

Admissibility in Strength-Based Argumentation: Complexity and Algorithms

Yohann BACQUEY^a Jean-Guy MAILLY^a Pavlos MORAITIS^{a,b} Julien ROSSIT^a

^a *Université Paris Cité, LIPADE, F-75006 Paris, France*

yohann.bacquey@etu.u-paris.fr; {jean-guy.mailly, pavlos.moraitis, julien.rossit}@u-paris.fr

^b *Argument Theory, Paris, France*

Abstract. Recently, Strength-based Argumentation Frameworks (StrAFs) have been proposed to model situations where some quantitative strength is associated with arguments. In this setting, the notion of accrual corresponds to sets of arguments that collectively attack an argument. Some semantics have already been defined, which are sensitive to the existence of accruals that collectively defeat their target, while their individual elements cannot. However, until now, only the surface of this framework and semantics have been studied. Indeed, the existing literature focuses on the adaptation of the stable semantics to StrAFs. In this paper, we push forward the study and investigate the adaptation of admissibility-based semantics. Especially, we show that the strong admissibility defined in the literature does not satisfy a desirable property, namely Dung's fundamental lemma. We therefore propose an alternative definition that induces semantics that behave as expected. We then study computational issues for these new semantics, in particular we show that complexity of reasoning is similar to the complexity of the corresponding decision problems for standard argumentation frameworks in almost all cases. We then propose a translation in pseudo-Boolean constraints for computing (strong and weak) extensions. We conclude with an experimental evaluation of our approach which shows in particular that it scales up well for solving the problem of providing one extension as well as enumerating them all.

Keywords. abstract argumentation, argument strength, accrual of arguments

1. Introduction

Among widespread knowledge representation and reasoning techniques proposed in the literature of Artificial Intelligence over the last decades, Abstract Argumentation [1] is an intuitive but yet powerful tool for dealing with conflicting information. Since then, the initial work of Dung has been actively extended and enriched in many directions, *e.g.* considering other kinds of relations between arguments [2] or additional information associated with arguments or attacks [3,4]. Among them, Strength-based Argumentation Frameworks (StrAFs) [5] allow to associate a quantitative information with each argument. This information is a weight that intuitively represents the intrinsic strength of an

argument, and is then naturally combined with attacks between arguments to induce a defeat relation that allows either to confirm an attack between two arguments or to cancel it, if the attacked argument is stronger than the attacker (w.r.t. their respective weights). StrAFs extend further this notion of defeat among arguments by building a defeat that is based on a collective attack of a group of arguments (or accrual) and by offering associate semantics. Within these semantics, arguments can collectively defeat arguments that they cannot defeat individually. Intuitively speaking, these accrual-sensitive semantics allow some kind of compensation among arguments, where the accumulation of weak arguments can create a synergy and get rid of a stronger one they collectively attack. This reasoning approach allows to produce extensions that are not considered when applying classical semantics. In [5] the authors presented the basics of StrAFs inspired by Dung’s semantics for abstract argumentation along with some theoretical and computational results concerning classical issues related to abstract argumentation (*i.e.* acceptability semantics, semantics inclusion, extensions existence and verification, etc.). In this paper we propose a state of the art advancement in StrAFs by presenting original theoretical and computational results related to different aspects. More particularly the contribution of this work lies into the following aspects. The semantics proposed in [5] exist in two versions (namely strong, and weak). Roughly speaking, a set is strongly conflict-free iff none of its elements attacks another one, whereas a set is weakly conflict-free iff it does not contain any (successful) accrual against one of its elements. After detecting that strong admissibility fails to satisfy a desirable property in Dung’s abstract argumentation frameworks, namely his Fundamental Lemma [1], we propose an alternative definition for strong admissibility in order to remedy this issue and we define new admissibility-based semantics for StrAFs. Furthermore, we study the complexity of reasoning with these semantics and in particular we show that, surprisingly, the complexity does not increase w.r.t. the complexity of reasoning with AFs. For computing the extensions under these semantics, we propose algorithms based on pseudo-Boolean constraints.

2. Background Notions

We assume that the reader is familiar with abstract argumentation [1]. We consider finite *argumentation frameworks* (AFs) $\langle \mathcal{A}, \mathcal{R} \rangle$, where \mathcal{A} is the set of *arguments*, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is the *attack relation*. We will use $cf(AF)$ and $ad(AF)$ to denote, respectively, the conflict-free and admissible sets of an AF AF , and $co(AF)$, $pr(AF)$ and $st(AF)$ for its extensions under the complete, preferred and stable semantics. For more details on the semantics of AFs, we refer the interested reader to [1,6]. A *Strength-based Argumentation Framework* (StrAF) [5] is a triple $StrAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ where \mathcal{A} and \mathcal{R} are arguments and attacks, and $\mathcal{S} : \mathcal{A} \rightarrow \mathbb{N}$ is a *strength function*. An example of such a StrAF is depicted at Figure 1, where nodes represent arguments, edges represent attacks, and the numbers close to the nodes represent the arguments strength. These strengths

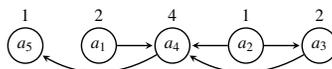


Figure 1. A StrAF Example

intuitively represent the intrinsic robustness associated with an argument and allow to

induce a *defeat* relation: an argument a defeats another argument b when a attacks b and the strength associated with b does not overcome that with a . This framework also offers the notion of collective defeat, *i.e.* sets of arguments that can jointly defeat their target while they cannot do so separately. First, we call an *accrual* a set of arguments that collectively attack a same target, *i.e.* a set $\kappa \subseteq \mathcal{A}$ s.t. $\exists c \in \mathcal{A}$ s.t. $\forall a \in \kappa, (a, c) \in \mathcal{R}$. Moreover, we say that κ is an *accrual that attacks c* . Then, for $\kappa' \subseteq \mathcal{A}$ an accrual, κ attacks κ' iff $\exists a \in \kappa'$ s.t. κ attacks a .

Example 1. Consider again StrAF from Figure 1. We can observe several examples of accruals, e.g. $\kappa_1 = \{a_1, a_2\}$ and $\kappa_2 = \{a_1, a_3\}$, that both attack a_4 . Notice that any attack $(a_i, a_j) \in \mathcal{R}$ induces an accrual $\{a_i\}$ attacking a_j .

We need to assess the collective strength of an accrual.

Definition 1 (Collective Strength). Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF and $\kappa = \{a_1, \dots, a_n\} \subseteq \mathcal{A}$ be an accrual. Then the collective strength associated with κ is $\text{coval}_{\oplus}(\kappa) = \oplus(\mathcal{S}(a_1), \dots, \mathcal{S}(a_n))$ where \oplus is an aggregation operator.

The operator \oplus must satisfy some properties discussed in [5]. An example of suitable operator is $\oplus = \Sigma$. If \oplus is clear from the context, we simply write coval for coval_{\oplus} .

Definition 2 (Collective Defeat). Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF, $a \in \mathcal{A}$, and \oplus an aggregation operator. Then, an accrual κ defeats a with respect to coval_{\oplus} , denoted by $\kappa \triangleright_{\oplus} a$, iff $\kappa \subseteq \mathcal{A}$ is an accrual that attacks a and $\text{coval}_{\oplus}(\kappa) \geq \mathcal{S}(a)$. If \oplus is clear from the context, we use $\kappa \triangleright a$ instead of $\kappa \triangleright_{\oplus} a$.

In the rest of the paper, we focus on $\oplus = \Sigma$ in examples and the pseudo-Boolean encoding defined in Section 4.2. But, unless explicitly stated otherwise, our results remain valid for any \oplus satisfying the properties from [5].

Definition 3. Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF, \oplus an aggregation operator and $\kappa \subseteq \mathcal{A}$, $\kappa' \subseteq \mathcal{A}$ two accruals. Then κ defeats κ' , denoted by $\kappa \triangleright_{\oplus} \kappa'$, iff $\exists a \in \kappa'$ s.t. $\kappa \triangleright_{\oplus} a$.

Example 2. Continuing Example 1, notice that $\text{coval}_{\Sigma}(\kappa_1) = 3 < \mathcal{S}(a_4)$, so $\kappa_1 \not\triangleright a_4$. On the contrary, $\text{coval}_{\Sigma}(\kappa_2) = 4 \geq \mathcal{S}(a_4)$, so $\kappa_2 \triangleright a_4$.

StrAF semantics rely on two possible adaptations of the notion of conflict-freeness:

Definition 4 (Conflict-freeness/Defence). Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ a StrAF, \oplus an aggregation operator, and $S \subseteq \mathcal{A}$,

- S is strongly conflict-free iff $\nexists a, b \in S$ s.t. $(a, b) \in \mathcal{R}$.
- S is weakly conflict-free iff there are no accruals $\kappa_1 \subseteq S$ and $\kappa_2 \subseteq S$ s.t. $\kappa_1 \triangleright_{\oplus} \kappa_2$.
- S defends an element $a \in \mathcal{A}$ iff for all accruals $\kappa_1 \subseteq \mathcal{A}$, if $\kappa_1 \triangleright_{\oplus} a$, then there exists an accrual $\kappa_2 \subseteq S$ s.t. $\kappa_2 \triangleright_{\oplus} \kappa_1$.

Intuitively, strongly conflict-free sets are “classically” conflict-free, *i.e.* there is no attack between two arguments members of such a set. On the contrary, weakly conflict-free sets are “defeat-free”: attacks between arguments are permitted as long as they do not result in a defeat neither individual nor collective. We use (respectively) cf_S^{\oplus} and cf_W^{\oplus} to denote these sets (or simply cf_S and cf_W when \oplus is clear from the context). Then, admissibility and extension-based semantics can be defined either strong or weak. Namely:

Definition 5 (Semantics for StrAFs [5]). *Given $StrAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ a StrAF, \oplus an aggregation operator, and $S \subseteq \mathcal{A}$ a strong (resp. weak) conflict-free set,*

- *S is a strong (resp. weak) admissible set iff S defends all elements of S .*
- *S is a strong (resp. weak) preferred extension iff S is a \subseteq -maximal strong (resp. weak) admissible set.*
- *S is a strong (resp. weak) stable extension iff $\forall a \in \mathcal{A} \setminus S, \exists \kappa \subseteq S$ s.t. $\kappa \triangleright_{\oplus} a$.*

For σ an extension-based semantics and $X \in \{S, W\}$ meaning respectively *strong* and *weak*, we use $\sigma_X^{\oplus}(StrAF)$ to denote the X - σ extensions of $StrAF$. We drop \oplus from the notation where there is no possible ambiguity. It is proven in [5] that Dung’s AFs are a subclass of StrAFs, where strong and weak semantics coincide. This result is useful for proving complexity results. However, in [5] authors only focus on complexity issues for the (weak and strong) stable semantics.

3. Admissibility-based Semantics for StrAFs

In our study, we investigate computational issues for admissibility-based semantics. A formal definition of (weak or strong) complete semantics is missing in [5], but matching the definition of (weak or strong) admissibility with the classical definition of the complete semantics, a straightforward definition can be stated as follows:

Definition 6. *Let $StrAF = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF, and \oplus an aggregation operator. A strong (resp. weak) admissible set $S \subseteq \mathcal{A}$ is a strong (resp. weak) complete extension of $StrAF$ if S contains all the arguments that it defends.*

Now we study these semantics and in particular, we show that surprisingly this intuitive definition of the complete semantics based on strong admissibility fails to satisfy a desirable property, namely the Fundamental Lemma, which states that admissible sets can be extended by the arguments that they defend. This leads us to redefine strong admissibility (and the associated complete and preferred semantics) in Section 3.1. On the contrary, the definition of weak admissibility is proved to be suitable in Section 3.2.

3.1. Revisiting Strong Admissibility

First, we observe that the usual inclusion relation between the preferred and complete semantics is not satisfied for the strong semantics of StrAFs. Moreover, the universal existence of complete extensions does not hold either.

Proposition 1. *There exists StrAF s.t. $pr_S(StrAF) \not\subseteq co_S(StrAF)$, and $co_S(StrAF) = \emptyset$.*

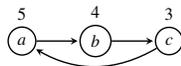


Figure 2. Example proving that $pr_S(StrAF) \not\subseteq co_S(StrAF)$

The StrAF from Figure 2 shows that the Fundamental Lemma does not hold for strong semantics of StrAFs: $\{a\}$ is strongly admissible, and defends c , but $\{a\} \cup \{c\}$ is not strongly admissible. A way to solve this issue is to redefine strong admissibility:

Definition 7 (Strong Semantics Revisited). *Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF and \oplus an aggregation operator. A set $S \in \text{cf}_S(\text{StrAF})$ strongly defends an argument a if S defends a against all the accruals that defeat it, i.e. $\forall \kappa \subseteq \mathcal{A}$ s.t. $\kappa \triangleright a$, $\exists \kappa' \subseteq S$ s.t. $\kappa' \triangleright \kappa$, and $S \cup \{a\}$ is strongly conflict-free. Then, a set $S \subseteq \mathcal{A}$ is strongly admissible if it is strongly conflict-free and it strongly defends all its elements. Moreover, S is a strong preferred extension iff S is a \subseteq -maximal strong admissible set; and S is a strong complete extension iff S strongly defends all its elements.*

If we consider again the StrAF from Figure 2, observe that this time, the strongly admissible set $\{a\}$ does not strongly defend c , since $\{a, c\}$ is not strongly conflict-free. Thus $\{a\}$ is a strong complete extension of this StrAF, following Definition 7. Now, Dung's Fundamental Lemma can be adapted to strong admissibility.

Lemma 1 (Fundamental Lemma for Strong Admissibility). *Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF and \oplus an aggregation operator. Let $S \subseteq \mathcal{A}$ be a strongly admissible set, and a, a' two arguments that are strongly defended by S against all their defeaters. Then, $S' = S \cup \{a\}$ is strongly admissible.*

Lemma 1 implies a relation between strong preferred and complete extensions:

Proposition 2. *For any StrAF and \oplus , $\text{pr}_S^\oplus(\text{StrAF}) \subseteq \text{co}_S^\oplus(\text{StrAF})$.*

This guarantees the existence of at least one strong complete extension for any StrAF: since \emptyset is a strong admissible set for any StrAF, then StrAF admits some \subseteq -maximal strong admissible sets, i.e. $\text{pr}_S(\text{StrAF}) \neq \emptyset$, which implies $\text{co}_S(\text{StrAF}) \neq \emptyset$.

Example 3. *Let us consider again the StrAF provided by Figure 1. Its strongly admissible sets are $\text{ad}_S(\text{StrAF}) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3, a_5\}\}$. Then, the strong preferred and complete extensions are $\text{pr}_S(\text{StrAF}) = \text{co}_S(\text{StrAF}) = \{\{a_1, a_2\}, \{a_1, a_3, a_5\}\}$.*

Finally, we prove that the new definition of strong admissibility does not change the fact that strong stable extensions are strongly admissible (and even strong preferred).

Proposition 3. *For any StrAF and \oplus , $\text{st}_S^\oplus(\text{StrAF}) \subseteq \text{pr}_S^\oplus(\text{StrAF})$.*

3.2. Properties of the Weak Semantics

Regarding now weak semantics as defined in [5], the usual result still holds for StrAFs.

Lemma 2 (Fundamental Lemma for Weak Admissibility). *Let $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a StrAF and \oplus an aggregation operator. Given $S \in \text{ad}_W(\text{StrAF})$, and a, a' two arguments that are defended by S , $S' = S \cup \{a\}$ is weakly admissible, and S' defends a' .*

Proposition 4. *For any StrAF and \oplus , $\text{pr}_W^\oplus(\text{StrAF}) \subseteq \text{co}_W^\oplus(\text{StrAF})$.*

Similarly to what we have noticed previously for strong admissibility, \emptyset is weakly admissible for any StrAF. This implies the existence of at least one weak preferred extension, and then one weak complete extension for any StrAF.

Example 4. *Consider again the StrAF from Figure 1. One identifies the weakly admissible sets $\text{ad}_W(\text{StrAF}) = \text{ad}_S(\text{StrAF}) \cup \{\{a_2, a_3\}, \{a_1, a_2, a_3\}, \{a_1, a_2, a_3, a_5\}\}$. Then, $\text{pr}_W(\text{StrAF}) = \text{co}_W(\text{StrAF}) = \{\{a_1, a_2, a_3, a_5\}\}$.*

Contrary to the case of stable semantics [5], we do not have $\text{co}_S(\text{StrAF}) \subseteq \text{co}_W(\text{StrAF})$. This comes from the fact that our strong and weak complete semantics are not based on the same notion of defense. However, we observe that, for any StrAF, each strong complete extension is included in some weak preferred (and complete) extension.

Proposition 5 (Strong/Weak Semantics Relationship). *Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and \oplus an aggregation operator, $\forall E \in \text{co}_S(\text{StrAF}), \exists E' \in \text{pr}_W(\text{StrAF})$ s.t. $E \subseteq E'$.*

We do not need a counterpart to Proposition 3: the definition of semantics based on weak admissibility is not modified, so [5, Proposition 1] still holds in this case.

3.3. Dung Compatibility

Previous work on StrAFs showed that this framework generalizes Dung’s AF, with a correspondence of StrAF semantics with AF semantics in this case. Following the new definition of strong admissible sets, one might fear that this property does not hold for strong admissibility-based semantics. However, we show here that it still does, as well as for weak complete semantics. Let us recall the transformation of an AF into a StrAF [5].

Definition 8. *Given an argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, the StrAF associated with AF is $\text{StrAF}_{AF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ with $\mathcal{S}(a) = 1, \forall a \in \mathcal{A}$ and $\text{coval} = \Sigma$.*

Now we can state the following proposition, that extends Dung Compatibility from [5] to the semantics studied in this paper.

Proposition 6 (Dung Compatibility). *Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, and $\text{StrAF}_{AF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ from Def. 8. For $\sigma \in \{\text{ad}, \text{pr}, \text{co}\}$, $\sigma(AF) = \sigma_X(\text{StrAF}_{AF})$, for $X \in \{S, W\}$.*

4. Complexity and Algorithms

Now we provide some insight on computational issues for admissibility-based semantics of StrAFs, *i.e.* we identify the computational complexity of several classical reasoning problems under these semantics, and we provide algorithms (based on pseudo-Boolean encoding) for solving them. While the complexity results are generic regarding the choice of \oplus , the algorithms focus on $\oplus = \Sigma$.

4.1. Complexity Analysis

We assume that the reader is familiar with basic notions of complexity, and otherwise we refer to [7] for details on complexity in argumentation, and [8] for a more general overview of complexity. We focus on three classical reasoning problems in abstract argumentation, namely *verification* (“Is a given set of arguments an extension?”), *credulous acceptability* (“Is a given argument in some extension?”) and *skeptical acceptability* (“Is a given argument in each extension?”). Formally, for $\sigma \in \{\text{ad}, \text{pr}, \text{co}\}$ and $X \in \{S, W\}$:

- σ -X-Ver: Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $S \subseteq \mathcal{A}$, is S a member of $\sigma_X(\text{StrAF})$?
- σ -X-Cred: Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $a \in \mathcal{A}$, is a in some $S \in \sigma_X(\text{StrAF})$?
- σ -X-Skep: Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $a \in \mathcal{A}$, is a in each $S \in \sigma_X(\text{StrAF})$?

We recall that these reasoning problems are already considered only for the (weak and strong) stable semantics in [5]. In the following, we assume a fixed \oplus , that can be computed in polynomial time. This is not a very strong assumption, since it is the case with the classical aggregation operators (e.g. \sum , \max , \dots). Proposition 6 implies that the complexity of reasoning with standard AFs provides a lower bound complexity of reasoning with StrAFs. So we focus on identifying upper bounds.

Proposition 7. *The complexity of the decision problems σ -X-Ver, σ -X-Cred and σ -X-Skep is as described in Table 1.*

	σ -X-Ver	σ -X-Cred	σ -X-Skep
ad_X	P	NP-c	Trivial
co_X	P	NP-c	in coNP
pr_X	coNP-c	NP-c	Π_2^P -c

Table 1. Complexity of reasoning for σ_X with $\sigma \in \{\text{ad}, \text{co}, \text{pr}\}$ and $X \in \{S, W\}$. Trivial means that all instances are trivially “NO” instances, and \mathcal{C} -c means \mathcal{C} -complete, for \mathcal{C} a complexity class in the polynomial hierarchy.

As for the strong (resp. weak) stable semantics [5], we prove here that the higher expressivity of StrAFs (compared to AFs) does not come at the price of a complexity blow-up. Only the case of skeptical acceptability under strong (resp. weak) complete semantics requires a deeper analysis, since we only provide the coNP upper bound. We observe that the choice of the weak or strong semantics does not impact on the complexity.

4.2. Algorithms

For computing the strong (resp. weak) admissible sets and complete extensions, we propose a translation of StrAF semantics in pseudo-Boolean (PB) constraints [9]. Such a constraint is an (in)equality $\sum_i w_i \times l_i \# k$ where w_i and k are positive integers, and $\# \in \{>, \geq, =, \neq, \leq, <\}$. l_i is a literal, i.e. $l_i = v_i$ or $l_i = \bar{v}_i = 1 - v_i$, where v_i is a Boolean variable. Determining whether a set of PB constraints has a solution is a NP-complete problem, that generalizes the Boolean satisfiability (SAT) problem. Despite the high complexity of this problem, it can be efficiently solved in many cases, see e.g. [10,11].

Strong and Weak Conflict-freeness. Now we describe our PB encoding of StrAF semantics. For ensuring self-containment of the paper, we recall the encoding of strong and weak conflict-freeness [5]. Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $\text{coval} = \sum$, we define a set of Boolean variables $\{x_i \mid a_i \in \mathcal{A}\}$ associated with each argument, where $x_i = 1$ means that a_i belongs to the set of arguments characterized by the solutions of the PB constraints. Then, strong and weak conflict-freeness are respectively encoded by (1) and (1’):

$$(1) \quad \forall (a_i, a_j) \in \mathcal{R}, \text{ add the constraint } x_i + x_j \leq 1$$

$$(1') \quad \forall a \in \mathcal{A}, \text{ add the constraint } \sum_{a_i \in \Gamma^-(a)} \mathcal{S}(a_i) \times x_i < x \times \mathcal{S}(a) + \bar{x} \times M$$

with M an arbitrary large natural number that is greater than the sum of the strengths of the arguments (i.e. $M > \sum_{a \in \mathcal{A}} \mathcal{S}(a)$), $\Gamma^-(a) = \{b \mid (b, a) \in \mathcal{R}\}$ is the set of attackers of $a \in \mathcal{A}$, and x is the Boolean variable associated with a .¹ A solution to the set of

¹Notice that the constraints referring to $\Gamma^-(a)$ must be added even when $\Gamma^-(a) = \emptyset$.

constraints **(1)** (resp. **(1')**) yields a strong (resp. weak) conflict-free set $E = \{a_i \mid x_i = 1\}$. We prove this claim with Proposition 8. First, let us introduce some notations. Given $S \subseteq \mathcal{A}$, $\omega_S : \{x_i \mid a_i \in \mathcal{A}\} \rightarrow \{0, 1\}$ is a mapping s.t. $\omega_S(x_i) = 1$ iff $a_i \in S$.

Proposition 8. *Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $S \subseteq \mathcal{A}$, $S \in \text{cf}_S(\text{StrAF})$ (resp. $S \in \text{cf}_W(\text{StrAF})$) iff ω_S satisfies the set of constraints **(1)** (resp. **(1')**).*

Strong and Weak Admissibility. For encoding strong (resp. weak) admissibility, one must add to the set of constraints **(1)** (resp. **(1')**) some new constraints that represent the strong defense (resp. defense) property. To do so, one needs to introduce new Boolean variables $\{y_i \mid a_i \in \mathcal{A}\}$ s.t. $y_i = 1$ means that a_i is defeated by the set of arguments characterized by the solution of the PB constraints. Then, three constraints are added (the same ones for strong and weak admissibility):

- (2)** $\forall a \in \mathcal{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathcal{S}(a_i) \times x_i \geq y \times \mathcal{S}(a)$
- (3)** $\forall a \in \mathcal{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathcal{S}(a_i) \times x_i \leq \bar{y} \times \mathcal{S}(a) + y \times M$
- (4)** $\forall a \in \mathcal{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathcal{S}(a_i) \times \bar{y}_i \leq x \times \mathcal{S}(a) + \bar{x} \times M$

The sets of constraints **(2)** and **(3)** ensure that $y = 1$ iff a is defeated by some $\kappa \subseteq E = \{a_i \mid x_i = 1\}$, and the constraints **(4)** ensure that E defends all its elements. The following proposition shows the correctness of the encodings.

Proposition 9. *Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $S \subseteq \mathcal{A}$, $S \in \text{ad}_S(\text{StrAF})$ (resp. $S \in \text{ad}_W(\text{StrAF})$) iff ω_S satisfies the sets of constraints **(1)** (resp. **(1')**), **(2)**, **(3)** and **(4)**.*

Strong and Weak Complete Semantics. Now, for computing the strong (resp. weak) extensions, one must consider the sets of constraints **(1)** (resp. **(1')**), **(2)**, **(3)** and **(4)**, and add a last set of constraints, respectively **(5)** for strong complete semantics, and **(5')** for weak complete semantics:

- (5)** $\forall a \in \mathcal{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} (\mathcal{S}(a_i) \times \bar{y}_i) + \sum_{a_i \in \Gamma^-(a)} (M \times x_i) + \sum_{a'_i \in \Gamma^+(a)} (M \times x'_i) \geq \bar{x} \times \mathcal{S}(a)$
- (5')** $\forall a \in \mathcal{A}$, add the constraint $\sum_{a_i \in \Gamma^-(a)} \mathcal{S}(a_i) \times \bar{y}_i \geq \bar{x} \times \mathcal{S}(a)$

where $\Gamma^+(a) = \{b \in \mathcal{A} \mid (a, b) \in \mathcal{R}\}$ is the set of arguments attacked by a . These constraints ensure that an argument is not accepted only if it is not (strongly) defended. Again, we prove the correctness of the encodings:

Proposition 10. *Given $\text{StrAF} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ and $S \subseteq \mathcal{A}$, $S \in \text{co}_S(\text{StrAF})$ (resp. $S \in \text{co}_W(\text{StrAF})$) iff ω_S satisfies the sets of constraints **(1)** (resp. **(1')**), **(2)**, **(3)**, **(4)** and **(5)** (resp. **(5')**).*

Acceptability and Verification. Obtaining one (resp. each) solution for one of the sets of constraints defined previously corresponds to obtaining one (resp. each) extension of the StrAF under the corresponding semantics. For checking whether a given argument a_i is credulously accepted, one simply needs to add the constraint $x_i = 1$. If a solution exists, then it corresponds to an extension that contains a_i , proving that this argument is credulously accepted. Otherwise, a_i is not credulously accepted. For skeptical acceptability, one needs to add the constraint $x_i = 0$. In this case, a solution exhibits an extension that does not contain a_i , thus this argument is not skeptically accepted. In the case where no

solution exists, then the argument is skeptically accepted. Finally, for checking whether a set of arguments $S \subseteq \mathcal{A}$ is an extension, one needs to add the constraints $x_i = 1$ for each $a_i \in S$, as well as $x_i = 0$ for each $a_i \in \mathcal{A} \setminus S$. A solution exists for the new set of constraints iff S is an extension under the considered semantics.

Strong and Weak Preferred Semantics. Finally, let us mention an approach to handle reasoning with strong and weak preferred semantics. Because of the higher complexity of skeptical reasoning under these semantics (recall Proposition 7), it is impossible (under the usual assumption that the polynomial hierarchy does not collapse) to find a (polynomial) encoding of these semantics in PB constraints. However, PB solvers can be used as oracles to find (with successive calls) preferred extensions. Algorithm 1 describes our method to do this for strong preferred semantics (replacing (1) by (1') provides an algorithm for weak preferred semantics). At start, we add the four constraints corresponding to a strong (resp. weak) admissible set and solve the instance, with the PB solver as a coNP oracle. Then we force the arguments within the extension to stay in the next one by adding the constraint on line 4. To avoid getting the same solution as in the previous step, we make sure that at least one argument outside the previous extension will be in the next one (line 5). This method iteratively extends an admissible set into a preferred extension, that is finally returned when the solver cannot find any (larger) solution.

Algorithm 1 Compute a strong preferred extension

```

P = PB problem with constraints (1), (2), (3) and (4)
while P.solve() ≠ null do
  E ← P.solve()
  P.add_constraint(x1 + x2 + ⋯ + xn = n), with E = {a1, a2, ..., an}
  P.add_constraint(x1 + x2 + ⋯ + xm ≥ 1), with  $\mathcal{A} \setminus E = \{a'_1, a'_2, \dots, a'_m\}$ 
end while
return E

```

5. Experimental Evaluation

For estimating the scalability of our method based on pseudo-Boolean constraints, we present now some results obtained from our experimental evaluation using two prominent PB solvers: Sat4j [12] and RoundingSat [11]. While Sat4j is based on saturation, RoundingSat uses the division rule (see [11] for a discussion on both approaches). We focus here on the most relevant results; full results are presented in [13].

Benchmark Generation. We generate benchmarks in a format adapted to StrAFs, inspired by ASPARTIX formalism [14]. We consider two classes of randomly generated graphs. First, with the Erdős–Rényi model (ER) [15], given a set of arguments \mathcal{A} , and $p \in [0, 1]$, we generate a graph such that for each $(a, b) \in \mathcal{A} \times \mathcal{A}$, a attacks b with a probability p . We consider two values for the probability, namely $p \in \{0.1, 0.5\}$. Then, with the Barabási–Albert (BA) model [16], a graph of n nodes is grown by attaching new nodes with m edges that are preferentially attached to existing nodes with a high degree. These types of graphs have been frequently used for studying computational aspects of formal argumentation, in particular during the ICCMA competitions [17]. The

choice of a generation model provides the arguments \mathcal{A} and attacks \mathcal{R} . We attach a random strength $\mathcal{S}(a) \in \{1, \dots, 20\}$ to each $a \in \mathcal{A}$. For each generation model, we build 20 StrAFs for each $|\mathcal{A}| \in \{5, 10, 15, \dots, 60\}$. Parameters ($p \in \{0.1, 0.5\}$ for ER, $m = 1$ for BA) are chosen to avoid graphs with a high density of attacks, that would prevent the existence of meaningful extensions (e.g. non-empty ones). Larger StrAFs (with $|\mathcal{A}| \in \{5, 10, \dots, 250\}$) have been generated with the same parameters ($p \in \{0.1, 0.5\}$ for ER, $m = 1$ for BA) for studying the problem of providing one extension.

Experimental Setting. The experiments were run on a Windows computer (using Windows Subsystem for Linux), with an Intel Core i5-6600K 3.50GHz CPU and 16GB of RAM. The timeout is set to 600 seconds (same as the timeout at ICCMA [18]).

Results. We are interested in the semantics σ_X , with $\sigma \in \{\text{pr}, \text{st}, \text{co}\}$ and $X \in \{S, W\}$. The encodings for st_X ($X \in \{S, W\}$) are those proposed in [5], while the encoding for the other semantics are those described in Section 4.2. For each generated *StrAF*, and each of these semantics σ_X , the two tasks we are interested in consist in enumerating all extensions and finding one extension. We first focus on the runtime for enumerating σ_X extensions, which provides an upper bound of the runtime for solving other classical reasoning tasks. To do so, we use a Python script that converts a StrAF into a set of PB constraints.² The set of extensions is then obtained in a classical iterative way: once an extension is returned by the PB solver, we add a new constraint that forbids this extension, and we call again the solver on this updated set of PB constraints. This process is repeated until the set of constraints becomes unsatisfiable, which means that all the extensions have been obtained. Concerning the preferred extensions, this iterative approach is combined with Algorithm 1. In order to measure the performance of our approach, and since there is no other computational approach for StrAF semantics yet, we also implemented a so-called *naive* algorithm that enumerates all sets of arguments and then checks, for each of them, if it is a σ_X extension. Figure 3 presents the average runtimes w.r.t. instance sizes (i.e. $|\mathcal{A}|$) for various semantics and StrAF families as described before. As a first result, we observe in Figure 3a that runtime for enumerating extensions (with the PB approach) is reasonable (i.e. less than a minute) for most of the cases considered in our study, when the PB approach is used, while the naive approach reaches the timeout for most of the large instances (in particular, all the instances with $|\mathcal{A}| \geq 45$). The average runtimes are higher in only two situations: the enumeration of strong preferred and strong complete extensions, with the BA graphs. However, even in such situations where the enumeration is harder (e.g. for pr_S -extensions on BA graphs, as depicted on Figure 3b), the PB solvers clearly outperform the naive algorithm, which reaches the timeout in every instance when $|\mathcal{A}| \geq 30$, while the PB approach can enumerate extensions for larger graphs.

We also study the classical problem of providing one extension, for StrAFs of larger sizes (recall that here $|\mathcal{A}| \in \{5, 10, \dots, 250\}$). Figure 4 shows that the PB solvers (in particular, Sat4j) provide one extension for these large graphs under two minutes, even for the preferred semantics (which is the hardest one, in our study, from the computational point of view). Concerning the respective performances of the two PB solvers, Figure 4 shows that RoundingSat processes faster for fast-to-compute instances (i.e. the smallest ones), while Sat4j outperforms it for instances of larger size. While we do not have explanations for this phenomenon, a plausible assumption is that it is related to the difference

²The script is available here: <https://github.com/BacqueyYohann/SolvingStrafs>.

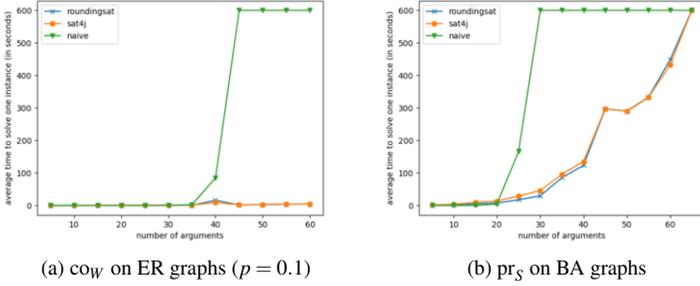


Figure 3. Enumeration runtime

of the underlying algorithms (saturation for Sat4j and division rule for RoundingSat). Similar things have been observed for SAT solvers used in the case of standard AFs [19].

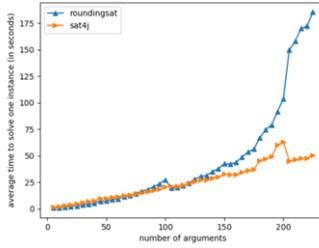


Figure 4. Finding one extension runtime under pr_S on BA graphs

As a general conclusion on our experimental analysis, we observe that the PB approach for reasoning with StrAFs generally scales up well, for both problems of enumerating extensions and providing one extension.

6. Conclusion

Strength-based Argumentation Frameworks (StrAFs) have originally been proposed in [5]. Contrary to this work, in this paper we focused on admissibility-based semantics. We showed that the weak admissibility-based semantics defined in the original work satisfy some expected properties, namely Dung’s Fundamental Lemma. However, the definition for strong admissibility proposed in [5] does not yield semantics that behave as expected. This has conducted us to revisit the definition of strong admissibility, and this allowed us to introduce strong complete and preferred semantics. We have also enhanced the StrAFs literature by studying the computational complexity of classical reasoning problems for these semantics, and we have shown that it is the same as for the corresponding tasks in Dung’s framework, in spite of the increase of expressivity. Then we have proposed a method based on pseudo-Boolean constraints for computing the extensions of a StrAF under the various semantics defined in this paper, and we have empirically evaluated the scalability of this approach for the new semantics defined in this

paper, as well as the (weak and strong) stable semantics from [5]. Future work include the study of (weak and strong) grounded semantics, and tight complexity results for the skeptical reasoning under the (weak and strong) complete semantics. We also plan to analyze the relation between StrAFs and other frameworks, in particular the comparison of the signatures of StrAFs semantics and SETAFs semantics [20,21,22]. Finally, we want to study argument strength and accrual in a context of structured argumentation.

References

- [1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif Intell.* 1995;77(2):321-58.
- [2] Cayrol C, Lagasque-Schieux M. Bipolarity in argumentation graphs: Towards a better understanding. *Int J Approx Reason.* 2013;54(7):876-99.
- [3] Amgoud L, Cayrol C. A Reasoning Model Based on the Production of Acceptable Arguments. *Annals of Mathematics and Artificial Intelligence.* 2002;34(1-3):197-215.
- [4] Bench-Capon TJM. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *J Log Comput.* 2003;13(3):429-48.
- [5] Rossit J, Maily JG, Dimopoulos Y, Moraitis P. United We Stand: Accruals in Strength-based Argumentation. *Argument Comput.* 2021;12(1):87-113.
- [6] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: *Handbook of Formal Argumentation.* College Publications; 2018. p. 159-236.
- [7] Dvorák W, Dunne PE. Computational Problems in Formal Argumentation and their Complexity. In: *Handbook of Formal Argumentation.* College Publications; 2018. p. 631-88.
- [8] Arora S, Barak B. *Computational Complexity - A Modern Approach.* Cambridge UP; 2009.
- [9] Roussel O, Manquinho VM. Pseudo-Boolean and Cardinality Constraints. In: *Handbook of Satisfiability;* 2009. p. 695-733.
- [10] Martins R, Manquinho VM, Lynce I. Open-WBO: A Modular MaxSAT Solver,. In: *Proc. of SAT'14;* 2014. p. 438-45.
- [11] Elffers J, Nordström J. Divide and Conquer: Towards Faster Pseudo-Boolean Solving. In: *Proc. of IJCAI'18;* 2018. p. 1291-9.
- [12] Le Berre D, Parrain A. The Sat4j library, release 2.2. *J Satisf Boolean Model Comput.* 2010;7(2-3):59-6.
- [13] Bacquey Y, Maily JG, Moraitis P, Rossit J. Admissibility in Strength-based Argumentation: Complexity and Algorithms (Extended Version with Proofs). arXiv; 2022. Available from: <https://arxiv.org/abs/2207.02258>.
- [14] Dvorák W, Gaggl SA, Rapberger A, Wallner JP, Woltran S. The ASPARTIX System Suite. In: *Proc. of COMMA'20;* 2020. p. 461-2.
- [15] Erdős P, Rényi A. On Random Graphs. I. *Publicationes Mathematicae.* 1959;6:290-7.
- [16] Barabasi AL, Albert R. Emergence of Scaling in Random Networks. *Science.* 1999;286(5439):509-12.
- [17] Gaggl SA, Linsbichler T, Maratea M, Woltran S. Design and results of the Second International Competition on Computational Models of Argumentation. *Artif Intell.* 2020;279.
- [18] Lagniez JM, Lonca E, Maily JG, Rossit J. Introducing the Fourth International Competition on Computational Models of Argumentation. In: *Proc. of SAFA'20;* 2020. p. 80-5.
- [19] Gning S, Maily JG. On the Impact of SAT Solvers on Argumentation Solvers. In: *Proc. of SAFA'20.* vol. 2672; 2020. p. 68-73.
- [20] Nielsen SH, Parsons S. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In: *Proc. of ArgMAS'06.* Springer; 2006. p. 54-73.
- [21] Dvorák W, Fandinno J, Woltran S. On the expressive power of collective attacks. *Argument Comput.* 2019;10(2):191-230.
- [22] Flouris G, Bikakis A. A comprehensive study of argumentation frameworks with sets of attacking arguments. *Int J Approx Reason.* 2019;109:55-86.