if every argument $b \in \text{Arg}$ that attacks a is itself attacked by some argument $c \in E$, i.e., if $a_F^- \subseteq E_F^+$.

Further, a set $E \subseteq \text{Arg}$ is *conflict-free* if $E \cap E_F^+ = \emptyset$. If a set $E \subseteq \text{Arg}$ is conflict-free and each $a \in E$ is defended by E , we call E *admissible* (ad). We call sets of jointly acceptable arguments *extensions*, which can be defined under various semantics. The classical semantics, following the seminal work by Dung [1], are defined as follows:

- A set $E \subseteq \text{Arg}$ is *complete* (co) iff it is admissible, and if E defends $a \in \text{Arg}$ then $a \in E$.
- A set $E \subseteq \text{Arg}$ is *grounded* (gr) iff E is complete and $\subseteq$ -minimal.
- A set $E \subseteq \text{Arg}$ is *preferred* (pr) iff E is complete and $\subseteq$ -maximal.
- A set $E \subseteq \text{Arg}$ is *stable* (st) iff it is complete and $E \cup E_F^+ = \text{Arg}$.

In addition, we define a set $E \subseteq \text{Arg}$ to be an *ideal* (id) extension [4] if E is admissible, for every preferred extension E' , it holds that $E \subseteq E'$, and E is $\subseteq$ -maximal with these two properties. Note that the grounded and the ideal extension of an AF are each a uniquely defined, and that the former is always a subset of the latter.

Typical problems in the field of abstract argumentation involve the enumeration of one extension (or all extensions), or deciding whether a given argument is included in one extension (or all extensions) w.r.t. a given semantics. Let $\Sigma = \{\text{co}, \text{gr}, \text{pr}, \text{st}, \text{id}\}$. An argument is *credulously* accepted w.r.t. $\sigma \in \Sigma$ if it is included in at least one σ extension, and it is *skeptically* accepted if it is included in all σ extensions. We denote the computational problem of credulous acceptability regarding a semantics σ as DC_σ , and the problem of skeptical acceptability regarding σ as DS_σ .

3. Challenges in Benchmark Generation

In the following, we provide a brief overview of existing benchmarks (which were used in past editions of ICCMA) and subsequently discuss challenges that arise in the development of new benchmark generation techniques.

3.1. Existing Benchmarks

Existing benchmarks for abstract argumentation can be roughly categorized into the following groups:

- **Random graphs.** Some benchmark instances, e.g., those provided by *AFBenchGen* [5], comprise AFs generated using random graph generation algorithms. A related approach consists of connecting multiple random graphs to model communities of arguments [6].
- **Graphs tailored for argumentation.** Some benchmarks are aimed at specific abstract argumentation problems. Examples are *AdmBuster* [7], which is targeted to the problems DC_{pr} and DC_{gr} , and *SemBuster* [8], which is designed for problems regarding the semi-stable semantics [9]. Further, the *GroundedGenerator* produces AFs with a large grounded extension, the *StableGenerator* produces AFs with many stable extensions, and the *SccGenerator* produces graphs with many strongly-connected components [10]. Although these generators are not necessarily aimed at a specific argumentation problem, they were designed with abstract argumentation as the target application in mind, and allow for investigating certain solver properties (e.g., whether a solver exploits the fact that an argument accepted under grounded semantics—which is computationally easy to obtain—is also accepted under other semantics).
- **Translations from other domains.** AFs can be created by transforming existing problems or data sets from other domains. Examples include benchmarks from planning [11], assumption-based argumentation [12], mass transit data [13], and (inconsistent) knowledge bases expressed in the *Datalog*[±] language [14].

3.2. Challenges in the Generation of Novel Benchmarks

When creating a benchmark data set, the overall goal should be to obtain a *diverse* set of argumentation frameworks. Since the term “diversity” allows for multiple perspectives, we discuss some key aspects in the following.

Graph-Theoretical Features In order to ensure diversity in a graph-theoretical sense, benchmarks for abstract argumentation should encompass a wide range of graph properties and characteristics. This includes various properties, for example the node degree, the occurrence and number of (odd) cycles, variations in connectivity patterns, such as different levels of connectedness, etc. When creating new benchmarks, an analysis regarding such graph properties is valuable in order to check how newly generated AFs differ from existing benchmarks from a graph-theoretical perspective. New graph generators may also offer the possibility of parameterizing a number of graph features (which is already possible, to a degree, with most random graph generators). On the other hand, this may not be applicable in some scenarios (e.g., when dealing with real-world data).

Relation to Real-World Scenarios Creating benchmarks that reflect real-world argumentation scenarios can be challenging. Abstract argumentation frameworks might abstract from the complexity of real-world arguments and their relationships. Moreover, different domains, such as law, politics, healthcare, or ethics, have unique argumentation characteristics and

requirements. Incorporating domain-specific considerations in benchmark generation allows for more targeted evaluations and comparisons of argumentation systems within their intended domains.

Semantic Aspects Benchmarks should also be geared towards evaluating solution approaches to the different problems related to abstract argumentation. Some benchmarks are already designed for such purposes (such as *SemBuster*, which is aimed at problems related to the semi-stable semantics), however, there are still numerous problems that have not been specifically addressed yet. As an example, it was recently demonstrated that in most ICCMA’17 instances, a majority of skeptically accepted arguments w.r.t. *pr* were also included in the grounded and the ideal extension. Since the computational complexity of deciding DS_{pr} is Π_2^P -complete [2], but problems related to *id* are “only” Θ_2^P -complete, and the grounded extension can be computed in polynomial time. Hence, even though the task of deciding DS_{pr} is computationally complex, it can still be computed relatively efficiently, due to the occurrence of many “easy” cases.

4. KWT Benchmark Generator

In the previous section, we identified a number of challenges that occur in the generation of benchmarks for abstract argumentation. Since it is not reasonable to address all concerns within one graph generator, we focus on a specific semantic aspect as an example, namely the issue that solving the Π_2^P -complete problem DS_{pr} can often be bypassed by checking if the given argument is accepted w.r.t. *gr* or *id*. Note that another “easy case” regarding DS_{pr} occurs when arguments are attacked by some admissible set—such arguments are never skeptically accepted w.r.t. *pr*—and deciding this is a problem in NP. In [3], we briefly introduced a possible solution for this problem. In the following, we provide a more thorough description of our approach.

We developed the *KWT generator*, which takes as parameters

- num_{args} : the total number of arguments,
- num_{pa} : the number of arguments to be skeptically accepted under preferred semantics,
- num_{cred} : the number of arguments to be contained in at least one preferred extension,
- num_{pref} : the number of preferred extensions,
- num_{ideal} : the number of arguments in the ideal extension,

and further parameters that control the probability of attacks between different sets of arguments. More precisely, these parameters set the probabilities of arguments in the ideal extension to be attacked and to attack back, respectively, the probabilities of credulously accepted arguments to be attacked and to attack back, the probabilities of skeptically accepted arguments that are not contained in the ideal extension to be attacked and to attack back, and the probability of further random attacks between unaccepted arguments. Given these parameters, a random AF F is generated as follows:

1. The set Arg of num_{args} arguments is created and arguments are associated to sets S_{pa} (skeptically accepted arguments w.r.t. preferred semantics), S_{ideal} (arguments in the ideal extension), S_{cred} (arguments that are credulously accepted w.r.t. preferred semantics),

S_{unacc} (arguments that are not credulously accepted w.r.t. preferred semantics), such that $S_{ideal} \subseteq S_{pa} \subseteq S_{cred}$, $S_{cred} \cup S_{unacc} = \text{Arg}$, and the corresponding cardinalities are respected. Finally, sets $E_1, \dots, E_{num_{pref}}$ (the preferred extensions) are created by adding all arguments from S_{pa} and randomly drawn arguments from $S_{cred} \setminus S_{pa}$.

2. For every argument $a \in S_{ideal}$, random attackers from S_{unacc} are sampled. For each of these attackers b , another argument from S_{ideal} is sampled that attacks b . This ensures that the grounded extension will be empty and that the ideal extension is capable of defending itself (thus forming an admissible set).
3. For every argument $a \in S_{pa} \setminus S_{ideal}$, attacks from unaccepted arguments are sampled in a similar way (to ensure an empty grounded extension). Furthermore, every such argument a must be defended by each preferred extension. Thus, for each preferred extension E , some arguments are sampled to defend a .
4. For every preferred extension E and $a \in E \setminus S_{pa}$, attackers for a are sampled from $\text{Arg} \setminus E$ and corresponding defenders are defined within E .
5. Additional random attacks are added between arguments in S_{unacc} .
6. In order to avoid having stable extensions (which may also ease computation of arguments that are skeptically accepted under preferred semantics, since every stable extension is also preferred), we add another self-attacking argument and some attacks between this argument and arguments from S_{unacc} .

Note that due to the random approach of generating an argumentation graph, it may not necessarily be the case that the number of skeptically/credulously accepted arguments (w.r.t. preferred semantics) as well as the number of arguments in the ideal extension exactly match the given parameters. However, our experiments in [3] showed that it is indeed relatively hard to decide skeptical acceptance (w.r.t. preferred semantics) for most arguments in the resulting graph.

The graph generator² and an example demonstrating its usage³ we used can be found online.

5. Conclusion

Throughout this paper we discussed how the generation of new benchmarks for abstract argumentation problems can be challenging from multiple perspectives. Although existing benchmarks for abstract argumentation already provide valuable resources for evaluating and comparing different frameworks and algorithms, they may not adequately capture the challenges and requirements posed by recent developments in the field. Moreover, existing benchmarks may have limitations in terms of the problem space they cover (e.g. concerning characteristics of different graph properties).

Overall, we would like to highlight that new benchmarking techniques should yield AFs that are indeed novel in some regard, i.e., which differ from existing data—for instance, in terms of graph-theoretical properties, by addressing previously little considered semantic aspects, or by incorporating new real-world problems. Combining all of these different facets in one

²http://tweetyproject.org/r/?r=kwg_gen

³http://tweetyproject.org/r/?r=kwg_gen_ex

single generator is presumably rather difficult, however, considering them individually might already lead to new insights. As an example, we presented the KWT generator, which generates particularly challenging AFs for the task of deciding skeptical acceptability w.r.t. preferred semantics.

Acknowledgments

The research reported in this work was supported by Deutsche Forschungsgemeinschaft under grant 375588274.

References

- [1] P. M. Dung, On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games, *Artificial Intelligence* 77 (1995) 321–358.
- [2] P. E. Dunne, T. J. M. Bench-Capon, Coherence in finite argument systems, *Artificial Intelligence* 141 (2002) 187–203.
- [3] I. Kuhlmann, T. Wujek, M. Thimm, On the impact of data selection when applying machine learning in abstract argumentation, in: *Computational Models of Argument*, IOS Press, 2022, pp. 224–235.
- [4] P. M. Dung, P. Mancarella, F. Toni, Computing ideal sceptical argumentation, *Artificial Intelligence* 171 (2007) 642–674.
- [5] F. Cerutti, M. Giacomin, M. Vallati, Generating challenging benchmark afs., *COMMA* 14 (2014) 457–458.
- [6] J.-M. Lagniez, E. Lonca, J.-G. Mailly, J. Rossit, Design and results of iccma 2021, *arXiv preprint arXiv:2109.08884* (2021).
- [7] M. Caminada, M. Podlaszewski, Admbuster: a benchmark example for (strong) admissibility, 2017. The Second International Competition on Computational Models of Argumentation (ICCMA’17).
- [8] M. Caminada, B. Verheij, Sembuster: a benchmark example for semi-stable semantics, 2017.
- [9] M. Caminada, W. A. Carnielli, P. E. Dunne, Semi-stable semantics, *J. Log. Comput.* 22 (2012) 1207–1254.
- [10] M. Thimm, S. Villata, The first international competition on computational models of argumentation: Results and analysis, *Artificial Intelligence* 252 (2017) 267–294.
- [11] F. Cerutti, M. Giacomin, M. Vallati, Exploiting planning problems for generating challenging abstract argumentation frameworks, URL: <http://argumentationcompetition.org/2017/Planning2AF.pdf> (2017).
- [12] T. Lehtonen, J. P. Wallner, M. Järvisalo, Assumption-based argumentation translated to argumentation frameworks, URL: <http://argumentationcompetition.org/2017/ABA2AF.pdf> (2017).
- [13] M. Diller, Traffic networks become argumentation frameworks, URL: <http://argumentationcompetition.org/2017/Traffic.pdf> (2017).

- [14] B. Yun, M. Croitoru, Benchmark on logic-based argumentation framework with datalog $^{\pm}$, 2019.

Abstract Argumentation Applied to Fair Resources Allocation: A Preliminary Study

Jean-Guy Mailly¹

¹Université Paris Cité, LIPADE
F-75006 Paris, France

Abstract

In this paper, we discuss the application of abstract argumentation mechanisms to resources allocation. We show how such problems can be modeled as abstract argumentation frameworks, such that specific sets of arguments corresponds to interesting solutions of the problem. By interesting solutions, here we mean *Local Envy-Free* (LEF) allocations. Envy-freeness is an important notion of fairness in resources allocation, assuming than no agent should prefer the resource allocated to another agent. We focus on LEF, a generalized form of envy-freeness, and we show that LEF allocations corresponds to some specific sets of arguments in our argument-based modeling of the problem. This work in progress paves the way to richer connections between the various models of argumentation and resources allocation problems.

Keywords

Abstract argumentation, Resources allocation, Fairness

1. Introduction

Fairness issues are important in multi-agent scenarios, including resources allocation. Among the various fairness criteria, one of them is *envy-freeness*, *i.e.* the fact that no agent is envious of another agent. A generalized version of envy-freeness is *local envy-freeness* (LEF) [1], where agents are part of a social network, and the goal is to assign each agent one object such that none of them is envious of one of her neighbours in the network. In this work, we study a transformation from LEF problems to abstract argumentation [2]. We show that there is a correspondence between LEF allocations and some specific extensions of an argumentation framework built from the LEF problem at hand. The representation of LEF problems as argumentation problems offers several advantages. First of all, there are plenty of efficient tools for computing extensions of argumentation frameworks, which is not the case with LEF problems. Then, the argumentation process can offer intuitive (and visual) explanations of why an allocation is LEF, or why there is no such allocation. Finally, the vast literature on argumentation provides tools for other fairness problems, or for identifying specific allocations, *e.g.* weighted argumentation frameworks can provide means to obtain optimal allocations (w.r.t. agents utility functions or w.r.t. the Pareto criterion).

Section 2 provides some background notions on LEF allocations and abstract argumentation.

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ jean-guy.mailly@u-paris.fr (J. Mailly)

ORCID [0000-0001-9102-7329](https://orcid.org/0000-0001-9102-7329) (J. Mailly)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Section 3 discusses our transformation from resources allocation problems to abstract argumentation, and shows the relation between LEF allocations and specific extensions. Finally Section 4 concludes the paper by discussing some interesting questions for future work.

2. Background

2.1. Local Envy-Freeness

We focus on a resource allocation scenario (previously studied in [1]) where agents may know some other agents, and have preferences over the set of resources that must be allocated. The other hypotheses in our scenario are the fact that the number of agents is equal to the number of resources, and the fact that resources are indivisible goods. Formally,

Definition 1. A preference-based allocation problem (PRAP) is a tuple $\mathcal{PRAP} = \langle \mathcal{O}, \mathcal{N}, \succ, \mathcal{G} \rangle$:

- $\mathcal{O} = \{o_1, \dots, o_n\}$ is the set of objects,
- $\mathcal{N} = \{1, \dots, n\}$ is the set of agents,
- $\succ$ is a set of binary relations $\{\succ_i \mid i \in \mathcal{N}\}$ where $\succ_i$ is a linear order expressing the preferences of agent i over $\mathcal{O}$,
- $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ is an undirected graph representing the social network of agents.

We are interested in the problem of local envy-freeness, *i.e.* whether we can assign each agent i an object o_k s.t. i does not prefer the object assigned to one of her neighbors, formally $\forall j \in \mathcal{N}$ s.t. $\{i, j\} \in \mathcal{E}$, $o_l \not\succ_i o_k$, where o_k and o_l are (respectively) the object assigned to i and the object assigned to j .

To characterize formally this concept, we represent an allocation as a set of pairs $(i, o_k) \in \mathcal{N} \times \mathcal{O}$. Given such a pair $x = (i, o_k)$, we use $\text{Ag}(x)$ and $\text{Obj}(x)$ to denote respectively the agent i and the object o_k of this pair. An allocation γ is *valid* if for each $x, y \in \gamma$, if $x \neq y$ then $\text{Ag}(x) \neq \text{Ag}(y)$ and $\text{Obj}(x) \neq \text{Obj}(y)$, *i.e.* no agent receives several objects, and no object is assigned to several agents. A valid allocation can be *partial* if $|\gamma| < |\mathcal{O}|$, and *total* if $|\gamma| = |\mathcal{O}| = |\mathcal{N}|$.

Definition 2. Given $\mathcal{PRAP} = \langle \mathcal{O}, \mathcal{N}, \succ, \mathcal{G} \rangle$, an allocation is local envy-free (LEF) iff it is a total valid allocation such that $\forall i, j \in \mathcal{N}$ s.t. $\{i, j\} \in \mathcal{E}$, if $(i, o_k), (j, o_l) \in \gamma$ then $o_l \not\succ_i o_k$.

Example 1. Figure 1a describes $\mathcal{PRAP} = \langle \mathcal{O}, \mathcal{N}, \succ, \mathcal{G} \rangle$ where the agents are $\mathcal{N} = \{A, B, C\}$, the objects are $\mathcal{O} = \{1, 2, 3\}$ with 1 representing some money, 2 a motorbike, and 3 a car. The social network $\mathcal{G}$ is shown at the top of the Figure, and the agents preferences $\succ$ are given underneath. We can easily find a LEF allocation by giving each agent her preferred object.

Now consider $\mathcal{PRAP}_2$ given at Figure 1b, where this time all the agents know each other, and agent C's preferences are slightly modified as well. Assume there is a LEF allocation γ . Neither A nor C can receive the object 1 (because otherwise the one receiving another object would be envious of the one receiving the object 1). Thus we must have $(B, 1) \in \gamma$. But in this case, both agents A and C are envious of agent B. So there is no LEF allocation for $\mathcal{PRAP}_2$.

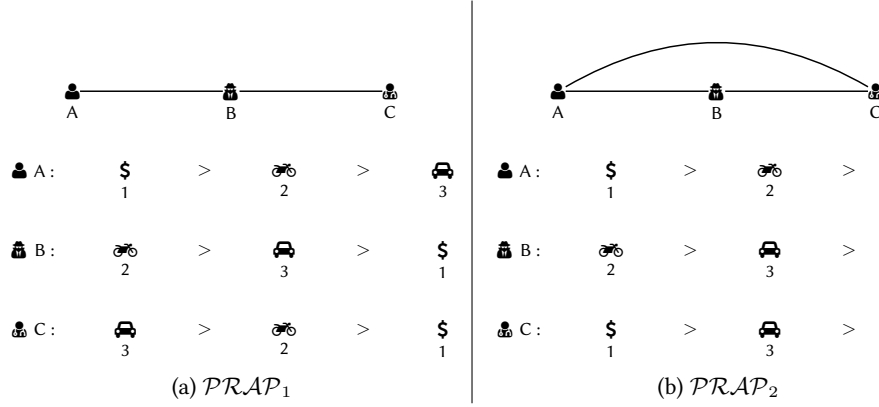


Figure 1: Social Networks and Preferences

2.2. Abstract Argumentation

Now let us recall basic notions of Dung's abstract argumentation [2].

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

Classical reasoning with AF uses the notion of *extensions*, i.e. sets of arguments that can be jointly accepted. Various semantics have been proposed to obtain the set of extensions of an AF. Formally, an extension-based semantics is a function σ such that for any AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, $\sigma(\mathcal{F}) \subseteq 2^{\mathcal{A}}$. In this paper, we only need the notions of conflict-free sets and stable extensions:

Definition 4. Given $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, $S \subseteq \mathcal{A}$ is conflict-free ($S \in \text{cf}(\mathcal{F})$) if $\forall a, b \in S, (a, b) \notin \mathcal{R}$. Then, S is a stable extension ($S \in \text{stb}(\mathcal{F})$) if $S \in \text{cf}(\mathcal{F})$ and $\forall b \in \mathcal{A} \setminus S, \exists a \in S$ s.t. $(a, b) \in \mathcal{R}$.

Among the various generalizations of Dung's argumentation framework, we are interested in *preference-based AFs* (PAFs) [3].

Definition 5. A preference-based argumentation framework is a tuple $\mathcal{P} = \langle \mathcal{A}, \mathcal{R}, \triangleright \rangle$ where $\langle \mathcal{A}, \mathcal{R} \rangle$ is an AF, and $\triangleright \subseteq \mathcal{A} \times \mathcal{A}$ is a preference relation over the set of arguments.

The preference relation is only assumed to be a pre-order, i.e. a reflexive and transitive binary relation. The main approach for reasoning with a PAF consists in reducing it into a standard AF by combining the attacks and the preferences into a *defeat* relation. Then, the extensions of the PAF for a semantics σ are the extensions of the defeat graph under the same semantics.

Definition 6. Given a PAF $\mathcal{P} = \langle \mathcal{A}, \mathcal{R}, \triangleright \rangle$, we define the defeat relation $\mathcal{D} = \{(x, y) \in \mathcal{R} \mid y \not\triangleright x\}$. Then, $\sigma(\mathcal{P}) = \sigma(\langle \mathcal{A}, \mathcal{D} \rangle)$.

3. Translation into Abstract Argumentation

In this section we show how to transform a PRAP into a PAF, such that there is a correspondence between LEF allocations and some sets of arguments, namely conflict-free sets of size $|\mathcal{N}|$, which are guaranteed to be stable extensions as well in our case.

Definition 7. Given $\mathcal{PRAP} = \langle \mathcal{O}, \mathcal{N}, \succ, \mathcal{G} \rangle$, we define the PAF $\mathcal{P}_{\text{LEF}} = \langle \mathcal{A}, \mathcal{R}, \triangleright \rangle$ with

- $\mathcal{A} = \{(i, o_j) \mid i \in \mathcal{N}, o_j \in \mathcal{O}\}$ (one argument $\simeq$ allocation of an object to an agent),
- $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$, with
 - $\mathcal{R}_1 = \{((i, o_j), (i, o_k)) \mid i \in \mathcal{N}, o_j, o_k \in \mathcal{O}\}$ (only one object per agent),
 - $\mathcal{R}_2 = \{((i, o_k), (j, o_k)) \mid i, j \in \mathcal{N}, o_k \in \mathcal{O}\}$ (only one agent per object),
 - $\mathcal{R}_3 = \{((i, o_k), (j, o_l)) \mid i, j \in \mathcal{N}, o_k, o_l \in \mathcal{O}, \{i, j\} \in \mathcal{E} \text{ and } o_l \succ_i o_k\}$ (envy),
- $\triangleright = \{((i, o_k), (i, o_j)) \mid i \in \mathcal{N}, o_k \succ_i o_j\}$ (preferences).

Obviously, any allocation γ corresponds to a set of arguments in $\mathcal{A}$. Observe also that the defeat relation will only remove some of the attacks in $\mathcal{R}_1$, namely, for each pair of arguments $x = (i, o_j)$ and $y = (i, o_k)$, the defeat relation will contain the defeat (x, y) if $o_j \succ_i o_k$, and the defeat (y, x) if $o_k \succ_i o_j$. The other attack relations $\mathcal{R}_2$ and $\mathcal{R}_3$ are not impacted by the preferences, so they are included in the defeat relation of this PAF.

Example 2. Let us consider again $\mathcal{PRAP}_1$ from Example 1. Figure 2 gives the defeat relation of the corresponding PAF $\mathcal{P}_{\text{LEF}}$. More precisely, Figure 2a gives the combination of $\mathcal{R}_1$ with the preferences, Figure 2b (resp. 2c) shows $\mathcal{R}_2$ (resp. $\mathcal{R}_3$). In Figure 2c, the red arrows correspond to the situations where agent A is envious (for instance, because she has received the object 2 while B has received the object 1), blue arrows are for agent B, and green arrows correspond to agent C.

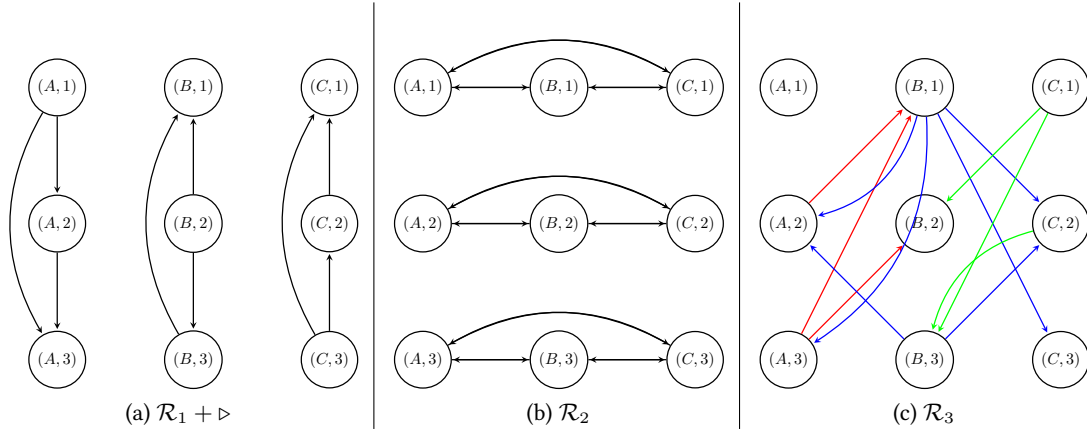


Figure 2: PAF Corresponding to a PRAP

The following lemmas will help us to prove the correspondance between LEF allocations and conflict-free sets (and stable extensions) of size $|\mathcal{N}|$.

Lemma 1. Given an allocation γ , if $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$ then γ is valid.

Proof. Assume γ is not valid. If $\exists x, y \in \gamma$ s.t. $\text{Ag}(x) = \text{Ag}(y)$, then $(x, y), (y, x) \in \mathcal{R}_1$, which implies that either (x, y) or (y, x) is in the defeat relation of $\mathcal{P}_{\text{LEF}}$, so $\gamma \notin \text{cf}(\mathcal{F}_{\text{LEF}})$. Similarly, if $\exists x, y \in \gamma$ s.t. $\text{Obj}(x) = \text{Obj}(y)$, then $(x, y) \in \mathcal{R}_2$, which implies $\gamma \notin \text{cf}(\mathcal{P}_{\text{LEF}})$. $\square$

Lemma 2. Given a valid allocation γ , if $\gamma \notin \text{cf}(\mathcal{P}_{\text{LEF}})$ then γ is not LEF.

Proof. Assume a valid allocation γ which is not conflict-free in $\mathcal{P}_{\text{LEF}}$. By construction, since γ is valid, there are no $x, y \in \gamma$ such that $(x, y) \in \mathcal{R}_1$ or $(x, y) \in \mathcal{R}_2$, so we deduce that $\exists x, y \in \gamma$ such that $(x, y) \in \mathcal{R}_3$ which implies that γ is not LEF. $\square$

Proposition 1. *Let γ be an allocation. γ is LEF iff $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$ and $|\gamma| = |\mathcal{N}|$.*

Proof. First assume that γ is a LEF allocation. From Lemma 2 we deduce that γ is conflict-free. By definition, since γ is LEF then γ is total. Hence the first part of the result.

Now assume that $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$ and $|\gamma| = |\mathcal{N}|$. From Lemma 1, we know that γ is valid. Then, since $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$, we can guarantee that there are no $x, y \in \gamma$ such that $(x, y) \in \mathcal{R}_3$. This means that, for any $i \in \mathcal{N}$ such that $(i, o_k) \in \gamma$, there is no $j \in \mathcal{N}$ such that $(j, o_l) \in \gamma$, $\{i, j\} \in \mathcal{E}$ and $o_l \succ_i o_k$. By definition, this means that γ is LEF. $\square$

Proposition 1 implies that LEF allocations can be easily computed thanks to a minor modification of a very classical approach for solving argumentation problems. Most of the efficient approaches for reasoning with abstract argumentation frameworks use SAT solvers. For the basic notion of conflict-freeness, it is enough to consider clauses which forbid to accept together arguments which are connected by an attack. We will use a MaxSAT version of this encoding [4], where the clauses usually corresponding to conflict-freeness will be hard clauses, and additional (unit) soft clauses will be added to ensure that the solver will return a maximal solution (in terms of cardinality). Formally,

Definition 8. *Given $\mathcal{PRAP} = \langle \mathcal{O}, \mathcal{N}, \succ, \mathcal{G} \rangle$, and $\mathcal{P}_{\text{LEF}} = \langle \mathcal{A}, \mathcal{R}, \triangleright \rangle$ the corresponding PAF. $\mathcal{D}$ denotes the defeat relation obtained from $\mathcal{R}$ and $\triangleright$. We define the following sets of hard clauses hc and soft clauses sc :*

$$hc = \{\neg x \vee \neg y \mid (x, y) \in \mathcal{D}\} \quad sc = \{(x, 1) \mid x \in \mathcal{A}\}$$

Given the sets of clauses hc and sc , a MaxSAT solver returns a conflict-free set of $\mathcal{P}_{\text{LEF}}$ of maximal cardinality. If this solution has a cardinality equal to $|\mathcal{N}|$, then it is a LEF allocation. Otherwise, there is no LEF allocation. Another possible approach consists in adding one cardinality constraint [5] $\sum_{x \in \mathcal{A}} x = |\mathcal{N}|$ to the set of hard clauses. In this case, if a LEF allocation exists then it will be provided by a SAT solver, otherwise the solver will answer UNSAT.

Notice that such an allocation also corresponds to a stable extension of cardinality $|\mathcal{N}|$.

Corollary 1. *Let γ be an allocation. γ is LEF iff $\gamma \in \text{stb}(\mathcal{P}_{\text{LEF}})$ and $|\gamma| = |\mathcal{N}|$.*

Proof. One side of the equivalence is obvious: if $\gamma \in \text{stb}(\mathcal{P}_{\text{LEF}})$, then $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$, so under the assumption that $|\gamma| = |\mathcal{N}|$, the result follows Proposition 1.

Now, assume that γ is a LEF allocation. From Proposition 1, we know that $\gamma \in \text{cf}(\mathcal{P}_{\text{LEF}})$ and $|\gamma| = |\mathcal{N}|$. For a given object o_k , there is an argument $a = (i, o_k) \in \gamma$, i.e. the object o_k has been assigned to agent i . By definition of $\mathcal{R}_2$, a defeats all the arguments of the form (j, o_k) for $j \neq i$. Since this is true for all the objects, any argument not in γ is defeated by some argument in γ , so $\gamma \in \text{stb}(\mathcal{P}_{\text{LEF}})$. $\square$

4. Discussion

Argumentation has already shown its interest for providing explanations to other problems, like e.g. scheduling [6] or case-based reasoning [7], so drawing connections between argumentation and resources allocation is a natural question.

The preliminary study of this connection allows us to envision deeper relations between both frameworks. For instance, it seems possible to assign numerical values to assignments (e.g. the preferred object o_k of agent i receives the value n , her second preferred object receives $n - 1$, etc.) in order to define a *Strength-based Argumentation Framework* (StrAF) [8] where the strength of an argument can intuitively correspond to the utility of allocating the object o_k to the agent i . Then using the semantics of StrAFs could induce interesting allocations. We plan to investigate this connection.

We are also interested in the methods allowing the explanation of arguments status in abstract argumentation (e.g. [9, 10]). They could allow to simply explain why the allocation of a specific object to an agent is necessary (or impossible). Also, the approach proposed by [11, 12] could provide interesting means to reduce the size of the argumentation graph, hence providing a better visual explanation of the (non-)existence of desirable allocation.

Another interesting way to go further in the study of argumentation applied to resources allocation consists in using the conflict-tolerant semantics of *Weighted Argumentation Frameworks* (WAFs) [13] in order to obtain optimal (non-LEF) allocations for instances which do not admit any LEF allocation.

These few ideas are only a small part of the possible connections between resources allocation and computational argumentation, and pave the way to a rich body of work that will allow to provide explainable solutions to fairness issues in resources allocation problem.

Acknowledgments

This work benefited from the support of the project AGGREEY ANR-22-CE23-0005 of the French National Research Agency (ANR).

References

- [1] A. Beynier, Y. Chevalere, L. Gourvès, A. Harutyunyan, J. Lesca, N. Maudet, A. Wilczynski, Local envy-freeness in house allocation problems, *Auton. Agents Multi Agent Syst.* 33 (2019) 591–627.
- [2] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.* 77 (1995) 321–358. URL: [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X). doi:10.1016/0004-3702(94)00041-X.
- [3] L. Amgoud, C. Cayrol, A reasoning model based on the production of acceptable arguments, *Ann. Math. Artif. Intell.* 34 (2002) 197–215. URL: <https://doi.org/10.1023/A:1014490210693>. doi:10.1023/A:1014490210693.
- [4] F. Bacchus, M. Jarvisalo, R. Martins, Maximum satisfiability, in: A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), *Handbook of Satisfiability - Second Edition*, volume 336

- of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2021, pp. 929–991. URL: <https://doi.org/10.3233/FAIA201008>. doi:10.3233/FAIA201008.
- [5] O. Roussel, V. M. Manquinho, Pseudo-boolean and cardinality constraints, in: A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2021, pp. 1087–1129. URL: <https://doi.org/10.3233/FAIA201012>. doi:10.3233/FAIA201012.
 - [6] K. Cyras, D. Letsios, R. Misener, F. Toni, Argumentation for explainable scheduling, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press, 2019, pp. 2752–2759. URL: <https://doi.org/10.1609/aaai.v33i01.33012752>. doi:10.1609/aaai.v33i01.33012752.
 - [7] G. Paulino-Passos, F. Toni, On monotonicity of dispute trees as explanations for case-based reasoning with abstract argumentation, in: K. Cyras, T. Kampik, O. Cocarascu, A. Rago (Eds.), 1st International Workshop on Argumentation for eXplainable AI co-located with 9th International Conference on Computational Models of Argument (COMMA 2022), Cardiff, Wales, September 12, 2022, volume 3209 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: <https://ceur-ws.org/Vol-3209/8465.pdf>.
 - [8] J. Rossit, J.-G. Mailly, Y. Dimopoulos, P. Moraitis, United we stand: Accruals in strength-based argumentation, *Argument Comput.* 12 (2021) 87–113. URL: <https://doi.org/10.3233/AAC-200904>. doi:10.3233/AAC-200904.
 - [9] A. Niskanen, M. Järvisalo, Smallest explanations and diagnoses of rejection in abstract argumentation, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, Rhodes, Greece, September 12–18, 2020, 2020, pp. 667–671. URL: <https://doi.org/10.24963/kr.2020/67>. doi:10.24963/kr.2020/67.
 - [10] M. Ulbricht, J. P. Wallner, Strong explanations in abstract argumentation, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2–9, 2021, AAAI Press, 2021, pp. 6496–6504. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16805>.
 - [11] P. Besnard, S. Doutre, T. Duchatelle, M. Lagasquie-Schiex, Explaining semantics and extension membership in abstract argumentation, *Intell. Syst. Appl.* 16 (2022) 200118. URL: <https://doi.org/10.1016/j.iswa.2022.200118>. doi:10.1016/j.iswa.2022.200118.
 - [12] S. Doutre, T. Duchatelle, M. Lagasquie-Schiex, Visual explanations for defence in abstract argumentation, in: N. Agmon, B. An, A. Ricci, W. Yeoh (Eds.), *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023*, London, United Kingdom, 29 May 2023 - 2 June 2023, ACM, 2023, pp. 2346–2348. URL: <https://dl.acm.org/doi/10.5555/3545946.3598929>. doi:10.5555/3545946.3598929.
 - [13] P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, M. J. Wooldridge, Weighted argument systems: Basic definitions, algorithms, and complexity results, *Artif. Intell.* 175 (2011) 457–486. URL: <https://doi.org/10.1016/j.artint.2010.09.005>. doi:10.1016/j.artint.2010.09.005.

Accessible Algorithms for Applied Argumentation

Daphne Odekerken^{1,2}, AnneMarie Borg¹ and Matti Berthold³

¹*Department of Information and Computing Sciences, Utrecht University, The Netherlands*

²*National Police Lab AI, Netherlands Police, The Netherlands*

³*ScaDS.AI Dresden/Leipzig, Universität Leipzig, Germany*

Abstract

Computational argumentation is a promising research area, yet there is a gap between theoretical contributions and practical applications. Bridging this gap could potentially raise interest in this topic even more. We argue that one part of the bridge could be an open-source package of implementations of argumentation algorithms, visualised in a web interface. Therefore we present a new release of PyArg, providing various new argumentation-based functionalities – including multiple generators, a learning environment, implementations of theoretical papers and a showcase of a practical application – in a new interface with improved accessibility.

Keywords

argumentation, implementation, visualisation, education

1. Introduction

Computational argumentation is a promising interdisciplinary research area, with applications in, e.g. the legal, medical and e-government domain [1]. Thanks to its natural connection with human cognition, its flexibility and its dialectic nature, argumentation seems to be particularly suitable as a logical foundation for human-centered artificial intelligence (AI) [2].

The construction of argumentation-based AI systems that are applicable to real-world use cases requires not only theoretical developments, but also effective solving of problems related to argumentation. Although there is a significant amount of work on theoretical aspects of computational argumentation, which include various argumentation formalisms, semantics and their properties, as well as a growing body of research on solvers that are sufficiently efficient to be applied in practice¹, we argue that the connection between these two areas could (and should) be strengthened. This is based on our observation that real-life applications use a different formalism than those mainly studied in the community [3] or require efficient algorithms for yet unexplored problems [4].

In addition, we hypothesise that the application of argumentation-based AI is more cumbersome for non-experts than, for example, the application of machine-learning based AI, which is

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

✉ d.odekerken@uu.nl (D. Odekerken); a.borg@uu.nl (A. Borg); berthold@informatik.uni-leipzig.de (M. Berthold)

🌐 <https://webpace.science.uu.nl/~3827887/> (D. Odekerken); <https://annemarieborg.nl/> (A. Borg);

<https://www.informatik.uni-leipzig.de/~berthold/> (M. Berthold)

🆔 0000-0003-0285-0706 (D. Odekerken); 0000-0002-7204-6046 (A. Borg); 0009-0006-9231-5115 (M. Berthold)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹In particular by submissions to the International Competition on Computational Models of Argumentation (ICCM): <http://argumentationcompetition.org/>

more convenient thanks to the availability of numerous software packages (such as Scikit-learn² and Tensorflow³) as well as user-friendly interfaces to try this AI.⁴ This makes argumentation less accessible to students, software developers and companies searching for AI-based solutions to domain-specific problems.

In order to improve both the connection within theoretical and practical work within the computational argumentation community and the connection with stakeholders outside the community, it would be helpful to have open-source software, paired with an accessible web interface. In this paper, we therefore present a new release of PyArg.

PyArg is an open-source software implementation in Python that not only provides practical algorithms for theoretical problems in various argumentation formalisms, but also makes (potential) applications of these algorithms visible in a user interface that is accessible from the internet without installation. PyArg is intended to be a software solution for researchers within the argumentation community, students who may become part of it, as well as stakeholders outside the community, thanks to the following features:

1. Open-source implementations of argumentation algorithms on GitHub can be validated and extended by community members (<https://github.com/DaphneOdekerken/PyArg>);
2. The Python package can be installed using `pip install python-argumentation` and is therefore directly usable for software developers;
3. The web interface on <https://pyarg.npai.science.uu.nl/> makes argumentation more accessible to those who wish to learn more about argumentation and serves as a platform for showcasing applications of argumentation to stakeholders outside the community.

The release presented here is a major update compared to the preliminary version earlier presented in [5]. In the following sections, we describe PyArg's new functionalities.

2. Support for more formalisms

Research on computational argumentation ranges over an ever-growing number of formalisms and extensions of formalisms. Whereas the initial version of PyArg [5] only supported abstract argumentation frameworks and ASPIC⁺, in this iteration we added support for assumption-based argumentation (ABA) [6, 7]. In particular, PyArg can now (1) compute the extensions of a given ABA framework under the prominent semantics and visualise them in the instantiated abstract argumentation framework; and (2) verify if a specific set of extensions can be realised under a given semantics (see Section 3.2).

3. Algorithms for argumentation problems

The new PyArg version still contains all algorithms presented in [5], including various implementations of formalisms and algorithms in both abstract argumentation [8] and ASPIC⁺ [9]. It provides algorithms for evaluating argumentation settings in different semantics [10], as well

²<https://scikit-learn.org/>

³<https://www.tensorflow.org/>

⁴e.g. <https://freegpt.cc/>

as explaining the (non-)acceptance of arguments and formulas in various ways [11, 12]. In the new version, we also provide algorithms for dynamic argumentation problems as well as for the construction of canonical representations.

3.1. Dynamic argumentation problems

Many problems in argumentation assume that all information required to decide upon, for example, the acceptance of an argument is present. This is however not always a realistic assumption in applied settings. Therefore, recently the problems of stability [4] and relevance [13] have been introduced for various formalisms, including ASPIC⁺. Informally, the stability status of a “topic” argument or formula represents the impossibility that its acceptability status may change in view of additional, yet uncertain information. For topics that are not stable, it is interesting to study relevance, where only information that can change the stability status of the topic is relevant. PyArg provides an implementation of the approximation algorithm for stability from [4], as well as an inexact but efficient algorithm for estimating relevance based on the labels from the aforementioned stability algorithm.

3.2. Canonical argumentation frameworks

The work by Dunne et al. [14] studies the *realisability* in abstract argumentation: given a semantics σ and a set of extensions $\mathbb{S}$, is there an argumentation framework F realising $\mathbb{S}$ under σ , i.e., such that $\sigma(F) = \mathbb{S}$ – and which characteristics determine whether such an argumentation framework exists? Similarly, realisability can be characterised for ABAFs [15]. PyArg now provides the algorithms to determine whether a given set extensions satisfies these characteristic properties and generates “canonical” argumentation frameworks or ABA frameworks, when they exist; see Figure 1.

4. Practical features

4.1. Generators

There are various situations in which it is useful to randomly generate an argumentation setting. One example would be for testing new algorithms: a large part of the data sets used in the ICCMA competition to assess runtime of algorithms is based on randomly generated argumentation frameworks. A second example is related to education: in order to assess a student’s understanding of, e.g., specific argumentation semantics, it is convenient to have a generator for automatically creating new exercises.

In order to address this demand, PyArg provides several generators. For generating ASPIC⁺ argumentation systems, PyArg uses the “layered” generator from [4, Section 4.2.5]. In addition, PyArg provides a basic random generator for abstract argumentation frameworks.

The generators can be found in the source code; in addition, the web interface provides functionality for generating a single argumentation framework or system, parameterised by the values given in input fields. The resulting argumentation setting can then be downloaded for further use within or outside PyArg.

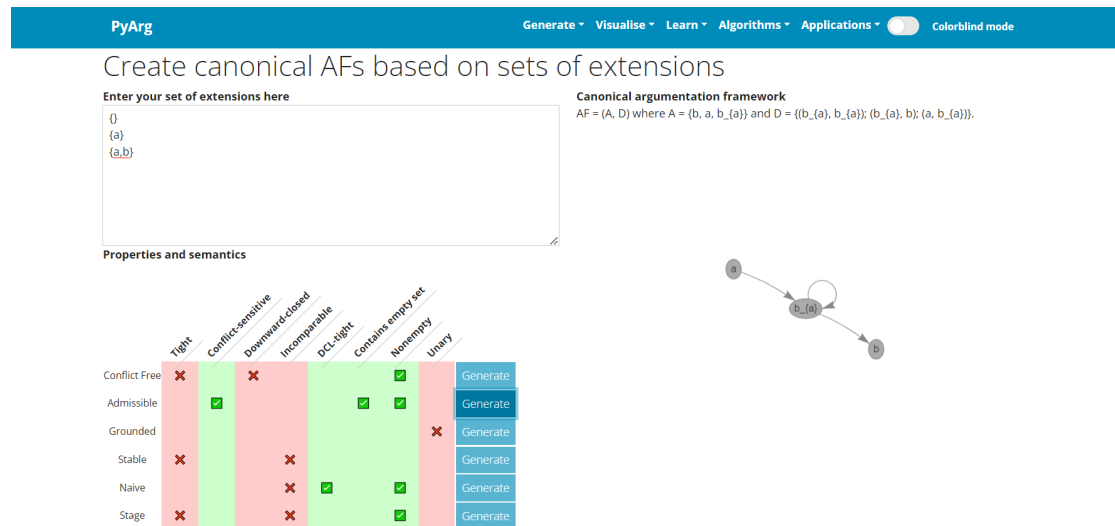


Figure 1: Given an extension set provided by the user, PyArg computes certain properties and generates a canonical argumentation framework when possible.

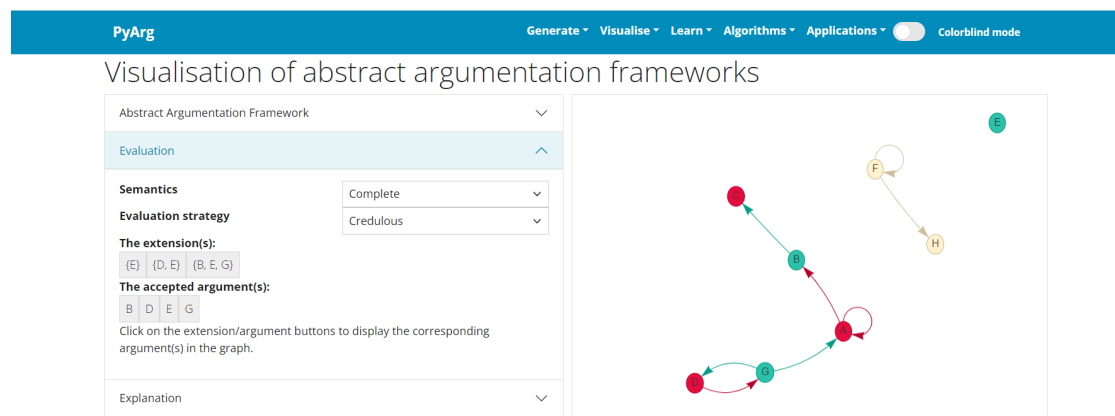


Figure 2: Improved visualisation of the abstract argumentation environment.

4.2. Improved visualisation

Compared to the previous version, PyArg's user interface has been made more user-friendly and accessible. A screenshot of the new visualisation for abstract argumentation frameworks is shown in Figure 2. PyArg now features both a *regular* mode (in which accepted arguments are coloured green, while other arguments are yellow or red) and a *colourblind-friendly* mode that uses an adapted colour palette.

4.3. Importers and exporters

PyArg provides various importers and exporters to convert argumentation settings to various formats. For abstract argumentation frameworks, it is possible to read from and write to all formats used in recent ICCMA competitions, that is: the ASPARTIX format (.APX), the Trivial Graph Format (.TGF) and the input format used for ICCMA 2023 (.ICCMA23). In addition, for both abstract argumentation frameworks and ASPIC⁺ argumentation systems, there are importers and exporters to and from JSON.

5. Applications of algorithms in the web interface

In order to demonstrate how PyArg's algorithms can be applied in various settings, we provide two use cases in the web interface: a learning environment and a chat interface.

5.1. Learning environment

The learning environment is intended for anyone interested in learning computational argumentation. In this functionality in the web interface, a learner can choose between various exercises. The current PyArg version features three exercises: identifying grounded, complete and preferred extensions in abstract argumentation. As the learner starts an exercise, PyArg generates a random abstract argumentation framework using its generators. The learner then gives the extensions, and PyArg uses its semantics algorithms for validating the learner's solutions.

5.2. Chat interface

The chat interface showcases an application for the algorithms for stability and relevance in inquiry dialogue. First, the user chooses an ASPIC⁺ argumentation system (which can be randomly generated, hand-made or the predefined fraud example), a set of queryables (e.g. formulas that can be asked in a dialogue), a topic formula for the chat and an initial knowledge base. PyArg then uses the stability algorithm to find out if it makes sense to ask for more information - if so, it uses the relevance algorithm for identifying relevant questions.

6. Related work

For an extended overview of software related to computational argumentation, we refer to [16]. In this section, we relate to implementations that are most similar to PyArg. Tweety [17] is a comprehensive collection of Java libraries that includes algorithms for both abstract and structured approaches to argumentation. It is in fact more comprehensive than PyArg, but does not have a visualisation option. The Online Argument Structures Tool (TOAST) [18] does provide a visualisation for ASPIC⁺, but the source code is not openly available. Gorgias Cloud [19] is a recent system that is similar to PyArg, but is based on the Gorgias argumentation system. NEXAS [20] is an alternative approach to visualising extensions of abstract argumentation frameworks, which, compared to PyArg, is more aimed towards (large) frameworks with many extensions. Finally, many algorithms for argumentation-related problems have been submitted

to the ICCMA competition. However, these are mostly focused on fast implementation of limited problems, mainly in the context of abstract argumentation.

7. Conclusion and future work

PyArg combines algorithms for argumentation problems with a web interface, aiming to improve the connections between theoretical and practical work on argumentation on the one hand, and inside and outside the community on the other hand.

The contributions of this version of PyArg are mainly focused on the front-end. In the back-end, the algorithms for computing the semantics are not state-of-the-art. This becomes noticeable if the visualisation is tested on hard (large) instances, as no output will appear on the screen within a reasonable amount of time. In a next version of PyArg, we plan to improve this performance aspect by calling more efficient solvers in the back-end. In addition, we aim to add support more formalisms, such as abstract dialectical frameworks [21], abstract frameworks with collective attacks [22] and abstract frameworks with support for claims [23], and to implement their intertranslations [24].

On a final note, we hope that the functionalities presented in this paper are just the beginning, as are the plans for future work mentioned above. We are open to any suggestions for additional functionalities, algorithms or other feedback, which we hope to incorporate in future releases of PyArg. Hopefully, this leads to an increase of interest and understanding of computational argumentation, both within and outside the community, eventually resulting in more responsible applications of artificial intelligence.

Acknowledgments

The authors have not been sorted alphabetically. They acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI.

References

- [1] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. Simari, M. Thimm, S. Villata, Towards artificial argumentation, *AI magazine* 38 (2017) 25–36.
- [2] E. Dietz, A. Kakas, L. Michael, Argumentation: A calculus for human-centric AI, *Frontiers in Artificial Intelligence* 5 (2022).
- [3] A. C. Kakas, P. Moraitis, N. I. Spanoudakis, Gorgias: Applying argumentation, *Argument & Computation* 10 (2019) 55–81.
- [4] D. Odekerken, F. Bex, A. Borg, B. Testerink, Approximating stability for applied argument-based inquiry, *Intelligent Systems with Applications* 16 (2022) 200110.
- [5] A. Borg, D. Odekerken, PyArg for solving and explaining argumentation in Python: Demonstration, in: *Proceedings of (COMMA-22)*, 2022, pp. 349–350.

- [6] A. Bondarenko, F. Toni, R. A. Kowalski, An assumption-based framework for non-monotonic reasoning, in: *Proceedings of (LPNMR-93)*, 1993, pp. 171–189.
- [7] K. Čyras, X. Fan, C. Schulz, F. Toni, Assumption-based argumentation: Disputes, explanations, preferences, in: *Handbook of Formal Argumentation*, volume 1, 2018, pp. 365–408.
- [8] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artificial Intelligence* 77 (1995) 321–357.
- [9] H. Prakken, An abstract framework for argumentation with structured arguments, *Argument & Computation* 1 (2010) 93–124.
- [10] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, *The Knowledge Engineering Review* 26 (2011) 365–410.
- [11] A. Borg, F. Bex, A basic framework for explanations in argumentation, *IEEE Intelligent Systems* 36 (2021) 25–35.
- [12] A. Borg, F. Bex, Necessary and sufficient explanations for argumentation-based conclusions, in: *Proceedings of (ECSQARU-21)*, Springer, 2021, pp. 45–58.
- [13] D. Odekerken, T. Lehtonen, A. Borg, J. P. Wallner, M. Järvisalo, Argumentative reasoning in ASPIC+ under incomplete information, in: *Proceedings of (KR-23)*, 2023, pp. 531–541.
- [14] P. E. Dunne, W. Dvořák, T. Linsbichler, S. Woltran, Characteristics of multiple viewpoints in abstract argumentation, *Artificial Intelligence* 228 (2015) 153–178.
- [15] M. Berthold, A. Rapberger, M. Ulbricht, On the expressive power of assumption-based argumentation, in: *Proceedings of (JELIA-23)*, 2023.
- [16] F. Cerutti, S. A. Gaggl, M. Thimm, J. Wallner, Foundations of implementations for formal argumentation, *IfCoLog Journal of Logics and their Applications* 4 (2017) 2623–2705.
- [17] M. Thimm, The formal argumentation libraries of Tweety, in: *Proceedings of (TAFE-17)*, Springer, 2018, pp. 137–142.
- [18] M. Snaith, C. Reed, TOAST: Online ASPIC+ implementation, in: *Proceedings of (COMMA-12)*, IOS Press, 2012, pp. 509–510.
- [19] N. I. Spanoudakis, G. Gligoris, A. Koumi, A. C. Kakas, Explainable argumentation as a service, *Journal of Web Semantics* (2023) 100772.
- [20] R. Dachsel, S. Gaggl, M. Krötzsch, J. Méndez, D. Rusovac, M. Yang, Nexas: A visual tool for navigating and exploring argumentation solution spaces, in: *Proceedings of (COMMA-22)*, volume 353, IOS Press, 2022, pp. 116–127.
- [21] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, S. Woltran, Abstract dialectical frameworks revisited, in: *Proceedings of (IJCAI-13)*, 2013, pp. 803–809.
- [22] S. H. Nielsen, S. Parsons, A generalization of dung’s abstract framework for argumentation: Arguing with sets of attacking arguments, in: *Proceedings of (ArgMAS-2006)*, Springer, 2006, pp. 54–73. doi:10.1007/978-3-540-75526-5_4.
- [23] W. Dvořák, A. Rapberger, S. Woltran, Argumentation semantics under a claim-centric view: Properties, expressiveness and relation to setafs, in: *Proceedings of (KR-20)*, 2020, pp. 341–350. doi:10.24963/kr.2020/35.
- [24] M. König, A. Rapberger, M. Ulbricht, Just a matter of perspective, in: *Proceedings of (COMMA-22)*, 2022, pp. 212–223. doi:10.3233/FAIA220154.