An Efficient Algorithm for Skeptical Preferred Acceptance in Dynamic Argumentation Frameworks

Gianvincenzo Alfano, **Sergio Greco** and **Francesco Parisi**[*]
Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, Italy
{g.alfano, greco, fparisi}@dimes.unical.it

Abstract

Though there has been an extensive body of work on efficiently solving computational problems for static Dung's argumentation frameworks (AFs), little work has been done for handling dynamic AFs and in particular for deciding the skeptical acceptance of a given argument. In this paper we devise an efficient algorithm for computing the skeptical preferred acceptance in dynamic AFs. More specifically, we investigate how the skeptical acceptance of an argument (goal) evolves when the given AF is updated and propose an efficient algorithm for solving this problem. Our algorithm, called SPA, relies on two main ideas: *i*) computing a small portion of the input AF, called "context-based" AF, which is sufficient to determine the status of the goal in the updated AF, and *ii*) incrementally computing the ideal extension to further restrict the context-based AF. We experimentally show that SPA significantly outperforms the computation from scratch, and that the overhead of incrementally maintaining the ideal extension pays off as it speeds up the computation.

1 Introduction

Abstract argumentation has emerged as one of the major fields in Artificial Intelligence [Bench-Capon and Dunne, 2007; Rahwan and Simari, 2009]. In particular, abstract argumentation frameworks (AFs) [Dung, 1995] are a simple, yet powerful formalism for modelling disputes between two or more agents. The formal meaning of an AF is given in terms of argumentation semantics, which intuitively tell us the sets of arguments (called *extensions*) that can collectively be used to support a point of view in a discussion.

Although the idea underlying AFs is very simple and intuitive, most of the argumentation semantics proposed so far suffer from a high computational complexity [Dunne and Wooldridge, 2009; Dunne, 2009; Kröll *et al.*, 2017]. In particular, skeptical reasoning under the well-known *preferred* semantics—one of the most popular semantics [Caminada *et al.*, 2016]—is in the second level of the polynomial hierarchy. Efficient algorithms for AFs have been deeply investigated in the literature, as witnessed by the International Competition on Computational Models of Argumentation (ICCMA) [1]. One of the more challenging tasks of ICCMA is deciding the skeptical preferred acceptance of a given argument (goal).

However, ICCMA competition as well as most research have focused on 'static' frameworks, whereas in practice AFs are dynamic systems [Baumann and Brewka, 2010; Baumann, 2011; Falappa *et al.*, 2011; Oikarinen and Woltran, 2011; Charwat *et al.*, 2015]. In fact, an AF often represents a temporary situation, as new arguments and attacks can be added/retracted to take into account new available knowledge. For instance, for disputes among users of online social networks [Kökciyan *et al.*, 2017], arguments/attacks are continuously added/retracted by users to express their point of view in response to the last moves made by the adversaries.

Recently, the definition of evaluation algorithms taking into account such dynamic aspects has received an increasing attention, as in these situations incremental computation techniques could greatly improve performance [Liao *et al.*, 2011; Baroni *et al.*, 2014; Alfano *et al.*, 2017; Bistarelli *et al.*, 2018a]. In this regard, a new track focusing on solvers processing dynamic AFs have been recently proposed for the next edition of ICCMA competition [Bistarelli *et al.*, 2018b].

In this paper, we propose an algorithm for incrementally solving the following computational task: Given an AF $\mathscr{A}_0$, a goal argument g whose skeptical preferred acceptance w.r.t. $\mathscr{A}_0$ is known, and an update u, decide whether g is skeptical preferred accepted w.r.t. the updated AF $u(\mathscr{A}_0)$, that is, decide if g belongs to every preferred extension of $u(\mathscr{A}_0)$. Thus, we explore how to efficiently and incrementally solve the ICCMA computational task DS-pr [Thimm and Villata, 2017].

Contributions. We make the following contributions:

- Given an update and an argument, we identify a set of arguments, called *supporting set*, which contains all the arguments whose acceptance status may change after the update and propagate up to the goal argument.
- Given the supporting set, we define the concept of *context-based AF* that allows us to compute the skeptical preferred acceptance of an argument by focusing on a smaller AF containing the supporting set as well as additional arguments and attacks representing auxiliary information on the external context.

[*] Contact Author

[1] http://argumentationcompetition.org

- We introduce SPA, an incremental algorithm for computing the Skeptical Preferred Acceptance of a goal within a dynamic AF. It enables the computation on context-based AFs by means of (non-incremental) state-of-the-art AF solvers. Our solution relies on incrementally maintaining the ideal extension of the given AF. However, to show the relevance of using the ideal extension, we also consider a simpler version of our algorithm (called SPA-base) which does not consider the information provided by the ideal extension.
- We perform a thorough experimental analysis showing the effectiveness of our approach, even in the case of sets of updates applied simultaneously. We compare both SPA and SPA-base with the solver that won the ICCMA'17 competition for the computational task DS-`pr` as well as between them. Both SPA and SPA-base significantly beat the computation from scratch, and SPA performs better than SPA-base on average.

To the best of our knowledge, this is the first paper addressing the problem of efficiently and incrementally computing skeptical acceptance for dynamic AFs.

2 Preliminaries

We assume the existence of a set *Arg* of *arguments*. An *(abstract) argumentation framework* [Dung, 1995] (*AF*) is a pair $\langle A, \Sigma \rangle$, where $A \subseteq Arg$ is a set of *arguments*, and $\Sigma \subseteq A \times A$ is a binary relation over A whose elements are called *attacks*.

Example 1 (Running example). *Figure 1(a) shows the graph of the AF* $\mathtt{AF}_0 = \langle \mathtt{A}_0, \Sigma_0 \rangle$ *where* $\mathtt{A}_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{l}\}$ *and* Σ_0 *includes, among others, attacks* $(\mathtt{a}, \mathtt{b})$*,* $(\mathtt{b}, \mathtt{a})$*, and* $(\mathtt{c}, \mathtt{d})$*.* ■

Given an AF $\langle A, \Sigma \rangle$ and arguments $a, b \in A$, we say that *a attacks b* iff $(a, b) \in \Sigma$, and that a set $S \subseteq A$ attacks b iff there is $a \in S$ attacking b. We use $S^+ = \{b \mid \exists a \in S : (a, b) \in \Sigma\}$ to denote the set of arguments attacked by S.

Moreover, we say that $S \subseteq A$ *defends a* iff $\forall b \in A$ such that *b attacks a*, there is $c \in S$ such that *c attacks b*.

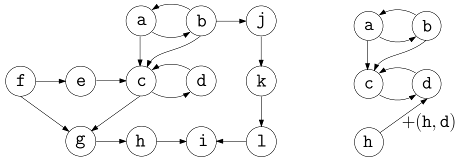
A set $S \subseteq A$ of arguments is said to be: (*i*) *conflict-free* if there are no $a, b \in S$ such that *a attacks b*; (*ii*) *admissible* if it is conflict-free and it defends all its arguments.

An argumentation semantics specifies the criteria for identifying a set of arguments that can be considered "reasonable" together, called *extension*. A *preferred* extension of an AF $\mathscr{A}$ is a maximal (w.r.t. $\subseteq$) admissible set of $\mathscr{A}$. The *ideal* extension of $\mathscr{A}$ is the biggest (w.r.t. $\subseteq$) admissible set of $\mathscr{A}$ which is contained in every preferred extension of $\mathscr{A}$.

It is well-known that every AF admits exactly one ideal extension which is contained in the intersection of the preferred extensions, which are at least one [Dung *et al.*, 2007].

Example 2. *The preferred extensions of* $\mathtt{AF}_0$ *are* $\mathtt{E}_{\mathtt{pr}} = \{\mathtt{a}, \mathtt{d}, \mathtt{f}, \mathtt{h}, \mathtt{j}, \mathtt{l}\}$ *and* $\mathtt{E}'_{\mathtt{pr}} = \{\mathtt{b}, \mathtt{d}, \mathtt{f}, \mathtt{h}, \mathtt{k}\}$*, while the ideal extension of* $\mathtt{AF}_0$ *is* $E_{id} = \{\mathtt{d}, \mathtt{f}, \mathtt{h}\}$*.* ■

Given an AF $\mathscr{A} = \langle A, \Sigma \rangle$ and an argument $g \in A$, we say that g is *skeptically accepted* w.r.t. $\mathscr{A}$ under the preferred semantics iff for each preferred extension E of $\mathscr{A}$ it holds that $g \in E$. In the following, we use $SA_{\mathscr{A}}(g)$ to denote the *skeptical acceptance* (either *true* or *false*) of g w.r.t. AF $\mathscr{A}$.



(*a*) (*b*)

Figure 1: (*a*) AF $\mathtt{AF}_0$, (*b*) $\mathsf{CBAF}(+(\mathtt{h}, \mathtt{d}), \mathtt{AF}_0, E_{id}, \mathtt{c})$

Example 3. *For AF* $\mathtt{AF}_0$ *of our running example, we have that the arguments skeptically accepted are* $\mathtt{d}$*,* $\mathtt{f}$*, and* $\mathtt{h}$*. Thus,* $SA_{\mathtt{AF}_0}(\mathtt{d})$ *is true, and so is for* $SA_{\mathtt{AF}_0}(\mathtt{f})$ *and* $SA_{\mathtt{AF}_0}(\mathtt{h})$*, while for any other argument x,* $SA_{\mathtt{AF}_0}(x)$ *=false.* ■

Fact 1. *Let* $\mathscr{A}$ *be an AF, E the ideal extension of* $\mathscr{A}$*, and g an argument of* $\mathscr{A}$*. If* $g \in E$ *then* $SA_{\mathscr{A}}(g)$ *=true. On the other hand, if* $g \in E^+$ *then* $SA_{\mathscr{A}}(g)$ *=false.*

2.1 Updates

Performing an update on an AF $\mathscr{A}_0$ means modifying it into an AF $\mathscr{A}$ by adding or removing arguments or attacks.

We use $+(a,b)$, with $a, b \in A_0$ and $(a, b) \notin \Sigma_0$, (resp. $-(a,b)$, with $(a,b) \in \Sigma_0$) to denote the addition (resp. deletion) of an attack (a, b), and $u(\mathscr{A}_0)$ to denote the application of update $u = \pm(a, b)$ to AF $\mathscr{A}_0$ (where ± means either + or −). Applying an update u to an AF $\mathscr{A}_0$ implies that the extensions prescribed by a given semantics, as well as the set of arguments that are skeptical accepted, may change.

Example 4. *Continuing with our running example, let* $u = +(\mathtt{h}, \mathtt{d})$*. The ideal extension of* $u(\mathtt{AF}_0)$ *is* $\{\mathtt{f}, \mathtt{h}\}$*, while the preferred extensions are* $\{\mathtt{a}, \mathtt{f}, \mathtt{h}, \mathtt{j}, \mathtt{l}\}$ *and* $\{\mathtt{b}, \mathtt{f}, \mathtt{h}, \mathtt{k}\}$*. Thus, only* $\mathtt{f}$ *and* $\mathtt{h}$ *are skeptically accepted w.r.t.* $u(\mathtt{AF}_0)$*.* ■

As for the addition (resp. deletion) of a set of *isolated* arguments (i.e., arguments not adjacent to any other argument in the graph), it is easy to see that if $\mathscr{A}$ is obtained from $\mathscr{A}_0$ through the addition (resp. deletion) of a set S of isolated arguments, then every argument in S is trivially skeptically accepted (resp., not accepted) w.r.t. $\mathscr{A}$. Indeed, if E_0 is an extension for $\mathscr{A}_0$, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$) is an extension for $\mathscr{A}$ containing every (resp., none) argument in S. Of course, if arguments in S are not isolated, for addition we can first add isolated arguments and then add attacks involving these arguments, while for deletion we can first delete all attacks involving arguments in S. Thus we do not consider these kinds of updates in the following, and focus on the addition and deletions of attacks.

2.2 Notation for reachability and other useful concepts

Given an AF $\mathscr{A} = \langle A, \Sigma \rangle$ and an argument x, we use $Reach_{\mathscr{A}}(x)$ to denote the set of arguments that are reachable from x in the AF $\mathscr{A}$. Moreover, we use $Reach^{-1}_{\mathscr{A}}(x)$ to denote the set of arguments from which x is reachable in $\mathscr{A}$. For instance, for the AF $\mathtt{AF}_0 = \langle \mathtt{A}_0, \Sigma_0 \rangle$ of our running example (see Figure 1(a)), we have that $Reach_{\mathtt{AF}_0}(\mathtt{d}) = \{\mathtt{d}, \mathtt{c}, \mathtt{g}, \mathtt{h}, \mathtt{i}\}$, and $Reach^{-1}_{\mathtt{AF}_0}(\mathtt{h}) = \mathtt{A}_0 \setminus \{\mathtt{i}, \mathtt{j}, \mathtt{k}, \mathtt{l}\}$. We write $Reach_{\mathscr{A}}(x) = \emptyset$ and $Reach^{-1}_{\mathscr{A}}(x) = \emptyset$ if x is not in $\mathscr{A}$.

We use $H(\mathscr{A},u)$ to denote the larger AF between $\mathscr{A}$ and $u(\mathscr{A})$, that is, $H(\mathscr{A},u)$ is (*i*) the updated AF $u(\mathscr{A})$ if u is an addition update (it includes the attack added through u), (*ii*) the original AF $\mathscr{A}$ if u is a deletion (the removed attack is still considered in $H(\mathscr{A},u)$). For instance, if $u = +(\mathtt{h},\mathtt{d})$ then $H(\mathtt{AF}_0,u) = \langle \mathtt{A}_0, \Sigma_0 \cup \{(\mathtt{h},\mathtt{d})\}\rangle$, while $H(\mathtt{AF}_0,u) = \mathtt{AF}_0$ for any deletion update u.

We use $\Pi(S,\mathscr{A})$ to denote the *restriction* of AF $\mathscr{A} = \langle A,\Sigma\rangle$ to a subset $S \subseteq A$ of its arguments [Baroni *et al.*, 2005], that is $\Pi(S,\mathscr{A}) = \langle S, \Sigma \cap (S \times S)\rangle$. For instance, if $S = \{\mathtt{c},\mathtt{d}\}$ then $\Pi(S,\mathtt{AF}_0) = \langle\{\mathtt{c},\mathtt{d}\},\{(\mathtt{c},\mathtt{d}),(\mathtt{d},\mathtt{c})\}\rangle$.

Finally, given $\mathscr{A}_1 = \langle A_1,\Sigma_1\rangle$ and $\mathscr{A}_2 = \langle A_2,\Sigma_2\rangle$, we denote as $\mathscr{A}_1 \sqcup \mathscr{A}_2 = \langle A_1 \cup A_2, \Sigma_1 \cup \Sigma_2\rangle$ the *union* of the two AFs.

3 Supporting Set

In this section, we introduce the novel concept of *supporting set* which intuitively consists of the set of arguments that needs to be taken into account in order to determine the skeptical acceptance of an argument of interest after performing an update. We provide a parametric definition of supporting set that will enable the characterization of different portions of a given AF, called *context-based AFs*, that will be used for two different purposes: (*i*) recompute the skeptical acceptance of a goal w.r.t. the updated AF, and (*ii*) recompute the ideal extension of the updated AF.

Before defining the supporting set, we introduce the auxiliary notion of *steadiness* of an argument. Given an AF $\mathscr{A} = \langle A,\Sigma\rangle$, the ideal extension E of $\mathscr{A}$, and an update $u = \pm(a,b)$, we first define $E(u)$ as the subset of E consisting of the arguments which are not reachable from b in $\mathscr{A}$, i.e., $E(u) = \{z \mid z \in E, z \notin Reach_{\mathscr{A}}(b)\}$. Intuitively, the acceptance status of the arguments in $E(u)$ is not affected by u as they are not reachable from it. Then, the set of *steady* arguments for $u = \pm(a,b)$ w.r.t. $\mathscr{A}$ is defined as $Std_{\mathscr{A}}(u) = (E(u))^+ \setminus \{b\}$, i.e., the arguments attacked by $E(u)$ in $\mathscr{A}$ and that will be still attacked by $E(u)$ in $u(\mathscr{A})$. Argument b is not included in $Std_{\mathscr{A}}(u)$ as it may be no longer attacked by $a \in E(u)$ after performing $u = -(a,b)$; however, it will be considered for positive updates in Definition 1. For the AF $\mathtt{AF}_0$ of our running example, where $E_{id} = \{\mathtt{d},\mathtt{f},\mathtt{h}\}$, if $u = +(\mathtt{h},\mathtt{d})$ then $E_{id}(u) = \{\mathtt{f}\}$ and $Std_{\mathtt{AF}_0}(u) = \{\mathtt{e},\mathtt{g}\} \subseteq E_{id}^+ = \{\mathtt{c},\mathtt{e},\mathtt{g},\mathtt{i}\}$.

Thus, if an argument is steady for update u w.r.t. AF $\mathscr{A}$ then its acceptance status does not depend on u, though it is reachable from an argument of u. Steady arguments limit the portion of the AF to be examined to define the supporting set.

Definition 1 (Supporting set). *Let $\mathscr{A} = \langle A,\Sigma\rangle$ be an AF, $u = \pm(a,b)$ an update, E the ideal extension of $\mathscr{A}$, and g an argument in A. Let*

- $Sup_0(u,\mathscr{A},E,g) = \begin{cases} \emptyset & \textit{if } u = +(a,b) \wedge b \in (E(u))^+; \\ \emptyset & \textit{if } b \notin Reach^{-1}_{H(\mathscr{A},u)}(g); \\ \{b\} & \textit{otherwise.} \end{cases}$
- $Sup_{i+1}(u,\mathscr{A},E,g) = Sup_i(u,\mathscr{A},E,g) \cup \{y \mid \exists (x,y) \in \Sigma \; s.t. \; x \in Sup_i(u,\mathscr{A},E,g) \wedge y \in Reach^{-1}_{H(\mathscr{A},u)}(g) \wedge y \notin Std_{\mathscr{A}}(u)\}$.

Let n be the natural number such that $Sup_n(u,\mathscr{A},E,g) = Sup_{n+1}(u,\mathscr{A},E,g)$. The supporting set *$Sup(u,\mathscr{A},E,g)$ is:*

$$Sup(u,\mathscr{A},E,g) = Sup_n(u,\mathscr{A},E,g) \cap Reach^{-1}_G(g) \qquad (1)$$

where $G = \Pi(Sup_n(u,\mathscr{A},E,g), H(\mathscr{A},u))$ is the restriction of $H(\mathscr{A},u)$ to $Sup_n(u,\mathscr{A},E,g)$.

Finally, when g is not specified, the supporting set, denoted as $Sup(u,\mathscr{A},E,\star)$, is defined as $Sup(u,\mathscr{A},E,g)$ except that all the checks concerning $Reach^{-1}$ are omitted.

Intuitively, $Sup(u,\mathscr{A},E,g)$ consists of the arguments whose status may change after performing an update u and such that their change can imply a change of the status of g.

The supporting set is iteratively defined by $n+1$ steps ($n \leq |A|$), each of them consisting of the addition of at least a non-steady argument attacked by the set built at the previous step and allowing to reach the goal g (if specified). More in detail, $Sup(u,\mathscr{A},E,g)$ for $u = \pm(a,b)$ and g consists of the arguments that (*i*) can be reached from b without using any steady argument y; and (*ii*) allow to reach g in $H(\mathscr{A},u)$ by using only the arguments in $Sup_n(u,\mathscr{A},E,g)$. In fact, Equation (1) entails that an argument of $Sup_n(u,\mathscr{A},E,g)$ will be in $Sup(u,\mathscr{A},E,g)$ only if it can reach g in the restriction of $H(\mathscr{A},u)$ to $Sup_n(u,\mathscr{A},E,g)$—the other arguments in $Sup_n(u,\mathscr{A},E,g)$ are not needed to determine the acceptance status of g, and thus they are pruned by Equation (1).

When no argument g is specified, the set $Sup(u,\mathscr{A},E,\star)$ is built by ignoring condition (*ii*) above. It is easy to see that, for any argument g, $Sup(u,\mathscr{A},E,g) \subseteq Sup(u,\mathscr{A},E,\star) \subseteq Reach_{\mathscr{A}}(b)$, where b is the argument in the update $u = \pm(a,b)$. Moreover, $Sup(u,\mathscr{A},E,g)$ may be empty even if $g \in Reach_{\mathscr{A}}(b)$. Finally, if $Sup(u,\mathscr{A},E,g) \neq \emptyset$ then the arguments of at least one path from b to g belong to $Sup(u,\mathscr{A},E,g)$.

Example 5 (Supporting set for $u = +(\mathtt{h},\mathtt{d})$)**.** *For the goal $\mathtt{c}$, we have that $Sup_0(u,\mathtt{AF}_0,E_{id},\mathtt{c}) = \{\mathtt{d}\}$, $Sup_1(u,\mathtt{AF}_0,E_{id},c) = \{\mathtt{c},\mathtt{d}\}$, and $Sup_2(u,\mathtt{AF}_0,E_{id},\mathtt{c}) = \{\mathtt{c},\mathtt{d}\}$ (the latter does not contain $\mathtt{g}$ since $\mathtt{g} \in Std_{\mathtt{AF}_0}(u)$). Thus, $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{c}) = \{\mathtt{c},\mathtt{d}\}$ as both $\mathtt{c}$ and $\mathtt{d}$ allow to reach $\mathtt{c}$ in the restriction of the updated AF to $\{\mathtt{c},\mathtt{d}\}$. Reasoning analogously, we have that $Sup(u,\mathtt{AF}_0,E_{id},\star) = \{\mathtt{c},\mathtt{d}\}$.*

Consider now what happens for the goal $\mathtt{h}$. Again $Sup_0(u,\mathtt{AF}_0,E_{id},\mathtt{h}) = \{\mathtt{d}\}$, and $Sup_1(u,\mathtt{AF}_0,E_{id},\mathtt{h}) = Sup_2(u,\mathtt{AF}_0,E_{id},\mathtt{h}) = \{\mathtt{c},\mathtt{d}\}$. However, $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{h}) = \emptyset$ as $\{\mathtt{c},\mathtt{d}\} \cap Reach^{-1}_G(\mathtt{h}) = \emptyset$, where $G = \Pi(\{\mathtt{c},\mathtt{d}\},u(\mathtt{AF}_0))$.

Also for the goal $\mathtt{a}$, we have that $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{a}) = \emptyset$. ■

Theorem 1. *Let $\mathscr{A}_0 = \langle A_0,\Sigma_0\rangle$ be an AF, E_0 the ideal extension of $\mathscr{A}_0$, $u = \pm(a,b)$ an update, $\mathscr{A} = u(\mathscr{A}_0)$ the updated AF, and x an argument in A_0. Therefore, if $Sup(u,\mathscr{A}_0,E_0,x) = \emptyset$ then $SA_{\mathscr{A}}(x) = SA_{\mathscr{A}_0}(x)$.*

Example 6. *Continuing with Example 5, since $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{h}) = \emptyset$ we can conclude that $SA_{u(\mathtt{AF}_0)}(\mathtt{h}) = SA_{\mathtt{AF}_0}(\mathtt{h})$ =true. Similarly, since $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{a}) = \emptyset$ then $SA_{u(\mathtt{AF}_0)}(\mathtt{a}) = SA_{\mathtt{AF}_0}(\mathtt{a})$ =false.* ■

4 Context-Based Argumentation Frameworks

The supporting set has been used so far to determine whether the status of the goal does not need to be recomputed. In this section, starting from the supporting set, we define a restriction of the AF which will be used to compute the status of the goal after an update. More specifically, given the supporting set $Sup(u,\mathscr{A},E,g)$ (resp. $Sup(u,\mathscr{A},E,\star)$), we define the *context-based AF* $\mathsf{CBAF}(u,\mathscr{A},E,g)$ (resp.

$\mathsf{CBAF}(u,\mathscr{A},E,\star)$). Moreover, while $\mathsf{CBAF}(u,\mathscr{A},E,\star)$ will be used to incrementally compute the ideal extension of the updated AF (with the aim of checking if one of the conditions of Fact 1 holds), $\mathsf{CBAF}(u,\mathscr{A},E,g)$ will be used to compute the skeptical acceptance $SA_{u(\mathscr{A})}(g)$ w.r.t. the updated AF.

Given an AF $\mathscr{A} = \langle A,\Sigma\rangle$, its ideal extension E, and a set $S \subseteq A$, we use $Nodes(\mathscr{A},S,E)$ to denote the set of the nodes $x \in A$ such that there are a node $y \in S$ and a path from x to y in $\mathscr{A}$ such that all nodes in the path except y do not belong to $E \cup E^+$ (i.e., they are *undecided*, using the *labelling* terminology [Baroni *et al.*, 2011]). Analogously, $Edges(\mathscr{A},S,E)$ is the set of edges $(x,z) \in \Sigma$ such that there are $y \in S$ and a path from x to y in $\mathscr{A}$ containing (x,z) such that all nodes in the path except y do not belong to $E \cup E^+$. Essentially, if S is the supporting set, to determine the status of nodes in S we must also consider all nodes and attacks occurring in paths (of any length) ending in S whose nodes outside S are undecided. The motivation to also consider "undecided" paths is that some of the undecided arguments occurring in such paths could belong to (or be attacked by) some preferred extension and, therefore, together they could determine a change in the status of nodes in S.

Definition 2 (Context-Based AF). *Let $\mathscr{A} = \langle A,\Sigma\rangle$ be an AF, $u = \pm(a,b)$, E the ideal extension of $\mathscr{A}$, and x either an argument in A or the symbol $\star$. Let $S = Sup(u,\mathscr{A},E,x)$. The context-based AF of $\mathscr{A}$ w.r.t. u and x is $\mathsf{CBAF}(u,\mathscr{A},E,x) = \Pi(Sup(u,\mathscr{A},E,x),u(\mathscr{A})) \sqcup T_1 \sqcup T_2$ where:*

- *T_1 is the union of the AFs $\langle\{c,d\},\{(c,d)\}\rangle$ s.t. (c,d) is an attack of $u(\mathscr{A})$ and $c \notin Sup(u,\mathscr{A},E,x)$, $c \in E$, and $d \in Sup(u,\mathscr{A},E,x)$;*
- *$T_2 = \langle Nodes(u(\mathscr{A}),S,E),\ Edges(u(\mathscr{A}),S,E)\rangle$.*

Example 7. *For $\mathtt{AF}_0$, where $E_{id} = \{\mathtt{d},\mathtt{f},\mathtt{h}\}$, and $u = +(\mathtt{h},\mathtt{d})$, we have seen in Example 5 that $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{c}) = \{\mathtt{c},\mathtt{d}\}$. Thus $\mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\mathtt{c}) = \langle\{\mathtt{c},\mathtt{d}\},\{(\mathtt{c},\mathtt{d}),(\mathtt{d},\mathtt{c})\}\rangle \sqcup T_1 \sqcup T_2$ where: $T_1 = \langle\{\mathtt{h},\mathtt{d}\},\{(\mathtt{h},\mathtt{d})\}\rangle$ since $\mathtt{h} \in E_{id}$ does not belong to $Sup(u,\mathtt{AF}_0,E_{id},\mathtt{c})$ while $\mathtt{d} \in Sup(u,\mathtt{AF}_0,E_{id},\mathtt{c})$; and $T_2 = \langle\{\mathtt{a},\mathtt{b},\mathtt{c}\},\{(\mathtt{a},\mathtt{b}),\ (\mathtt{b},\mathtt{a}),(\mathtt{a},\mathtt{c}),(\mathtt{b},\mathtt{c})\}\rangle$ since there are paths starting from the undecided arguments $\mathtt{a}$ and $\mathtt{b}$ ($\{\mathtt{a},\mathtt{b}\} \not\subseteq (E_{id} \cup E^+_{id})$) and ending in $\mathtt{c} \in Sup(u,\mathtt{AF}_0,E_{id},\mathtt{c})$. Thus, $\mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\mathtt{c})$ is the AF shown in Figure 1(b).*

Also, $\mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\star) = \mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\mathtt{c})$. ∎

In general, $\mathsf{CBAF}(u,\mathscr{A},E,g)$ is a subgraph of $\mathsf{CBAF}(u,\mathscr{A},E,\star)$ since $Sup(u,\mathscr{A},E,g) \subseteq Sup(u,\mathscr{A},E,\star)$.

Theorem 2. *Let $\mathscr{A}_0 = \langle A_0,\Sigma_0\rangle$ be an AF, E_0 the ideal extension of $\mathscr{A}_0$, $u = \pm(a,b)$ an update, $\mathscr{A} = u(\mathscr{A}_0)$ the updated AF, and x an argument in A_0. Thus, if $Sup(u,\mathscr{A}_0,E_0,x) \neq \emptyset$ then x is skeptically accepted w.r.t. $\mathscr{A}$ iff it is skeptically accepted w.r.t. the context-based AF $\mathsf{CBAF}(u,\mathscr{A}_0,E_0,x)$.*

Example 8. *Continuing from Example 7, we can conclude that argument $\mathtt{c}$ is not skeptically accepted w.r.t. the updated AF $u(\mathtt{AF}_0)$ because it is not skeptically accepted w.r.t. the context-based AF $\mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\mathtt{c})$ of Figure 1(b) whose preferred extensions are $\{\mathtt{a},\mathtt{h}\}$ and $\{\mathtt{b},\mathtt{h}\}$ (only $\mathtt{h}$ is sceptically accepted w.r.t. the context-based AF).* ∎

Theorem 3. *Let $\mathscr{A}_0 = \langle A_0,\Sigma_0\rangle$ be an AF, E_0 the ideal extension of $\mathscr{A}_0$, $u = \pm(a,b)$ an update, and $\mathscr{A} = u(\mathscr{A}_0)$ the updated AF. Then, the ideal extension E of $\mathscr{A}$ is such that $E = (E_0 \setminus Sup(u,\mathscr{A}_0,E_0,\star)) \cup E'$, where E' is the ideal extension of the context-based AF $\mathsf{CBAF}(u,\mathscr{A}_0,E_0,\star)$.*

Example 9. *Continuing from Example 7, the ideal extension $\{\mathtt{f},\mathtt{h}\}$ of $u(\mathtt{AF}_0)$ is equal to $(\{\mathtt{d},\mathtt{f},\mathtt{h}\} \setminus \{\mathtt{c},\mathtt{d}\}) \cup \{\mathtt{h}\}$ where $\{\mathtt{h}\}$ is the ideal extension of $\mathsf{CBAF}(u,\mathtt{AF}_0,E_{id},\star)$.* ∎

5 Incremental Computation

The results of Theorems 1 and 2, along with those of Theorem 3 and Fact 1, allow us to define SPA (see Algorithm 1) to decide the skeptical acceptance of a goal g w.r.t. an AF $\mathscr{A}_0$ updated by $u = \pm(a,b)$. Given the initial skeptical acceptance $SA_{\mathscr{A}_0}(g)$ of g and the ideal extension E_0 of $\mathscr{A}_0$, both $SA_{u(\mathscr{A}_0)}(g)$ and the ideal extension E of the updated AF $u(\mathscr{A}_0)$ are incrementally computed, thus enabling consecutive invocations of the algorithm to perform sequences of updates.

Algorithm SPA works as follows. First, the supporting set $S_\star = Sup(u,\mathscr{A}_0,E_0,\star)$ is computed at Line 1, and using Theorem 3 the ideal extension E of the updated AF is computed by invoking an external solver $\mathsf{ID\text{-}Solver}(\mathscr{A}_{id})$, computing the ideal extension of the context-based AF $\mathsf{CBAF}(u,\mathscr{A}_0,E_0,\star)$ (Line 3). Then, using Fact 1, if g belongs to E, then g is skeptically accepted and the algorithm returns *true* along with the ideal extension of the updated AF (Line 5). Similarly, if g belongs to the set of arguments attacked by an argument in E, then g is not skeptically accepted and the algorithm returns *false* along with E (Line 7). Otherwise, the set $S_g = Sup(u,\mathscr{A}_0,E_0,g)$ is built (it can be efficiently done by starting from $S_\star$), and it is checked if it is empty. If this is the case, using Theorem 1, we can conclude that the acceptance status of g does not change after the update (Line 10). Otherwise, the context-based AF is built at Line 11 and, using Theorem 2, the skeptical acceptance of g is recomputed by invoking an external solver $\mathsf{SA\text{-}Solver}(\mathscr{A}_{sa},g)$ which tells us if g is skeptically accepted w.r.t. the context-based AF $\mathsf{CBAF}(u,\mathscr{A}_0,E_0,g)$ (Line 12).

Theorem 4. *If $\mathsf{ID\text{-}Solver}$ and $\mathsf{SA\text{-}Solver}$ are sound and complete, for any goal g Algorithm 1 computes $SA_{u(\mathscr{A}_0)}(g)$ w.r.t. the updated AF $u(\mathscr{A}_0)$ and the ideal extension of $u(\mathscr{A}_0)$.*

$\mathsf{SPA\text{-}base}$: A version of SPA not using the ideal extension. $\mathsf{SPA\text{-}base}$ is obtained from SPA by skipping lines 1–7 of Algorithm 1 and assuming $E_0 = \emptyset$ at lines 8 and 11 to compute S_g and $\mathscr{A}_{sa}$ respectively. Also, no ideal extension is returned (i.e., $E = \perp$). Notice that, similarly to $\mathsf{SPA\text{-}base}$, SPA does not use the information provided by the initial ideal extension when $E_0 = \emptyset$, though SPA always incrementally computes the ideal extension of the updated AF.

Multiple Updates

The application of a set U of updates can be reduced to performing a single attack update as follows. Let $\mathscr{A} = \langle A,\Sigma\rangle$ be an AF, and

- $\Sigma^+ = \{(a_1,b_1),...,(a_n,b_n)\} \subseteq (A \times A) \setminus \Sigma$, and
- $\Sigma^- = \{(a'_1,b'_1),...,(a'_m,b'_m)\} \subseteq \Sigma$

such that $\Sigma^+ \cap \Sigma^- = \emptyset$ be two sets of attacks. Let $U = \{+(a_i,b_i)\,|(a_i,b_i) \in \Sigma^+\} \cup \{-(a_j,b_j)\,|(a_j,b_j) \in \Sigma^-\}$ be a set of updates. We define $\mathscr{A}^U = \langle A^U,\Sigma^U\rangle$ where:

Algorithm 1 SPA($\mathcal{A}_0, g, SA_{\mathcal{A}_0}(g), u, E_0$)

```
Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, argument $g \in A_0$,
       skeptical acceptance $SA_{\mathcal{A}_0}(g)$ of $g$ w.r.t. $\mathcal{A}_0$,
       update $u = \pm(a,b)$, ideal extension $E_0$ of $\mathcal{A}_0$.
Output: skeptical acceptance $SA_{u(\mathcal{A}_0)}(g)$ of $g$ w.r.t. $u(\mathcal{A}_0)$,
        ideal extension $E$ of $u(\mathcal{A}_0)$.
Let $S_\star = Sup(u, \mathcal{A}_0, E_0, \star)$
Let $\mathcal{A}_{id} =$ CBAF$(u, \mathcal{A}_0, E_0, \star)$
Let $E = (E_0 \setminus S_\star) \cup$ ID-Solver$(\mathcal{A}_{id})$
if $g \in E$ then
    return $\langle true, E \rangle$
if $g \in E^+$ then
    return $\langle false, E \rangle$
Let $S_g = Sup(u, \mathcal{A}_0, E_0, g)$
if $S_g$ is empty then
    return $\langle SA_{\mathcal{A}_0}(g), E \rangle$
Let $\mathcal{A}_{sa} =$ CBAF$(u, \mathcal{A}_0, E_0, g)$
return $\langle$SA-Solver$(\mathcal{A}_{sa}, g), E \rangle$
```

- $A^U = A \cup \{x_i, y_i \mid +(a_i, b_i) \in U\} \cup \{x'_j, y'_j \mid -(a_j, b_j) \in U\} \cup \{v, w, w'\}$, where all x_i, y_i, x'_j, y'_j, w, w', and v are new arguments not occurring in A, and
- $\Sigma^U = (\Sigma \setminus \Sigma^-) \cup \{(a_i, x_i), (x_i, y_i), (y_i, b_i) \mid +(a_i, b_i) \in U\} \cup \{(a_j, x'_j), (x'_j, y'_j), (y'_j, b_j) \mid -(a_j, b_j) \in U\} \cup \{(w, y_i) \mid +(a_i, b_i) \in U\} \cup \{(w', y'_j) \mid -(a_j, b_j) \in U\} \cup \{(w, w')\}$.

Proposition 1. *Let $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$ be an AF, and U a set of updates. Let $\mathcal{A}$ be the AF obtained from $\mathcal{A}_0$ by performing all updates in U on it. Then,*

- *$E = E^U \cap A_0$ is the ideal extension of $\mathcal{A}$ iff E^U is the ideal extension of $+(v, w)(\mathcal{A}^U)$.*
- *Given an argument g of $\mathcal{A}_0$, $SA_{\mathcal{A}}(g) = SA_{+(v,w)(\mathcal{A}^U)}(g)$.*

6 Implementation and Experiments

We have implemented a C++ prototype and compared our incremental technique with *ArgSemSAT* [Cerutti *et al.*, 2014], the solver that won the last ICCMA competition for the task DS-`pr` of determining the skeptical preferred acceptance.

Datasets and Methodology

We used benchmarks from the DS-`pr` track of ICCMA'17, that is, the dataset $A2$ consisting of 50 AFs with a number of arguments $|A| \in [61, 20K]$ and a number of attacks $|\Sigma| \in [97, 358K]$, and the dataset $A3$ consisting of 100 AFs with $|A| \in [39, 100K]$ and $|\Sigma| \in [72, 1.26M]$.

For each AF $\mathcal{A}_0$ in the dataset, we randomly selected an update u (or a set U of updates), and an argument g. Then, we computed $SA_{u(\mathcal{A}_0)}(g)$ by using 1) SPA, that is Algorithm 1 where ID-Solver is *pyglaf* [Alviano, 2017] and SA-Solver is *ArgSemSAT*; 2) SPA-base where only *ArgSemSAT* is used; and 3) *ArgSemSAT* (from scratch).

For AF $\mathcal{A}_0$, update u, and argument g, let t_A and t_B be the amount of time required by SPA and SPA-base, respectively, to compute $SA_{u(\mathcal{A}_0)}(g)$. Let t_S be the time required by *ArgSemSAT* to compute $SA_{u(\mathcal{A}_0)}(g)$ from scratch. Then, the *improvements* of SPA and SPA-base over *ArgSemSAT* are defined as $\frac{t_S}{t_A}$ and $\frac{t_S}{t_B}$, respectively. Thus, an improvement equal to x means that the incremental computation is x times faster than the computation from scratch.

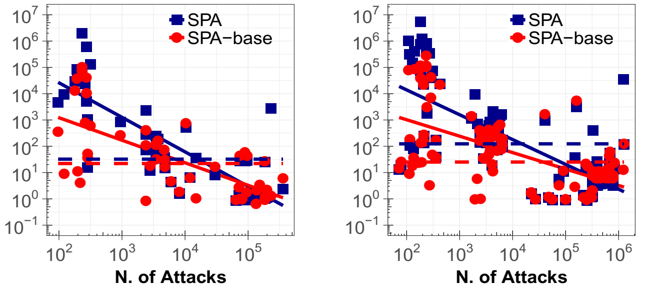
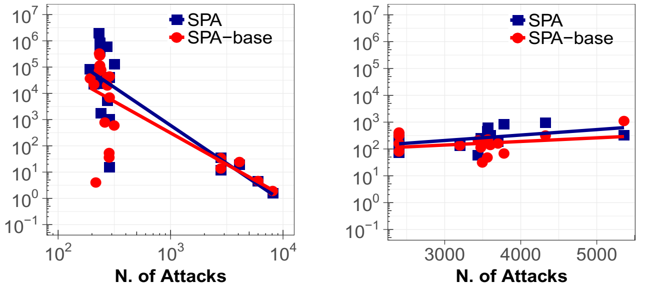


Figure 2: (Experiment 1). Improvement of SPA and SPA-base over *ArgSemSAT* on datasets $A2$ (left-hand side) and $A3$ (right-hand side)



(a) (b)

Figure 3: (Experiment 2). Improvement of SPA and SPA-base over *ArgSemSAT* on AFs from datasets $A2$ and $A3$ having (a) a number of arguments in $[160, 200]$; (b) an average degree in $[5, 10]$

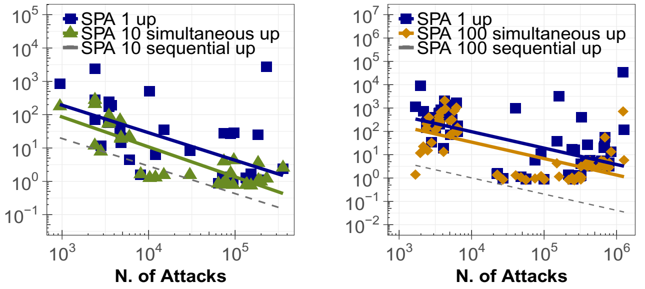


Figure 4: (Experiment 3). Improvement of SPA over *ArgSemSAT* for 10 updates (LHS) and 100 updates (RHS). Dashed gray lines are improvements for 10 and 100 updates applied sequentially

In the figures, each data-point is the average of 10 runs, and solid lines are obtained by linear regression.

Experiment 1. Figure 2 reports the improvement (log scale) of SPA and SPA-base over *ArgSemSAT* on datasets $A2$ (LHS) and $A3$ (RHS) for single updates versus the size of the AFs, i.e., the number of attacks. Both SPA and SPA-base significantly outperform the computation from scratch, though the improvement decreases as the number of attacks increases. This behavior, which is in line with that of algorithms for computing argumentation semantics in the static setting [Liao, 2013; Liao and Huang, 2013; Alfano *et al.*, 2019], is analysed in Experiment 2. Considering the averages of the improvements, SPA and SPA-base turn out to be 5 and 4 orders of magnitude faster than *ArgSemSAT*, respectively. However, as this can be skewed by extremely large values of improvements (e.g. 10^6), we also considered the medians of improvements for SPA (32 on $A2$, 134 on $A3$) and SPA-base (27 on $A2$, 40 on $A3$) (see dashed line in Figure 2), which confirm the significance of the gain in efficiency. The experiments show that SPA is generally faster than SPA-base, except for a few AFs whose initial ideal extension is empty.

Experiment 2. We analysed the performances of SPA and SPA-base by varying the number of attacks and keeping almost constant either the number of arguments or the average degree (i.e., N. of attacks / N. of arguments). To this end, we selected as many AFs as possible from the two datasets having these properties. Figure 3 reports the improvement for AFs having (*a*) a number of arguments in $[160, 200]$ and (*b*) an average degree in $[5, 10]$. Figure 3(a) shows that the performance gets worse when the increasing of the number of attacks is mainly due to the increasing of nodes' degree (in our experiment it varies from 1.5 to 40) and the AFs become more and more dense. Figure 3(b) shows that when the average degree does not change significantly and the increasing of the number of attacks is mainly caused by the increasing of the number of arguments (in our experiment it has more than doubled, from $2.4K$ to $5.4K$), the improvement does not decrease. This can be explained by looking at the ratio ρ between the size of the context-based AF and that of the initial AF. Indeed, ρ increases from 4% to 95% when the average degree of the input AFs varies from 2 to 40 in Figure 3(a). On the other hand, for Figure 3(b), ρ is almost constant (i.e., in $[60\%, 65\%]$). Thus, the performance gets worse when the ratio between the size of the context-based AF and that of the initial AF becomes very large because of the increasing density of the initial AFs.

Experiment 3. Figure 4 reports the improvement of SPA over *ArgSemSAT* on datasets $A2$ (LHS) and $A3$ (RHS) for 10 and 100 updates performed simultaneously. To preserve the structure of the AFs in the datasets, we changed at most 1% (resp., 10%) of the number of attacks for the AFs of $A2$ (resp., $A3$), that is, we considered AFs whose size is greater than $1K$. Figure 4 also reports the improvement for 1 update, as well as for 10 and 100 updates applied sequentially, i.e., one after another (see dashed lines). The results show that SPA remains faster than the competitor even when 10 or 100 updates are performed simultaneously. Moreover, despite the overhead of the construction given before Proposition 1, applying updates simultaneously is faster than applying them sequentially.

7 Conclusions and Future Work

There has been an extensive body of work on managing changes in argumentation (a recent survey can be found in [Doutre and Mailly, 2018]). Besides the works cited in the introduction, other significant efforts coping with dynamics aspects of AFs include [Baroni *et al.*, 2005; Boella *et al.*, 2009; Cayrol *et al.*, 2008; Amgoud and Vesic, 2012; Bisquert *et al.*, 2013]. Similarly to what is done in this paper, some approaches focused on local computation in dynamic AFs [Liao *et al.*, 2011; Baroni *et al.*, 2014; Greco and Parisi, 2016a; 2016b; Alfano *et al.*, 2017] but with the aim of recomputing extensions. To the best our knowledge, this is the first paper proposing an efficient technique for the incremental computation of skeptical acceptance in dynamic AFs.

We have proposed a new algorithm, called SPA, which incrementally computes the skeptical preferred acceptance by maintaining the ideal extension (that can be considered the best available short-cut to compute preferred extensions when it is known [Dunne *et al.*, 2016]). For a better understanding of the relevance of computing the ideal extension, we have also considered its simpler variant SPA-base which does not rely on the ideal extension. Both algorithms outperform the computation from scratch, and SPA is generally faster than SPA-base. However, as the experiments showed, SPA may be slower than SPA-base when the initial ideal extension is empty. Thus, a first direction for future work is devising heuristics to take advantages of both algorithms.

Analogously to classical AF solvers, our approach allows us to determine the skeptical acceptance of a single argument. However, the definition of supporting set and context-based AF can be extended to sets of arguments—a way to do it is using the union of the supporting sets to compute the context-based AF. As a further line for future work, we plan to extend our algorithm to take as input a set of arguments. In this direction, SPA always computes the ideal extension, which being contained in every preferred extension, already provides additional skeptical preferred arguments other than the goal.

Finally, we plan to extend our technique to other argumentation semantics. In this regard, we point out that our technique can be used as it is for skeptical acceptance under the *semi-stable* semantics, but a deeper investigation is needed for extending the technique to deal with the *stable* semantics.

References

[Alfano *et al.*, 2017] Gianvincenzo Alfano, Sergio Greco, and Francesco Parisi. Efficient computation of extensions for dynamic abstract argumentation frameworks: An incremental approach. In *IJCAI*, pages 49–55, 2017.

[Alfano *et al.*, 2019] Gianvincenzo Alfano, Sergio Greco, and Francesco Parisi. On scaling the enumeration of the preferred extensions of abstract argumentation frameworks. In *SAC*, pages 1147–1153, 2019.

[Alviano, 2017] Mario Alviano. The pyglaf argumentation reasoner. In *ICLP*, pages 2:1–2:3, 2017.

[Amgoud and Vesic, 2012] Leila Amgoud and Srdjan Vesic. Revising option status in argument-based decision systems. *J. Log. Comput.*, 22(5):1019–1058, 2012.

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

[Baroni *et al.*, 2011] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[Baroni *et al.*, 2014] Pietro Baroni, Massimiliano Giacomin, and Beishui Liao. On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method. *Artif. Intell.*, 212:104–115, 2014.

[Baumann and Brewka, 2010] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. In *COMMA*, pages 75–86, 2010.

[Baumann, 2011] Ringo Baumann. Splitting an argumentation framework. In *LPNMR*, pages 40–53, 2011.

[Bench-Capon and Dunne, 2007] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in AI. *Artif. Intell.*, 171(10–15):619 – 641, 2007.

[Bisquert *et al.*, 2013] Pierre Bisquert, Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Characterizing change in abstract argumentation systems. In *Trends in Belief Revision and Argumentation Dynamics*, volume 48, pages 75–102. 2013.

[Bistarelli *et al.*, 2018a] Stefano Bistarelli, Francesco Faloci, Francesco Santini, and Carlo Taticchi. Studying dynamics in argumentation with Rob. In *COMMA*, pages 451–452, 2018.

[Bistarelli *et al.*, 2018b] Stefano Bistarelli, Lars Kotthoff, Francesco Santini, and Carlo Taticchi. Containerisation and dynamic frameworks in ICCMA'19. In *SAFA Workshop co-located with COMMA*, pages 4–9, 2018.

[Boella *et al.*, 2009] Guido Boella, Souhila Kaci, and Leendert W. N. van der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In *ECSQARU*, pages 107–118, 2009.

[Caminada *et al.*, 2016] Martin W. A. Caminada, Wolfgang Dvorák, and Srdjan Vesic. Preferred semantics as socratic discussion. *J. Log. Comput.*, 26(4):1257–1292, 2016.

[Cayrol *et al.*, 2008] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. Revision of an argumentation system. In *KR*, pages 124–134, 2008.

[Cerutti *et al.*, 2014] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. ArgSemSAT: Solving argumentation problems using SAT. In *COMMA*, pages 455–456, 2014.

[Charwat *et al.*, 2015] Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *Artif. Intell.*, 220:28–63, 2015.

[Doutre and Mailly, 2018] Sylvie Doutre and Jean-Guy Mailly. Constraints and changes: A survey of abstract argumentation dynamics. *Argument & Computation*, 9(3):223–248, 2018.

[Dung *et al.*, 2007] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[Dung, 1995] 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.

[Dunne and Wooldridge, 2009] Paul E. Dunne and Michael Wooldridge. Complexity of abstract argumentation. In *Argumentation in AI*, pages 85–104. 2009.

[Dunne *et al.*, 2016] Paul E. Dunne, Christof Spanring, Thomas Linsbichler, and Stefan Woltran. Investigating the relationship between argumentation semantics via signatures. In *IJCAI*, pages 1051–1057, 2016.

[Dunne, 2009] Paul E. Dunne. The computational complexity of ideal semantics. *Artif. Intell.*, 173(18):1559–1591, 2009.

[Falappa *et al.*, 2011] Marcelo A. Falappa, Alejandro Javier Garcia, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. On the evolving relation between belief revision and argumentation. *The Knowledge Engineering Review*, 26(1):35–43, 2011.

[Greco and Parisi, 2016a] Sergio Greco and Francesco Parisi. Efficient computation of deterministic extensions for dynamic abstract argumentation frameworks. In *ECAI*, pages 1668–1669, 2016.

[Greco and Parisi, 2016b] Sergio Greco and Francesco Parisi. Incremental computation of deterministic extensions for dynamic argumentation frameworks. In *JELIA*, pages 288–304, 2016.

[Kökciyan *et al.*, 2017] Nadin Kökciyan, Nefise Yaglikci, and Pinar Yolum. An argumentation approach for resolving privacy disputes in online social networks. *ACM Trans. Internet Techn.*, 17(3):1–22, 2017.

[Kröll *et al.*, 2017] Markus Kröll, Reinhard Pichler, and Stefan Woltran. On the complexity of enumerating the extensions of abstract argumentation frameworks. In *IJCAI*, pages 1145–1152, 2017.

[Liao and Huang, 2013] Beishui Liao and Huaxin Huang. Partial semantics of argumentation: basic properties and empirical results. *J. Log. Comput.*, 23(3):541–562, 2013.

[Liao *et al.*, 2011] Bei Shui Liao, Li Jin, and Robert C. Koons. Dynamics of argumentation systems: A division-based method. *Artif. Intell.*, 175(11):1790–1814, 2011.

[Liao, 2013] Beishui Liao. Toward incremental computation of argumentation semantics: A decomposition-based approach. *Ann. Math. Artif. Intell.*, 67(3-4):319–358, 2013.

[Oikarinen and Woltran, 2011] Emilia Oikarinen and Stefan Woltran. Characterizing strong equivalence for argumentation frameworks. *Artif. Intell.*, 175(14-15):1985–2009, 2011.

[Rahwan and Simari, 2009] Iyad Rahwan and Guillermo R. Simari. *Argumentation in AI*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[Thimm and Villata, 2017] Matthias Thimm and Serena Villata. The first international competition on computational models of argumentation: Results and analysis. *Artif. Intell.*, 252:267–294, 2017.