Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Reasoning over Attack-incomplete AAFs in the Presence of Correlations

Bettina Fazzinga[1] , **Sergio Flesca**[2] , **Filippo Furfaro**[2]

[1]ICAR-CNR, DICES-UNICAL
[2]DIMES-UNICAL
fazzinga@icar.cnr.it, {furfaro, flesca}@dimes.unical.it

Abstract

Attack-Incomplete Abstract Argumentation Frameworks (att-iAAFs) are a popular extension of AAFs where attacks are marked as uncertain when they are not unanimously perceived by different agents reasoning on the same arguments. We here extend att-iAAFs with the possibility of specifying correlations involving the uncertain attacks. This feature supports a unified and more precise representation of the different scenarios for the argumentation, where, for instance, it can be stated that an attack α has to be considered only if an attack β is considered, or that α and β are alternative, and so on. In order to provide a user-friendly language for specifying the correlations, we allow the argumentation analyst to express them in terms of a set of elementary dependencies, using common logical operators (namely, OR, NAND, CHOICE, $\Rightarrow$). In this context, we focus on the problem of verifying extensions under the possible perspective, and study the sensitivity of its computational complexity to the forms of correlations expressed and the semantics of the extensions.

1 Introduction

The need of suitably modeling the uncertainty characterizing the real world has led to the introduction of several variants of Dung's *Abstract Argumentation Frameworks* (AAFs) (Dung 1995). In particular, *attack-incomplete Abstract Argumentation Frameworks* (att-iAAFs) have proved effective in enabling a "qualitative" representation of the uncertainty involving the attacks between the arguments occurring in disputes. Basically, an att-iAAF is an AAF where the set of attacks is partitioned into the sets of *certain* and *uncertain* attacks: an attack is marked as *certain* if its presence in the argumentation is guaranteed, and *uncertain* otherwise. As observed in (Baumeister et al. 2018; Coste-Marquis et al. 2007; Cayrol, Devred, and Lagasquie-Schiex 2007), resorting to att-iAAFs is natural in the case where the argumentation involves several agents having different subjective views. Here, different opinions on the meaning of the various arguments or on the trustworthiness of who claimed the arguments may yield to different opinions on which attacks make sense. Thus, when the analyst examines the "global" view of the argumentation, they find it natural to model the attacks that are not unanimously perceived as "uncertain". Analogously, when several agents may contribute to a dispute by claiming arguments and by elaborating on why some arguments (not necessarily claimed by them) attack other arguments, it is natural to consider as "uncertain" the attacks motivated by agents whose participation is not guaranteed.

It is easy to see that an att-iAAF compactly encodes a set of alternative scenarios for the argumentation: each scenario is called "completion" and is an AAF containing all the arguments and the certain attacks of the att-iAAF, plus a subset of its uncertain attacks. In order to take into account the fact that, differently from "traditional" AAFs, multiple scenarios are possible, the traditional notion of extension for an AAF has been re-formulated in terms of *i^*-extension* (Fazzinga, Flesca, and Furfaro 2020b): *A set of arguments S is a possible (resp., necessary) i^*-extension of the att-iAAF IF if, for some (resp., every) completion F of IF, the set S is an extension of F.*

Example 1. *Consider the att-iAAF IF over the set of arguments $A = \{a, b, c\}$ and the set of attacks $D = \{(a, b), (b, a), (c, b)\}$ depicted in Fig. 1 (disregard the dotted edges for now). All the attacks are uncertain, thus IF has the following 8 completions, denoted as pairs $\langle$ arguments, attacks $\rangle$:*
$F_1 = \langle A, \emptyset\rangle$;
$F_2 = \langle A, \{(a, b)\}\rangle$;
$F_3 = \langle A, \{(b, a)\}\rangle$;
$F_4 = \langle A, \{(c, b)\}\rangle$;
$F_5 = \langle A, \{(a, b), (b, a)\}\rangle$;
$F_6 = \langle A, \{(b, a), (c, b)\}\rangle$;
$F_7 = \langle A, \{(a, b), (c, b)\}\rangle$;
$F_8 = \langle A, \{(a, b), (b, a), (c, b)\}\rangle$.
It is easy to see that every subset of A is a possible i^*-extension under the admissible semantics, since it is an admissible extension in F_1. The only necessary i^*-extensions (under the admissible semantics) are $\emptyset$ and $\{c\}$.

A limit of the representation paradigm of att-iAAFs is that it does not take into account possible correlations between the uncertain attacks. For instance, the argumentation analyst cannot specify that, if an attack α is considered when reasoning on the acceptability of (sets of) arguments, then also the attack β should be considered (since the motivations that lead an agent to believe that α holds imply that also β holds), or that α and β cannot co-exist in any realistic scenario (since the motivations that lead an agent to believe that α or β hold are antithetical).

In fact, the presence of such dependencies between the

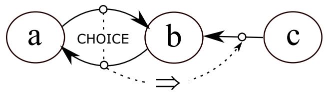

Figure 1: ⇒- and CHOICE- labeled edges/lines represent dependencies, the edges marked with a white circle are uncertain attacks

attacks can imply that some completions encoded by the att-iAAF represent scenarios that cannot actually occur, and this may deeply affect the reasoning process, as shown in Example 2.

Example 2. *Consider the att-iAAF of Example 1 and assume that it has been defined for modeling a television debate between two parties: the anarchists and the authorities. Assume that the arguments b and c are claimed by two distinct representatives of the authorities (Bob and Charlotte, respectively), while a is claimed by Anthony, a representative of the anarchists.*

The analyst who defined the att-iAAF wants to reason on the debate by considering the possible subjective views of the audience. Specifically, the analyst knows that Charlotte is an authoritative and popular personality, so that the capability of her argument c to attack b can be perceived by every type of audience, independently from the fact that they are supporters of anarchists or authorities. Moreover, the analyst knows that if the audience perceives the attack (b, a) as valid, they will also agree on the attack (c, b), since the reasons of (b, a) imply those of (c, b). Finally, since the analyst wants to focus on two "antithetic" and polarized types of audience (that is, the supporters of anarchists and those of authorities), the attacks (a, b) and (b, a) are mutually exclusive, as the former will be present in the subjective view of the supporters of the anarchists, and the latter in the authorities' view. Overall, this means that the anarchists's supporters will consider the attack (a, b) and possibly the attack (c, b), while the authorities' supporters will consider the attacks (b, a) and (c, b) in their view.

The above-described correlations between the attacks of the att-iAAF can be formally written as $\text{CHOICE}\big((a, b), (b, a)\big)$ *and* $(b, a) \Rightarrow (c, b)$*, and can be easily represented in a graphical form: in Figure 1, they are represented as dotted edges over the argumentation graph, labeled with the logical connective. Taking into account the dependencies makes the att-iAAF encode the only three possible scenarios that can occur, according to the system of belief of the audience. In fact, considering the dependencies, many of the completions listed in Example 1 encode scenarios that do not correspond to any subjective view. The only "valid" completions are F_2, F_6 and F_7. In the light of this, under the admissible semantics, the sets $\{a, b\}$ and $\{b, c\}$ are no more possible i*-extensions, and the sets $\{a\}$ and $\{a, c\}$ become necessary i*-extensions.*

As shown in Example 2, the fact that att-iAAFs have no mechanism for expressing correlations can cause an imprecise representation of the possible scenarios for the argumentation, and this can yield wrong assessments. For instance, under the possible perspective, it may happen that the completions witnessing that a set S is an i*-extension are not scenarios that can occur in practice (i.e. scenarios compatible with some of the subjective views summarized in the att-iAAF). Hence, disregarding the correlations may lead the analyst to wrongly consider S as a set of arguments that is "robust" according to at least one way of perceiving the attacks (as happens in examples 1 and 2 for $\{a, b\}$ and $\{b, c\}$).

An analogous issue was the motivation of the framework in our previous work (Fazzinga, Flesca, and Furfaro 2021), where a language for specifying correlations between arguments in *argument incomplete AAFs* was introduced. In fact, this work can be viewed as a natural continuation of (Fazzinga, Flesca, and Furfaro 2021): we here introduce "*att-iAAFs with dependencies*" (d-att-iAAFs), where the same language for expressing correlations exploited in *argument incomplete AAFs* is now used to specify dependencies between uncertain attacks. This language offers a set of classical n-ary logical connectives (namely, OR, NAND, CHOICE, ⇒) that allow "elementary" dependencies over sets of attacks to be expressed in a user-friendly manner.

Given this, we first extend the notion of i*-extension to the case of d-att-iAAFs (so that, when deciding whether a set is an extension or not, only the completions consistent with the dependencies are included). Then, we characterize the complexity of the verification problem for *possible* i*-extensions for a d-att-iAAF under Dung's semantics (admissible, stable, complete, grounded and preferred). We perform a thorough investigation, where we study the sensitivity of the computational complexity to the forms of dependency used to specify the correlations and to the semantics of extensions. Interestingly, we show that, for some combinations ⟨dependency, semantics⟩, the complexity is P (the same as the verification problem over att-iAAFs in the absence of correlations), while for others the complexity moves to NP-complete. In order to give an insight on the sources of complexity, we even study the impact of the arity of the logical connectives on the complexity, and show that for some connectives (i.e., NAND, CHOICE) the verification is hard even for the lowest arity, while for ⇒ the verification problem becomes solvable in polynomial time if we suitably limit the number of operands. Table 1 reports a synopsis of the results proved in the paper.

2 Preliminaries

An *abstract argumentation framework* (*AAF*) is a pair $\langle A, D\rangle$, where A is a finite set, whose elements are called *arguments*, and $D \subseteq A \times A$ is a binary relation over A, whose elements are called *attacks*. Given a set of arguments S and an argument a, we say that "*S attacks a*" if there is an argument b in S such that b attacks a, and that "*a attacks S*" if there is an argument $b \in S$ such that a attacks b. Moreover, we say that *a is acceptable w.r.t. S* if every argument attacking a is attacked by S, and say that S is *conflict-free* if there is no attack between its arguments.

Several semantics for AAFs have been proposed to identify "reasonable" sets of arguments, called *extensions* (Dung 1995). A set $S \subseteq A$ is: an *admissible extension* (ad) iff S is conflict-free and all its arguments are acceptable w.r.t. S;

a *stable extension* (st) iff S is conflict-free and S defeats each argument in $A \setminus S$; a *complete extension* (co) iff S is admissible and S contains all the arguments that are acceptable w.r.t. S; a *grounded extension* (gr) iff S is a minimal (w.r.t. $\subseteq$) complete set of arguments; a *preferred extension* (pr) iff S is a maximal (w.r.t. $\subseteq$) complete set of arguments.

We recall the notion of *attack-incomplete Abstract Argumentation Framework* (att-iAAF) (Baumeister, Rothe, and Schadrack 2015).

Definition 1 (att-iAAF)**.** *An* attack-incomplete Abstract Argumentation Framework *is a tuple $\langle A, D, D^? \rangle$, where A is a set of arguments and D and $D^?$ are disjoint sets of attacks between arguments in A. The attacks in D (resp., $D^?$) are said to be* certain *(resp.,* uncertain*), i.e. they are guaranteed (resp., not guaranteed) to occur in the argumentation.*

An att-iAAF compactly represents the alternative scenarios for the argumentation, i.e. all the possible combinations of arguments and attacks that can occur according to what is certain and uncertain. Each scenario is called *completion*.

Definition 2 (Completion)**.** *Given an att-iAAF $IF = \langle A, D, D^? \rangle$, a completion for IF is an AAF $F = \langle A, D' \rangle$ where $D \subseteq D' \subseteq (D \cup D^?)$.*

In (Fazzinga, Flesca, and Furfaro 2020b), *i*-extensions* were introduced to adapt the notion of extension to the case of att-iAAFs. Specifically, since an att-iAAF encodes several alternative scenarios, i*-extensions were defined under both the possible and the necessary perspective, where the condition of being extension is required to be true in *at least one* and *every* scenario, respectively. Example 1 contains examples of possible and necessary i*-extensions over att-iAAFs.

Definition 3 (i*-extension)**.** *Given an att-iAAF IF and a semantics σ, a set S is a possible (resp., necessary) i*-extension for IF (under σ) if, for at least one (resp., for every) completion F of IF, the set S is an extension of F under σ.*

3 Augmenting att-iAAFs with Correlations

The original paradigm of att-iAAFs assumes that the attacks are independent from one another: the occurrence of an attack does not have any effect on the occurrence of other attacks. As we have shown in Example 2, the independence assumption may not be valid in general, since it can happen that some correlation is known to exist between uncertain attacks, and this has important consequences.

Indeed, introducing dependencies may have the effect of discarding some completions, as they turn out to describe scenarios that cannot occur. This has a strong impact on the reasoning. In fact, under the possible perspective, a set that is an i*-extension when dependencies are not considered may be no longer an i*-extension when dependencies are taken into account. For instance, Example 2 shows that this happens with $\{a, b\}$, since the completion F_1, that is the only completion witnessing that $\{a, b\}$ is an admissible extension, turns out to be an impossible scenario when the CHOICE-dependency is taken into account. Under the necessary perspective, a set that, with no dependency, is not an i*-extension may become an i*-extension when the dependencies are considered. In Example 2 the sets $\{a\}$ and $\{a, c\}$ become necessary admissible i*-extensions as the only completions representing possible scenarios are F_2, F_6 and F_7.

In order to encode the correlations among attacks, we resort to the language introduced in (Fazzinga, Flesca, and Furfaro 2021) in the analogous scenario where the correlations involve the arguments in argument-incomplete AAFs. We use this language since, as we will see later on, it is expressive enough to allow every form of correlation to be specified and, moreover, it is easy and intuitive. The language offers a set of n-ary logical connectives, so that correlations are expressed in terms of dependencies of the following form.

Definition 4 (Dependency)**.** *A* dependency *δ over an att-iAAF $IF = \langle A, D, D^? \rangle$ is an expression $X \Rightarrow Y$ ($\Rightarrow$-dependency) or* OP(X) *(*OP*-dependency), where* OP $\in$ {OR, NAND, CHOICE} *and X, Y are non-empty subsets of $D^?$.*

Specifying dependencies makes some scenarios become invalid. We now define the "valid" completions, that are those that satisfy the dependencies.

Definition 5 (Valid completion)**.** *A completion $F = \langle A, D' \rangle$* is valid w.r.t *a dependency δ (written $F \models \delta$) iff*
- *δ is* OR(X) *and $X \cap D' \neq \emptyset$,*
- *δ is* NAND(X) *and $X \cap D' \subset X$,*
- *δ is* CHOICE(X) *and $|X \cap D'| = 1$,*
- *δ is $X \Rightarrow Y$ and, if $X \subseteq D'$, then $Y \cap D' \neq \emptyset$.*

F is valid *w.r.t. a set of dependencies Δ if $\forall \delta \in \Delta$ $F \models \delta$.*

Thus, an OR- (resp., CHOICE-) dependency imposes that at least (resp., exactly) one of the specified attacks is in the completion; a NAND-dependency imposes that the specified attacks cannot occur all together; an implication $\Rightarrow$ means that if a completion contains all the attacks on the left-hand side, then it must contain at least one of the attacks of the right-hand side. An implication whose right-hand side consists of only one attack is called *disjunction free*. The attacks on the left-had side of the implication are called *implicants*. We did not consider AND and NOR as here they make no sense: an AND- (resp., NOR-) dependency requires that each (resp., none) of the specified attacks is in the completion, but this can be done by putting these attacks in D (resp., removing these attacks from $D^?$).

Definition 6 (d-att-iAAF)**.** *An attack-incomplete Abstract Argumentation Framework with dependencies (d-att-iAAF) is a pair $DIF = \langle IF, \Delta \rangle$, where IF is an att-iAAF and Δ a set of dependencies over IF.*

Note that the completions of a d-att-iAAF $DIF = \langle IF, \Delta \rangle$ are the completions of IF, and the valid completions of DIF are the completions of DIF valid w.r.t. Δ. We now adapt the notion of i*-extension to d-att-iAAFs considering only valid completions in the reasoning.

Definition 7 (i*-extensions over d-att-iAAFs)**.** *Given a d-att-iAAF DIF and a semantics σ, a set of arguments S is a possible (resp., necessary) i*-extension for DIF (under σ)*

if, for at least one (resp., for every) valid completion F of DIF, the set S is an extension of F under σ.

Once the notions of dependency, valid completion, and i*-extensions for d-att-iAAFs have been introduced, we can formalize and summarize the observations made at the beginning of this section in Proposition 1, which states that, as the set of dependencies grows, the set of possible i*-extensions gets smaller and the set of necessary i*-extensions gets larger:

Proposition 1. *Let $DIF = \langle IF, \Delta\rangle$ and $DIF' = \langle IF, \Delta'\rangle$ be two d-att-iAAFs, where $\Delta \subseteq \Delta'$. Let σ be a semantics in $\{ad, st, co, gr, pr\}$, and let* PExt$(\cdot)$ *and* NExt$(\cdot)$ *return the sets of possible and necessary i*-extensions of a d-att-iAAF under σ, respectively. It holds that:*

1. PExt$(DIF) \supseteq$ PExt(DIF');
2. NExt$(DIF) \subseteq$ NExt(DIF').[1]

The introduction of d-att-iAAFs gives a great benefit to the analysts: augmenting att-iAAFs with the possibility of defining correlations among uncertain attacks enables analysts to reason more precisely over att-iAAFs. As a matter of fact, the dependencies introduced in Definition 4 are an easy and compact tool for expressing elementary correlations, that do not require to write complicate formulas. In this regard, we point out that using a set of dependencies is a general mechanism to specify any correlation expressible by means of a propositional formula whose variables represent the presence/absence of uncertain attacks. In fact, any propositional formula over a set of variables $\{x_1, \dots, x_n\}$ representing the presence/absence of attacks can be translated into a set of dependencies reasoning as follows. Let $\Phi = C^0 \wedge \cdots \wedge C^k$ be a propositional formula in CNF over $\{x_0, \dots, x_n\}$. Every clause C^i can be translated into a single dependency, reasoning by cases on the form of C^i:

- $C^i = x^i_1 \vee \cdots \vee x^i_h$ (i.e. C^i contains only positive literals): C^i is equivalent to $\textsc{or}(x^i_1, \cdots, x^i_h)$,
- $C^i = \neg x^i_1 \vee \cdots \vee \neg x^i_h$ (i.e. C^i contains only negative literals): C^i is equivalent to $\textsc{nand}(x^i_1, \dots, x^i_h)$,
- $C^i = \neg x^i_1 \vee \cdots \vee \neg x^i_m \vee x^i_{m+1} \vee \cdots \vee x^i_h$ (i.e. C^i contains both positive and negative literals): C^i is equivalent to $x^i_1, \dots, x^i_m \Rightarrow x^i_{m+1}, \dots, x^i_h$.

Moreover, in practical cases, encoding the correlations into Δ allows for distinguishing the various forms of correlations imposed by the analyst, which could be otherwise "hidden" if a general propositional formula were used. This allows us to provide the contribution presented in the next section: a fine-grained analysis of the impact of correlations on the computational complexity of the fundamental reasoning problem over att-iAAFs. This contribution is still of interest if a propositional formula is used to encode correlations instead of the set of dependencies: our study can be viewed as a sensitivity analysis of the complexity of the reasoning tasks to some syntactic restrictions of practical interest.

[1]Note that this holds also in the case where no valid completions exists, due to the definition of necessary extension. If one changes the definition of necessary i*-extensions adding the requirement that they are extensions in at least one completion, then the second statement does not hold anymore.

4 The Verification Problem over d-att-iAAFs and its Computational Complexity

Starting from the definition of i*-extension for d-att-iAAFs (Definition 7), it is natural to adapt the classical verification problem to d-att-iAAFs as follows:

Definition 8 (PDVER$^\sigma(DIF, S)$)**.** *Let DIF be a d-att-iAAF, S a set of arguments, and σ a semantics for extensions.* PDVER$^\sigma(DIF, S)$ *(Possible-perspective Dependency-aware* VER*ification) is the problem of verifying if S is a possible i*-extension for DIF under σ.*

Observe that we focus on the possible perspective, and defer the study of the verification of necessary i*-extensions to future work. In what follows, any instance of PDVER$^\sigma(DIF, S)$ will be denoted as $\langle DIF, S\rangle$ where $DIF = \langle IF, \Delta\rangle$ is a d-att-iAAF and S a set of arguments (thus, this notation considers the semantics σ implied by the context).

In the rest of this section, we provide the main contribution of this work: a thorough study of the computational complexity of PDVER$^\sigma(DIF, S)$, where we investigate its sensitivity to the form of dependencies appearing in the d-att-iAAF and to the semantics σ. In order to obtain fine-grain insights on the sources of complexity, our analysis will consider some restrictions on the arity of the logical connectives occurring in the dependencies of Definition 4. In order to refer to these restrictions, we use the following notations: given $\textsc{op} \in \{\textsc{or}, \textsc{nand}, \textsc{choice}\}$, we denote as $\textsc{op}^x$-dependency an $\textsc{op}$-dependency where $\textsc{op}$ is applied over a set X such that $|X| = x$; moreover, we denote as ${}_m{\Rightarrow}_n$-dependency an $\Rightarrow$-dependency of the form $X \Rightarrow Y$ where $|X| = m$ and $|Y| = n$.

Our results are summarized in Table 1. Here, EMPTY means $\Delta = \emptyset$, and the corresponding row is a result from (Baumeister et al. 2018), where the verification problem over att-iAAFs (with no dependencies) was shown to be in P for $\sigma \in \{\texttt{ad}, \texttt{st}, \texttt{co}, \texttt{gr}\}$ and Σ^p_2-complete for $\sigma = \texttt{pr}$. ANY OTHER stands for "*any combination of* 2 *or more forms of dependencies different from the combinations in the other rows*". Observe that the combination $\textsc{or}{+}_1{\Rightarrow}_n$ is not included in ANY OTHER, but is in a distinguished row, as its complexity is different from the other combinations.

4.1 Upper Bounds

We start the presentation of our results by stating an upper bound for the computational complexity of PDVER$^\sigma(DIF, S)$.

Theorem 1 (Upper bound)**.** PDVER$^\sigma(DIF, S)$ *is in NP for any $\sigma \in \{ad, st, co, gr\}$ and in Σ^p_2 for $\sigma = pr$.*

Proof. An instance $\langle\langle IF, \Delta\rangle, S\rangle$ of PDVER$^\sigma(DIF, S)$ can be solved by guessing a completion F of IF and then verifying if $F \models \Delta$ and if S is a σ-extension over F. Then, the statement follows from the fact that checking $F \models \Delta$ is in P, and that verifying extensions is in P for any $\sigma \in \{\texttt{ad}, \texttt{st}, \texttt{co}, \texttt{gr}\}$ and in *coNP* for $\sigma = \texttt{pr}$. □

		PDVER$^\sigma(DIF, S)$			
		ad, st	co	gr	pr
1)	EMPTY	P	P	P	Σ_2^p-c
2)	OR (even OR^2)	P	NP-c	NP-c	Σ_2^p-c
3)	NAND (even NAND^2)	NP-c	NP-c	NP-c	Σ_2^p-c
4)	${}_m{\Rightarrow}_n$ (any $m \geq 2, n \geq 1$)	NP-c	NP-c	NP-c	Σ_2^p-c
5)	${}_1{\Rightarrow}_n$ (any $n \geq 1$)	P	NP-c	NP-c	Σ_2^p-c
6)	CHOICE (even CHOICE^2)	NP-c	NP-c	NP-c	Σ_2^p-c
7)	OR + ${}_1{\Rightarrow}_n$ (even OR^2, $n \geq 1$)	P	NP-c	NP-c	Σ_2^p-c
8)	ANY OTHER	NP-c	NP-c	NP-c	Σ_2^p-c

Table 1: Complexity of PDVER$^\sigma(DIF, S)$ over att-iAAFs

4.2 Lower Bounds: Intractability Results

We now consider lower bounds, and we first address the case of the preferred semantics. Here, the Σ_2^p-hardness trivially follows from the above mentioned result in (Baumeister et al. 2018), where the verification problem was shown to be Σ_2^p-hard over att-iAAFs, that is for d-att-iAAFs with $\Delta = \emptyset$.

Proposition 2. *Under $\sigma = pr$,* PDVER$^\sigma(DIF, S)$ *is Σ_2^p-hard, even if $\Delta = \emptyset$.*

We now focus on the other semantics and show the combinations $\langle$ *form of dependency, semantics* $\rangle$ for which the NP upper bound stated in Theorem 1 is also a lower bound. All the proofs in the rest of this section are based on reductions from 3SAT. Hence, for the sake of brevity, in each proof we avoid repeating that we show a reduction from 3SAT, and we start the proof as if we already said that we are given a 3CNF formula Φ over the variables $x_1, \ldots, x_n$ and the clauses $C_1, \ldots, C_m$, where each C_j is of the form $C_j = l_1^j \vee l_2^j \vee l_3^j$, and each l_k^j is a literal of the form x_i or $\neg x_i$.

We start with Theorems 2 and 3, presenting the cases where the hardness for NP holds even under the admissible and stable semantics. That is, we show that considering only $\Rightarrow$- or only NAND- or only CHOICE- dependencies suffices to make PDVER$^\sigma(DIF, S)$ hard, even under $\sigma \in \{\texttt{ad}, \texttt{st}\}$ and even if we impose the strongest limit to the arity of the connectives NAND and CHOICE, and we limit $\Rightarrow$ to ${}_2{\Rightarrow}_1$ (i.e. two implicants and no disjunction in the head).

Theorem 2. *Under any $\sigma \in \{\texttt{ad}, \texttt{st}, \texttt{co}, \texttt{gr}\}$, if Δ contains only $\Rightarrow$-dependencies, then* PDVER$^\sigma(DIF, S)$ *is NP-hard, even if the $\Rightarrow$-dependencies are disjunction-free and have at least two implicants.*

Proof. We consider $\sigma = \texttt{ad}$ (it is easy to see that the same proof holds for the other semantics). Let $DIF(\Phi) = \left\langle \langle A, D, D^? \rangle, \Delta \right\rangle$ be the d-att-iAAF constructed as follows. A contains three arguments y, z, Φ, and, for each $i \in [1..n]$, the arguments x_i and $\neg x_i$, and, for each $j \in [1..m]$, the argument C_j. Then, D contains, for each $j \in [1..m]$, the attack (C_j, Φ). Moreover, $D^?$ contains the uncertain attack (y, z) and, for each $j \in [1..m]$, an uncertain attack towards C_j from every literal x_i or $\neg x_i$ occurring in C_j. Finally, for each pair of attacks $(x_i, C_{j'})$, $(\neg x_i, C_{j''})$ (i.e., attacks originating from opposite literals of the same variable) Δ contains $(x_i, C_{j'}), (\neg x_i, C_{j''}) \Rightarrow (y, z)$. An example of construction is in Figure 2. We prove the equivalence: "Φ *is satisfiable*" $\Leftrightarrow$ "$S = \{x_1, \neg x_1, \ldots, x_n, \neg x_n, y, z, \Phi\}$ *is an admissible i*-extension of $DIF(\Phi)$*".

$\Rightarrow$: Let t be a truth assignment for $x_1, \ldots, x_n$ making Φ evaluate to true, and $F = \langle A, D \cup D_F \rangle$ the completion of $DIF(\Phi)$ where D_F is the subset of $D^?$ containing, for each x_i such that $t(x_i) = $ *true* (resp., *false*), the attack (x_i, C_j) (resp., $(\neg x_i, C_j)$). It is easy to see that, since t assigns exactly one truth value to each variable, no implication in Δ is triggered: therefore, the presence of the attack (y, z) is not implied, and then F (which does not contain (y, z)) is a valid completion. Moreover, since t makes Φ satisfied, every argument C_j in F is attacked by at least one argument (of the form x_i or $\neg x_i$) in S, and this implies that Φ is defended by S from the attacks $(C_1, \Phi), \ldots, (C_k, \Phi)$. This implies, along with the fact that S is conflict-free, that S is an admissible extension of F.

$\Leftarrow$: Let F be a valid completion of $DIF(\Phi)$ such that S is an admissible extension of F. Since S is admissible, it means that F does not contain the attack (y, z) (otherwise, S would not be conflict-free in F). Hence, since F is a valid completion, no implication in Δ is triggered in F, and this means that, for every $i \in [1..n]$, F may contain attacks outgoing from either x_i or $\neg x_i$, but not from both. This entails that the three rows in the following definition of t are alternative cases:

$$t(x_i) = \begin{cases} \text{true} & \text{if } \exists j \in [1..m] \mid (x_i, C_j) \in D_F; \\ \text{false} & \text{if } \exists j \in [1..m] \mid (\neg x_i, C_j) \in D_F; \\ \text{false} & \text{otherwise} \end{cases}$$

Hence, t is a function from $\{x_1, \ldots, x_n\}$ to $\{$*true*, *false*$\}$, that is, a truth assignment over the variables $x_1, \ldots, x_n$. Observe that the third row in the definition of t means that, if

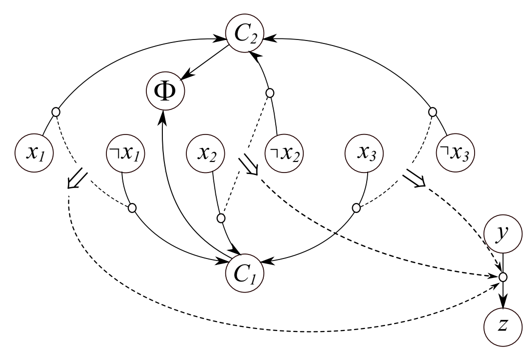

Figure 2: Construction of Theorem 2, for $\Phi = C_1 \wedge C_2$, where $C_1 = \neg x_1 \vee x_2 \vee x_3$ and $C_2 = x_1 \vee \neg x_2 \vee \neg x_3$ (we recall that uncertain attacks are marked with a white circle)

there is no attack in F outgoing from x_i or $\neg x_i$, we assign *false* to variable x_i: however, it is easy to see that the validity of the proof is independent from the value assigned to x_i in this case.

Now, since S is admissible in F, all the attacks towards Φ must be counter-attacked by S, and this means that every C_j is attacked in F by some x_i or some $\neg x_i$. By construction, the attacks (x_i, C_j) and $(\neg x_i, C_j)$ mean that $x_i = $ *true* and x_i= *false* makes C_j evaluate to true, respectively. This means that t makes every C_j evaluate to true, and then Φ is satisfiable. □

Theorem 3. *Under any $\sigma \in \{ad, st, co, gr\}$, if Δ contains only* NAND*-dependencies or only* CHOICE*-dependencies, then* PDVER$^\sigma(DIF, S)$ *is NP-hard, even if the dependencies are binary.*

Proof. The statement can be proved with minor changes to the strategy used in the case of ⇒-dependencies. In the construction of the proof of Theorem 2, arguments y and z must be removed from A, as well as the attack (y, z) from $D^?$. Moreover, in order to prove the case of NAND- (resp., CHOICE-) dependencies, every dependency $(x_i, C_{j'}), (\neg x_i, C_{j''}) \Rightarrow (y, z)$ must be replaced with NAND $\big((x_i, C_{j'}), (\neg x_i, C_{j''})\big)$ (resp., CHOICE $\big((x_i, C_{j'}), (\neg x_i, C_{j''})\big)$). Finally, the set S to be verified is $S = \{x_1, \neg x_1, \ldots, x_n, \neg x_n, \Phi\}$. □

As using implications with two implicants (even without exploiting the disjunction in the head) makes PDVER$^\sigma(DIF, S)$ *NP*-hard (as proved above), it is natural to ask what happens if we use only implications with one implicant. The following theorem gives a first answer to this question, as it states that PDVER$^\sigma(DIF, S)$ remains *NP*-hard with this restriction under $\sigma \in \{\texttt{co}, \texttt{gr}\}$. In the following, we will see that this restriction makes PDVER$^\sigma(DIF, S)$ solvable in polynomial time under $\sigma \in \{\texttt{ad}, \texttt{st}\}$.

Theorem 4. *Under any $\sigma \in \{co, gr\}$, if Δ contains only ⇒-dependencies, then* PDVER$^\sigma(DIF, S)$ *is NP-hard, even if the ⇒-dependencies are disjunction-free and have at most one implicant.*

Proof. We prove the case $\sigma = \texttt{co}$ (the same construction works for the case $\sigma = \texttt{gr}$). Let $DIF(\Phi) = \Big\langle \langle A, D, D^?\rangle, \Delta \Big\rangle$ be the d-att-iAAF constructed as follows. A contains, for each $j \in [1..m]$, the argument C_j, and, for each $i \in [1..n]$, the arguments x_i, $\neg x_i$, i, i', i''. Then, D contains, for each $i \in [1..n]$, the attacks (i', i), (i'', i) and the self-attacks (i', i'), (i'', i''). Moreover, $D^?$ contains, for each $i \in [1..n]$, the attacks (x_i, i'), $(\neg x_i, i'')$, and, for each $j \in [1..m]$, an attack towards C_j from every x_i and $\neg x_i$ occurring in C_j. As for Δ, it contains:

1) for each (x_i, C_j) in $D^?$, the dependency $(x_i, C_j) \Rightarrow (x_i, i')$;

2) for each $(\neg x_i, C_j)$ in $D^?$, the dependency $(\neg x_i, C_j) \Rightarrow (\neg x_i, i'')$.

We prove the equivalence: "Φ *is satisfiable*" ⇔ "$S = \{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$ *is a complete i*-extension of* $DIF(\Phi)$".

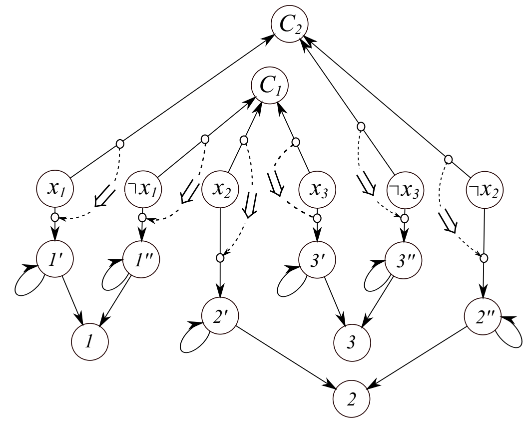

Figure 3: Construction of Theorem 4, for $\Phi = C_1 \wedge C_2$, where $C_1 = \neg x_1 \vee x_2 \vee x_3$ and $C_2 = x_1 \vee \neg x_2 \vee \neg x_3$

⇒: Let t be a truth assignment for $x_1, \ldots, x_n$ making Φ evaluate to true, and $F = \langle A, D \cup D_F\rangle$ the completion of $DIF(\Phi)$ where D_F consists of every attack outgoing from x_i (if $t(x_i) = $ *true*) or from $\neg x_i$ (if $t(x_i) = $ *false*). It is easy to see that this construction entails that $F \models \Delta$. Since t assigns exactly one truth value to each variable, we have that, for each $i \in [1..n]$, one of the attacks (x_i, i'), $(\neg x_i, i'')$ is not in D_F. This implies that no complete extension of F can contain any argument i, since one of the attacks (i', i), (i'', i) cannot be counter-attacked. Moreover, since t makes Φ evaluate to true, every C_j in F is attacked by at least one argument x_i or $\neg x_i$, and this implies that no C_j can belong to any complete extension of F. Considering this jointly with the facts that i' and i'' are self-attacked (for each $i \in [1..n]$) and that S is admissible (as F contains no attack towards S), we have that S is a complete extension of F.

⇐: Let F be a valid completion of $DIF(\Phi)$ such that S is a complete extension of F. This means that, for each $i \in [1..n]$, F does not contain at least one of the attacks (x_i, i') and $(\neg x_i, i'')$ (otherwise S would not be complete, as some argument i would be acceptable w.r.t. S). Due to the implications in Δ, this entails that, for each $i \in [1..n]$, F does not simultaneously contain attacks outgoing from x_i and attacks outgoing from $\neg x_i$. This implies that the three rows in the following definition of t are alternative cases:

$$t(x_i) = \begin{cases} \text{true} & \text{if } \exists j \in [1..m] \mid (x_i, C_j) \in D_F; \\ \text{false} & \text{if } \exists j \in [1..m] \mid (\neg x_i, C_j) \in D_F; \\ \text{false} & \text{otherwise} \end{cases}$$

and then t is a function from $\{x_1, \ldots, x_n\}$ to $\{$*true*,*false*$\}$. The third case in the definition of t means that, if there is no attack in F outgoing from x_i or $\neg x_i$, we assign *false* to variable x_i (however, it is easy to see that the validity of the proof is independent from the value assigned to x_i in this case). Since S is a complete extension of F, every C_j in F is attacked by some argument x_i or $\neg x_i$. In turn, since the existence of (x_i, C_j) (resp., $(\neg x_i, C_j)$) in $D^?$ means that assigning *true* to x_i (resp., *false* to x_i) makes C_j true, this implies that t makes Φ evaluate to true. □

To complete the picture of NP-hard cases, we now show that the use of OR-dependencies makes PDVER$^\sigma(DIF, S)$ hard under the complete and grounded semantics.

Theorem 5. *Under any* $\sigma \in \{co, gr\}$*, if* Δ *contains only* OR*-dependencies, then* PDVER$^\sigma(DIF, S)$ *is NP-hard, even if the dependencies are binary.*

Proof. We consider $\sigma = \texttt{co}$, but the same proof holds for $\sigma = \texttt{gr}$. Let $DIF(\Phi) = \Big\langle\langle A, D, D^?\rangle, \Delta\Big\rangle$ be the d-att-iAAF constructed as follows. A contains an argument s as well as, for each $i \in [1..n]$, the arguments x_i and $\neg x_i$, and, for each $j \in [1..m]$, the argument C_j. Then, D contains, for each $i \in [1..n]$, the self-attacks (x_i, x_i) and $(\neg x_i, \neg x_i)$, and, for each clause C_j containing x_i (resp., $\neg x_i$), the attack (x_i, C_j) (resp., $(\neg x_i, C_j)$). Moreover, $D^?$ contains, for each $i \in [1..n]$, the uncertain attacks (s, x_i) and $(s, \neg x_i)$. Basically, the **absence** of (s, x_i) and of $(s, \neg x_i)$ in a completion will be used to simulate the assignment of the values *true* and *false* to the variable x_i, respectively. Finally, Δ contains, for each $i \in [1..n]$, the OR^2-dependency OR$\Big((s, x_i), (s, \neg x_i)\Big)$. We prove the equivalence: "Φ *is satisfiable*" $\Leftrightarrow$ "$\{s\}$ *is a complete i***-extension of* $DIF(\Phi)$".
$\Rightarrow$: Let t be a truth assignment for $x_1, \ldots, x_n$ making Φ evaluate to true, and $F = \langle A_F, D_F\rangle$ the completion of $DIF(\Phi)$ containing, for each $i \in [1..n]$, the attack (s, x_i) (resp., $(s, \neg x_i)$) if and only if $t(x_i)$ = *false* (resp., $t(x_i)$ =*true*). It is straightforward to see that $F \models D$, since the fact that t is a truth assignment implies that exactly one between (s, x_i) and $(s, \neg x_i)$ is in D_F, for each $i \in [1..n]$. Moreover, since t makes Φ satisfied and we do not put in D_F any attack (s, x_i) (resp., $(s, \neg x_i)$) if x_i = *true* (resp., x_i = *false*) makes C_j true, we have that every C_j in F is attacked by at least one argument of the form x_i or $\neg x_i$ that is not attacked by s. Hence, no C_j can belong to an admissible extension of F. Considering this jointly with the fact that every argument of the form x_i or $\neg x_i$ is self-attacked, we obtain that the only $\subseteq$-maximal admissible extension of F is $\{s\}$.
$\Leftarrow$: Let $F = \langle A_F, D_F\rangle$ be a valid completion of $DIF(\Phi)$ admitting $\{s\}$ as a complete extension. Since $F \models \Delta$, there is no $i \in [1..n]$ such that both $(s, x_i) \notin D_F$ and $(s, \neg x_i) \notin D_F$. Hence, the three rows in the following definition of t are alternative cases:

$$t(x_i) = \begin{cases} \text{true} & \text{if } (s, x_i) \notin D_F; \\ \text{false} & \text{if } (s, \neg x_i) \notin D_F; \\ \text{false} & \text{if } (s, x_i) \in D_F \wedge (s, \neg x_i) \in D_F. \end{cases}$$

Hence, t is a truth assignment over the variables $x_1, \ldots, x_n$, as it unambiguously assigns one truth value to each x_i. Given this, as $\{s\}$ is a complete extension of F, every C_j in F is attacked by at least one x_i or $\neg x_i$ that is not counterattacked by s. This means that t makes every C_j evaluate to true, and thus Φ is satisfiable (observation: this property does not depend on which truth value is used in the third case of the definition of t). □

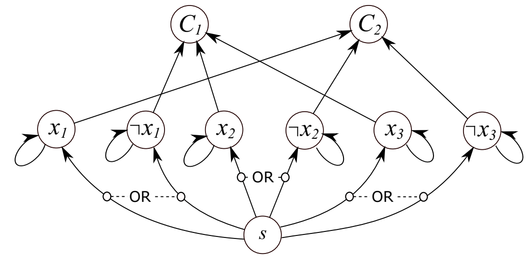

Figure 4: Construction of Theorem 5, for $\Phi = C_1 \wedge C_2$, where $C_1 = \neg x_1 \vee x_2 \vee x_3$ and $C_2 = x_1 \vee \neg x_2 \vee \neg x_3$

4.3 Upper Bounds: Polynomiality Results

We here show that, in the (even simultaneous) presence of OR- and $_1\!\Rightarrow_n$- dependencies, PDVER$^\sigma(DIF, S)$ becomes solvable in polynomial time under both the admissible and stable semantics. This will be proved by showing that Algorithm 1, that uses a constructive strategy for checking that S is an admissible i*-extension, is correct and runs in polynomial time. Basically, Algorithm 1 starts from considering the completion of DIF whose set of attacks D^* contains all the certain and uncertain attacks (line 1). Then, it refines D^* by removing the uncertain attacks (if any) making S conflicting (lines 2– 9) or attacked by arguments that are not counter-attacked (lines 10– 17). After removing an attack, all the attacks on the right-hand side of some $_1\Rightarrow_n$-dependency $(x, y) \Rightarrow B$ may turn out to be absent from D^*, thus the algorithm removes the left-hand side (x, y) from D^* in order to pursue the validity of the completion (lines 9, 17). The removal of attacks is done iteratively, as removing an attack to force the admissibility of S or the consistency with the $\Rightarrow$-dependencies may trigger the violation of another dependency. Observe that the removal of an attack is done only if it is uncertain and if it does not make D^* violate any OR-dependency (note that C denotes the set of the other attacks involved in the OR-dependency); otherwise (lines 3, 7, 15), the algorithm returns *No*. If Algorithm 1 manages to force the admissibility of S while guaranteeing that the dependencies are satisfied, it eventually returns *Yes*.

Theorem 6. *Under* $\sigma \in \{\texttt{ad}, \texttt{st}\}$*, if* Δ *contains only* OR*-dependencies and* $_1\Rightarrow_n$*-dependencies, then* PDVER$^\sigma(DIF, S)$ *is in P.*

Proof. Case $\sigma = \texttt{ad}$. We prove the statement by showing that Algorithm 1 is correct and runs in polynomial time.

The correctness of Algorithm 1 derives from the fact that all the attacks denoted as (a, b) or (x, y) that are detected at lines 2, 6, 10, 14, and that are removed or that trigger the answer *No*, cannot belong to any valid completion where S is admissible. In fact, these attacks contrast the admissibility of S or imply the presence of further attacks that contrast the admissibility of S. Moreover, the answer *Yes* is provided only if the AAF containing all the arguments and all the attacks of the input d-att-iAAF but the attacks (a, b) or (x, y) mentioned above is a valid completion. In fact, if the answer *Yes* is returned, it means that the AAF $\langle A, D^*\rangle$ that reaches line 18 is a valid completion, as every attack removal that has been performed to obtain it is preceded by a

Algorithm 1 Deciding PDVER$^\sigma(DIF, S)$ (under $\sigma = \texttt{ad}$ and only OR- and $_1{\Rightarrow}_n$-dependencies)

Require: An instance $\langle DIF, S\rangle$ of PDVER$^\sigma(DIF, S)$ under $\sigma = \texttt{ad}$ (where $DIF = \langle\langle A, D, D^?\rangle, \Delta\rangle$, $S \subseteq A$, and Δ contains only OR- and $_1{\Rightarrow}_n$-dependencies)
Ensure: The answer to $\langle DIF, S\rangle$

1: $D^* = D \cup D^?$
2: **while** $\exists(a, b) \in D^*$ s.t. $\{a, b\} \subseteq S$ **do**
3:   **if** $(a, b) \notin D^? \vee \exists\ \text{OR}((a, b), C) \in \Delta$ s.t. $C \cap D^* = \emptyset$ **then**
4:     **return** *No*
5:   $D^* = D^* \setminus \{(a, b)\}$
6:   **while** $\exists(x, y) \in D^*$ s.t. $\exists\ (x, y) \Rightarrow B \in \Delta$ s.t. $B \cap D^* = \emptyset$ **do**
7:     **if** $(x, y) \notin D^? \vee \exists\ \text{OR}((x, y), C) \in \Delta$ s.t. $C \cap D^* = \emptyset$ **then**
8:       **return** *No*
9:     $D^* = D^* \setminus \{(x, y)\}$
10: **while** $\exists(a, b) \in D^*$ s.t. $b \in S$ and $\nexists s \in S$ s.t. $(s, a) \in D^*$ **do**
11:   **if** $(a, b) \notin D^? \vee \exists\ \text{OR}((a, b), C) \in \Delta$ s.t. $C \cap D^* = \emptyset$ **then**
12:     **return** *No*
13:   $D^* = D^* \setminus \{(a, b)\}$
14:   **while** $\exists(x, y) \in D^*$ s.t. $\exists\ (x, y) \Rightarrow B \in \Delta$ s.t. $B \cap D^* = \emptyset$ **do**
15:     **if** $(x, y) \notin D^? \vee \exists\ \text{OR}((x, y), C) \in \Delta$ s.t. $C \cap D^* = \emptyset$ **then**
16:       **return** *No*
17:     $D^* = D^* \setminus \{(x, y)\}$
18: **return** *Yes*

check of the fact that it does not raise a dependency violation. Moreover, since S is an admissible extension for the completion $\langle A, D^*\rangle$ that reaches line 18 (as otherwise the while-condition at line 10 would be still true), we have that S is an admissible i*-extension for DIF.

Finally, the polynomiality of Algorithm 1 trivially follows from the fact that the number of removals is linear in the number of attacks, and each removal requires a number of tests that is linear in the number of dependencies in Δ.

Case $\sigma = \texttt{st}$. The proof is even simpler than the previous case. We can compute the answer to PDVER$^\sigma(DIF, S)$ by invoking an algorithm that performs lines 1–9, and then simply checks if there is an attack from S towards every argument outside S. This algorithm obviously runs in polynomial time. Its correctness follows from these facts: 1) as shown in the above case, the attack removals at lines 2–9 are necessary and sufficient to make S conflict-free, and 2) if, after performing these removals, S is not stable, there is no way to make it attack every external argument by removing further attacks. □

4.4 Discussion of the Results

The results reported so far give a complete picture of how the complexity of reasoning on extensions is affected by the introduction of dependencies.

As a baseline, PDVER$^\sigma(DIF, S)$ is in P if $\Delta = \emptyset$ under all the Dung's semantics but the preferred one, under which it is Σ_2^p-complete. It is easy to see that the presence of dependencies does not make PDVER$^\sigma(DIF, S)$ harder under $\sigma = \texttt{pr}$ (this derives from the fact that the cost of verifying the dependencies is dominated by the cost of verifying preferred extensions over a standard AAF).

As for the other semantics, PDVER$^\sigma(DIF, S)$ remains in P under $\sigma \in \{\texttt{ad}, \texttt{st}\}$ if we allow OR- and $_1{\Rightarrow}_n$- dependencies to be specified (even simultaneously, and with no limit on the arity of OR). However, under $\sigma \in \{\texttt{co}, \texttt{gr}\}$, these limitations on the allowed forms of dependencies do not prevent the explosion of the complexity. In fact, even if we impose the strongest restrictions on the arities of OR and $\Rightarrow$ and forbid their combined use, the complexity explodes: under $\sigma \in \{\texttt{co}, \texttt{gr}\}$, PDVER$^\sigma(DIF, S)$ is NP-hard even if only OR^2- or only $_1{\Rightarrow}_1$- dependencies are allowed.

In the presence of any other form of dependency (namely, NAND, CHOICE, and $_m\Rightarrow_n$, with $m \geq 2$ and $n \geq 1$), PDVER$^\sigma(DIF, S)$ is NP-hard, under every semantics. While, as recalled above, PDVER$^\sigma(DIF, S)$ becomes tractable if the arity of the implication is further limited, there is no way to make PDVER$^\sigma(DIF, S)$ in P by reducing the arity of NAND and CHOICE, since the NP-hardness holds also for binary dependencies of these forms.

Thus, further investigation is needed to find islands of tractability for the forms of dependencies in the presence of which PDVER$^\sigma(DIF, S)$ is NP-hard even under $\sigma \in \texttt{ad}$. Interestingly, even the acyclicity of the argumentation graph is not a condition making PDVER$^\sigma(DIF, S)$ easier in these cases: the argumentation graphs exploited in the proofs of theorems 2 and 3 are acyclic. On the other hand, the acyclicity of the argumentation graph makes the complexity of PDVER$^\sigma(DIF, S)$ under $\sigma \in \{\texttt{co}, \texttt{gr}, \texttt{pr}\}$ in the presence of OR- and $_1{\Rightarrow}_n$- dependencies (row 7 of Table 1) move in P, since the acyclicity makes the complete, grounded, and preferred semantics collapse with the stable semantics (under which PDVER$^\sigma(DIF, S)$ is in P).

	PDVER$^\sigma(DIF, S)$			
	ad,st	co	gr	pr
1 EMPTY	P	P	P	Σ_2^p-c
2 OR	P	P	NP-c	Σ_2^p-c
3 NAND	P	NP-c	NP-c	Σ_2^p-c
4 $_m{\Rightarrow}_n$	NP-c	NP-c	NP-c	Σ_2^p-c
5 $_m{\Rightarrow}_1$	P	NP-c	NP-c	Σ_2^p-c
6 CHOICE	NP-c	NP-c	NP-c	Σ_2^p-c
7 CHOICE^2	P	NP-c	NP-c	Σ_2^p-c
8 OR + NAND	NP-c	NP-c	NP-c	Σ_2^p-c
9 OR + XOR^2	NP-c	NP-c	NP-c	Σ_2^p-c
10 NAND + XOR^2	NP-c	NP-c	NP-c	Σ_2^p-c
11 $_m{\Rightarrow}_1$ + XOR^2	NP-c	NP-c	NP-c	Σ_2^p-c
12 $_m{\Rightarrow}_1$ + OR	NP-c	NP-c	NP-c	Σ_2^p-c
13 $_m{\Rightarrow}_1$ + NAND	P	NP-c	NP-c	Σ_2^p-c
14 ANY OTHER	NP-c	NP-c	NP-c	Σ_2^p-c

Table 2: Complexity of PDVER$^\sigma(DIF, S)$ for arg-iAAFs

By comparing the results in this paper with our previous work (Fazzinga, Flesca, and Furfaro 2021), whose results on the computational complexity of $\text{PDV}\textsc{er}^\sigma(DIF, S)$ over arg-iAAFs are summarized in Table 2, we observe that the correlations between attacks make the reasoning harder than the case of correlations involving arguments: the complexity of $\text{PDV}\textsc{er}^\sigma(DIF, S)$ over arg-iAAFs is a lower bound for its complexity over att-iAAFs (if the same dependencies are allowed). Besides this, the main differences are:

- OR-dependencies and $\sigma = \texttt{co}$: $\text{PDV}\textsc{er}^\sigma(DIF, S)$ is in P over arg-iAAFs, but NP-complete over att-iAAFs, even for arity 2;
- NAND-dependencies and $\sigma \in \{\texttt{ad}, \texttt{st}\}$: $\text{PDV}\textsc{er}^\sigma(DIF, S)$ is in P over arg-iAAFs, but NP-complete over att-iAAFs, even for arity 2;
- CHOICE-dependencies and $\sigma \in \{\texttt{ad}, \texttt{st}\}$: over arg-iAAFs, $\text{PDV}\textsc{er}^\sigma(DIF, S)$ is in P for arity 2, and NP-complete for arity greater than 2, while over att-iAAFs it is NP-complete for any arity;
- ${}_m\Rightarrow_n$-dependencies and $\sigma \in \{\texttt{ad}, \texttt{st}\}$ and $n = 1$: $\text{PDV}\textsc{er}^\sigma(DIF, S)$ is in P over arg-iAAFs for any m, while, over att-iAAFs, $\text{PDV}\textsc{er}^\sigma(DIF, S)$ is in P if $m = 1$, and NP-complete otherwise.

5 Related Work

Uncertain attacks were first introduced as the core of *Partial Argumentation Frameworks* (PAFs) in (Coste-Marquis et al. 2007; Cayrol, Devred, and Lagasquie-Schiex 2007), to allow a form of ignorance to be encoded when deciding on the type of interaction between arguments. att-iAAFs have been introduced as a special case of iAAFs (where both arguments and attacks can be uncertain) in (Baumeister et al. 2018), where the computational complexity of the verification problem has been studied. Here, a completion-based semantics for extensions of iAAFs has been introduced, later denoted in the literature as "*i-extensions*". In (Fazzinga, Flesca, and Furfaro 2020b), i*-extensions have been introduced as a revisitation of i-extensions to fix some counterintuitive behaviors. However, in the case of att-iAAFs (where the arguments are certain), i- and i*- extensions coincide. As for the acceptance problem over iAAFs, it was defined (by implicitly resorting to i*-extensions) in (Baumeister et al. 2021), where it was shown to be NP-hard under Dung's semantics. This means that embedding dependencies in iAAFs does not increase the complexity of the acceptance, and that the practical algorithms for this problem in (Baumeister et al. 2021) are prone to be adapted, since dependencies can be suitably encoded in the propositional formula used to represent the completions in these algorithms.

Reasoning over incomplete AAFs is also related to revising AAFs to enforce the existence of an extension (Baumann and Ulbricht 2019), or to make a set an extension (Coste-Marquis et al. 2015), as well as to the credulous/skeptical conclusion problems in *Control Argumentation Frameworks* (Dimopoulos, Mailly, and Moraitis 2018).

As for the quantitative approaches to the representation of uncertainty, they allow the specification of preferences and/or weights (Amgoud and Vesic 2011; Bench-Capon 2003; Brewka, Polberg, and Woltran 2014; Coste-Marquis et al. 2012; Dunne et al. 2011; Kaci, van der Torre, and Villata 2018; Modgil 2009) and/or trust degrees (Fazzinga, Flesca, and Furfaro 2020a), or probabilities, according to the "epistemic" (Thimm 2012; Hunter and Thimm 2014), or the "constellation" paradigm (Hunter 2014; Dung and Thang 2010; Doder and Woltran 2014; Dondio 2014; Hunter 2012; Li, Oren, and Norman 2011; Fazzinga, Flesca, and Parisi 2015; Fazzinga, Flesca, and Furfaro 2019a; Fazzinga, Flesca, and Furfaro 2019b). The last ones are more related to our framework, as they can be seen as iAAFs where a probability distribution is defined over the completions.

Other related works are those where, even if with different purposes, constraints are embedded in the argumentation framework, such as (Coste-Marquis, Devred, and Marquis 2006) (where constraints refine the set of extensions), (Alfano et al. 2021) (where weak constraints define a ranking of the extensions), (Brewka et al. 2017) (where the logical formulas occurring in *Abstract Dialectical Frameworks* can be viewed as constraints defining acceptance conditions), and (Wallner 2020) (where constraints are used to limit the admitted structural modifications in dynamic scenarios). Finally, our study is also related with works investigating the effects of removing/adding terms of the argumentation on the status of (sets of) arguments (Alfano and Greco 2021).

6 Conclusions and Future Work

In continuation with our previous work on argument-incomplete AAFs (Fazzinga, Flesca, and Furfaro 2021), we have investigated a generalization of *attack-incomplete AAFs*, where correlations between the uncertain attacks can be specified in terms of "elementary" dependencies (expressed by means of common logical connectives). This widens the scope of traditional attack-incomplete AAFs, whose representation paradigm, although recognized as a suitable formalism for merging different subjective views on the interactions between the arguments, is intrinsically limited by the assumption of independence between the attacks (as already observed in (Neugebauer 2019)). Our research has been focused on the computational complexity of the verification problem under the possible perspective and for Dung's semantics. In this regard, we have performed a thorough investigation, by studying the sensitivity of the complexity to the semantics of extensions and the form of correlations. In the future work, we plan to address the verification problem under the necessary perspective.

Acknowledgements

This work was partially supported by the European Commission funded project *Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us* (grant no. 820437).

References

Alfano, G., and Greco, S. 2021. Incremental skeptical preferred acceptance in dynamic argumentation frameworks. *IEEE Intell. Syst.* 36(2):6–12.

Alfano, G.; Greco, S.; Parisi, F.; and Trubitsyna, I. 2021. Argumentation frameworks with strong and weak constraints: Semantics and complexity. In *Proc. Conf. on Artificial Intelligence (AAAI)*.

Amgoud, L., and Vesic, S. 2011. A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.* 63(2):149–183.

Baumann, R., and Ulbricht, M. 2019. If nothing is accepted - repairing argumentation frameworks. *J. Artif. Intell. Res.* 66:1099–1145.

Baumeister, D.; Neugebauer, D.; Rothe, J.; and Schadrack, H. 2018. Verification in incomplete argumentation frameworks. *Artif. Intell.* 264:1–26.

Baumeister, D.; Järvisalo, M.; Neugebauer, D.; Niskanen, A.; and Rothe, J. 2021. Acceptance in incomplete argumentation frameworks. *Artificial Intelligence* 295.

Baumeister, D.; Rothe, J.; and Schadrack, H. 2015. Verification in argument-incomplete argumentation frameworks. In *Proc. of Algorithmic Decision Theory - 4th International Conference (ADT)*, volume 9346, 359–376.

Bench-Capon, T. J. M. 2003. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.* 13(3):429–448.

Brewka, G.; Ellmauthaler, S.; Strass, H.; Wallner, J. P.; and Woltran, S. 2017. Abstract dialectical frameworks. an overview. *FLAP* 4(8).

Brewka, G.; Polberg, S.; and Woltran, S. 2014. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems* 29(1):30–38.

Cayrol, C.; Devred, C.; and Lagasquie-Schiex, M. 2007. Handling ignorance in argumentation: Semantics of partial argumentation frameworks. In *Proc. Conf. Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), Hammamet, Tunisia*, 259–270.

Coste-Marquis, S.; Devred, C.; Konieczny, S.; Lagasquie-Schiex, M.; and Marquis, P. 2007. On the merging of dung's argumentation systems. *Artif. Intell.* 171(10-15):730–753.

Coste-Marquis, S.; Konieczny, S.; Marquis, P.; and Ouali, M. A. 2012. Weighted attacks in argumentation frameworks. In *Proc. Principles of Knowledge Representation and Reasoning (KR), Rome, Italy*.

Coste-Marquis, S.; Konieczny, S.; Mailly, J.-G.; and Marquis, P. 2015. Extension enforcement in abstract argumentation as an optimization problem. In *Proc. Int. Conf. on Artificial Intelligence (IJCAI)*, 2876—-2882.

Coste-Marquis, S.; Devred, C.; and Marquis, P. 2006. Constrained argumentation frameworks. In *Proc. Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, 112–122. AAAI Press.

Dimopoulos, Y.; Mailly, J.; and Moraitis, P. 2018. Control argumentation frameworks. In *Proc. Conf. on Artificial Intelligence (AAAI), New Orleans, USA*, 4678–4685.

Doder, D., and Woltran, S. 2014. Probabilistic argumentation frameworks - A logical approach. In *Proc. Conf. Scalable Uncertainty Management (SUM), Oxford, UK, Sept. 15-17*, 134–147.

Dondio, P. 2014. Toward a computational analysis of probabilistic argumentation frameworks. *Cybernetics and Systems* 45(3):254–278.

Dung, P. M., and Thang, P. M. 2010. Towards (probabilistic) argumentation for jury-based dispute resolution. In *Proc. Computational Models of Argument (COMMA)*, 171–182.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2):321–358.

Dunne, P. E.; Hunter, A.; McBurney, P.; Parsons, S.; and Wooldridge, M. 2011. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artif. Intell.* 175(2):457–486.

Fazzinga, B.; Flesca, S.; and Furfaro, F. 2019a. Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence. *Artif. Intell.* 268:1–29.

Fazzinga, B.; Flesca, S.; and Furfaro, F. 2019b. Vispraf: A system for visually specifying and reasoning over probabilistic abstract argumentation frameworks. In Santini, F., and Toniolo, A., eds., *Proc. Workshop on Advances In Argumentation In Artificial Intelligence (AI^3)*, 85–99.

Fazzinga, B.; Flesca, S.; and Furfaro, F. 2020a. Embedding the trust degrees of agents in abstract argumentation. In *Proc. Eur. Conf. on Artificial Intelligence (ECAI)*.

Fazzinga, B.; Flesca, S.; and Furfaro, F. 2020b. Revisiting the notion of extension over incomplete abstract argumentation frameworks. In *Proc. Joint Conf. on Artificial Intelligence (IJCAI)*, 1712–1718. ijcai.org.

Fazzinga, B.; Flesca, S.; and Furfaro, F. 2021. Reasoning over argument-incomplete aafs in the presence of correlations. In *Proc. Joint Conf. on Artificial Intelligence (IJCAI)*. ijcai.org.

Fazzinga, B.; Flesca, S.; and Parisi, F. 2015. On the complexity of probabilistic abstract argumentation frameworks. *ACM Trans. Comput. Log.* 16(3):22:1–22:39.

Hunter, A., and Thimm, M. 2014. Probabilistic argumentation with incomplete information. In *Proc. Eur. Conf. on Artificial Intelligence (ECAI), Prague, Czech Republic*, 1033–1034.

Hunter, A. 2012. Some foundations for probabilistic abstract argumentation. In *Proc. Computational Models of Argument (COMMA)*, 117–128.

Hunter, A. 2014. Probabilistic qualification of attack in abstract argumentation. *Int. J. Approx. Reasoning* 55(2):607–638.

Kaci, S.; van der Torre, L. W. N.; and Villata, S. 2018. Preference in abstract argumentation. In *Proc. Computational Models of Argument (COMMA)*, 405–412.

Li, H.; Oren, N.; and Norman, T. J. 2011. Probabilistic argumentation frameworks. In *Proc. Int. Workshop on Theory and Applications of Formal Argumentation (TAFA), Barcelona, Spain*, 1–16.

Modgil, S. 2009. Reasoning about preferences in argumentation frameworks. *Artif. Intell.* 173(9-10):901–934.

Neugebauer, D. 2019. *Bridging the Gap between Online Discussions and Formal Models of Argumentation*. Ph.D. Dissertation, University of Düsseldorf, Germany.

Thimm, M. 2012. A probabilistic semantics for abstract argumentation. In *Proc. Eur. Conf. on Artificial Intelligence (ECAI), France*, 750–755.

Wallner, J. P. 2020. Structural constraints for dynamic operators in abstract argumentation. *Argument Comput.* 11(1-2):151–190.