Bipolar Abstract Argumentation with Dual Attacks and Supports

Nico Potyka
University of Stuttgart, Germany
nico.potyka@ipvs.uni-stuttgart.de

Abstract

Bipolar abstract argumentation frameworks allow modeling decision problems by defining pro and contra arguments and their relationships. In some popular bipolar frameworks, there is an inherent tendency to favor either attack or support relationships. However, for some applications, it seems sensible to treat attack and support equally. Roughly speaking, turning an attack edge into a support edge, should just invert its meaning. We look at a recently introduced bipolar argumentation semantics and two novel alternatives and discuss their semantical and computational properties. Interestingly, the two novel semantics correspond to stable semantics if no support relations are present and maintain the computational complexity of stable semantics in general bipolar frameworks.

1 Introduction

Abstract argumentation (Dung 1995) studies the acceptability of arguments abstracted from their content, just based on their relationships. In the classical framework from (Dung 1995), arguments can only attack each other. However, in many applications it seems reasonable that arguments cannot only contradict, but can also support each other. Therefore, in bipolar argumentation, an additional support relation is considered (Amgoud et al. 2008; Oren and Norman 2008; Boella et al. 2010; Cayrol and Lagasquie-Schiex 2013). However, supports are often not considered as direct counterparts of attacks, but rather as meta-relations with a special meaning. In bipolar argumentation with *deductive support*, support relations are basically used to derive indirect attacks. For example, if an argument directly or indirectly supports an attacker of an argument, this is considered as a *supported attack* (Cayrol and Lagasquie-Schiex 2013). Derived attacks like this are then used to translate the bipolar argumentation framework to an attack-only framework that can be interpreted by semantics for the standard framework. While this is an elegant way to reduce bipolar argumentation to well established argumentation approaches, it causes an asymmetry between attack and support as we will discuss later. Another example is bipolar argumentation with *evidential support* (Oren and Norman 2008). Here, nothing can be accepted unless directly or indirectly supported by special prima-facie arguments. Furthermore, attacks can only be successful if they are supported. While this approach explicitly takes account of the positive meaning of support, attacks and supports are still treated quite differently.

Our focus here is on *bipolar argumentation frameworks with dual attacks and supports*. The intuitive idea is that attack and support should be dual notions: replacing an attack edge with a support edge should just invert its meaning. While an attack should decrease the credibility of an argument, a support should increase its credibility by the same amount. It is difficult to make this idea formally precise. The idea that we follow here to accomplish this duality is to define symmetrical constraints for attack and support relations. While complete symmetry can result in rather weak semantics, a slight asymmetry in the definition suffices to get meaningful semantics that still treat attacks and supports equally.

We will start our discussion with deductive labellings that have been introduced in (Potyka 2020). While deductive labellings are based on a completely symmetrical definition of attack and support, the definition is rather weak and admits labellings that can be very sceptical and indecisive. We consider two alternatives called *s-deductive* and *m-deductive* labellings that introduce a slight asymmetry. Similar to complete semantics in attack-only argumentation frameworks, they give preference to accepting arguments. As it turns out, if there are no support relations, *s-deductive* and *m-deductive* labellings actually correspond to common stable labellings. In bipolar frameworks with supports, they differ in the way how they handle conflicts between accepted attackers and supporters. S-deductive labellings label an argument that is both attacked and supported by accepted arguments undecided. M-deductive labellings try to resolve the conflict by means of a majority decision. We demonstrate how s-deductive and m-deductive labellings treat attack and support equally, illustrate their differences and discuss some of their general properties.

2 Background

A Dung-style *(finite) abstract argumentation framework (AAF)* is a tuple $(\mathcal{A}, \mathrm{Att})$, where $\mathcal{A}$ is a finite set of arguments and $\mathrm{Att} \subseteq \mathcal{A} \times \mathcal{A}$ is the *attack relation* (Dung 1995). If $(A, B) \in \mathrm{Att}$, we say that A attacks B. With a slight abuse of notation, we let $\mathrm{Att}(A) = \{B \mid (B, A) \in \mathrm{Att}\}$ denote the set of attackers of A. Semantics of argumentation frameworks can be defined in terms of extensions

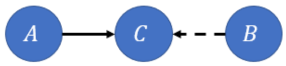

Figure 1: BAF $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ with $\mathcal{A} = \{A, B, C\}$, $\mathrm{Att} = \{(A, C)\}, \mathrm{Sup} = \{(B, C)\}$

or labellings in an equivalent way (Caminada and Gabbay 2009). We will use labellings here. A labelling is a function $\mathrm{L} : \mathcal{A} \to \{\mathrm{in}, \mathrm{out}, \mathrm{und}\}$ that assigns to each argument a label. We say that an argument A is accepted if $\mathrm{L}(A) = \mathrm{in}$ and that A is rejected if $\mathrm{L}(A) = \mathrm{out}$. Following (Caminada and Gabbay 2009), we call a labelling

Complete: if L satisfies

1. $\mathrm{L}(A) = \mathrm{in}$ if and only if $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$.
2. $\mathrm{L}(A) = \mathrm{out}$ if and only if $\mathrm{L}(B) = \mathrm{in}$ for some $B \in \mathrm{Att}(A)$.

Furthermore, we say that a complete labelling is

Grounded: if $\mathrm{in}(\mathrm{L})$ is minimal with respect to set inclusion.

Preferred: if $\mathrm{in}(\mathrm{L})$ is maximal with respect to set inclusion.

Semi-stable: if $\mathrm{und}(\mathrm{L})$ is minimal with respect to set inclusion.

Stable: if $\mathrm{und}(\mathrm{L}) = \emptyset$.

In many applications it is useful to consider not only attack relations, but also support relations. A *bipolar argumentation framework (BAF)* is a tuple $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$, where $\mathcal{A}$ is a finite set of arguments, $\mathrm{Att} \subseteq \mathcal{A} \times \mathcal{A}$ is the *attack relation* as before and Sup is the support relation (Cayrol and Lagasquie-Schiex 2013). We also assume that $\mathrm{Att} \cap \mathrm{Sup} = \emptyset$. We let again $\mathrm{Sup}(A) = \{B \mid (B, A) \in \mathrm{Sup}\}$ denote the set of supporters of A. Graphically, we denote attack relations by solid edges and support relations by dashed edges. Figure 1 shows an example BAF with one attack and one support. For example, Figure 1 could be instantiated with the following arguments:

C: Schools should be closed to slown down the spread of COVID-19.

A: School shutdowns may force parents to ask older relatives for help and may thus increase the number of infections in a high-risk group.

B: Slowing down the spread of the virus is vital to avoid a collapse of the health system.

The meaning of the support relation can be defined in different ways. The intuitve idea of *deductive support* (Boella et al. 2010; Cayrol and Lagasquie-Schiex 2013) is that if an argument is accepted, every argument that it supports must be accepted as well. In this way, arguments can also indirectly support arguments via chains of support relations. The dual idea (if an argument is accepted, all its supporters must be accepted as well) is referred to as *necessary support* (Cayrol and Lagasquie-Schiex 2013). However, deductive support relations can be tranlsated to necessary support relations by just reversing their direction, so we will focus on deductive support here. A deeper discussion of both and other notions of support can be found in (Cayrol and Lagasquie-Schiex 2013). One way to give formal meaning to deductive support relations is to translate the bipolar argumentation framework to an abstract argumentation framework with additional attacks (Cayrol and Lagasquie-Schiex 2013). These new attacks correspond to indirect attacks that are created by the interplay between attack and support relations. The following indirect attacks have been considered for this purpose in (Cayrol and Lagasquie-Schiex 2013):

Supported Attack from A to B: there is a sequence of arguments $A_1, \dots, A_n$ such that $A_1 = A$, $A_n = B$, $(A_i, A_{i+1}) \in \mathrm{Sup}$ for $1 \leq i \leq n-2$ and $(A_{n-1}, A_n) \in \mathrm{Att}$.

Mediated Attack from A to B: there is a sequence of arguments $A_1, \dots, A_n$ such that $A_1 = B$, $A_n = A$, $(A_i, A_{i+1}) \in \mathrm{Sup}$ for $1 \leq i \leq n-2$ and $(A_n, A_{n-1}) \in \mathrm{Att}$.

Intuitively, there is a supported attack from A to B iff A directly or indirectly supports an attacker of B. There is a mediated attack from A to B iff A attacks an argument that is directly or indirectly supported by B. The *Dung framework associated with* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ is then defined as the Dung framework $(\mathcal{A}, \mathrm{Att} \cup \mathrm{Att}^s \cup \mathrm{Att}^m)$, where Att^s and Att^m contain additional attacks that correspond to supported and mediated attacks in $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ (Cayrol and Lagasquie-Schiex 2013). This is an elegant way to extend established semantics to bipolar argumentation frameworks. However, this solution does not treat attack and support equally. For example, in Figure 1, it seems that C could as well be accepted as rejected. However, the translation takes only account of the mediated attack from A to B and ignores the fact that C has a supporter. Hence, the only complete labelling of the associated Dung framework labels A in and B and C out. The disparity between attack and support becomes more prevalent when we keep our attacker A, but add n supporters B_i of C. Then A will be accepted and C and all of its supporters B_i will be rejected even when C is supported by thousands of undisputed arguments.

3 Towards Bipolar Argumentation with Dual Attacks and Supports

For decision-making problems, where the final decision is based on pro (support) and contra (attack) arguments, it seems reasonable that attack and support are treated equally. That is, an attack relation should decrease the chance of acceptance in the same way as a support relation increases the chance of acceptance. For example, in Figure 1, one may prefer the contra argument A. However, it seems too restrictive to condemn B to be rejected just because it is a pro argument. In this scenario, accepting A and rejecting B as well as accepting B and rejecting A should be a valid option. I would even argue that it can be reasonable to accept both A and B. Then C should be undecided until further arguments for or against C have been proposed. For our COVID-19

Labelling	A	B	C
L_1	in	out	out
L_2	out	in	in
L_3	out	out	in
L_4	out	out	out
L_5	out	out	und
L_6	out	und	und
L_7	out	und	in
L_8	und	out	out
L_9	und	out	und
L_{10}	und	und	und

Table 1: Deductive labellings for the BAF in Figure 1.

instantiation that we considered before, this means that we have to collect further arguments that speak for or against a school closure or one of the existing arguments in order to make a decision.

In order to obtain dual behaviour of attack and support, the following definition has been proposed in (Potyka 2020): we call a labelling

Deductive: if L satisfies

1. If $\mathrm{L}(A) = \mathrm{in}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$.
2. If $\mathrm{L}(A) = \mathrm{out}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Sup}(A)$.
3. If $\mathrm{L}(B) = \mathrm{in}$ for some $B \in \mathrm{Att}(A)$, then $\mathrm{L}(A) = \mathrm{out}$.
4. If $\mathrm{L}(B) = \mathrm{in}$ for some $B \in \mathrm{Sup}(A)$, then $\mathrm{L}(A) = \mathrm{in}$.

Conditions 1 and 2 correspond to necessary conditions for acceptance and rejection. We can accept (reject) only when all attackers (supporters) are out. Conditions 3 and 4 correspond to sufficient conditions. If one attacker (supporter) is in, the argument must be out (in). Condition 1 and 3 make sure that deductive labellings respect classical attacks. Conditions 2 and 4 make sure that they interpret support in a dual manner. As shown in (Potyka 2020), deductive labellings also respect supported and mediated attacks.

Proposition 1 (from (Potyka 2020))**.** *Let* L *be a* deductive *labelling.*

1. *If* $\mathrm{L}(A) = \mathrm{in}$ *and there is a supported attack from A to B, then* $\mathrm{L}(B) = \mathrm{out}$.
2. *If* $\mathrm{L}(A) = \mathrm{in}$ *and there is a mediated attack from A to B, then* $\mathrm{L}(B) = \mathrm{out}$.

While the definition of deductive labellings defines the meaning of attack and support edges in a symmetrical fashion, it is rather weak and admits many labellings as we demonstrate in the following example.

Example 1. Consider again the BAF in Figure 1. We can compute all labellings by going through the definition step by step. For example, if A is accepted, B must be rejected because of condition 3 of deductive labellings. Then C must be rejected because of condition 1. By going through the remaining cases, we can find all deductive labels that are shown in Table 1. We can see that many labellings are rather sceptical and indecisive. This is, in particular, reflected by L_4 and L_{10}.

In order to make deductive labellings more credulous and decisive, they can be refined by considering grounded, preferred, semi-stable and stable deductive labellings as before by minimizing or maximizing particular labels. For example, a deductive labelling is called stable if it does not label any argument undecided. A discussion of stable deductive labellings can be found in (Potyka 2020). While stable deductive labellings are more decisive, one may argue that they remain too sceptical because the labelling l_{out} that labels every argument out is always deductive.

Let us look at grounded, preferred and semi-stable deductive labellings in more detail. To begin with, we note that grounded deductive labellings are overly sceptical. Since l_{out} is always deductive, grounded deductive labellings never accept any argument.

Preferred deductive labellings are more interesting. However, they do not treat attack and support equally. While the deductive support approach discussed in (Cayrol and Lagasquie-Schiex 2013) favors attacks, preferred deductive labellings favor supports. To see this, consider again the BAF in Figure 1. The only preferred deductive labelling labels B and C in and A out (labelling l_2 in Table 1).

Semi-stable deductive labellings actually do not add anything to stable deductive labellings. This is because the labelling l_{out} is deductive. Therefore, there are always deductive labellings that label zero arguments undecided. So stable deductive labellings always exist and coincide with semi-stable deductive labellings. A discussion of stable deductive labellings can be found in (Potyka 2020). They behave quite well for small examples. The stable deductive labellings for the BAF in Figure 1 are the labellings $\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_3$ and L_4 from Table 1. However, one may argue that they are still overly sceptical because they can reject arguments without reason. This is an artifact of the desire to have a symmetrical definition of attacks and supports. If one demands that everything that is not attacked is accepted, one may argue that we should symmetrically demand that every argument that is not supported must be rejected. However, these two constraints are inconsistent in many cases (every argument without parents must be accepted and rejected).

Because of the aforementioned shortcomings of deductive labellings and their refinements, we will look at some alternatives in the following sections. In order to obtain more decisive labellings, we will add a necessary condition for labeling an argument undecided. In order to obtain more credulous labellings, we will revise the sufficient conditions and favor accepting arguments over rejecting them. This slight asymmetry seems necessary in order to prevent an overly sceptical behaviour.

4 S-Deductive Labellings

In AAFs, it is natural to demand that an argument is accepted when all attackers are rejected. However, in BAFs, we have to be more careful. This can be seen from the BAF in Figure 1 already. If we want to treat attack and support equally and accept every argument that is unattacked, we would have to accept both A and B and declare C undecided. While this is a reasonable labelling, I would argue that it is also reasonable to accept only one of A and B and to reject the other to

Labelling	A	B	C
L_1	in	und	in
L_2	in	out	out
L_3	out	in	in

Table 2: S-deductive labellings for the BAF in Figure 1.

allow for different viewpoints. We will therefore consider a stronger precondition that not only considers direct attackers of an argument, but also more indirect conflicts between arguments. We say that argument A *contradicts* argument B if there is an argument C such that

1. $(A, C) \in \mathrm{Att}$ and $(B, C) \in \mathrm{Sup}$ or
2. $(B, C) \in \mathrm{Att}$ and $(A, C) \in \mathrm{Sup}$.

The *opponents* of an argument are defined as the set of arguments that contradict it, that is, $\mathrm{Opp}(A) = \{B \in \mathcal{A} \mid B \textit{ contradicts } A\}$. In the following definition, we use a variant of complete labellings' sufficient condition for acceptance that demands not only that all attackers are out, but also that all opponents are out. We call a labelling

S-Deductive: if L satisfies

1. If $\mathrm{L}(A) = \mathrm{in}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$.
2. If $\mathrm{L}(A) = \mathrm{out}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Sup}(A)$.
3. If $\mathrm{L}(A) = \mathrm{und}$, then $\mathrm{L}(B) = \mathrm{in}$ for some $B \in \mathrm{Att}(A)$ and $\mathrm{L}(B) = \mathrm{in}$ for some $B \in \mathrm{Sup}(A)$.
4. $\mathrm{L}(A) = \mathrm{in}$ whenever
   (a) $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$ and
   (b) $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Opp}(A)$.

The first two conditions are the necessary conditions from the deductive labelling definition. The third condition is a new necessary condition for undecidedness to make the semantics more decisive. We demand that an argument can only be undecided if there is a reason for both accepting and rejecting the argument. The fourth condition is the sufficient condition for acceptance that we discussed at the beginning of this section.

Example 2. Consider again the BAF in Figure 1. The only s-deductive labelling that can be undecided is C. This is only possible if both A and B are accepted. This is indeed possible since they are both unattacked.

If A is accepted, C cannot be accepted because of condition 1. C must be undecided if B is accepted and rejected if B is not accepted.

Another s-deductive labelling dually accepts B and C and rejects A. Table 2 shows the three s-deductive labellings for the BAF in Figure 1. L_2 is the grounded s-deductive labelling. L_1 and L_3 are the preferred s-deductive labellings. Both L_2 and L_3 are semi-stable and stable s-deductive labellings.

As opposed to deductive labellings, the definition of s-deductive labellings does not contain explicit statements about what must happen to an argument if a single attacker or a single supporter is accepted. This allows to accept both A and B for the BAF in Figure 1 by labelling C undecided. The following proposition sheds some light on the influence of single attack and support relations.

Proposition 2. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF and let* L *be a corresponding s-deductive labelling.*

1. *If* $(A, B) \in \mathrm{Att}$ *and* $\mathrm{L}(A) = \mathrm{in}$, *then* $\mathrm{L}(B) \in \{\mathrm{out}, \mathrm{und}\}$.
2. *If* $(A, B) \in \mathrm{Att}$, $\mathrm{L}(A) = \mathrm{in}$ *and* $\mathrm{L}(C) = \mathrm{out}$ *for all* $C \in \mathrm{Sup}(B)$, *then* $\mathrm{L}(B) = \mathrm{out}$.
3. *If* $(A, B) \in \mathrm{Sup}$ *and* $\mathrm{L}(A) = \mathrm{in}$, *then* $\mathrm{L}(B) \in \{\mathrm{in}, \mathrm{und}\}$.
4. *If* $(A, B) \in \mathrm{Sup}$, $\mathrm{L}(A) = \mathrm{in}$ *and* $\mathrm{L}(C) = \mathrm{out}$ *for all* $C \in \mathrm{Att}(B)$, *then* $\mathrm{L}(B) = \mathrm{in}$.

Proof. 1. Since $\mathrm{L}(A) = \mathrm{in} \neq \mathrm{out}$, the contrapositive of condition 1 of s-deductive labellings implies that $\mathrm{L}(B) \neq \mathrm{in}$ and therefore $\mathrm{L}(B) \in \{\mathrm{out}, \mathrm{und}\}$.

2. From item 1, we know that $\mathrm{L}(B) \in \{\mathrm{out}, \mathrm{und}\}$. Since $\mathrm{L}(C) = \mathrm{out} \neq \mathrm{in}$ for all $C \in \mathrm{Sup}(B)$, the contrapositive of condition 3 of s-deductive labellings implies that $\mathrm{L}(B) \neq \mathrm{und}$ and thus $\mathrm{L}(B) = \mathrm{out}$.

3 and 4 follow analogously using condition 2 instead of condition 1 of s-deductive labellings. □

Item 1 says that if an argument is attacked by an accepted argument, then it must be either rejected or undecided. This is a weaker statement than in AAFs, but I would argue that it makes sense in BAFs because an attack can be cancelled by a support now. In this case, labelling the argument undecided seems reasonable. If there is no accepted argument that cancels the attack, the argument must indeed be rejected as explained in item 2. Items 3 and 4 state dual properties for supports.

It is interesting to note that, in AAFs, S-deductive labellings are always complete labellings.

Proposition 3. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF such that* $\mathrm{Sup} = \emptyset$. *Then every s-deductive labelling is a complete labelling in the corresponding AAF* $(\mathcal{A}, \mathrm{Att})$.

Proof. Let L be an s-deductive labelling. Note that condition 4b) of s-deductive labellings is trivially satisfied if $\mathrm{Sup} = \emptyset$. Therefore, condition 4 and 1 together state that an argument is accepted if and only if all attackers are rejected. This corresponds to condition 1 of complete labellings.

Since $\mathrm{Sup} = \emptyset$, the contrapositive of condition 4 of s-deductive labellings implies that if $\mathrm{L}(A) = \mathrm{out} \neq \mathrm{in}$, then $\mathrm{L}(B) \neq \mathrm{out}$ for some $B \in \mathrm{Att}(A)$. However, there can be no undecided labellings if $\mathrm{Sup} = \emptyset$ because of condition 3 of s-deductive labellings. Hence, if A is out, there must be an attacker that is in. Furthermore, item 2 of Proposition 2 and the fact that $\mathrm{Sup} = \emptyset$ implies that A is out whenever an attacker is in. Hence, the second condition of complete labellings is satisfied as well and L is a complete labelling. □

As explained in the previous proof, s-deductive labellings do not label anything undecided if there are no supporters.

Corollary 1. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF such that* $\mathrm{Sup} = \emptyset$ *and let* L *be an s-deductive labelling. Then* $\mathrm{L}(A) \neq \mathrm{und}$ *for all* $A \in \mathcal{A}$.

Since complete labellings can label arguments undecided, not every complete labelling corresponds to an s-deductive labelling. In particular, in AAFs where no stable labelling exists, no s-deductive labelling exists either. The standard example is an attack cycle of length 3 given by the AAF $(\{A, B, C\}, \{(A, B), (B, C), (C, A)\})$. As it turns out, stable labellings are actually always s-deductive labellings.

Proposition 4. *Let* $(\mathcal{A}, \mathrm{Att})$ *be an AAF and let* L *be a stable labelling. Then* L *is an s-deductive labelling in the corresponding BAF* $(\mathcal{A}, \mathrm{Att}, \emptyset)$.

Proof. Note that L is, in particular, a complete labelling and does not label any arguments undecided. Condition 1 and 4 of s-deductive labellings follow immediately from condition 1 of complete labellings and the fact that no opponents exist in AAFs. Condition 2 of s-labellings follows immediately from condition 2 of complete labellings. Condition 3 of s-labellings is satisfied trivially because no argument is labelled undecided. □

Taken together, our results show that, for AAFs, s-deductive labellings coincide with stable labellings.

Corollary 2. *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF such that* $\mathrm{Sup} = \emptyset$ *and let* $\mathcal{G}' = (\mathcal{A}, \mathrm{Att})$. *Then the s-deductive labellings of* $\mathcal{G}$ *are exactly the stable labellings of* $\mathcal{G}'$.

Proof. If L is an s-deductive labelling for $\mathcal{G}$, Proposition 3 and Corollary 1 imply that L is a complete labelling for $\mathcal{G}'$ that does not label any arguments undecided. Therefore, L is a stable labelling for $\mathcal{G}'$.

Conversely, if L is a stable labelling for $\mathcal{G}'$, Proposition 4 implies that L is an s-deductive labelling for $\mathcal{G}$. □

As demonstrated in Example 2 for the BAF in Figure 1, for general bipolar graphs, arguments can be labelled undecided. However, this can only happen if the argument is both supported and attacked by an accepted argument.

We also note that, as opposed to deductive labellings, s-deductive labellings do not respect supported and mediated attacks in the strong sense of Proposition 1. This is because support chains can be broken. If a deductive labelling accepts an argument A, it has to accept every argument along a support chain that starts at A. This is no longer true for s-deductive labellings as we already saw in Example 2 for the BAF in Figure 1. In this example, even though there is a mediated attack from A to B, we can accept both A and B by labelling C undecided. As an example for a supported attack that is not respected under s-deductive semantics, consider the BAF at the top of Figure 2. There is an s-deductive labelling that accepts A, A' and B and labels C undecided. Hence, even though A is accepted and there is a supported attack from A to C, C is not rejected. However, supported attacks are respected in a weaker sense that is very similar to the effect of classical attacks described in Proposition 2.

Proposition 5. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF and let* L *be a corresponding s-deductive labelling. Assume further that there is a supported Attack from* A_1 *to* A_n *given by the sequence of arguments* $A_1, \ldots, A_n$ *such that* $(A_i, A_{i+1}) \in \mathrm{Sup}$ *for* $1 \leq i \leq n-2$ *and* $(A_{n-1}, A_n) \in \mathrm{Att}$.

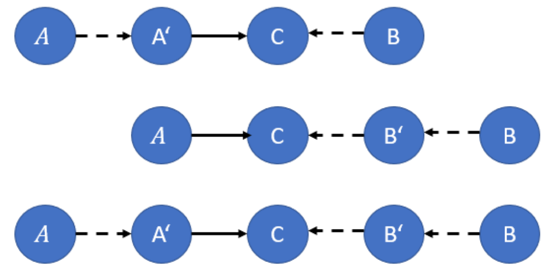

Figure 2: Supported attack vs. support (top), attack vs. supported support (middle) and supported attack vs. supported support (bottom).

1. *If* $\mathrm{L}(A_1) = \mathrm{in}$, *then* $\mathrm{L}(A_n) \in \{\mathrm{out}, \mathrm{und}\}$.
2. *If* $\mathrm{L}(A_1) = \mathrm{in}$ *and* $\mathrm{L}(C) = \mathrm{out}$ *for all* $C \in \mathrm{Sup}(A_n) \cup \bigcup_{i=2}^{n-1} \mathrm{Att}(A_i)$, *then* $\mathrm{L}(A_n) = \mathrm{out}$.

Proof. 1. Since $\mathrm{L}(A_1) = \mathrm{in} \neq \mathrm{out}$, the contrapositive of condition 2 of s-deductive labellings implies that $\mathrm{L}(A_2) \neq \mathrm{out}$. It follows inductively from the same argument that $\mathrm{L}(A_2) \neq \mathrm{out}$ for $1 \leq i \leq n-1$. Since $\mathrm{L}(A_{n-1}) \neq \mathrm{out}$, the contrapositive of condition 1 of s-deductive labellings implies that $\mathrm{L}(A_n) \neq \mathrm{in}$, thus $\mathrm{L}(A_n) \in \{\mathrm{out}, \mathrm{und}\}$.

2. Since no attacker of any A_i, $i = 2, \ldots, n-1$, is in, the necessary condition for undecided (condition 3 of s-deductive labellings) cannot be satisfied. Hence, item 1 implies that $\mathrm{L}(A_i) = \mathrm{in}$ for $1 \leq i \leq n-1$. Since $\mathrm{L}(A_{n-1}) = \mathrm{in}$ and all supporters of A_n are out by assumption, item 2 of Proposition 2 implies that $\mathrm{L}(A_n) = \mathrm{out}$. □

Item 1 says that if there is a supported attacked from A_1 to A_n and A_1 is accepted, then A_n must be either rejected or undecided. Item 2 explains that A_n must be rejected when all arguments that could break the support chain from A_1 to A_{n-1} or cancel the attack from A_{n-1} to A_n are out.

As we saw in Example 2, simple attacks and supports are treated equally under s-deductive semantics. It is interesting to note that supported attacks are stronger than simple supports and supported supports are stronger than simple attacks. Furthermore, supported attacks and supported supports are equally strong. We demonstrate this in the following example.

Example 3. Figure 2 shows some basic cases of supported attacks and supported supports. For our COVID-19 instantiation, the additional supporters may correspond to model calculations or previous studies that back the supporting or attacking argument. Consider again the BAF at the top of Figure 2. There is a supported attack from A to C and a support from B to C. Let us compute all s-deductive labellings. A does not have any attackers nor opponents and thus must be accepted according to condition 4 of deductive labellings. Therefore, A' cannot be rejected according to condition 2. It cannot be undecided either because of condition 3. Hence, A' must be accepted as well. B cannot be undecided because of condition 3. If B is out, C must be rejected because it is attacked but not supported by an accepted argument. If B is in, C must be undecided. Hence, there are two labellings.

One rejects C, the other one labels C undecided. The supported attack is stronger than the simple support in the sense that C cannot be accepted.

In the BAF in the middle of Figure 2, there is a supported support from B to C and an attack from A to C. We can check as before that there are two m-deductive labellings. Now, one accepts C, the other one labels C undecided. Dual to the previous case, C cannot be rejected. The supported support is stronger than the simple attack.

Finally, in the BAF at the bottom of Figure 2, we have a supported attack from A to C and a supported support from B to C. Similar to before, we can convince ourselves that A, A', C, C' must be in and thus B must be undecided. This is actually the only s-deductive labelling. From this example, we can see that supported attacks and supported supports are equally strong. However, as opposed to the interaction between simple supports and simple attacks, we do not have labellings that accept either the attacker or the supporter position. This can be justified by the fact that both positions are supported by undisputed arguments.

Since stable s-deductive labellings do not admit any undecided arguments, Propositions 2 and 5 immediately imply that they interpret simple and supported attacks in the strong sense.

Corollary 3. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF and let* L *be a corresponding stable s-deductive labelling.*

1. *If* $(A, B) \in \mathrm{Att}$ *and* $\mathrm{L}(A) = \mathrm{in}$*, then* $\mathrm{L}(B) = \mathrm{out}$.
2. *If there is a supported attack from* A *to* B *and* $\mathrm{L}(A) = \mathrm{in}$*, then* $\mathrm{L}(B) = \mathrm{out}$.

From our discussion in Example 3, we can see that for the BAFs at the top and in the middle of Figure 2, there is exactly one stable s-deductive labelling. The one for the top accepts A and rejects B and C, the one for the middle accepts B and C and rejects A. For the BAF at the bottom, no stable s-deductive labelling exists.

5 M-Deductive Labellings

S-deductive labellings allow accepting both an argument and one of its opponents if the argument that is supported by one and attacked by the other is undecided. In applications like e-government, it seems reasonable to make the labellings more decisive by resolving conflicts between accepted arguments by a majority decision. This motivates the following definition. We say that a labelling is

M-Deductive: if L satisfies

1. If $\mathrm{L}(A) = \mathrm{in}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$ or $|\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) \neq \mathrm{out}\}| < |\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}|$.
2. If $\mathrm{L}(A) = \mathrm{out}$, then $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Sup}(A)$ or $|\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) \neq \mathrm{out}\}| < |\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) = \mathrm{in}\}|$.
3. If $\mathrm{L}(A) = \mathrm{und}$, then $|\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| = |\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| > 0$.
4. $\mathrm{L}(A) = \mathrm{in}$ whenever
   (a) $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$ and
   (b) $\mathrm{L}(B) = \mathrm{out}$ for all $B \in \mathrm{Opp}(A)$.

Figure 3: Two BAFs that allow resolving conflicts by majority decisions.

Labelling	A	A'	B	C
L_1	in	in	in	und
L_2	in	in	out	out
L_3	in	out	in	und
L_4	out	in	in	und
L_5	out	out	in	in

Table 3: S-deductive labellings for the BAF on the left in Figure 3.

As opposed to s-deductive labellings, we weaken the necessary conditions for labelling an argument in or out, but strengthen the necessary condition for labelling an argument undecided. An argument can be accepted if all attackers are out or if the number of attackers that are not out is smaller than the number of supporters that are in. Dually, an argument can be rejected if all supporters are out or if the number of supporters that are not out is smaller than the number of attackers that are in. Labelling an argument undecided is only possible if the number of accepted attackers equals the number of accepted supporters and is non-zero.

For the BAF in Figure 1, the s-deductive and m-deductive labellings are equal (see Table 2 for the labellings). However, if a conflict can be resolved by a majority decision, the labellings differ.

Example 4. Consider the BAF on the left in Figure 3. Table 3 shows the s-deductive labellings for this example. $\mathrm{L}_1, \mathrm{L}_3$ and L_4 label C undecided because it is both supported and attacked. However, since L_1 accepts more attackers than supporters, one may argue that C should be rejected in this case. This is indeed what happens for m-deductive labellings. Table 4 shows all m-deductive labellings. For the BAF on the right in Figure 3, the behaviour is dual. For the m-deductive labelling, where B, B' and A are accepted, C must also be accepted now.

As opposed to s-deductive labellings, it is no longer true that an argument cannot be accepted if an attacker is accepted. This is because an accepted attacker can now be overruled by accepted supporters. However, we still have

Labelling	A	A'	B	C
L_1	in	in	in	out
L_2	in	in	out	out
L_3	in	out	in	und
L_4	out	in	in	und
L_5	out	out	in	in

Table 4: M-deductive labellings for the BAF on the left in Figure 3.

the following guarantees for the meaning of attack and support edges.

Proposition 6. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF and let* L *be a corresponding m-deductive labelling.*

1. *If* $|\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| > |\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}|$*, then* $\mathrm{L}(A) = \mathrm{out}$.
2. *If* $|\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| > |\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) = \mathrm{in}\}|$*, then* $\mathrm{L}(A) = \mathrm{in}$.
3. *If* $|\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| = |\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| > 0$*, then* $\mathrm{L}(A) = \mathrm{und}$.

Proof. 1. Since the number of accepted attackers is larger than the number of accepted supporters, A cannot be undecided because of condition 3 of m-deductive labellings. It cannot be accepted either because of condition 1 of m-deductive labellings. Hence, it must be rejected.

2 follows analogously using condition 2 instead of condition 1 of m-deductive labellings.

3. A can be neither accepted nor rejected because of conditions 1 and 2 of m-deductive labellings and thus must be undecided. □

For the special case of AAFs, the m-deductive and s-deductive labellings are actually always equal.

Proposition 7. *Let* $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF such that* $\mathrm{Sup} = \emptyset$*. Then every s-deductive labelling is an m-deductive labelling and vice versa.*

Proof. Note that condition 2 of s-deductive and m-deductive labellings is trivially satisfied in AAFs because there are no support edges. Note also that condition 4 is equal for s-deductive and m-deductive labellings, so it suffices to check conditions 1 and 3.

Consider an s-deductive labelling L. Condition 1 of s-deductive labellings is stronger and thus implies that L satisfies conditions 1 of m-deductive labellings. Condition 3 of m-deductive labellings is trivially satisfied because s-deductive labellings do not label any arguments undecided (Corollary 1). Hence, L is an m-deductive labelling.

Conversely, consider an m-deductive labelling L. If $\mathrm{L}(A) = \mathrm{in}$, condition 1 of m-deductive labellings implies that we must have $L(B) = \mathrm{out}$ for all $B \in \mathrm{Att}(A)$ because $|\{B \in \mathrm{Att}(A) \mid \mathrm{L}(B) \neq \mathrm{out}\}| < |\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| = 0$ can never be satisfied. Hence, L satisfies conditon 1 of s-deductive labellings. Since $|\{B \in \mathrm{Sup}(A) \mid \mathrm{L}(B) = \mathrm{in}\}| = 0$ in AAFs, m-deductive labellings cannot label anything undecided in AAFs and so condition 3 of s-deductive labellings is satisfied as well. Hence, L is an s-deductive labelling. □

Hence, in AAFs, we have the same guarantees for s-deductive and m-deductive labellings. In particular, m-deductive labellings correspond to stable labellings again.

Corollary 4. *Let* $\mathcal{G} = (\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ *be a BAF such that* $\mathrm{Sup} = \emptyset$ *and let* $\mathcal{G}' = (\mathcal{A}, \mathrm{Att})$*. Then the m-deductive labellings of* $\mathcal{G}$ *are exactly the stable labellings of* $\mathcal{G}'$.

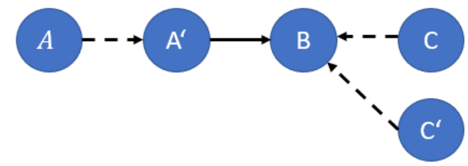

Figure 4: BAF with conflict between supported attack from A to B and supporters of B.

Like s-deductive labellings, m-deductive labellings do not respect supported and mediated attacks in the strong sense of Proposition 1. This can be seen from the same counterexamples that we discussed in the previous section. We could state a result similar to Proposition 5. However, the preconditions become rather long, so we refrain from doing so. Roughly speaking, if there is a supported attack from A to B and neither the supports nor the final attack along the supported attack path between A and B are overruled by other accepted arguments, then B must be out. There is no analogue to Corollary 3 for stable m-deductive labellings, though. This is because for m-deductive labellings, the arguments along a supported attack path cannot only become undecided, but may also become in if an attacker is overruled or out if a supporter is overruled.

Example 5. Consider the BAF in Figure 4. Let us first look at s-deductive labellings. A has neither attackers nor opponents and thus must be accepted. Therefore, A' cannot be rejected nor undecided and thus must be accepted as well. Since C and C' have opponents but no attackers, they can be either accepted or rejected. If they are both rejected, B must be rejected. Otherwise, B must be undecided. It is interesting to note that the situation does not change if we add an arbitrary number of simple supporters of B. Thus, under s-deductive semantics, a supported attack is stronger than an arbitrary number of simple supporters. In particular, there is a unique stable s-deductive labelling that rejects B, C and C'.

For m-deductive labellings, again A and A' must be accepted and C and C' can be either accepted or rejected. However, for the labelling that accepts both C and C', B will be accepted now. Hence, under m-deductive semantics, multiple simple supports can overrule a supported attack. In particular, there are two stable m-deductive labellings. Both accept A and A'. One rejects C, C' and B, the other one accepts C, C' and B. Hence, under stable m-deductive semantics, a supported attack cannot be overruled by a simple support, but by two simple supports.

We have again dual relationships between supported supports and simple attacks.

6 Computing S-deductive and M-deductive Labellings

As explained in Corollaries 2 and 4, for the special case that there are no support edges, s-deductive and m-deductive labellings correspond to stable labellings in AAFs. Therefore, inference under stable semantics in AAFs can be seen as a special case of inference under s-deductive and m-deductive labellings and all lower complexity bounds for stable semantics can be transferred to s-deductive and m-deductive labellings. We consider the following inference problems discussed in (Dunne and Wooldridge 2009) for AAFs:

EX: Given a BAF $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$, decide whether an $\mathcal{S}$-labelling exists.

CA (Credulous Acceptance): Given a BAF $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ and an argument $A \in \mathcal{A}$, decide whether $\mathrm{L}(A) = \mathrm{in}$ for at least one $\mathcal{S}$-labelling.

SA (Sceptical Acceptance): Given a BAF $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$ and an argument $A \in \mathcal{A}$, decide whether $\mathrm{L}(A) = \mathrm{in}$ for all $\mathcal{S}$-labelling.

We have the following results.

Proposition 8. *For both s-deductive and m-deductive semantics*

1. *EX is NP-complete.*
2. *CA is NP-complete.*
3. *SA is coNP-complete.*

Proof. 1. Since EX is NP-hard for stable semantics (Dunne and Wooldridge 2009), it suffices to give a polynomial-time reduction from EX for stable semantics to EX for s-deductive/m-deductive labellings to prove hardness. Given an AAF $(\mathcal{A}, \mathrm{Att})$, create the corresponding BAF $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$. According to Corollaries 2 and 4, there is a stable labelling for $(\mathcal{A}, \mathrm{Att})$ iff there is a s-deductive/m-deductive labelling for $(\mathcal{A}, \mathrm{Att}, \mathrm{Sup})$.

For membership, just notice that every s-deductive/m-deductive labelling is a yes-certificate. It is easy to check that it can be verified in polynomial time by just checking that it satisfies all necessary conditions of s-deductive/m-deductive labellings.

2. Hardness follows with the same reduction as in 1 from the fact that CA is NP-hard for stable semantics (Dunne and Wooldridge 2009). Since stable and s-deductive/m-deductive labelling are equal for AAFs according to Corollaries 2 and 4, an argument is accepted by a stable labelling if and only if it is accepted by an s-deductive/m-deductive labelling.

Membership follows as before by noting that checking if a labelling labels an argument in adds only linear additional cost.

3. Hardness follows analogously to 2 from the fact that SA is coNP-hard for stable semantics (Dunne and Wooldridge 2009).

Membership follows from observing that every labelling that is not s-deductive/m-deductive is a no-certificate. It is easy to check that it can be verified in polynomial time by checking that it violates a necessary condition of s-deductive/m-deductive labellings. □

A general implementation for these problems is work in progress. In principle, we can apply ideas similar to AAFs and solve the inference problems using solvers for SAT problems (Alviano 2018; Dvorák et al. 2012; Beierle, Brons, and Potyka 2015), ASPs (Egly, Gaggl, and Woltran 2008), CSPs (Lagniez, Lonca, and Mailly 2015) or Markov networks (Potyka 2020). While this often results in fast solutions due to the existence of sophisticated solvers for these problems, we note that s-deductive and m-deductive labellings feature a more complicated structure than argumentation problems under stable semantics in AAFs. This is because constraints are not only based on the predecessors of arguments (attackers and supporters), but can also depend on the predecessors of their successors (potential opponents).

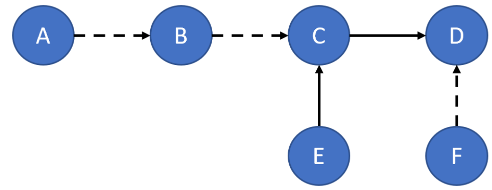

Figure 5: Example BAF from (Amgoud et al. 2008).

7 Related Work

Several bipolar argumentation approaches have been discussed in the past, we sketch only a few. We already gave a brief overview of *deductive support* (Cayrol and Lagasquie-Schiex 2013) and *evidential support* (Oren and Norman 2008) in the introduction. A more elaborate discussion of both approaches can be found in (Cayrol and Lagasquie-Schiex 2013). Some other bipolar semantics have been studied in (Amgoud et al. 2008). In this work, the authors do not only consider supported attacks, but also supported supports. In particular, they considered a refinement of admissible sets that demands that the set is conflict-free, defends all its elements and is closed under the support relation. Figure 5 shows an example from (Amgoud et al. 2008). The only admissible set is $\{D, E, F\}$ (Amgoud et al. 2008). To see this, note that C cannot be defended against the attack by C. Therefore, it cannot be accepted. Since accepting A or B would entail accepting C, they cannot be accepted either. This example shows that attack is again stronger than support under their semantics because C is necessarily rejected even though it is both attacked and supported. In contrast, our deductive, s-deductive and m-deductive labellings allow accepting C. Note that there is actually a supported support from B to C that is stronger than the simple attack from E to C under s-deductive and m-deductive semantics, so that they will actually never reject C.

Bipolar argumentation also plays a prominent role in gradual argumentation frameworks as discussed in (Amgoud et al. 2008; Baroni et al. 2015; Rago et al. 2016; Amgoud and Ben-Naim 2016; Potyka 2018). Roughly speaking, in these frameworks, a strength value for arguments is computed iteratively based on an initial base score and the strength of attackers and supporters. If this procedure converges, we can assign to every argument a well-defined strength value. Since arguments are evaluated numerically and we do not have to think about several extensions, the idea of duality is easier to formalize for these frameworks. Roughly speaking, we can demand that an attack's negative effect on the base score equals a support's positive effect, see (Potyka 2019) Definition 5.1 and Proposition 5.2 for a more precise statement. A discussion of the relevance of bipolar argumentation in the context of probabilistic argumentation can be found in (Polberg and Hunter 2018). While pure AAFs can be sufficient for interesting applications, support edges are vital for many recent applications of argumentation frameworks. For example, in review aggregation (Cocarascu, Rago, and Toni 2019), it is important that both positive and negative aspects of a review are taken into account. In fake news detection (Kotonya and Toni 2019), claims in an article can be contradicted or supported by other sources. As a final example, in product recommendation (Rago, Cocarascu, and Toni 2018), there can be features that make the product more (support) or less (attack) interesting for a user.

8 Discussion and Conclusions

We discussed several ideas for implementing bipolar argumentation with dual attacks and supports. The idea of duality is that attacks and supports should be treated equally. While this may not be necessary for every application, it seems reasonable in domains where we want to balance pro and contra arguments without favoring one or the other. While the deductive semantics from (Potyka 2020) offers perfect symmetry between attack and support, it admits rather sceptical and indecisive labellings. The problem can be mitigated by considering stable deductive labellings, but even then labellings can reject arguments even though accepting them would not cause any conflict. The rationale for this is that when we accept arguments that are not attacked, we should dually reject arguments that are not supported in order to maintain symmetry. However, in order to define more credulous and decisive semantics for bipolar argumentation frameworks, we may have to give up the demand for complete symmetry.

We discussed s-deductive and m-deductive labellings that introduce a slight asymmetry in favor of accepting arguments. Arguments whose attackers and opponents are all rejected, must be accepted. Even though there is no counterpart for supports, these labellings still feature symmetrical behaviour of attack and support in many cases. One important difference to deductive labellings is that an accepted attacker does not condemn an argument to be rejected. The attack can be cancelled by an accepted supporter, which allows the argument to become undecided. If there is conflicting evidence for the state of an argument, s-deductive labellings generally label the argument undecided. M-deductive labellings try to resolve the conflict by means of a majority decision. While deductive labellings always respect mediated and supported attacks as discussed in (Cayrol and Lagasquie-Schiex 2013), s-deductive and m-deductive labellings interpret them in a weaker sense. If all counterarguments along complex attack chains are rejected, mediated and supported attack behave as usual. However, if counterarguments along the argumentation chain are accepted, the attack can be cancelled. Even though complex attacks are defeasible in general, supported attacks are stronger than simple supports. In particular, supported supports behave dually and are stronger than simple attacks.

It is also interesting to note that, from a computational complexity perspective, reasoning with s-deductive and m-deductive labellings in BAFs is not harder than reasoning with stable labellings in AAFs. In particular, the three semantics are actually equivalent for AAFs (BAFs with empty support relation). However, since the constraints for s-deductive and m-deductive labellings can become more complex in real BAFs, computing them can be algorithmically more challenging than computing stable labellings. An implementation and empirical runtime evaluation is currently work in progress.

References

Alviano, M. 2018. The pyglaf argumentation reasoner. In *International Conference on Logic Programming (ICLP)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Amgoud, L., and Ben-Naim, J. 2016. Evaluation of arguments from support relations: Axioms and semantics. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pp–900.

Amgoud, L.; Cayrol, C.; Lagasquie-Schiex, M.-C.; and Livet, P. 2008. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems* 23(10):1062–1093.

Baroni, P.; Romano, M.; Toni, F.; Aurisicchio, M.; and Bertanza, G. 2015. Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation* 6(1):24–49.

Beierle, C.; Brons, F.; and Potyka, N. 2015. A software system using a sat solver for reasoning under complete, stable, preferred, and grounded argumentation semantics. In *Joint German/Austrian Conference on Artificial Intelligence (KI)*, 241–248. Springer.

Boella, G.; Gabbay, D. M.; van der Torre, L.; and Villata, S. 2010. Support in abstract argumentation. In *International Conference on Computational Models of Argument (COMMA)*, 40–51. Frontiers in Artificial Intelligence and Applications, IOS Press.

Caminada, M. W., and Gabbay, D. M. 2009. A logical account of formal argumentation. *Studia Logica* 93(2-3):109.

Cayrol, C., and Lagasquie-Schiex, M.-C. 2013. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning* 54(7):876–899.

Cocarascu, O.; Rago, A.; and Toni, F. 2019. Extracting dialogical explanations for review aggregations with argumentative dialogical agents. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1261–1269.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77(2):321–357.

Dunne, P. E., and Wooldridge, M. 2009. Complexity of abstract argumentation. In *Argumentation in artificial intelligence*. Springer. 85–104.

Dvorák, W.; Järvisalo, M.; Wallner, J. P.; and Woltran, S. 2012. Cegartix: A sat-based argumentation system. In *Pragmatics of SAT Workshop (POS)*.

Egly, U.; Gaggl, S. A.; and Woltran, S. 2008. Aspartix: Implementing argumentation frameworks using answer-set programming. In *International Conference on Logic Programming (ICLP)*, 734–738. Springer.

Kotonya, N., and Toni, F. 2019. Gradual argumentation evaluation for stance aggregation in automated fake news detection. In *Proceedings of the 6th Workshop on Argument Mining*, 156–166.

Lagniez, J.-M.; Lonca, E.; and Mailly, J.-G. 2015. Coquiaas: A constraint-based quick abstract argumentation solver. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, 928–935. IEEE.

Oren, N., and Norman, T. J. 2008. Semantics for evidence-based argumentation. In *International Conference on Computational Models of Argument (COMMA)*, 276–284. IOS Press.

Polberg, S., and Hunter, A. 2018. Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. *International Journal of Approximate Reasoning* 93:487–543.

Potyka, N. 2018. Continuous dynamical systems for weighted bipolar argumentation. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 148–157.

Potyka, N. 2019. Extending modular semantics for bipolar weighted argumentation. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 1722–1730.

Potyka, N. 2020. Abstract argumentation with markov networks. In *European Conference on Artificial Intelligence (ECAI)*, in press.

Rago, A.; Toni, F.; Aurisicchio, M.; and Baroni, P. 2016. Discontinuity-free decision support with quantitative argumentation debates. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 63–73.

Rago, A.; Cocarascu, O.; and Toni, F. 2018. Argumentation-based recommendations: Fantastic explanations and how to find them. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1949–1955.