

Raisonnement Approximatif pour l'Acceptabilité des Arguments en Argumentation Abstraite

Jérôme Delobelle¹ Jean-Guy Mailly² Julien Rossit¹

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

² Université Toulouse Capitole, IRIT, Toulouse, France

{jerome.delobelle,julien.rossit}@u-paris.fr, jean-guy.mailly@irit.fr

Résumé

Définir des algorithmes efficaces pour des tâches de raisonnement complexe est un enjeu majeur dans l'argumentation abstraite. Une approche récente propose de définir des algorithmes approximatifs fournissant une réponse qui peut ne pas toujours être correcte, mais qui surpasse les algorithmes exacts en termes de temps de calcul. Plus particulièrement, elle propose d'utiliser la sémantique de base, qui est calculable en temps polynomial, comme point de départ pour déterminer si les arguments sont acceptés (de manière crédule ou sceptique) par rapport à différentes sémantiques. Dans cet article, nous allons plus loin dans cette idée en définissant différentes approches permettant d'évaluer l'acceptabilité des arguments qui ne sont ni dans l'extension de base, ni attaqués par celle-ci. Nous avons implémenté ces approches afin d'établir une évaluation empirique de celles-ci. Ce papier est un résumé du papier publié à la conférence ECSQARU 2023 [2].

1 État de l'art

Formellement, un **système d'argumentation** (AF, pour *Argumentation Framework*) [3] est un couple $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ où \mathcal{A} est un ensemble fini d'arguments et $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ est la relation d'attaque entre arguments. Des *sémantiques à base d'extensions* ont été définies pour déterminer l'acceptabilité des arguments. Elles sont en général basées sur deux principes : un ensemble acceptable $E \subseteq \mathcal{A}$ doit 1) être *sans conflit* ($\text{cf}(\mathcal{F})$) : $\forall a, b \in E, (a, b) \notin \mathcal{R}$; 2) *défendre tous ses éléments* : $\forall a \in E, \forall b \in \mathcal{A}$, si $(b, a) \in \mathcal{R}$ alors $\exists c \in E$ tel que $(c, b) \in \mathcal{R}$. Un ensemble qui satisfait ces deux propriétés est appelé *admissible* ($\text{ad}(\mathcal{F})$). Les extensions *complètes* ($\text{co}(\mathcal{F})$) sont les ensembles admissibles qui contiennent tous les arguments qu'ils défendent; l'extension *de base* ($\text{gr}(\mathcal{F})$) est l'extension complète minimale pour l'inclusion; les extensions *préférées* ($\text{pr}(\mathcal{F})$) sont les extensions complètes maximales pour l'inclusion;

les extensions *stables* ($\text{stb}(\mathcal{F})$) sont les ensembles sans conflit qui attaquent tous les arguments n'étant pas dans l'ensemble. Étant donné une sémantique à base d'extensions σ , un argument est *crédulement* (resp. *sceptiquement*) accepté par rapport à σ s'il appartient à au moins une (resp. toutes les) extension(s). Les problèmes liés au fait de savoir si un argument est *crédulement* ou *sceptiquement* accepté sont DC- σ et DS- σ respectivement.

Notons que le calcul de la sémantique de base (gr) est réalisé en temps polynomial, alors que les autres sémantiques ont une complexité plus importante [4]. Ceci a conduit à une évaluation empirique montrant la similarité non-négligeable entre la sémantique de base et les autres sémantiques à base d'extensions [1]. Cela a amené au développement du solveur approximatif Harper++ [5] pour les problèmes DC- σ et DS- σ . Pour les sémantiques usuelles, un argument qui appartient à l'extension de base sera nécessairement dans toutes les extensions, tandis qu'un argument attaqué par un élément de l'extension de base ne sera dans aucune extension. Il convient donc de se concentrer sur le troisième type d'arguments, qu'on notera UNDEC(\mathcal{F}) et qui correspond à l'ensemble des arguments n'étant ni dans $\text{gr}(\mathcal{F})$, ni attaqué par celle-ci. Étant donné un AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ et $a \in \mathcal{A}$, l'algorithme de Harper++ répond donc : YES si $a \in \text{gr}(\mathcal{F})$, NO si a est attaqué par $b \in \text{gr}(\mathcal{F})$, et pour les éléments de UNDEC(\mathcal{F}) l'algorithme répond YES si le problème est DC- σ , et NO si le problème est DS- σ .

2 ARIPOTER

Notre but est d'étendre Harper++, notamment sur l'évaluation des arguments appartenant à UNDEC(\mathcal{F}), en introduisant deux familles de solveurs approximatifs nommées ARIPOTER (ARgumentation apPrOximaTE Reasoning).

Le solveur ARIPOTER-degrees capture l'idée qu'un argument qui attaque plus d'arguments que le nombre de ses attaquants directs a de bonnes chances de se défendre, et donc d'être accepté. Ainsi, un argument de UNDEC(\mathcal{F}) sera considéré comme accepté si le nombre d'arguments qu'il attaque (i.e. *out-degree*) est au moins k fois plus élevé que le nombre d'arguments qui l'attaquent (i.e. *in-degree*).

Définition 1 *Étant donné un AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, $a \in \mathcal{A}$ et $k \in \mathbb{R}$, la fonction $Acc^{Out/In}$ est défini comme suit :*

$$Acc^{Out/In}(\mathcal{F}, a, k) = \begin{cases} YES & \text{si } a \in gr(\mathcal{F}) \text{ ou} \\ & (a \in UNDEC(\mathcal{F}) \text{ et} \\ & |a^+| \geq k \times |a^-|), \\ NO & \text{sinon.} \end{cases}$$

avec $a^+ = \{b \in \mathcal{A} \mid (a, b) \in \mathcal{R}\}$ et $a^- = \{b \in \mathcal{A} \mid (b, a) \in \mathcal{R}\}$ l'ensemble des arguments attaqués par a ou attaquant a respectivement.

Notre seconde approche, ARIPOTER-hcat, accepte les arguments de UNDEC(\mathcal{F}) dont le score retourné par la sémantique h-categorizer est supérieur à une valeur τ . Cette sémantique graduée utilise une fonction dont le but est d'attribuer une valeur qui capture la force relative d'un argument en tenant compte de la force de ses attaquants. Formellement, étant donné un AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ et $a \in \mathcal{A}$, $h\text{-cat}(\mathcal{F}, a) = \frac{1}{1 + \sum_{b \in a^-} h\text{-cat}(\mathcal{F}, b)}$.

Définition 2 *Étant donné un AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, $a \in \mathcal{A}$ et $\tau \in [0, 1]$, la fonction $Acc^{h\text{-cat}}$ est défini comme suit :*

$$Acc^{h\text{-cat}}(\mathcal{F}, a, \tau) = \begin{cases} YES & \text{si } a \in gr(\mathcal{F}) \text{ ou} \\ & (a \in UNDEC(\mathcal{F}) \text{ et} \\ & h\text{-cat}(\mathcal{F}, a) \geq \tau), \\ NO & \text{sinon.} \end{cases}$$

avec $h\text{-cat}(\mathcal{F}, a) = \frac{1}{1 + \sum_{b \in a^-} h\text{-cat}(\mathcal{F}, b)}$.

3 Analyse Empirique

Concernant les instances utilisés pour évaluer et comparer nos deux approches et Harper++, nous avons sélectionné les deux familles d'instances suivantes :

iccma19 : ensemble des instances difficiles "2019" de la compétition ICCMA 2021, composé de 107 AF, avec un nombre d'arguments entre 102 et 8034 arguments.

randomAF : ensemble de 9460 AF réparties entre trois familles de graphes (Erdős-Rényi, Barabási-Albert et Watts-Strogatz) utilisés lors des compétitions ICCMA. Le nombre d'attaques, de cycles et d'arguments (entre 10 et 100) varie selon la famille étudiée.

Notre étude s'est concentrée sur quatre problèmes de décision : DC-stb, DS-stb, DC-pr and DS-pr. La table 1 résume la précision des différents solveurs approximatifs sur randomAF et iccma19. Observons que les deux sol-

Instances	Solveur	DC-pr	DC-stb	DS-pr	DS-stb
randomAF	Harper++	0.4161	0.414	0.9794	0.4738
	ARIPOTER-degrees	0.8306 (8)	0.8306 (10)	0.9797 (8)	0.7802 (0.1)
	ARIPOTER-hcat	0.9321 (0.5)	0.9304 (0.5)	0.9794 (1)	0.7648 (0.1)
iccma19	Harper++	0.7549	0.757	0.9712	0.8269
	ARIPOTER-degrees	0.7941 (\mathcal{A})	0.8131 (\mathcal{A})	0.9712 (\mathcal{A})	0.5481 (0.1)
	ARIPOTER-hcat	0.7941 (0.5)	0.8131 (0.5)	0.9712 (1)	0.5385 (0.1)

TABLE 1 – Comparaison de la précision des trois solveurs approximatifs pour randomAF et iccma19, avec les valeurs de k (pour ARIPOTER-degrees) ou de τ (pour ARIPOTER-hcat) retournant les meilleurs résultats entre parenthèses.

veurs ARIPOTER retournent une bien meilleure précision que Harper++ pour les instances randomAF avec des résultats à plus de 93% pour tous les problèmes étudiés excepté pour DS-stb (78%). Pour les instances iccma19, les résultats de nos deux solveurs sont très proches pour les quatre problèmes étudiés. En comparaison avec les résultats précédents, la précision est légèrement plus faible pour les problèmes DC, mais reste de l'ordre de 80% de réponses correctes. Cependant, cette diminution est plus importante pour le problème DS-stb où nos solveurs obtiennent une précision qui est de l'ordre de 54% contrairement à Harper++ qui est de plus de 82%.

Après ces résultats très encourageants, nous prévoyons d'étendre notre approche à d'autres sémantiques graduées et d'étudier plus en détails la précision de ces solveurs sur d'autres familles de graphes.

Remerciements. Ce travail est supporté par l'ANR (AG-GREEY ANR-22-CE23-0005 et AIDAL ANR-22-CPJ1-0061-01).

Références

- [1] Federico Cerutti, Matthias Thimm, and Mauro Vallati. An experimental analysis on the similarity of argumentation semantics. *Arg. Comput.*, 11(3) :269–304, 2020.
- [2] Jérôme Delobelle, Jean-Guy Mailly, and Julien Rossit. Revisiting approximate reasoning based on grounded semantics. In *Proc. of ECSQARU'23*, 2023.
- [3] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2) :321–358, 1995.
- [4] Wolfgang Dvorák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In *Handbook of Formal Argumentation*, pages 631–688. College Publications, 2018.
- [5] Matthias Thimm. Harper++ : Using grounded semantics for approximate reasoning in abstract argumentation. <http://argumentationcompetition.org/2021/downloads/harper++.pdf>, 2021.