Optimisation and Approximation in Abstract Argumentation: The Case of Stable Semantics

Matthias Thimm
Artificial Intelligence Group, University of Hagen, Germany
matthias.thimm@fernuni-hagen.de

Abstract

We analyse two soft notions of stable extensions in abstract argumentation, one that weakens the requirement of having full range and one that weakens the requirement of conflict-freeness. We then consider optimisation problems over these two notions that represent optimisation variants of the credulous reasoning problem with stable semantics. We investigate the computational complexity of these two problems in terms of the complexity of solving the optimisation problem exactly and in terms of approximation complexity. We also present some polynomial-time approximation algorithms for these optimisation problems and investigate their approximation quality experimentally.

1 Introduction

Abstract argumentation frameworks (AAFs) [Dung, 1995] are an approach for modelling argumentative scenarios that represents arguments as nodes in a directed graph, where a directed edge represents an attack from one argument to another. Reasoning in AAFs consists of identifying sets of arguments (*extensions*) that form a *plausible* outcome of the argumentation. Several different semantical approaches for formalising plausibility in this context have been proposed, see e. g. [Dung, 1995; Baroni *et al.*, 2018]. In this work, we focus on *stable semantics*. A stable extension is a conflict-free set of arguments (no argument in the set attacks some other argument in the set) that attacks all arguments not contained in the set. Stable semantics is a very strong semantical notion and stable extensions are not guaranteed to exist [Dung, 1995].

In this work, we consider two soft notions of stable semantics that weaken the requirements of stable semantics by allowing to consider sets that do not necessarily attack all other arguments or are not necessarily conflict-free. While *qualitative* approaches for this motivation have been investigated with the *stage* [Verheij, 1996] and *semi-stable semantics* [Caminada, 2006] before, we consider here a *quantitative* treatment. In particular, a set of arguments is called a *k-stable extension* (with $k \in \mathbb{N}$) iff is conflict-free and the number of arguments contained in the set and attacked by the set is at least k. Therefore, k-stable extensions with larger values of k are "closer" of being a stable extension. The other variant we consider is the *k-stable* extension*, which attacks all arguments it does not contain and "respects" at least k attacks (meaning that for at least k attacks, it is not the case that both arguments involved in the attack are in the extension). These notions have several applications. For one, they allow to rank sets of arguments in terms of their closeness of being stable extensions, see e. g. [Skiba *et al.*, 2021] for a general treatment of this problem area. This allows to derive a graded notion of the acceptability of arguments and therefore judge their strength in a more fine-grained manner, as in ranking-based semantics [Amgoud and Ben-Naim, 2013; Bonzon *et al.*, 2016]. Another application is in *enforcement* [Baumann *et al.*, 2021], which addresses the problem of how to change an AAF in order to enforce arguments to become acceptable. Here, a set of arguments that is already "close" of being stable is likely to be easier to enforced than a set of arguments that are not "close" of being stable. The approaches considered in this work can therefore be used to guide practical approaches to enforcement.

Our analysis of the above two notions in this paper focusses on computational issues and, in particular, on the problem of *approximation*. The term *approximation* has received quite some attention in the context of AAFs in recent years [Kuhlmann and Thimm, 2019; Craandijk and Bex, 2020; Malmqvist *et al.*, 2020; Malmqvist, 2022; Delobelle *et al.*, 2023; Thimm, 2021] and the *International Competition on Computational Models of Argumentation* (ICCMA)[1] even featured *approximation tracks* in 2021 and 2023. However, these tracks and the aforementioned works are actually about *heuristic algorithms* for solving decision problems (such as the acceptability of arguments wrt. certain semantics) by relying, e. g., on heuristics informed by machine learning models [Kuhlmann and Thimm, 2019; Craandijk and Bex, 2020; Malmqvist *et al.*, 2020; Malmqvist, 2022]. But in theoretical computer science the term *approximation* [Vazirani, 2003] actually refers to algorithms that feature a theoretically guaranteed *approximation quality* when solving *optimisation problems*. We consider this definition as well in the present work. In particular, we consider the optimisation problems of finding the maximal value k, such that a given argument is in a k-stable or k-stable* extension, which thus represent

[1] http://argumentationcompetition.org/

graded variants of the classical credulous reasoning problem for stable semantics. We study the computational complexity of these optimisation problems from two perspectives. First, we determine the complexity for solving these optimisation problems *exactly* and show that they are $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete. Moreover, we analyse the hardness of approximation and show that the problem is Poly-APX-complete for k-stable semantics and Log-APX-hard for k-stable* semantics (both under PTAS-reductions). So both problems are, in general, very hard to approximate. Nonetheless, we present several simple polynomial-time approximation algorithms for these problems and show experimentally that their *average* approximation quality is rather good.

To summarise, the contributions of the paper are as follows.

1. We present two soft notions of stable semantics for abstract argumentation and analyse their general properties (Section 3).
2. We consider the two corresponding optimisation problems and show that they are both $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete (Section 4).
3. We analyse the complexity of approximation of these optimisation problems and show that they are Poly-APX-complete and Log-APX-hard (under PTAS-reductions), respectively (Section 5).
4. We present four polynomial-time approximation algorithms, two for k-stable semantics and two for k-stable* semantics (Section 6).
5. We analyse the general feasibility of these approximation algorithms in an experimental study and show that they perform quite good in practice (Section 7).

Section 2 gives some necessary background on abstract argumentation and Section 8 concludes the paper. Proofs of technical results can be found in an online appendix.[2]

2 Abstract argumentation

We consider abstract argumentation frameworks [Dung, 1995] defined as follows.

Definition 1. An *abstract argumentation framework* (AAF) F is a tuple $F = (A, R)$ where A is a (finite) set of *arguments* and $R \subseteq A \times A$ is the *attack relation*.

For $(a, b) \in R$ we also write aRb and say that a is an *attacker* of b. For an AF $F = (A, R)$ and any set $S \subseteq A$ we write

$$S_F^+ = \{b \in A \mid \exists a \in S : aRb\}$$
$$S_F^- = \{b \in A \mid \exists a \in S : bRa\}$$

to denote the set of attacked arguments by S (S_F^+) and the set of arguments attacking S (S_F^-). For singleton sets $\{a\} \subseteq A$ we write a_F^+ resp. a_F^- instead of $\{a\}_F^+$ resp. $\{a\}_F^-$. The *range* $S_F^\oplus$ of a set $S \subseteq A$ is defined as $S_F^\oplus = S \cup S_F^+$.

Semantics are given to an AF $F = (A, R)$ via extensions, i. e., sets of arguments that form a plausible outcome of the argumentation represented by F. In this paper, we focus on the *stable semantics* [Dung, 1995].

[2] https://mthimm.de/misc/ijcai24_opt_mt.pdf

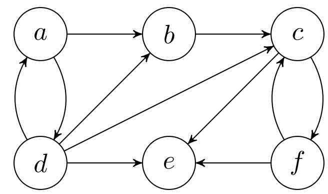

Figure 1: The AAF F_1 from Example 1.

Definition 2. Let $F = (A, R)$ be an AF and let $S \subseteq A$ be a set of arguments. We say that

- S is *conflict-free set* iff $S \cap S_F^+ = \emptyset$ and
- S is a *stable extension* iff S is conflict-free and $S_F^\oplus = A$.

Let $\mathsf{cf}(F)$ denote the set of all conflict-free sets of S and let $\mathsf{st}(F)$ denote the set of all stable extensions of F. Note that stable extensions do not necessarily exist for a given AAF.

Example 1. Consider the AAF F_1 in Figure 1. We have $\mathsf{st}(F_1) = \{\{a, c\}, \{a, f\}, \{d, f\}\}$.

We consider the following classical reasoning problems for AAFs [Dvořák and Dunne, 2018] (let σ be a semantics such as st):

$\textsc{Ver}_\sigma$:	**Input**	AF $F = (A, R)$, $S \subseteq A$
	Output	YES iff $S \in \sigma(F)$
$\textsc{Cred}_\sigma$:	**Input**	AF $F = (A, R)$, $a \in A$
	Output	YES iff $a \in \bigcup \sigma(F)$
$\textsc{Skep}_\sigma$:	**Input**	AF $F = (A, R)$, $a \in A$
	Output	YES iff $a \in \bigcap \sigma(F)$
$\textsc{Exists}_\sigma$:	**Input**	AF $F = (A, R)$
	Output	YES iff $\sigma(F) \neq \emptyset$
$\textsc{Exists}_\sigma^{\neg\emptyset}$:	**Input**	AF $F = (A, R)$
	Output	YES iff there is $S \in \sigma(F)$ with $S \neq \emptyset$

The computational complexity of the above problems for stable semantics are as follows, cf. [Dimopoulos and Torres, 1996; Dvořák and Dunne, 2018].

Theorem 1. $\textsc{Ver}_{\mathsf{st}}$ *is in P,* $\textsc{Cred}_{\mathsf{st}}$ *is NP-complete,* $\textsc{Skep}_{\mathsf{st}}$ *is coNP-complete,* $\textsc{Exists}_{\mathsf{st}}$ *is NP-complete,* $\textsc{Exists}_{\mathsf{st}}^{\neg\emptyset}$ *is NP-complete.*

3 Soft notions of stable extensions

In this section, we will investigate soft notions[3] of stable extensions that allow us to assess the acceptability of arguments in a more relaxed manner as with strict stable semantics. This allows us to judge arguments as "closer" being acceptable wrt. stable semantics than others.

A stable extension S is characterised by two properties: it has full range ($S_F^\oplus = A$) and it is conflict-free ($S \cap S_F^+ = \emptyset$). Our first approach (Section 3.1) weakens the first property and our second approach (Section 3.2) weakens the second.

[3] We use the term "soft" instead of the probably more appropriate "weak" to avoid confusion with the non-related notion of weak stable semantics from [Baumann *et al.*, 2022].

3.1 Softening the requirement of full range

We now consider a semantical notion that requires a set to be conflict-free (this is a strict requirement) and that at least a minimum number of arguments are contained in its range.

Definition 3. Let $F = (A, R)$ be an AF, $k \in \mathbb{N}$, and let $S \subseteq A$. We say that S is a *k-stable extension* iff S is conflict-free and $|S_F^{\oplus}| \geq k$. Let $\mathsf{st}_k(F)$ denote the set of k-stable extensions of F.

In other words, a conflict-free set S is a k-stable extension if the number of arguments in S and the number of arguments attacked by S is at least k.

Some simple observations are as follows.

Proposition 1. *Let $F = (A, R)$ be an AF, $k \in \mathbb{N}$, $S \subseteq A$, and $a \in A$.*

1. *S is a $|A|$-stable extension iff S is a stable extension.*
2. *If S is a k-stable extension then S is a k'-stable extension for all $k' < k$.*
3. *$S \in \mathsf{cf}(F)$ iff S is a 0-stable extension.*
4. *$(a, a) \notin R$ iff $\{a\}$ is a 1-stable extension.*
5. *$\emptyset$ is a 0-stable extension.*

Due to items 1–3 from above we obtain

$$\mathsf{st}(F) = \mathsf{st}_{|A|}(F) \subseteq \mathsf{st}_{|A|-1}(F) \subseteq \ldots \subseteq \mathsf{st}_0(F) = \mathsf{cf}(F),$$

so k-stable extensions cover the complete spectrum between stable extensions and conflict-free sets of any AF F, for $k = 0, \ldots, |A|$.

Example 2. We consider again the AAF F_1 in Figure 1. Since $\mathsf{st}(F_1) = \mathsf{st}_{|A_1|}(F_1)$ we have $\mathsf{st}_6(F_1) = \{\{a, c\}, \{a, f\}, \{d, f\}\}$. By allowing extensions to contain less arguments in the range we obtain

$$\begin{aligned}
\mathsf{st}_5(F_1) &= \mathsf{st}_6(F_1) \cup \{\{d\}\} \\
\mathsf{st}_4(F_1) &= \mathsf{st}_5(F_1) \cup \{\{a, e\}, \{b, f\}\} \\
\mathsf{st}_3(F_1) &= \mathsf{st}_4(F_1) \cup \{\{a\}, \{b, e\}, \{c\}, \{f\}\} \\
\mathsf{st}_2(F_1) &= \mathsf{st}_3(F_1) \cup \{\{b\}\} \\
\mathsf{st}_1(F_1) &= \mathsf{st}_2(F_1) \cup \{\{e\}\} \\
\mathsf{st}_0(F_1) &= \mathsf{st}_1(F_1) \cup \{\emptyset\}
\end{aligned}$$

Reasoning with k-stable semantics is as hard as reasoning with stable semantics when considering k as an additional input parameter.

Theorem 2. *Let $k \in \mathbb{N}$ be an additional input parameter to the following problems.* $\textsc{Ver}_{\mathsf{st}_k}$ *is in P,* $\textsc{Cred}_{\mathsf{st}_k}$ *is NP-complete,* $\textsc{Skep}_{\mathsf{st}_k}$ *is coNP-complete,* $\textsc{Exists}_{\mathsf{st}_k}$ *is NP-complete,* $\textsc{Exists}^{\neg\emptyset}_{\mathsf{st}_k}$ *is NP-complete.*

3.2 Softening the requirement of conflict-freeness

Our second variant for a softening of stable extensions requires that a set has full range (that is a strict requirement) and that it "satisfies" a maximal number of attacks, where a set of arguments *satisfies* an attack if it does not contain both the attacker and the attacked argument of that attack. More formally, for $F = (A, R)$ and a set $S \subseteq A$ we denote by

$$S_F^{\circledast} = \{(a, b) \in R \mid a \notin S \text{ or } b \notin S\}$$

the set of *satisfied attacks* of F by S. Any attack $(a, b) \in R \setminus S_F^{\circledast}$ is also called *violated attack* of F by S. Note that a set $S \subseteq A$ is conflict-free if and only if $S_F^{\circledast} = R$.

Definition 4. Let $F = (A, R)$ be an AF, $k \in \mathbb{N}$, and let $S \subseteq A$. We say that S is a *k-stable* extension* iff $S_F^{\oplus} = A$ and $|S_F^{\circledast}| \geq k$. Let $\mathsf{st}^*_k(F)$ denote the set of k-stable* extensions of F.

In other words, a set S with full range is a k-stable* extension if the number of satisfied attacks of S is at least k.

We make some simple observations about k-stable* extensions.

Proposition 2. *Let $F = (A, R)$ be an AF, $k \in \mathbb{N}$, $S \subseteq A$, and $a \in A$.*

1. *S is a $|R|$-stable* extension iff S is a stable extension.*
2. *If S is a k-stable* extension then S is a k'-stable* extension for all $k' < k$.*
3. *A is a 0-stable* extension.*
4. *If $a \in A$ is not attacked in F, then $a \in S$ for every k-stable* extension S.*

Example 3. We consider again the AAF F_1 in Figure 1. Since $\mathsf{st}(F_1) = \mathsf{st}^*_{|R_2|}(F_1)$ we have $\mathsf{st}^*_{11}(F_1) = \{\{a, c\}, \{a, f\}, \{d, f\}\}$. Due to the large number of extensions, we only give a complete list for $k = 10, 9, 8$:

$$\begin{aligned}
\mathsf{st}^*_{10}(F_1) = \mathsf{st}^*_{11}(F_1) \cup \{&\{a, b, f\}, \{a, c, e\}, \{a, e, f\}, \\
&\{b, d, f\}, \{c, d\}\} \\
\mathsf{st}^*_9(F_1) = \mathsf{st}^*_{10}(F_1) \cup \{&\{a, b, e, f\}, \{a, b, c\}, \{a, c, f\}, \\
&\{a, d, f\}, \{d, e, f\}\} \\
\mathsf{st}^*_8(F_1) = \mathsf{st}^*_9(F_1) \cup \{&\{b, d, e, f\}, \{a, b, c, e\}\}, \{a, c, d\}, \\
&\{b, c, d\}, \{c, d, e\}, \{c, d, f\}\}
\end{aligned}$$

Complexity-wise, k-stable* semantics behaves as k-stable and stable semantics.

Theorem 3. *Let $k \in \mathbb{N}$ be an additional input parameter to the following problems.* $\textsc{Ver}_{\mathsf{st}^*_k}$ *is in P,* $\textsc{Cred}_{\mathsf{st}^*_k}$ *is NP-complete,* $\textsc{Skep}_{\mathsf{st}^*_k}$ *is coNP-complete,* $\textsc{Exists}_{\mathsf{st}^*_k}$ *is NP-complete,* $\textsc{Exists}^{\neg\emptyset}_{\mathsf{st}^*_k}$ *is NP-complete.*

4 Complexity of optimisation

The notions of k-stable and k-stable* extensions "approach" the notion of stable extensions: the larger the value k the "closer" these extensions are to being stable extensions. Given an AAF $F = (A, R)$ and an argument $a \in A$, we are interested in finding the maximum value $k^\dagger$ such that a is contained in a $k^\dagger$-stable/$k^\dagger$-stable* extension. Formally, the main reasoning problems we are interested in are the following optimisation problems[4]:

$\textsc{MaxStable}$
Input AF $F = (A, R)$, $a \in A$
Output $\mathsf{mst}(F, a) := \max_k\{k \mid a \in \bigcup \mathsf{st}_k(F)\}$

$\textsc{MaxStable*}$
Input AF $F = (A, R)$, $a \in A$
Output $\mathsf{mst}^*(F, a) := \max_k\{k \mid a \in \bigcup \mathsf{st}^*_k(F)\}$

[4] We define $\max \emptyset = -\infty$

Note for MaxStable, the case that no k exists with $a \in \bigcup \mathsf{st}_k(F)$ can only happen when a attacks itself. For MaxStable*, the output is always a finite number as $a \in A \in \mathsf{st}^*_{|R|}(F)$ (see item 3 of Proposition 2).

The two problems above are optimisation variants of credulous reasoning wrt. stable semantics: we are seeking the maximum number $k^\dagger$ such that a given argument a can be credulously inferred wrt. $k^\dagger$-stable/$k^\dagger$-stable* semantics. An analysis for the optimisation variants of skeptical reasoning (where we seek the maximum $k^\dagger$ such that a given argument a is contained in *all* $k^\dagger$-stable/$k^\dagger$-stable* extensions) is left for future work.

Observe furthermore that solving MaxStable is not equivalent to reasoning with *stage semantics* [Verheij, 1996]. For an AAF $F = (A, R)$ recall that a set $S \subseteq A$ is a *stage extension* if S is conflict-free and $S^\oplus_F$ is *maximal wrt. set inclusion* among all conflict-free sets. In particular, stage semantics is defined in terms of maximality wrt. set inclusion while MaxStable seeks sets where the range is maximal wrt. cardinality. Moreover, MaxStable is defined wrt. to a given argument and every argument has a defined value, while an argument is credulously inferred wrt. stage semantics only if there is a stage extension containing that argument. However, a simple relationship between these two notions is captured in the following proposition.

Proposition 3. *Let $F = (A, R)$ be an AAF and $a \in A$. If there is a stage extension S with $a \in S$ then $\mathsf{mst}(F, a) \geq |S^\oplus_F|$.*

Another soft notion of a stable extension is that of a *semi-stable extension* [Caminada, 2006], which is an admissible[5] set that has maximal range (wrt. set inclusion). Since admissibility is a stronger notion than conflict-freeness, the bound given in Proposition 3 also applies when using semi-stable extensions instead of stage extensions. As k-stable extensions are not necessarily admissible, no more general relationships can be stated, however.

Example 4. We continue Example 2 and consider again the AAF F_1 from Figure 1. Here we get

$$\begin{aligned}\mathsf{mst}(F_1, a) &= \mathsf{mst}(F_1, c) = \mathsf{mst}(F_1, d) \\ &= \mathsf{mst}(F_1, f) = 6\end{aligned}$$

$$\mathsf{mst}(F_1, b) = \mathsf{mst}(F_1, e) = 4$$

and

$$\begin{aligned}\mathsf{mst}^*(F_1, a) &= \mathsf{mst}^*(F_1, c) = \mathsf{mst}^*(F_1, d) \\ &= \mathsf{mst}^*(F_1, f) = 11\end{aligned}$$

$$\mathsf{mst}^*(F_1, b) = \mathsf{mst}^*(F_1, e) = 10$$

Since credulous reasoning with k-stable/k-stable* semantics is NP-complete (see Theorems 2 and 3), the problems MaxStable and MaxStable* are naturally in the complexity class NPO, i. e., the class containing optimisation problems where the corresponding decision problems are in NP. We will now investigate the computational complexity of MaxStable and MaxStable* from two perspectives in more detail. First, we consider the complexity of (exact) optimisation and show that both problems are $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete. In Section 5 we will consider the complexity of *approximation* and show that MaxStable is Poly-APX-complete (under PTAS-reductions) while MaxStable* is Log-APX-complete (under PTAS-reductions).

The class $\mathsf{FP}^{\mathsf{NP}[\log]}$ is the class of functional problems that have a polynomial-time algorithm that may make logarithmically many calls to an NP-oracle. Membership of our problems MaxStable and MaxStable* for this class can be shown easily through an algorithm that does a binary search for the maximal value $k^\dagger$ and iteratively solving the problems $\textsc{Cred}_{\mathsf{st}_k}$ and $\textsc{Cred}_{\mathsf{st}^*_k}$, respectively. Hardness is shown through a reduction from the $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete problem MaxSATSize, i. e., the problem of determining the maximal number of clauses of a SAT instance that can be jointly satisfied.

Theorem 4. MaxStable *is $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete.*

Theorem 5. MaxStable* *is $\mathsf{FP}^{\mathsf{NP}[\log]}$-complete.*

5 Complexity of approximation

We will now consider the problem of approximating the solutions to MaxStable and MaxStable*, i. e., we are interested in algorithms that maximise the *approximation ratio* AR in the following sense:

$$\mathsf{AR} = \frac{\mathsf{APP}}{\mathsf{OPT}}$$

where OPT is the solution to either MaxStable or MaxStable* (given some $F = (A, R)$ and $a \in A$) and $\mathsf{APP} \leq \mathsf{OPT}$ is the solution of the considered approximation algorithm. An approximation ratio of 1 (which is the maximal value for AR) therefore indicates an optimal algorithm and lower values of AR indicate worse approximation quality. Note that $\mathsf{OPT} = 0$ is a possible solution for MaxStable* (e. g., it is the solution for b in $F = (\{a, b\}, \{(a, b)\})$). Then we also necessarily have $\mathsf{APP} = 0$ (due to $0 \leq \mathsf{APP} \leq \mathsf{OPT}$) and we just define $AR = 1$ in that case.

We now investigate the approximation complexity of MaxStable. For that, observe that the notion of a k-stable extension is related to another well-known notion from graph theory, namely *independent sets*.

Definition 5. Let $G = (V, E)$ be an undirected graph. A set $S \subseteq V$ is an *independent set* iff $(S \times S) \cap E = \emptyset$.

In other words, an independent set S of an undirected graph $G = (V, E)$ is a set of vertices of G, such that no two vertices of S are adjacent in G. The optimisation problem MaxIndepSet is defined as:

MaxIndepSet
Input $G = (V, E)$
Output $\max_{S \subseteq V} \{|S| \mid S \text{ is an independent set in } G\}$

In other words, the problem MaxIndepSet[6] for a given

[5] A set S is admissible iff S is conflict-free and for every $a \in A$ and $b \in S$ with aRb there is $c \in S$ with cRa.

[6] This problem is also called the *maximum independent set* problem and should not be confused with the *maximal independent set* problem that seeks an independent set that is maximal wrt. set inclusion (the latter problem can also be solved in polynomial time).

undirected graph $G = (V, E)$ asks for finding the largest number k such that there is an independent set $S \subseteq V$ with $|S| = k$. This is very much related to the problem MAXSTABLE. However, for MAXSTABLE we are seeking sets of vertices of a *directed graph* (instead of an undirected graph), where the cardinality of the *range* (instead of the cardinality of the set) is maximal. These differences actually pose a significant challenge in relating approximative approaches for MAXINDEPSET and MAXSTABLE in the proof of Theorem 6 below.

The problem MAXINDEPSET is Poly-APX-complete [Bazgan *et al.*, 2005], and therefore very hard to approximate (given P $\neq$ NP). The complexity class Poly-APX consists of optimisation problems, which can be approximated with polynomial-time algorithms that admit an approximation ratio that is bounded by a polynomial dependent on the size of the input. We use this result to also show Poly-APX-completeness of MAXSTABLE. So in our case, the best general lower bound of the approximation ratio we can guarantee is $\mathsf{AR} \geq 1/f(|A|)$ where f is a polynomial and A is the set of arguments of the input $F = (A, R)$.

We show Poly-APX-completeness under *PTAS-reductions* [Crescenzi, 1997], which have also been used to show Poly-APX-completeness of MAXINDEPSET in [Bazgan *et al.*, 2005]. Informally speaking, a PTAS-reduction is a polynomial-time reduction from a problem A to a problem B that preserves, to some degree, the approximation ratio of an algorithm for problem B to problem A. The formal definition can be found in the proof of the theorem below (see the supplementary material).

Theorem 6. MAXSTABLE *is Poly-APX-complete under PTAS-reductions.*

We now turn to the approximation complexity of MAXSTABLE*. The k-stable* extension also has a well-known counterpart in graph theory: the *dominating set*.

Definition 6. Let $G = (V, E)$ be an undirected graph. A set $S \subseteq V$ is a *dominating set* iff every $v \in V$ is either in S or adjacent to S.

So a dominating set S of an undirected graph $G = (V, E)$ is a set of vertices of G, such that all vertices of G are either in S or adjacent to it. The optimisation problem MINDOMSET is defined as:

MINDOMSET
Input $G = (V, E)$
Output $\min_{S \subseteq V}\{|S| \mid S \text{ is a dominant set in } G\}$

In other words, the problem MINDOMSET for a given undirected graph $G = (V, E)$ asks for finding the smallest number k such that there is a dominating set $S \subseteq V$ with $|S| = k$. The problem MINDOMSET is Log-APX-complete [Lund and Yannakakis, 1994]. The complexity class Log-APX consists of optimisation problems, which can be approximated with polynomial-time algorithms that admit an approximation ratio that is bounded by the logarithm of a polynomial dependent on the size of the input, i. e., we have $\mathsf{AR} \geq 1/\log(f(|I|))$ where f is a polynomial and $|I|$ is the size of the instance.

Again, we can observe that the problem MINDOMSET is related to our problem MAXSTABLE*. Observe first that although MAXSTABLE* is defined as a maximisation problem, we can consider an equivalent formalisation where we minimise the number of violated attacks (instead of maximising the number of satisfied attacks).[7] Comparing MINDOMSET with MAXSTABLE*, we see that in the latter we are seeking sets of vertices of a *directed graph* (instead of an *undirected graph*), where the cardinality of the set of edges between members in the set is minimal (instead of the cardinality of the set itself). While these differences also pose some interesting challenges in the proof, we are still able to reduce the problem MINDOMSET to MAXSTABLE* and therefore obtain Log-APX-hardness of MAXSTABLE*. Unfortunately, proving the membership of MAXSTABLE* to Log-APX is quite elusive, so we only give the lower bound.

Theorem 7. MAXSTABLE* *is Log-APX-hard under PTAS-reductions.*

6 Approximation algorithms

Despite the quite negative theoretical results from the previous two sections, we will now discuss four simple approximation algorithms (based on greedy search) for the problems MAXSTABLE and MAXSTABLE*, for which we will investigate the empirical *average* approximation ratio in Section 7.

6.1 Growing conflict-free sets (GCF)

Our first approximation algorithm addresses the problem MAXSTABLE, i. e., given $F = (A, R)$ and $a \in A$ we are looking for a conflict-free set S such that $a \in S$ and $|S^{\oplus}_F|$ is maximised. We do this by greedily adding arguments to a set, as long as this set remains conflict-free. Algorithm 1 depicts our GCF algorithm (*growing conflict-freeness*). In lines 1–2 we first check for the special case of a self-attacking argument and return $-\infty$ as the (optimal) solution in that case. We add our query argument a to a set S in line 3 and repeat lines 5–6 as long as there are arguments, which can be added to S without violating conflict-freeness (so as long as $S \cup S^+_F \cup S^-_F \neq A$). From those arguments, we pick one in

Algorithm 1 GCF algorithm for MAXSTABLE.

Input: $F = (A, R)$ and $a \in A$
Output: $k \in \mathbb{N}$ such that there is a conflict-free set S with $a \in S$ and $k = |S^{\oplus}_F|$.
1: **if** $(a, a) \in R$ **then**
2: **return** $-\infty$
3: $S \leftarrow \{a\}$
4: **while** $S \cup S^+_F \cup S^-_F \neq A$ **do**
5: Let $b \in A \backslash (S \cup S^+_F \cup S^-_F)$ s.t. $|(S \cup \{b\})^{\oplus}_F|$ is maximal
6: $S \leftarrow S \cup \{b\}$
7: **return** $|S^{\oplus}_F|$

[7] Note, however, that approximation ratios differ between the representations as minimisation and maximisation problems. We only consider the approximation ratio for the representation as maximisation problem.

line 5 that maximises the number of arguments in the range and add it to S (line 6).

Proposition 4. *Let k be the output of the* GCF *algorithm on input $F = (A, R)$ and $a \in A$. Let furthermore $k^\dagger$ be the optimal solution to* MAXSTABLE *on F and a. Then*

1. *there is a conflict-free set S with $a \in S$ and $k = |S_F^{\oplus}|$,*
2. *$k \geq k^\dagger/|A|$, and*
3. *k has been determined by* GCF *in polynomial time.*

6.2 Shrinking conflict-free sets (SCF)

Our second approximation algorithm for MAXSTABLE starts from the set of all arguments and iteratively removes arguments until we end up with a conflict-free set. Algorithm 2 depicts this SCF algorithm (*shrinking conflict-free sets*). Lines 1–2 again cover the special case of a self-attacking query argument. We initialise the set S in line 3 with the whole set of arguments from A. In line 5 we select an argument b (which has to be different from the query argument) that, upon removal, will produce a maximal number of satisfied attacks. We terminate once we reach a conflict-free set (line 4).

Algorithm 2 SCF algorithm for MAXSTABLE.

Input: $F = (A, R)$ and $a \in A$
Output: $k \in \mathbb{N}$ such that there is a conflict-free set S with $a \in S$ and $k = |S_F^{\oplus}|$.

1: **if** $(a, a) \in R$ **then**
2:     **return** $-\infty$
3: $S \leftarrow A$
4: **while** $S \cap S_F^{+} \neq \emptyset$ **do**
5:     Let $b \in S \setminus \{a\}$ s.t. $|(S \setminus \{b\})_F^{\circledast}|$ is maximal
6:     $S \leftarrow S \setminus \{b\}$
7: **return** $|S_F^{\oplus}|$

Proposition 5. *Let k be the output of the* SCF *algorithm on input $F = (A, R)$ and $a \in A$. Let furthermore $k^\dagger$ be the optimal solution to* MAXSTABLE *on F and a. Then*

1. *there is a conflict-free set S with $a \in S$ and $k = |S_F^{\oplus}|$,*
2. *$k \geq k^\dagger/|A|$, and*
3. *k has been determined by* SCF *in polynomial time.*

6.3 Growing full-range sets (GFR)

We now consider the problem MAXSTABLE*, i. e., given $F = (A, R)$ and $a \in A$ we are looking for a set S with $a \in S$ and $S_F^{\oplus} = A$ such that $S_F^{\circledast}$ is maximised. We provide a variant of the GCF algorithm (Section 6.1) that starts from the set S containing just a and iteratively adds arguments until $S_F^{\oplus} = A$. Algorithm 3 depicts this GFR algorithm (*growing full-range sets*). In line 1, the set S is initialised with a and line 3 adds the argument that will maximise the range. The algorithm terminates, when full range is achieved (line 2) and returns the number of satisfied attacks $S_F^{\circledast}$.

Proposition 6. *Let k be the output of the* GFR *algorithm on input $F = (A, R)$ and $a \in A$. Let furthermore $k^\dagger$ be the optimal solution to* MAXSTABLE* *on F and a. Then*

Algorithm 3 GFR algorithm for MAXSTABLE*.

Input: $F = (A, R)$ and $a \in A$
Output: $k \in \mathbb{N}$ such that there is a set S with $a \in S$, $S_F^{\oplus} = A$ and $k = |S_F^{\circledast}|$.

1: $S \leftarrow \{a\}$
2: **while** $S_F^{\oplus} \neq A$ **do**
3:     Let $b \in A \setminus S$ s.t. $|(S \cup \{b\})_F^{\oplus}|$ is maximal
4:     $S \leftarrow S \cup \{b\}$
5: **return** $|S_F^{\circledast}|$

1. *there is a set S with $a \in S$, $S_F^{\oplus} = A$, and $k = |S_F^{\circledast}|$,*
2. *$k \geq k^\dagger/|R|$, and*
3. *k has been determined by* GFR *in polynomial time.*

Item 2 of the above proposition only gives a polynomial bound on the approximation ratio. As mentioned right before Theorem 7, we suspect that MAXSTABLE* is Log-APX-complete and, moreover, that the GFR algorithm is a witness for Log-APX-membership. So the bound in item 2 can likely be improved to $k^\dagger/O(\log |R|)$, but we were not yet able to formally prove this.

6.4 Shrinking full-range sets (SFR)

We now provide a variant of the SCF algorithm (Section 6.2) for the problem MAXSTABLE* that starts from the set S containing all arguments and iteratively removes arguments until $S_F^{\oplus} = A$ can no longer be sustained. Algorithm 4 depicts this SFR algorithm (*shrinking full-range sets*). In line 1, the set S is initialised with A. As long as there is an argument b (that is not the query argument a), such that removing b from S will retain full range (line 3), such a b is removed from S that also restores a maximal number of satisfied attacks (line 3).

Algorithm 4 SFR algorithm for MAXSTABLE*.

Input: $F = (A, R)$ and $a \in A$
Output: $k \in \mathbb{N}$ such that there is a set S with $a \in S$, $S_F^{\oplus} = A$ and $k = |S_F^{\circledast}|$.

1: $S \leftarrow A$
2: **while** there is $b \in S \setminus \{a\}$ with $(S \setminus \{b\})_F^{\oplus} = A$ **do**
3:     Let b be as above s.t. $|(S \setminus \{b\})_F^{\circledast}|$ is maximal
4:     $S \leftarrow S \setminus \{b\}$
5: **return** $|S_F^{\circledast}|$

Proposition 7. *Let k be the output of the* SFR *algorithm on input $F = (A, R)$ and $a \in A$. Let furthermore $k^\dagger$ be the optimal solution to* MAXSTABLE* *on F and a. Then*

1. *there is a set S with $a \in S$, $S_F^{\oplus} = A$, and $k = |S_F^{\circledast}|$,*
2. *$k \geq k^\dagger/|R|$, and*
3. *k has been determined by* SFR *in polynomial time.*

Also for the SFR algorithm, we believe that the bound in item 2 above can be improved to $k^\dagger/O(\log |R|)$.

7 Experiments

We now report on a small feasibility study that analyses the *average* approximation ratio of the algorithms from above on existing data sets.

7.1 Experimental setup

We implemented[8] the algorithms from Sections 6.1–6.4 in Java, building on existing implementations for AAFs in *TweetyProject*[9]. We also implemented algorithms OPT and OPT* that *optimally* solve the problems MAXSTABLE and MAXSTABLE*, respectively. The algorithms for OPT and OPT* rely on straightforward MaxSAT encodings of MAXSTABLE and MAXSTABLE* and were implemented in Java as well, using the MaxSAT solver open-wbo 2.1[10] as backend solver. Since the main aim of the feasibility study here is to analyse the approximation quality of our heuristic approaches, we did not optimise the implementations for runtime performance (which also explains why we did not use a more performant programming language) and will not go into further details on the exact design of these algorithms.

We used the ICCMA17 and the ICCMA19 datasets as benchmarks.[11] Note that the benchmarks of ICCMA17 are clustered in groups and group B contains the benchmarks for problems related to stable semantics, so we only used that group, indicated by ICCMA17-B in the following. We also did some preliminary experiments with the datasets from ICCMA21 and ICCMA23, but since the optimal approaches timed out quite often on these datasets, an analysis of the approximation ratio was not feasible.

For each AAF in the above datasets, we selected 10 arguments[12] at random and asked all algorithms to determine the solution of MAXSTABLE, resp. MAXSTABLE* with respect to these query arguments. In the following, a pair of AAF and query argument will be referred to as *instance*. The experiments were conducted on a server with Intel Xeon E5-2643 v3 3.40-GHz CPUs (each algorithm is single-threaded) with 192GB RAM and we set a timeout of 10 minutes per instance.

7.2 Results

Table 1 summarises the results of our feasibility study on the ICCMA17-B (containing 3492 instances) and ICCMA19 (containing 3224 instances) dataset. The column "#TO" lists the number of timeouts of the corresponding approach, the column "RT" gives the total runtime (in seconds) on all instances that did not result in a timeout, and the column "AR" gives the average approximation ratio. The numbers in parentheses in the columns "RT" and "AR" are the number of instances, where these metrics are based upon. In particular, note that the average approximation ratio of a heuristic approach is only over those instances were both the heuristic approach and the optimal approach did not time out.

[8]http://tweetyproject.org/r/?r=ijcai24_opt_mt

[9]http://tweetyproject.org

[10]https://github.com/sat-group/open-wbo

[11]http://argumentationcompetition.org

[12]Both datasets contained AAFs with less than 10 arguments, for those we selected all arguments.

Approach	#TO	RT	AR
ICCMA17-B (3492)			
OPT	719	84801.7 (2773)	1.000 (2773)
GCF	47	4960.8 (3445)	0.964 (2736)
SCF	70	26119.4 (3422)	0.906 (2720)
OPT*	1372	75558.0 (2120)	1.000 (2120)
GFR	55	5620.8 (3437)	0.925 (2081)
SFR	70	72483.9 (3422)	0.984 (2075)
ICCMA19 (3224)			
OPT	4	8271.7 (3220)	1.000 (3220)
GCF	0	54.9 (3224)	0.980 (3220)
SCF	0	3734.4 (3224)	0.889 (3220)
OPT*	9	19808.3 (3215)	1.000 (3215)
GFR	0	229.8 (3224)	0.961 (3215)
SFR	0	2390.4 (3224)	0.989 (3215)

Table 1: Results on the ICCMA17-B and ICCMA19 datasets.

We can notice that the average approximation ratios of our heuristic approaches are surprisingly large, given the quite negative theoretical observations in the previous section. In fact, all heuristic approaches come close to the optimal solution with an average approximation ratio of 0.889 to 0.989. For the MAXSTABLE problem, we see that the GCF approach outperforms the SCF approach both in terms of approximation ratio and runtime, the latter showing a quite significant difference. For the MAXSTABLE* problem, the situation is a bit different. Here, the SFR approach outperforms the GFR approach in terms of approximation ratio, but not in terms of runtime. It is also quite clear, that all heuristic approaches outperform their corresponding optimal approaches in terms of timeouts and runtimes. However, since optimising runtime performance is not the goal of our implementations and this feasibility study, we will not go into a deeper analysis of the runtime behaviour. However, the results in Table 1 show, despite the negative worst-case behaviour of approximation algorithms highlighted in Theorems 6 and 7, that heuristic approaches for MAXSTABLE and MAXSTABLE* give promising results on average for existing datasets.

8 Summary and future work

We introduced two soft notions of stable semantics, analysed their general properties, and investigated the computational complexity of optimisation and approximation. We also presented some polynomial-time approximation algorithms for the corresponding optimisation problems and analysed their empirical approximation quality. While we showed that the theoretical approximatibility is rather bad, our algorithms performed rather well on existing datasets.

For future work, we will investigate soft notions and the corresponding optimisation problems for other classical semantical notions such as *admissibility* and the resulting classical *complete* and *preferred* semantics. We will also consider skeptical variants of the optimisation problems MAXSTABLE and MAXSTABLE* and investigate their complexity. Finally, we will apply the developed soft notions to the problems of *ranking* [Skiba *et al.*, 2021; Amgoud and Ben-Naim, 2013; Bonzon *et al.*, 2016] and *enforcement* [Baumann *et al.*, 2021].

References

[Amgoud and Ben-Naim, 2013] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In Weiru Liu, V S Subrahmanian, and Jef Wijsen, editors, *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM 2013)*, volume 8078 of *Lecture Notes In Artificial Intelligence*, Washington, DC, USA, September 2013.

[Baroni *et al.*, 2018] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract argumentation frameworks and their semantics. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, pages 159–236. College Publications, 2018.

[Baumann *et al.*, 2021] Ringo Baumann, Sylvie Doutre, Jean-Guy Mailly, and Johannes P. Wallner. Enforcement in formal argumentation. In Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors, *Handbook of Formal Argumentation*, volume 2, chapter 8. College Publications, August 2021.

[Baumann *et al.*, 2022] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Shedding new light on the foundations of abstract argumentation: Modularization and weak admissibility. *Artificial Intelligence*, 310:103742, 2022.

[Bazgan *et al.*, 2005] Cristina Bazgan, Bruno Escoffier, and Vangelis Th. Paschos. Completeness in standard and differential approximation classes: Poly-(D)APX- and (D)PTAS-completeness. *Theoretical Computer Science*, 339(2-3):272–292, 2005.

[Bonzon *et al.*, 2016] Elise Bonzon, Jerome Delobelle, Sebastien Konieczny, and Nicolas Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 2016.

[Caminada, 2006] Martin Caminada. Semi-stable semantics. In *The First Conference on Computational Models of Argument (COMMA'06)*, pages 121–130, 2006.

[Craandijk and Bex, 2020] Dennis Craandijk and Floris Bex. Deep learning for abstract argumentation semantics. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1667–1673. ijcai.org, 2020.

[Crescenzi, 1997] Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 262–273. IEEE Computer Society, 1997.

[Delobelle *et al.*, 2023] Jérôme Delobelle, Jean-Guy Mailly, and Julien Rossit. Revisiting approximate reasoning based on grounded semantics. In Zied Bouraoui and Srdjan Vesic, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 17th European Conference, ECSQARU 2023, Arras, France, September 19-22, 2023, Proceedings*, volume 14294 of *Lecture Notes in Computer Science*, pages 71–83. Springer, 2023.

[Dimopoulos and Torres, 1996] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1–2):209–244, 1996.

[Dung, 1995] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[Dvořák and Dunne, 2018] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 14. College Publications, February 2018.

[Kuhlmann and Thimm, 2019] Isabelle Kuhlmann and Matthias Thimm. Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: A feasibility study. In Nahla Ben Amor, Benjamin Quost, and Martin Theobald, editors, *Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'19)*, volume 11940, pages 24–37. Springer International Publishing, December 2019.

[Lund and Yannakakis, 1994] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[Malmqvist *et al.*, 2020] Lars Malmqvist, Tommy Yuan, Peter Nightingale, and Suresh Manandhar. Determining the acceptability of abstract arguments with graph convolutional networks. In Sarah Alice Gaggl, Matthias Thimm, and Mauro Vallati, editors, *Proceedings of the Third International Workshop on Systems and Algorithms for Formal Argumentation co-located with the 8th International Conference on Computational Models of Argument (COMMA 2020), September 8, 2020*, volume 2672 of *CEUR Workshop Proceedings*, pages 47–56. CEUR-WS.org, 2020.

[Malmqvist, 2022] Lars Malmqvist. *Approximate Solutions to Abstract Argumentation Problems Using Graph Neural Networks*. PhD thesis, University of York, 2022.

[Skiba *et al.*, 2021] Kenneth Skiba, Tjitze Rienstra, Matthias Thimm, Jesse Heyninck, and Gabriele Kern-Isberner. Ranking extensions in abstract argumentation. In Zhi-Hua Zhou, editor, *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, pages 2047–2053, August 2021.

[Thimm, 2021] Matthias Thimm. Harper++: Using grounded semantics for approximate reasoning in abstract argumentation. In *The Fourth International Competition on Computational Models of Argumentation (ICCMA'21)*, May 2021.

[Vazirani, 2003] Vijay V. Vazirani. *Approximation Algorithms*. Springer, Berlin, 2003.

[Verheij, 1996] Bart Verheij. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In J.-J.Ch. Meyer and L.C. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, 1996.