An Efficient Algorithm for Skeptical Preferred Acceptance in Dynamic Argumentation Frameworks

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi

Department of Informatics, Modeling, Electronics and System Engineering
University of Calabria
Italy
{g.alfano, greco, fparisi}@dimes.unical.it

28^{th} International Joint Conference on Artificial Intelligence

August 10-16, 2019

Macao, China

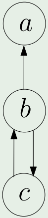
Argumentation in AI

- A general way for representing arguments and relationships (attacks) between them
- It allows representing dialogues, making decisions, and handling inconsistency and uncertainty

Abstract Argumentation Framework (AF) [Dung1995]: arguments are abstract entities (no attention is paid to their internal structure) that may attack and/or be attacked by other arguments

Example (a simple AF)

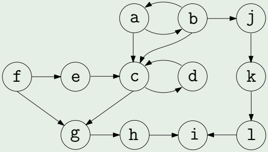
a =	Our friends will have great fun at our party on Saturday
b =	Saturday will rain (according to the weather forecasting service 1)
c =	Saturday will be sunny (according to the weather forecasting service 2)



Argumentation Semantics

- Several semantics (such as *preferred*, and *ideal*) have been proposed to identify “reasonable” sets of arguments, called *extensions*.
- A preferred extension of an AF $\mathcal{A}$ is a maximal admissible set of $\mathcal{A}$.
- The ideal extension of $\mathcal{A}$ is the biggest admissible set of $\mathcal{A}$ which is contained in every preferred extension of $\mathcal{A}$.

Example (AF $\mathcal{A}_0$)



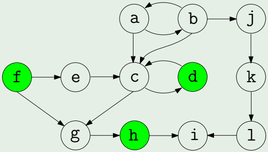
Semantic $\mathcal{S}$	Set of extensions of $\mathcal{A}_0$
preferred ($\mathtt{pr}$)	$\{\{\mathtt{a},\mathtt{d},\mathtt{f},\mathtt{h},\mathtt{j},\mathtt{l}\},\{\mathtt{b},\mathtt{d},\mathtt{f},\mathtt{h},\mathtt{k}\}\}$
ideal ($\mathtt{id}$)	$\{\{\mathtt{d},\mathtt{f},\mathtt{h}\}\}$

- An argument *g* is skeptically preferred accepted w.r.t. $\mathcal{A}$ (denoted as $SA_{\mathcal{A}}(\mathtt{g}) = true$) iff it appears in every $\mathtt{pr}$-extension of $\mathcal{A}$.
- In our example $SA_{\mathcal{A}}(\mathtt{d}) = SA_{\mathcal{A}}(\mathtt{f}) = SA_{\mathcal{A}}(\mathtt{h}) = true$.

Argumentation Semantics

- Several semantics (such as *preferred*, and *ideal*) have been proposed to identify “reasonable” sets of arguments, called *extensions*.
- A preferred extension of an AF $\mathcal{A}$ is a maximal admissible set of $\mathcal{A}$.
- The ideal extension of $\mathcal{A}$ is the biggest admissible set of $\mathcal{A}$ which is contained in every preferred extension of $\mathcal{A}$.

Example (AF $\mathcal{A}_0$)



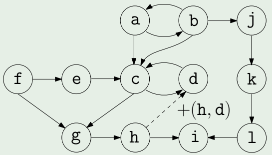
Semantic $\mathcal{S}$	Set of extensions of $\mathcal{A}_0$
preferred ($\mathtt{pr}$)	$\{\{\mathtt{a},\mathtt{d},\mathtt{f},\mathtt{h},\mathtt{j},\mathtt{l}\},\{\mathtt{b},\mathtt{d},\mathtt{f},\mathtt{h},\mathtt{k}\}\}$
ideal ($\mathtt{id}$)	$\{\{\mathtt{d},\mathtt{f},\mathtt{h}\}\}$

- An argument *g* is skeptically preferred accepted w.r.t. $\mathcal{A}$ (denoted as $SA_{\mathcal{A}}(\mathtt{g}) = true$) iff it appears in every $\mathtt{pr}$-extension of $\mathcal{A}$.
- In our example $SA_{\mathcal{A}}(\mathtt{d}) = SA_{\mathcal{A}}(\mathtt{f}) = SA_{\mathcal{A}}(\mathtt{h}) = true$.

Dynamic Abstract Argumentation Frameworks

- Most argumentation frameworks are dynamic systems, which are often updated by adding/removing arguments/attacks.
- For each semantics, extensions may change if we update the initial AF by adding/removing arguments/attacks.

Example (Updated AF $\mathcal{A} = +(h, d)(\mathcal{A}_0)$)



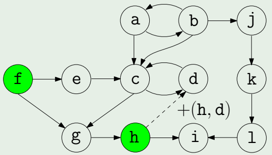
$\mathcal{S}$	Set of extensions of $\mathcal{A}_0$	Set of extensions of $\mathcal{A}$
pr	$\{\{a, d, f, h, j, l\}, \{b, d, f, h, k\}\}$	?
id	$\{\{d, f, h\}\}$	?

- Should we recompute the skeptical acceptance of an argument w.r.t. an updated AF from scratch?

Dynamic Abstract Argumentation Frameworks

- Most argumentation frameworks are dynamic systems, which are often updated by adding/removing arguments/attacks.
- For each semantics, extensions may change if we update the initial AF by adding/removing arguments/attacks.

Example (Updated AF $\mathcal{A} = +(h, d)(\mathcal{A}_0)$)



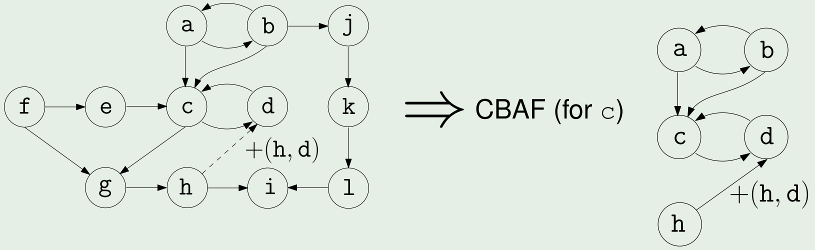
$\mathcal{S}$	Set of extensions of $\mathcal{A}_0$	Set of extensions of $\mathcal{A}$
pr	$\{\{a, d, f, h, j, l\}, \{b, d, f, h, k\}\}$	$\{\{a, \mathbf{f}, \mathbf{h}, j, l\}, \{b, \mathbf{f}, \mathbf{h}, k\}\}$
id	$\{\{d, f, h\}\}$	$\{\{f\}\}$

- **Should we recompute the skeptical acceptance of an argument w.r.t. an updated AF from scratch?**

Context-based AF (CBAF)

- We show that the skeptical preferred acceptance of an argument w.r.t an updated AF can be efficiently computed by looking only at a small part of the AF, called the *context-based* AF, which contains arguments whose acceptance status may change after the update.

Example (From the updated AF to the CBAF)



Incremental Algorithm

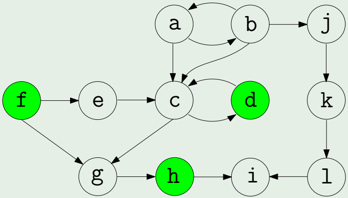
1. We formally define the *CBAF*
   - Sub-AF consisting of the arguments whose status could change after an update
   - It depends on both the update, the initial ideal extension, and the goal argument.
2. We present an incremental algorithm for recomputing the skeptical preferred acceptance of a goal argument of an updated AF
   - It calls state of the art solvers to compute the skeptical preferred acceptance of the goal argument and the ideal extension of the CBAF
   - It incrementally maintains the ideal extension using the CBAF.
3. We present a thorough experimental analysis showing the effectiveness of our approach
   - Our technique outperforms the computation from scratch even when using the best available solver for determine the skeptical preferred acceptance.

Outline

Supporting set: Intuition

- $Sup(u, \mathcal{A}, E, g)$ is the set of arguments whose status may change after performing update u and s.t. they may imply a change of the status of g.
- Given $u = \pm(a, b)$, an argument is *steady* if it is attacked by an argument appearing in the initial ideal extension that is not reachable from b.
- Informal definition: $Sup(u, \mathcal{A}, E, g)$ for $u = \pm(a, b)$ and g consists of the arguments that (*i*) can be reached from b without using any steady argument; and (*ii*) allow to reach the goal g by using only the selected arguments.

Example (For update $u = +(\mathrm{h}, \mathrm{d})$)

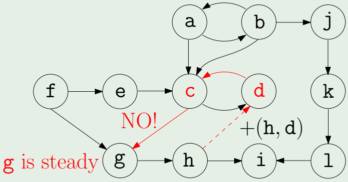


g is steady since it is attacked by $\mathrm{f} \in E_{id}$ and f is not reachable from d.

Supporting set: Intuition

- *Sup*$(u, \mathcal{A}, E, g)$ is the set of arguments whose status may change after performing update u and s.t. they may imply a change of the status of g.
- Given $u = \pm(a, b)$, an argument is *steady* if it is attacked by an argument appearing in the initial ideal extension that is not reachable from b.
- Informal definition: *Sup*$(u, \mathcal{A}, E, g)$ for $u = \pm(a, b)$ and g consists of the arguments that (*i*) can be reached from b without using any steady argument; and (*ii*) allow to reach the goal g by using only the selected arguments.

Example (For update $u = +(\mathtt{h}, \mathtt{d})$)



For the goal c the supporting set is:
$Sup(u, \mathrm{AF}_0, E_{id}, \mathtt{c}) = \{\mathtt{c}, \mathtt{d}\}$

Supporting set: Formal Definition

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

– $Sup_0(u, \mathcal{A}, E, g) = \begin{cases} \emptyset & \text{if } u = +(a, b) \wedge b \in (E(u))^+; \\ \emptyset & \text{if } b \notin Reach^{-1}_{H(\mathcal{A},u)}(g); \\ \{b\} & \text{otherwise.} \end{cases}$

– $Sup_{i+1}(u, \mathcal{A}, E, g) = Sup_i(u, \mathcal{A}, E, g) \cup \{y \mid \exists (x, y) \in \Sigma \ s.t. x \in Sup_i(u, \mathcal{A}, E, g) \ \wedge y \in Reach^{-1}_{H(\mathcal{A},u)}(g) \wedge y \notin Std_{\mathcal{A}}(u)\}$.

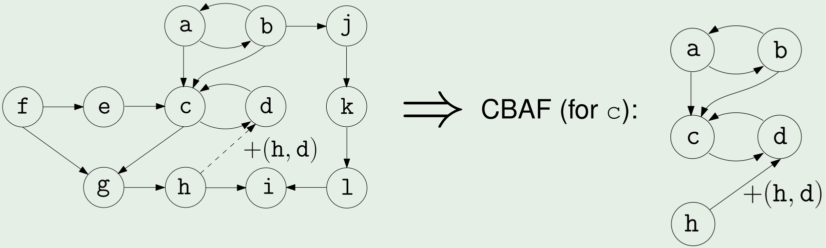
Let n be the natural number such that $Sup_n(u, \mathcal{A}, E, g) = Sup_{n+1}(u, \mathcal{A}, E, g)$.

- The *supporting set* is $Sup(u, \mathcal{A}, E, g) = Sup_n(u, \mathcal{A}, E, g) \cap Reach^{-1}_G(g)$ where $G = \Pi(Sup_n(u, \mathcal{A}, E, g), H(\mathcal{A}, u))$ is the restriction of $H(\mathcal{A}, u)$ to $Sup_n(u, \mathcal{A}, E, g)$.
- If g is not specified, the supporting set, denoted as $Sup(u, \mathcal{A}, E, \star)$, is defined as $Sup(u, \mathcal{A}, E, g)$ except that all the checks concerning $Reach^{-1}$ are omitted.

Context-based AF (CBAF)

- Using the supporting set we define the Context-based AF (CBAF).
- It is a restriction of the AF used to compute:
  1) The status of the goal after an update
  2) The updated ideal extension

Example (From the updated AF to the CBAF)



Incremental Algorithm

Algorithm 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⋆ = Sup(u, A0, E0, ⋆) // Supporting set for computing the updated ideal extension
Let Aid = CBAF(u, A0, E0, ⋆)// CBAF for computing the updated ideal extension
Let E = (E0 \ S⋆) ∪ ID-Solver(Aid)// Computing the updated ideal extension using the CBAF
if g ∈ E then
    return ⟨true, E⟩// g is in the ideal extension, thus skeptical accepted
if g ∈ E+ then
    return ⟨false, E⟩// g is attacked by the ideal extension, thus it is not skeptically accepted
Let Sg = Sup(u, A0, E0, g) // Supporting set for determining the skeptical acceptance of g
if Sg is empty then
    return ⟨SA_A0(g), E⟩ // If the supporting set is empty, then the skeptical acceptance is preserved (result in the paper)
Let Asa = CBAF(u, A0, E0, g)// CBAF for determining the skeptical acceptance of g
return ⟨SA-Solver(Asa, g), E⟩// If the supporting set is not empty, it suffices to compute the skeptical acceptance only on the CBAF (result in the paper)
```

Outline

Datasets and Metodology

Datasets: ICCMA'17 benchmarks for the task DS-pr of determining the skeptical preferred acceptance.

- $A2$ consists of 50 $A \in [61, 20K]$ and $\Sigma \in [97, 358K]$
- $A3$ consists of 100 $A \in [39, 100K]$ and $\Sigma \in [72, 1.26M]$.

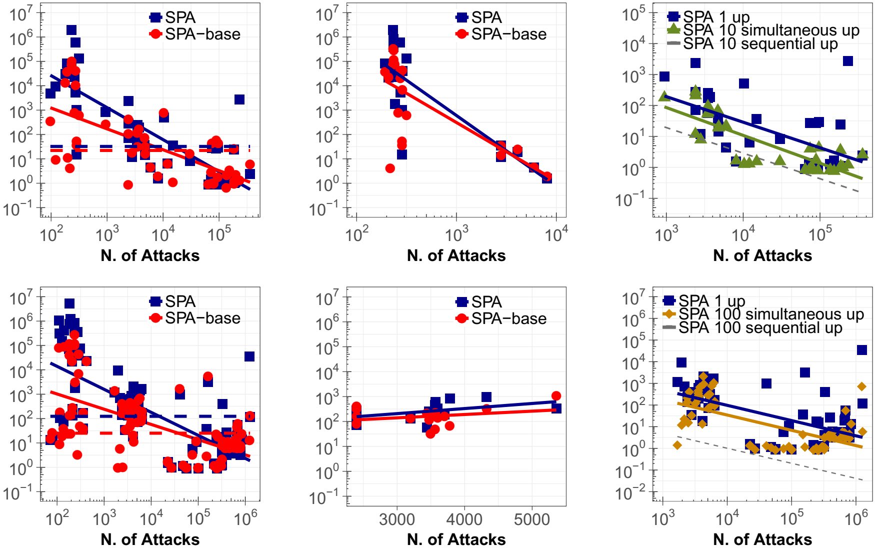
Methodology: For each AF we randomly selected an update u (or a set), and a goal argument g. Then, we computed $SA_{u(\mathcal{A}_0)}(g)$ by using

1. SPA, where ID-Solver is pyglaf [Alviano, 2017] and SA-Solver is ArgSemSAT [Cerutti et al., 2014], the solver that won the the DS-pr track;
2. SPA-base where the ideal extension is not used; and
3. ArgSemSAT (from scratch).

We report on the improvements: ■ $\frac{Time(3)}{Time(1)}$ ■ $\frac{Time(3)}{Time(2)}$

Experimental Results

Experiment 1 **Experiment 2** **Experiment 3**



Results

- **Experiment 1:**
  - SPA and SPA-base turn out to be on average 5 and 4 orders of magnitude faster than ArgSemSAT, respectively—dashed lines reports median values (32 on A2, 134 on A3) and SPA-base (27 on A2, 40 on A3).
  - SPA generally faster than SPA-base—not so if initial ideal extension is empty.
- **Experiment 2:**
  - We analyzed the performances of SPA and SPA-base by varying the number of attacks and keeping constant either the number of arguments or the average degree.
  - 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—from 4% to 95%.
- **Experiment 3:**
  - SPA remains faster than the competitor even when 10 or 100 updates are performed simultaneously.
  - Applying updates simultaneously is faster than applying them sequentially (dashed grey lines).

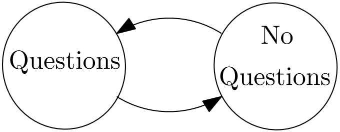
Outline

Conclusions and Future Work

- To the best our knowledge, this is the first paper proposing an efficient technique for the incremental computation of skeptical acceptance in dynamic AFs.
- The technique can be used for general (multiple) updates
- We identified a tighter portion of the updated AF to be examined for the recomputation.
- Both SPA and SPA-base 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.
- We plan to extend our technique to other argumentation semantics.

see you at the poster!

Thanks...



Which one is your preferred extension?!