Computing Stable and Preferred Extensions of Dynamic Bipolar Argumentation Frameworks

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi

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

1st Workshop on Advances In Argumentation In Artificial Intelligence

November 14-17, 2017

Bari, Italy

Argumentation in AI

- A general way for representing arguments and relationships between them
- It allows representing dialogues, making decisions, and handling inconsistency and uncertainty
- **Abstract Bipolar Argumentation Framework** (BAF)

Example (a simple BAF)

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

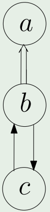
Semantics for Bipolar Argumentation Frameworks: "reasonable" sets of arguments, called *extensions*. We focused on preferred and stable.

Argumentation in AI

- A general way for representing arguments and relationships between them
- It allows representing dialogues, making decisions, and handling inconsistency and uncertainty
- **Abstract Bipolar Argumentation Framework** (BAF)

Example (a simple BAF)

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



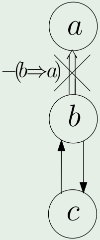
Semantics for Bipolar Argumentation Frameworks: “reasonable” sets of arguments, called *extensions*. We focused on preferred and stable.

Dynamic Argumentation Frameworks

Many argumentation frameworks are highly dynamic in practice.

Example (a simple BAF)

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



Should we recompute the semantics from scratch?

Contributions

1) We identify early-termination conditions.

2) We define an incremental algorithm for computing extensions of dynamic BAFs by leveraging on the incremental technique proposed in [Alfano,Greco,Parisi IJCAI 2017].

3) Experimental analysis comparing with fastest solvers from ICCMA 2015.

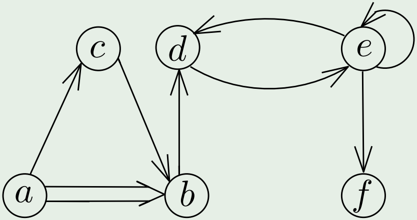
Outline

Bipolar Argumentation Frameworks

- An *abstract bipolar argumentation framework* (*BAF* for short) [Amgoud et al. 2004] is a triple $\langle A, \Sigma, \Pi \rangle$, where
  - $A \subseteq Arg$ is a (finite) set whose elements are referred to as *arguments*,
  - $\Sigma \subseteq A \times A$ is a binary relation over A whose elements are called *attacks*,
  - $\Pi \subseteq A \times A$ is a binary relation over A whose elements are called *supports*, and
- $\Sigma \cap \Pi = \emptyset$. Thus, a Dung's argumentation framework (AF) [Dung 1995] is a BAF of the form $\langle A, \Sigma, \emptyset \rangle$.

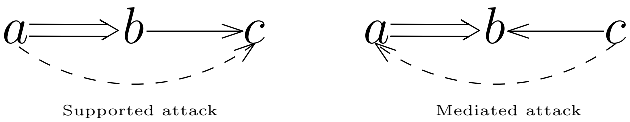
Example (BAF)

$A = \{a, b, c, d, e, f\}$
$\Sigma = \{(a, c),\ (c, b),\ (b, d),\ (d, e)$
$(e, d),\ (e, e),\ (e, f)\}$
$\Pi = \{(a, b)\}$



Implicit Attacks

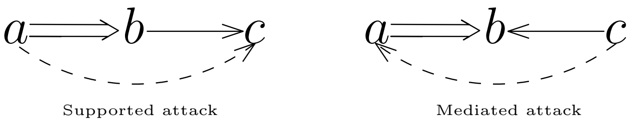
The coexistence of the support and attack relations in BAFs entails that new kinds of “implicit” attacks should be considered.



- A set $S \subseteq A$ *set-attacks* an argument $b \in A$ iff there exists a supported or mediated attack for b by an argument $a \in S$.
- $S \subseteq A$ *defends* an argument $a \in A$ iff for each $b \in A$ such that $\{b\}$ set-attacks a, it is the case that S set-attacks b

Implicit Attacks

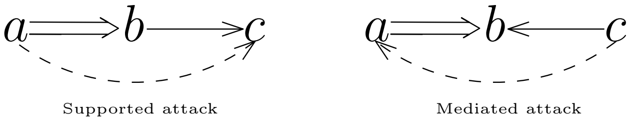
The coexistence of the support and attack relations in BAFs entails that new kinds of “implicit” attacks should be considered.



- A set $S \subseteq A$ *set-attacks* an argument $b \in A$ iff there exists a supported or mediated attack for b by an argument $a \in S$.
- $S \subseteq A$ *defends* an argument $a \in A$ iff for each $b \in A$ such that $\{b\}$ set-attacks a, it is the case that S set-attacks b

Implicit Attacks

The coexistence of the support and attack relations in BAFs entails that new kinds of “implicit” attacks should be considered.



- A set $S \subseteq A$ *set-attacks* an argument $b \in A$ iff there exists a supported or mediated attack for b by an argument $a \in S$.
- $S \subseteq A$ *defends* an argument $a \in A$ iff for each $b \in A$ such that $\{b\}$ set-attacks a, it is the case that S set-attacks b

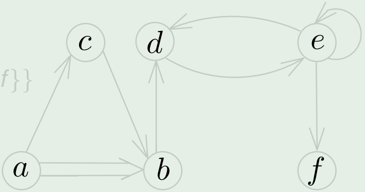
Semantics for Bipolar Abstract Argumentation

A semantics identifies "reasonable" sets of arguments, called *extensions*

- A set $S \subseteq A$ is *conflict-free* iff there are no two arguments $a, b \in S$ such that $\{a\}$ set-attacks b.
- A conflict-free set $S \subseteq A$ is said to be ***admissible*** iff it defends all of its arguments.
- A ***preferred extension*** ($\texttt{pr}$) for a BAF is an admissible set which is maximal (w.r.t $\subseteq$).
- A conflict-free set $S \subseteq A$ is a ***stable extension*** ($\texttt{st}$), if and only if it set-attacks all the arguments in $A \setminus S$.

Example (semantics for BAF)

admissible sets: $\{\{\emptyset\}, \{a\}, \{c\}, \{a, b\}, \{c, d\}, \{c, d, f\}\}$
preferred extensions: $\{\{a, b\}, \{c, d, f\}\}$
stable extension: $\{\{c, d, f\}\}$



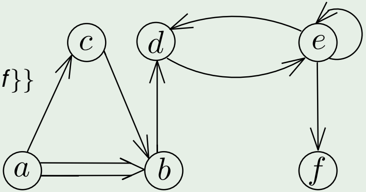
Semantics for Bipolar Abstract Argumentation

A semantics identifies "reasonable" sets of arguments, called *extensions*

- A set $S \subseteq A$ is *conflict-free* iff there are no two arguments $a, b \in S$ such that $\{a\}$ set-attacks b.
- A conflict-free set $S \subseteq A$ is said to be ***admissible*** iff it defends all of its arguments.
- A ***preferred extension*** ($\mathtt{pr}$) for a BAF is an admissible set which is maximal (w.r.t $\subseteq$).
- A conflict-free set $S \subseteq A$ is a ***stable extension*** ($\mathtt{st}$), if and only if it set-attacks all the arguments in $A \setminus S$.

Example (semantics for BAF)

admissible sets: $\{\{\emptyset\}, \{a\}, \{c\}, \{a, b\}, \{c, d\}, \{c, d, f\}\}$
preferred extensions: $\{\{a, b\}, \{c, d, f\}\}$
stable extension: $\{\{c, d, f\}\}$



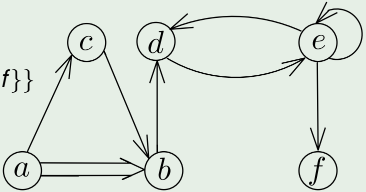
Semantics for Bipolar Abstract Argumentation

A semantics identifies "reasonable" sets of arguments, called *extensions*

- A set $S \subseteq A$ is *conflict-free* iff there are no two arguments $a, b \in S$ such that $\{a\}$ set-attacks b.
- A conflict-free set $S \subseteq A$ is said to be ***admissible*** iff it defends all of its arguments.
- A ***preferred extension*** ($\mathtt{pr}$) for a BAF is an admissible set which is maximal (w.r.t $\subseteq$).
- A conflict-free set $S \subseteq A$ is a ***stable extension*** ($\mathtt{st}$), if and only if it set-attacks all the arguments in $A \setminus S$.

Example (semantics for BAF)

admissible sets: $\{\{\emptyset\}, \{a\}, \{c\}, \{a, b\}, \{c, d\}, \{c, d, f\}\}$
preferred extensions: $\{\{a, b\}, \{c, d, f\}\}$
stable extension: $\{\{c, d, f\}\}$



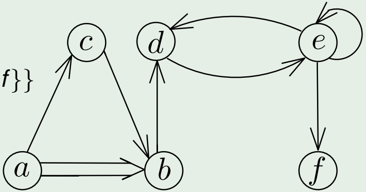
Semantics for Bipolar Abstract Argumentation

A semantics identifies “reasonable” sets of arguments, called *extensions*

- A set $S \subseteq A$ is *conflict-free* iff there are no two arguments $a, b \in S$ such that $\{a\}$ set-attacks b.
- A conflict-free set $S \subseteq A$ is said to be ***admissible*** iff it defends all of its arguments.
- A ***preferred extension*** ($\mathtt{pr}$) for a BAF is an admissible set which is maximal (w.r.t $\subseteq$).
- A conflict-free set $S \subseteq A$ is a ***stable extension*** ($\mathtt{st}$), if and only if it set-attacks all the arguments in $A \setminus S$.

Example (semantics for BAF)

admissible sets: $\{\{\emptyset\}, \{a\}, \{c\}, \{a, b\}, \{c, d\}, \{c, d, f\}\}$
preferred extensions: $\{\{a, b\}, \{c, d, f\}\}$
stable extension: $\{\{c, d, f\}\}$

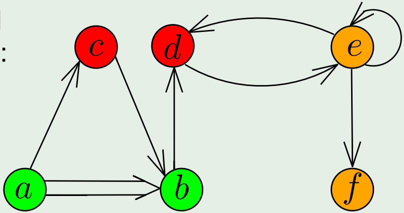


Extensions and labellings

- Semantics can be also defined in terms of *labelling*.
- Function $L : A \rightarrow \{\text{IN}, \text{OUT}, \text{UN}\}$ assigns a label to each argument
  - $L(a) = \text{IN}$ means a is accepted
  - $L(a) = \text{OUT}$ means a is rejected
  - $L(a) = \text{UN}$ means that a is undecided

Example (Preferred extension and labelling)

Preferred extension: $\{a, b\}$



Preferred labelling:
$\{a, b\}$ are IN (green nodes)
$\{c, d\}$ are OUT (red nodes)
$\{e, f\}$ are UN (orange nodes)

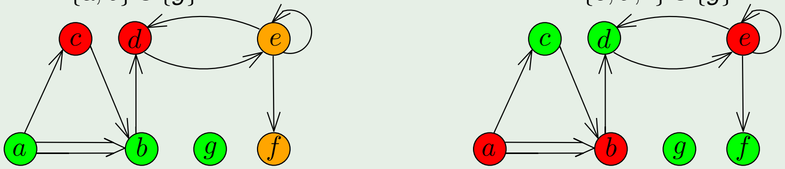
Updates

- An *update* u for a BAF $\mathcal{B}_0$ allow us to change $\mathcal{B}_0$ into a BAF $\mathcal{B}$ by adding or removing an argument, an attack, or a support.
- If E_0 is an extension for $\mathcal{B}_0$ and $\mathcal{B}$ is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition (+) and deletion (−) of an attack ($a \rightarrow b$) or a support ($a \Rightarrow b$).
- $u(\mathcal{B}_0)$ denotes the application of update $u = \pm(a \rightarrow b)$ *or* $\pm(a \Rightarrow b)$ to $\mathcal{B}_0$.

Example (Extensions/labellings after adding the isolated argument g)

preferred extension: $\{a, b\} \cup \{g\}$

stable extension: $\{c, d, f\} \cup \{g\}$

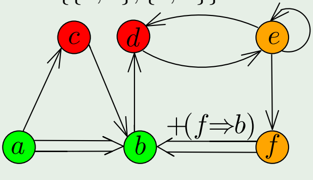


Updates

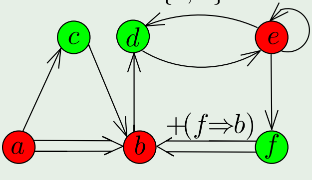
- An *update* u for a BAF $\mathcal{B}_0$ allow us to change $\mathcal{B}_0$ into a BAF $\mathcal{B}$ by adding or removing an argument, an attack, or a support.
- If E_0 is an extension for $\mathcal{B}_0$ and $\mathcal{B}$ is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition $(+)$ and deletion $(-)$ of an attack $(a \rightarrow b)$ or a support $(a \Rightarrow b)$.
- $u(\mathcal{B}_0)$ denotes the application of update $u = \pm(a \rightarrow b)$ *or* $\pm(a \Rightarrow b)$ to $\mathcal{B}_0$.

Example (Extensions/labellings after adding the support $+(f \Rightarrow b)$)

preferred extension:
$\{\{a, b\}, \{c, d\}\}$

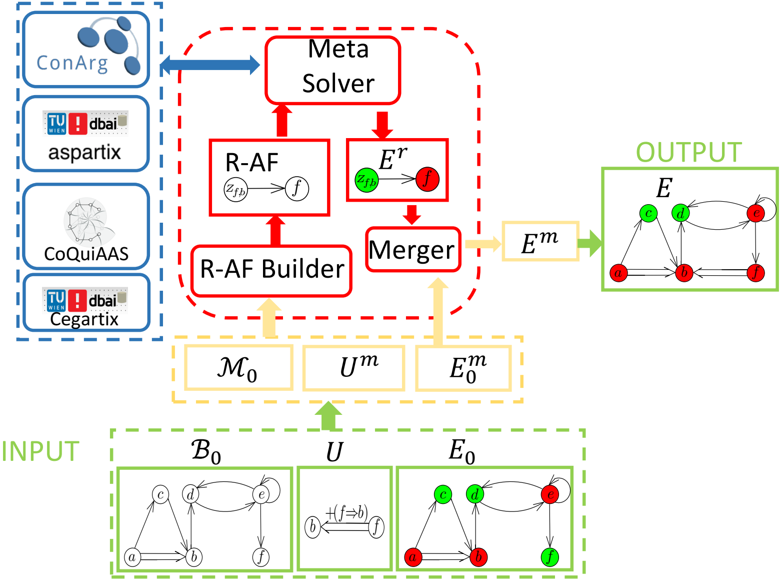


stable extension:
$\{c, d\}$



Outline

Overview of the approach



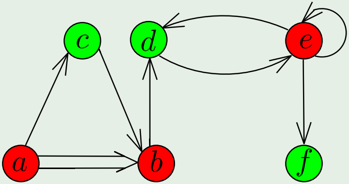
Extension preservation for addition/deletion of an attack/support

- Cases for which E_0 is still an extension of the updated BAF after a negative update.

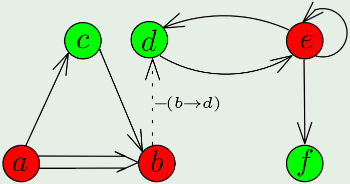
update $-(a \rightarrow b)$		$L_0(b)$		
		IN	UN	OUT
$L_0(a)$	IN	NA	NA	
	UN	NA		pr
	OUT	pr,st	pr	pr,st

update $-(a \Rightarrow b)$		$L_0(b)$		
		IN	UN	OUT
$L_0(a)$	IN	pr,st	NA	NA
	UN	pr		NA
	OUT	pr,st	pr	

Example (For $-(b \rightarrow d)$ the initial preferred extension $E_0 = \{c, d, f\}$ is preserved ($L_0(b) = \text{OUT}$ and $L_0(d) = \text{IN}$))



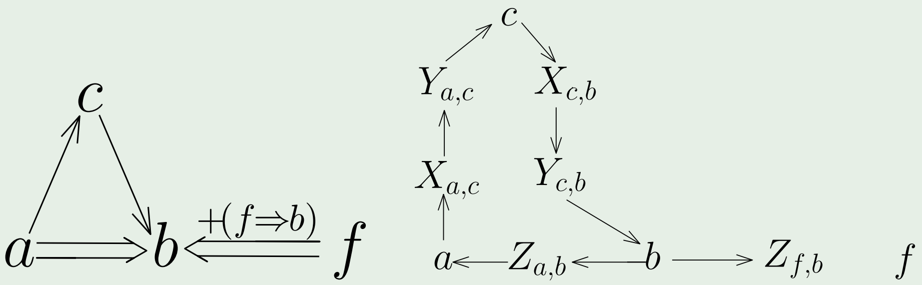
Preferred extension: $\{c, d, f\}$



The Meta-Argumentation Framework: an example

Our definition of meta-AF builds on that proposed in [BoellaGTV10] and consider additional (meta)arguments (e.g., $Z_{f,b}$) and attacks (e.g., $(b, Z_{f,b})$) that will allow us to simulate (positive) updates to be performed on BAF $\mathcal{B}_0$ by means of updates performed on the corresponding the meta-AF $\mathcal{M}_0$.

Example (Meta AF $\mathcal{M}_0$ for the BAF $\mathcal{B}_0$ w.r.t. the update $u = +(f \Rightarrow b)$.)

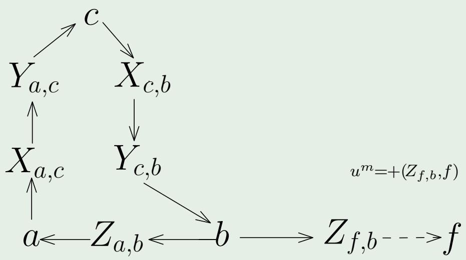


Updates for the Meta AF

Let $\mathcal{B} = \langle A, \Sigma, \Pi \rangle$ be a BAF, and u an update for $\mathcal{B}$ of the form $u = \pm(c \rightarrow d)$ or $u = \pm(e \Rightarrow f)$. The corresponding update u^m for the meta-AF $\mathcal{M}$ for $\mathcal{B}$ w.r.t. u is as follows:

$$u^m = \begin{cases} +(Z_{e,f}, e) \text{ if } u = +(e \Rightarrow f) \\ -(Z_{e,f}, e) \text{ if } u = -(e \Rightarrow f) \\ +(Y_{c,d}, d) \text{ if } u = +(c \rightarrow d) \\ -(Y_{c,d}, d) \text{ if } u = -(c \rightarrow d) \end{cases}$$

Example (for $u = +(f \Rightarrow b)$ is $u^m = +(Z_{f,b}, f)$)



Incremental Algorithm

Algorithm Incr-BAF($\mathcal{B}_0, u, E_0, \mathcal{S}$, Solver$_\mathcal{S}$)

Input: BAF $\mathcal{B}_0 = \langle A_0, \Sigma_0\ \Pi_0 \rangle$,
update u of the form $u = \pm(a \Rightarrow b)$ or $u = \pm(a \rightarrow b)$,
an initial $\mathcal{S}$-extension E_0,
semantics $\mathcal{S} \in \{\mathtt{pr}, \mathtt{st}\}$,
function Solver$_\mathcal{S}(\mathcal{A})$ returning an $\mathcal{S}$-extension for AF $\mathcal{A}$ if it exists, $\perp$ otherwise

Output: An $\mathcal{S}$-extension E for $u(\mathcal{B}_0)$ if it exists, $\perp$ otherwise;

if *checkProp*($\mathcal{B}_0, u, E_0, \mathcal{S}$) **then**
  return E_0 // Extension preservation
Let $\mathcal{M}_0 = \langle A^m, \Sigma^m \rangle$ be the the meta-AF for $\mathcal{B}_0$ w.r.t. u // Compute the meta AF
Let u^m be the update for $\mathcal{M}_0$ corresponding to u // Translate u in u^m
Let E_0^m be the initial $\mathcal{S}$-extension for $\mathcal{M}_0$ corresponding to E_0 // Convert the initial extension of the BAF in an extension for the meta AF
Let E^m = Incr-Alg($\mathcal{M}_0, u^m, \mathcal{S}, E_0^m$, Solver$_\mathcal{S}$) [Alfano,Greco,Parisi IJCAI 2017]// Compute an $\mathcal{S}$-extension for the meta AF by calling Incr-Alg
if $(E^m \neq \perp)$ **then**
  return $E = (E^m \cap A_0)$;
else
  return $\perp$; // An extension for the the meta AF could not exists (e.g., stable semantics)

Outline

Methodology

Datasets:
Generated set of BAFs from AF used as ICCMA'15 benchmarks.
Given a percentage $p \in \{10\%, 20\%\}$ of support, for each AF $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle$ in the ICCMA dataset, we generate two BAFs $\mathcal{B}_0 = \langle A_d, \Sigma^p, \Pi^p \rangle$ as follows. We selected $p \times |\Sigma_d|$ attacks in Σ_d in a random way, and converted them into supports by randomly choosing in $\{(a,b),(b,a)\}$ to Π^p.

Methodology
The average run time of our algorithm to compute an $\mathcal{S}$-extension was compared with the average run time of the best ICCMA solver to compute an $\mathcal{S}$-extension for $u^m(\mathcal{M}_0)$ from scratch.

- As Solver$_\mathcal{S}$ for computing an $\mathcal{S}$-extension for the reduced AF we used the solver that won the ICCMA'15 competition for the task $\mathcal{S}$-SE
- *Cegartix* [Dvorák et al. 2014] for $\mathcal{S} = \mathtt{pr}$
- *ASPARTIX-D* [Gaggl and Manthey 2015] for $\mathcal{S} = \mathtt{st}$.

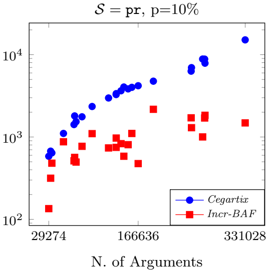
Methodology

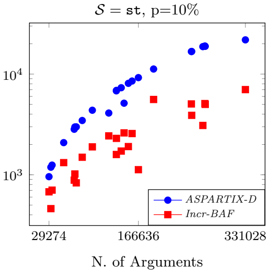
Datasets:
Generated set of BAFs from AF used as ICCMA'15 benchmarks.
Given a percentage $p \in \{10\%, 20\%\}$ of support, for each AF $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle$ in the ICCMA dataset, we generate two BAFs $\mathcal{B}_0 = \langle A_d, \Sigma^p, \Pi^p \rangle$ as follows. We selected $p \times |\Sigma_d|$ attacks in Σ_d in a random way, and converted them into supports by randomly choosing in $\{(a, b), (b, a)\}$ to Π^p.

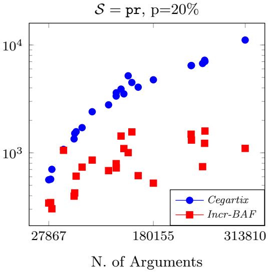
Methodology
The average run time of our algorithm to compute an $\mathcal{S}$-extension was compared with the average run time of the best ICCMA solver to compute an $\mathcal{S}$-extension for $u^m(\mathcal{M}_0)$ from scratch.

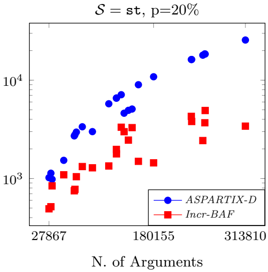
- As Solver$_\mathcal{S}$ for computing an $\mathcal{S}$-extension for the reduced AF we used the solver that won the ICCMA'15 competition for the task $\mathcal{S}$-SE
- *Cegartix* [Dvorák et al. 2014] for $\mathcal{S} = \mathtt{pr}$
- *ASPARTIX-D* [Gaggl and Manthey 2015] for $\mathcal{S} = \mathtt{st}$.

Experimental Results









Results

- Our algorithm outperforms the competitors that compute the extensions from scratch. In particular, the time saved by the incremental computation increases exponentially with respect to the size of the input BAF.
- The improvements obtained for the two semantics (preferred and stable) are similar. That is, our incremental approach is quite insensitive w.r.t. the semantics adopted.
- The improvements obtained increase when increasing the percentage of support from 10% to 20%. In fact, for a given fixed number $n = |\Sigma_0| + |\Pi_0|$ of the edges in the interaction graph for BAF $\mathcal{B}_0$, it is the case that increasing the percentage of edges in Π_0 (and thus decreasing $|\Sigma_0|$) yields to smaller meta AFs.

Outline

Conclusions and future work

- We introduced a technique for the incremental computation of extensions of dynamic BAFs
- We identified a tighter portion of the updated BAF to be examined for recomputing the semantics
- Our experiments showed that the incremental technique outperforms the computation from scratch
- Future work 1: Although in this paper we focused on updates consisting of adding/removing one attack/support, our technique can be extended to deal with sets of updates performed simultaneously.
- Future work 2: our technique can be extended to consider *second-order attacks* [BoellaGTV10] for BAFs, that is, attacks from an argument or an attack to another attack and attacks from an argument to a support.
- Future work 3: we also plan to extend our technique to deal with other interpretations of support, particularly the approach in [CayrolL09, CayrolL10] where meta AFs are also adopted to cope with bipolarity in argumentation.

Conclusions and future work

- We introduced a technique for the incremental computation of extensions of dynamic BAFs
- We identified a tighter portion of the updated BAF to be examined for recomputing the semantics
- Our experiments showed that the incremental technique outperforms the computation from scratch
- Future work 1: Although in this paper we focused on updates consisting of adding/removing one attack/support, our technique can be extended to deal with sets of updates performed simultaneously.
- Future work 2: our technique can be extended to consider *second-order attacks* [BoellaGTV10] for BAFs, that is, attacks from an argument or an attack to another attack and attacks from an argument to a support.
- Future work 3: we also plan to extend our technique to deal with other interpretations of support, particularly the approach in [CayrolL09, CayrolL10] where meta AFs are also adopted to cope with bipolarity in argumentation.

Thank you!

... any ~~question~~ argument?

Selected References

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.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex.
Bipolar abstract argumentation systems.
Argumentation in Artificial Intelligence, 65–84 (2009)

Baroni, P., Giacomin, M., Liao, B.
On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method.
Artificial Intelligence 212, 104–115 (2014)

Wolfgang Dvorák, Matti Järvisalo, Johannes Peter Wallner, and Stefan Woltran.
Complexity-sensitive decision procedures for abstract argumentation.
AI, 206:53–78, 2014.

Sarah Alice Gaggl and Norbert Manthey.
ASPARTIX-D ready for the competition, 2015.

Stefano Bistarelli, Fabio Rossi, and Francesco Santini.
ConArg: A tool for classical and weighted argumentation.
In *COMMA*, 2016.

Leila Amgoud, Claudette Cayrol, and Marie-Christine Lagasquie-Schiex.
On the bipolarity in argumentation frameworks.
In *NMR*, pages 1–9, 2004.

Guido Boella and Dov M. Gabbay and Leendert W. N. van der Torre and Serena Villata.
Support in Abstract Argumentation.
In *COMMA*, 2010, 111–122.

Gianvincenzo Alfano and Sergio Greco and Francesco Parisi.
Efficient Computation of Extensions for Dynamic Abstract Argumentation Frameworks: An Incremental Approach.
In *IJCAI*, pages 49–55, 2017.