Probabilistic Control Argumentation Frameworks

Fabrice Gaignier
University of Paris
fgaignier@hotmail.com

Yannis Dimopoulos
University of Cyprus
yannis@cs.ucy.ac.cy

Jean-Guy Mailly
University of Paris
jean-guy.mailly@u-paris.fr

Pavlos Moraitis
University of Paris and Argument Theory
pavlos.moraitis@u-paris.fr

ABSTRACT

In this paper we present Probabilistic Control Argumentation Frameworks (PCAFs) that extend classical Control Argumentation Frameworks (CAFs) to take into account probabilistic information in the reasoning process. We show that probabilities can be used to optimally control CAFs that cannot be controlled otherwise. We introduce the notion of controlling power, that represents the probability that a control configuration reaches its target. A computational method based on Monte Carlo simulations for computing the controlling power of control configurations is defined. We experimentally show that PCAFs outperform w.r.t runtime classical CAFs and in a large number of situations they can reach the target with a high probability while the classical CAFs fail.

KEYWORDS

Abstract Argumentation; Uncertainty; Probabilistic Reasoning

ACM Reference Format:
Fabrice Gaignier, Yannis Dimopoulos, Jean-Guy Mailly, and Pavlos Moraitis. 2021. Probabilistic Control Argumentation Frameworks. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Computational argumentation has become a major sub-domain in the field of autonomous agents and multi-agent systems during the last decades. It has applications like reasoning in presence of inconsistent knowledge [9], decision making [4, 5, 23], or legal reasoning [8]. It has also applications in the modeling of agent dialogues such as negotiation [14], persuasion [28], etc. A classical abstract argumentation framework [16] is a directed graph where the nodes are the arguments, and edges are attacks between arguments. The evaluation of arguments acceptability is made through the concept of extensions, *i.e.* sets of arguments that are collectively acceptable.

In the last decade, argumentation dynamics [6, 15, 35] and uncertainty in argumentation [7, 11] have gained interest. Bridging the gap between these matters, Control Argumentation Frameworks (CAFs) [13, 26, 27] have been defined to provide an approach that allows an agent to enforce a set of arguments (called the target) as a subset of some (or each) extension in presence of uncertainty.

As said before argumentation has shown its value in automated negotiation (see e.g. [3, 14]). In this context, it was natural to use CAFs for proposing a new negotiation framework [12], where CAFs are used to model the uncertain knowledge that an agent has about the profile of its opponent. Roughly speaking, a proponent agent makes an offer to its opponent if this offer is supported in the opponent's uncertain theory, by an argument that is expected to be accepted by the opponent. The CAFs reasoning mechanism aims at guaranteeing that the supporting argument is accepted whatever the current ("real") opponent theory. When this arrives we say that we have a controllable CAF.

Experimental results presented in [12] have shown that using CAFs allows to improve the quality of negotiation, *i.e.* the percentage of agreement between agents. But there are situations where a CAF is not controllable, meaning that the agent cannot make an offer, because it is not sure that the arguments supporting offers can be accepted by its opponent (even with the support of additional arguments sent to the opponent).

However, the CAFs used in negotiation are based on a qualitative representation of uncertainty. The work on negotiation proposed in [12] has thus motivated us for proposing a quantitative approach for representing uncertainty in CAFs by introducing *Probabilistic CAFs (PCAFs)*. Indeed, we do expect this will improve the overall quality of the negotiation (see e.g. observations in [29]). For instance, in situations where the negotiation dialogue is submitted to time and/or number of rounds limits constraints, it is worth focusing on the "best" completions (corresponding to the most probable profiles) instead of considering the whole set of completions (i.e. all possible profiles) for computing the arguments that will support the desired offer in the "real" opponent's theory.

Besides automated negotiation, *Probabilistic CAFs* could also be used in persuasion dialogues but they have natural applications in many other domains, since conflicting information and uncertainty are omnipresent in real world problems such as automated risk and threat management and mitigation, automated management of regulatory compliance, crisis risk management, strategic risk management, etc., where quantitative information on uncertainty can be gathered and profitable. For example in the context of strategic risk management, fixed arguments could represent strategic goals (conclusions of arguments) and scenarios established by the business analysts that support these goals (supporting evidence). Strategic goals are usually conflicting. Scenarios may be then in conflict with potential external events (represented by uncertain arguments) that could occur. But these events could then be mitigated by actions the company may take to enforce and protect its strategy (represented by control arguments). So being able to calculate the probability of success of a set of such actions through

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

probabilistic controllability, will determine the probability of success of the strategy, and if this probability is too low then it would help the decision makers to change their current strategy.

While argumentation frameworks with probabilities already exist [19–21, 24, 25, 30–33], our PCAF is the first one to incorporate argumentation dynamics and probabilistic argumentation. In this paper we experimentally show that our PCAFs approach outperforms (in most cases) the classical CAFs one w.r.t. runtime and target reachability.

2 PRELIMINARIES

2.1 Background on AFs and CAFs

We assume that the reader is familiar with abstract argumentation [16]. We use $AF = \langle A, R \rangle$ to denote an argumentation framework (AF), and $st(AF)$ for its set of stable extensions. The stable semantics is chosen for a question of exemplification, but all concepts defined here can be directly adapted to other semantics. See [16] for other extension semantics.

Let us now introduce control argumentation frameworks (CAFs) [13]. Besides the classical arguments and attacks, a CAF is made of uncertain information (that can represent the information that an agent has about the environment, or about the other agents) and control arguments (that represent the possible ways for the agent to interact with the environment or the other agents). Formally, $CAF = \langle F, C, U \rangle$ with $F = \langle A_F, \rightarrow \rangle$; $C = \langle A_C, \Rrightarrow \rangle$; $U = \langle A_U, \dashrightarrow \cup \leftrightarrows \rangle$. A_F, A_C and A_U are disjoint sets of arguments, while $\rightarrow$, $\Rrightarrow$, $\dashrightarrow$ and $\leftrightarrows$ are disjoint sets of attacks.

F is the fixed part, made of arguments A_F and attacks $\rightarrow$ that are certain to exist. The uncertain part U is made of arguments A_U that may appear in the system, but could be absent. Similarly, the attacks from $\dashrightarrow$ are uncertain. Finally, the symmetric conflict relation $\leftrightarrows$ allows to state that we are sure that two arguments a, b are conflicting, but we are not sure whether the direction of the attack is from a to b, from b to a, or even both at the same time. The uncertain part represents elements that can be actually part of the system (or not) w.r.t. the evolution of a dynamic environment, or w.r.t. the actions of other agents. Finally, the control part C is made of arguments A_C and attacks $\Rrightarrow$ that the agent can use for defending arguments in the fixed part from attacks coming from the uncertain part. When it is the case, the agent defines a *control configuration* $cc \subseteq A_C$ and then only the control arguments in cc (and the associated attacks) are considered by the system; this defines the CAF configured by cc.

Similarly to incomplete argumentation frameworks [7, 11], a CAF can be associated with a set of *completions*, that are classical AFs compatible with the (uncertain) information carried by CAF. Each completion represents a possible state of the system that is modeled by CAF.

The notion of controllability of a CAF is related to a target, *i.e.* a set of arguments $T \subseteq A_F$. This target has to be (skeptically or credulously) accepted in each completion of CAF. For instance, [12] propose a method for automated negotiation where the target is an argument that supports the agent's preferred offer; making the target accepted in this context means that the agent is certain that its opponent will accept this offer. We say that CAF is *skeptically (resp. credulously) controllable* w.r.t. T and σ if there exist $cc \subseteq A_C$ s.t. T is included in each (resp. some) σ-extension of CAF configured by cc. Since the agent expects the target to be included in each (or some) extension, we can focus on conflict-free targets.

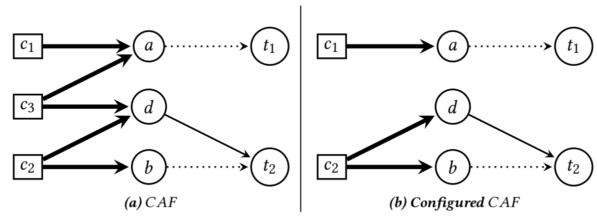


Figure 1: *CAF* **and one possible configuration**

Example 1. *In CAF given at Figure 1a, the fixed part is* $A_F = \{a, b, d, t_1, t_2\}$, $\rightarrow = \{(d, t_2)\}$. *The uncertain part is here* $A_U = \emptyset$, $\leftrightarrows = \emptyset$, *and* $\dashrightarrow = \{(a, t_1), (b, t_2)\}$. *Finally, the control part is* $A_C = \{c_1, c_2, c_3\}$, *and* $\Rrightarrow = \{(c_1, a), (c_2, b), (c_2, d), (c_3, a), (c_3, d)\}$. *Figure 1b corresponds to CAF configured by* $cc = \{c_1, c_2\}$. *This (configured) CAF has four completions, since both* (a, t_1) *and* (b, t_2) *can either belong to the completion or not. We notice that the target* $\{t_1, t_2\}$ *is included in each extension of each completion of the configured CAF (for any of the semantics considered in this paper). So, CAF is skeptically controllable w.r.t.* $\{t_1, t_2\}$ *the stable semantics.*

2.2 New Definitions and Notations

Let us now introduce some new concepts and notations related to CAFs, that are not defined in [13], and that will be useful when working with probabilities.

First, we distinguish between "basic" control configurations, that consist in selecting a subset of A_C, and a valid configuration that allows to guarantee the acceptance of a target set of arguments.

Definition 1. *Let* $T \subseteq A_F$ *be a target set of arguments, and* σ *a semantics. For a given* $CAF = \langle F, C, U \rangle$, $cc \subseteq A_C$ *is a skeptical (resp. credulous) valid control configuration of CAF w.r.t.* T *and* σ *if* T *is included in each (resp. at least one)* σ*-extension of every completion of* $CAF' = \langle F, C', U \rangle$, *with* $C' = \langle cc, \{(a_i, a_j) \in \Rrightarrow \mid a_i, a_j \in (A_F \cup A_U \cup cc)\} \rangle$.

For instance, in CAF given at Figure 1a, $cc = \{c_1, c_2\}$ is a valid configuration: in every completion, t_1 is defended against the (potential) attack from a, and t_2 is defended against the (potential) attack from t_2. On the contrary, $cc' = \{c_1\}$ is not valid, since t_2 is not defended against b in completions where (b, t_2) is there. Controlling a CAF is then the capacity to find at least a valid control configuration for the CAF.

Let us now introduce a notation for the set of completions of a CAF configured by a control configuration:

Definition 2. *We define* $Comp(CAF)$ *as the set of all possible completions of CAF. Then,* $Comp_{cc}(CAF) = Comp(CAF')$ *where* $CAF' = \langle F, C', U \rangle$, *with* $C' = \langle cc, \{(a_i, a_j) \in \Rrightarrow \mid a_i, a_j \in (A_F \cup A_U \cup cc)\} \rangle$.

Again, when we consider CAF from Figure 1a, for $cc = \{c_1, c_2\}$, $Comp_{cc}(CAF)$ is the set of completions of the configured CAF drawn at Figure 1b.

Definition 3. *Given $CAF = \langle F, C, U \rangle$, we define CAF_U, the CAF without control part derived from it: $CAF_U = \langle F, \emptyset, U \rangle$. A root completion of CAF is a completion of the associated CAF_U.*

A root completion of a CAF is then a completion built with no control arguments. Intuitively, this concept will be useful in the computation of probabilities of completions: since control arguments are not attached with a probability (but they are only used if they are selected by the agent), the probability of any completion is the same as the probability of the associated root completion.

Finally, given a semantics σ and $T \subseteq A_F$ a target, cc is a skeptical (resp. credulous) valid configuration for a root completion $AF_r = \langle A, R \rangle$ if T belongs to each (resp. at least one) extension of the AF $\langle A \cup cc, R \cup \{(a_i, a_j) \in \Rightarrow \mid a_i, a_j \in (A_F \cup A_U \cup cc)\} \rangle$.

$CC^T_{\sigma,X}(CAF)$ denotes the set of all X valid control configurations for the CAF w.r.t. T and σ, where $X \in \{sk, cr\}$ stands for "skeptical" or "credulous". Similarly, $CC^T_{\sigma,X}(AF_r)$ is the set of all X valid control configurations for the root completion AF_r w.r.t. T and σ. If it is clear from the context, we drop σ and X to keep the notation lighter.

3 PROBABILISTIC CAF

Since a CAF contains an uncertain part, it makes sense to study how a probability distribution can be added to the uncertain part and how extensions and control configurations can be qualified using this distribution. While probabilistic argumentation has already been studied, our work is the first approach that is related to argumentation dynamics (more precisely, we recall that CAFs are closely related to extension enforcement defined by [6]). We discuss related work in a later section of the paper.

Here, we consider the case where the probabilities of different arguments and attacks are independent. Indeed, having conditional probabilities is meaningful when structured argumentation is considered. For instance, the probability of an argument to appear in the framework is related to the probability of its sub-arguments. But this kind of approach is out of the scope of our work, since we consider abstract argumentation frameworks where arguments are self-contained.

Therefore we can give a formal definition of a probabilistic CAF.

Definition 4. *A Probabilistic Control Argumentation Framework is a tuple $PCAF = \langle F, C, U, p_1, p_2, p_3 \rangle$ where $\langle F, C, U \rangle$ is a CAF, and:*

- *p_1 is a mapping from A_U to $]0;1[$ that expresses the probability of presence of uncertain arguments;*
- *p_2 is a mapping from $\dashrightarrow$ to $]0;1[$, that gives the probability of presence of uncertain attacks, if both arguments are present (implicit conditional probability);*
- *p_3 is a mapping from $\rightleftarrows$ to $]0;1[\times]0;1[$, corresponding to the probability that an undirected attack appears in one or the other direction when both arguments are present (again, there is an implicit conditional probability). Since there certainly is a conflict between the arguments, $p_3(x)[0] + p_3(x)[1] \leq 1$ holds.*

For $(a,b) \in \leftrightarrows$ an undirected conflict, $p_3((a,b))$ is a pair s.t. $p_3((a,b))[0]$ is the probability that only (a,b) is in the system, $p_3((a,b))[1]$ is the probability that only (b,a) is in the system, and $1 - p_3((a,b))[0] - p_3((a,b))[1]$ is the probability that both attacks are in the system.

Here we need to clarify that none of the individual probabilities of uncertain elements can be 0 or 1. If the probability is 0, we just need to remove the element from the PCAF definition, and if the probability is 1, we have to move the element to the fixed part F.

A simple example is given at Figure 2, where the uncertainty only concerns two attacks: (a, t_1) belongs to the system with a probability of 0.2, while (b, t_2) has a probability of 0.8 to appear.

Considering $PCAF$ and a root completion $AF_r = \langle A_r, D_r \rangle$ of $PCAF$, s.t. $A_r = A_{rU} \cup A_F$ and $A_{rU} \subseteq A_U$, we define the following random events:

- $on(a) \iff a \in A_r$
- $off(a) \iff a \notin A_r$
- $on((a_i, a_j)) \iff (a_i, a_j) \in D_r$
- $off((a_i, a_j)) \iff (a_i, a_j) \notin D_r$ and $a_i, a_j \in A_r$
- $dir_1((a_i, a_j)) \iff (a_i, a_j) \in D_r$ and $(a_j, a_i) \notin D_r$
- $dir_2((a_i, a_j)) \iff (a_j, a_i) \in D_r$ and $(a_i, a_j) \notin D_r$
- $dir_{12}((a_i, a_j)) \iff (a_i, a_j) \in D_r$ and $(a_j, a_i) \in D_r$

We define as well the following sets:

- $A_1 = A_{rU}$
- $A_2 = A_U \setminus A_{rU}$
- $R_1 = \{(a_i, a_j) \in D_r \mid a_i, a_j \in A_r \text{ and } (a_i, a_j) \in \dashrightarrow\}$
- $R_2 = \{(a_i, a_j) \notin D_r \mid a_i, a_j \in A_r \text{ and } (a_i, a_j) \in \dashrightarrow\}$
- $R_3 = \{(a_i, a_j) \in D_r, (a_j, a_i) \notin D_r \mid a_i, a_j \in A_r, (a_i, a_j) \in \leftrightarrows\}$
- $R_4 = \{(a_j, a_i) \in D_r, (a_i, a_j) \notin D_r \mid a_i, a_j \in A_r, (a_i, a_j) \in \leftrightarrows\}$
- $R_5 = \{(a_j, a_i) \in D_r, (a_i, a_j) \in D_r \mid a_i, a_j \in A_r, (a_i, a_j) \in \leftrightarrows\}$

Knowing that (1) $P(off(x)) = 1 - P(on(x))$, (2) $P(off((a_i, a_j))) = 1 - P(on((a_i, a_j)))$, and (3) $P(dir_{12}(a_i, a_j)) = 1 - P(dir_1(a_i, a_j)) - P(dir_2(a_i, a_j))$, we observe the following result.

Observation 1.

$$\begin{aligned} P(AF_r) = \; & \prod_{a \in A_1} P(on(a)) \times \prod_{a' \in A_2} [1 - P(on(a'))] \\ & \times \prod_{(a_i, a_j) \in R_1} P[on((a_i, a_j))] \\ & \times \prod_{(a_k, a_l) \in R_2} [1 - P[on((a_k, a_l)]] \\ & \times \prod_{(a_m, a_n) \in R_3} P[dir_1((a_m, a_n))] \\ & \times \prod_{(a_o, a_p) \in R_4} P[dir_2((a_o, a_p))] \\ & \times \prod_{(a_q, a_r) \in R_5} [1 - P[dir_1((a_q, a_r))] \\ & - P[dir_2((a_q, a_r))]] \end{aligned}$$

Observation 1 states that the probability of a root completion is computed as the product of the probabilities of its elements. The probability distribution for the root completions follows the Kolmogorov axioms.

4 CONTROLLING UNDER UNCERTAINTY

4.1 Discussion on Probabilistic Controllability

When a PCAF has valid control configurations, the exact methods proposed by [13, 27], based on Quantified Boolean Formulas (QBFs) or iterative SAT solving, can be used to compute these configurations. But in the case where the PCAF is not controllable, the problem is different. We need to identify the "best" control configuration(s). In this section, we propose two different approaches to solve this problem.

The reasons for a PCAF to be uncontrollable are of two kinds:

- There is one (or more) element of the target $T = \{t_1, \ldots, t_n\}$ for which the PCAF is not controllable taken individually.

If *PCAF* is not controllable w.r.t. $t_i \in T$, then *PCAF* is not controllable w.r.t. T.

- Each individual $t_i \in T$ can be controlled individually, but there is no way to control all of them at the same time. Therefore the intersection of control configurations with regard to each individual target element of T is empty.

From the above, it is clear that even if it is not possible to control a PCAF w.r.t. a target T, we may still be able to control the PCAF w.r.t. $T' \subset T$. The following example illustrates the issue of "optimally controlling" an uncontrollable PCAF.

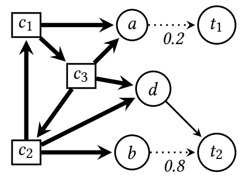


Figure 2: Uncontrollable PCAF

Example 2. *The PCAF depicted at Figure 2 is not controllable with regards to $T = \{t_1, t_2\}$. It has exactly four possible root completions. If we choose the control configuration $\{c_2\}$, we will control T with a probability of 80%. The only case where the PCAF is not controlled w.r.t. T is when the attack (a, t_1) is present. But if we choose this control configuration, t_2 will always be in all stable extensions whereas t_1 will be present only 80% of the time. If now we set a priority on the elements of the target, and it turns out that t_1 is more important to protect than t_2, we would rather choose $\{c_3\}$ as the best control configuration. Now, t_1 is sure to be accepted, while t_2 is only accepted in situations where the attack (b, t_2) is not in the system.*

Both approaches are valid. An agent could have a policy where all target elements are equally important, but in other situations, it could decide to set priorities on the target elements. For instance, in an autonomous car, the target to protect pedestrians is certainly more important than not damaging the car, even if at the end it is preferable to have both targets protected if possible. In order to formalize both approaches, we introduce two fundamental concepts: *controlling power* and *supporting power*.

4.2 Maximising Controlling Power

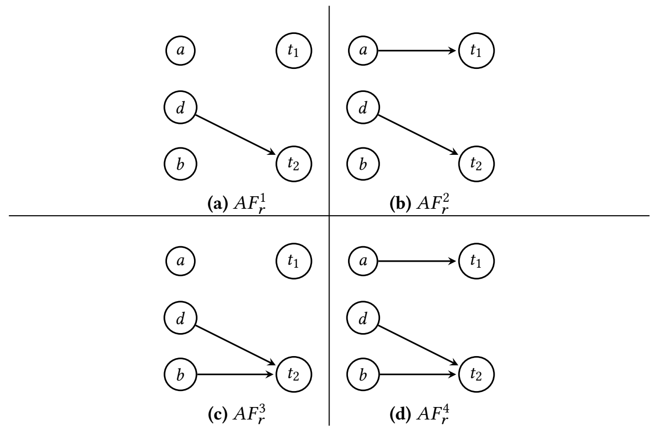
The controlling power of a control configuration *cc* in *PCAF*, w.r.t. a semantics and a target T, is the probability that the target is accepted when *PCAF* is configured by *cc*. This concept is formalized as follows:

Definition 5. *Given $PCAF = \langle F, C, U, p_1, p_2, p_3 \rangle$, a target T and a control configuration $cc \subseteq A_C$, we define $f_T^\sigma : Comp_{cc}(PCAF) \rightarrow \{0, 1\}$ s.t. $f_T^\sigma(AF_{cc}) = 1$ iff T is accepted according to the semantics σ for the completion $AF_{cc} \in Comp_{cc}(PCAF)$.*
The controlling power *of cc w.r.t. T is:*

$$c_{power}^T(cc) = \sum_{AF_{cc} \in Comp_{cc}(PCAF)} P(AF_{cc}) \times f_T^\sigma(AF_{cc})$$

The controlling power of a configuration *cc*, w.r.t. a target T, is the sum of the probabilities of the completions where T is accepted when the PCAF is configured with *cc*. Said otherwise, this is the probability that the target is reached if the PCAF is configured by *cc*.

Since we know that there is always a root configuration AF_r with the exact same probability as AF_{cc} obtained by removing all control arguments and control attacks from AF_{cc}, we have $c_{power}^T(cc) = \sum_{AF_{cc} \in Comp_{cc}(PCAF)} P(AF_r) * f_T^\sigma(AF_{cc})$, where AF_r is the corresponding root completion of AF_{cc}.



(a) AF_r^1 **(b)** AF_r^2 **(c)** AF_r^3 **(d)** AF_r^4

Figure 3: Four root completions of *PCAF*

Example 3. *We consider again the PCAF described at Figure 2. This PCAF has four root completions, that are represented at Figure 3. These completions have probabilities: $P(AF_r^1) = (1 - 0.2) \times (1 - 0.8) = 0.16$, $P(AF_r^2) = 0.2 \times (1 - 0.8) = 0.04$, $P(AF_r^3) = (1 - 0.2) \times 0.8 = 0.64$ and $P(AF_r^4) = 0.2 \times 0.8 = 0.16$. We start with the target $T = \{t_1, t_2\}$. With $cc = \{c_2\}$, T is accepted in the completions built from AF_r^1 and AF_r^3 (since c_2 does not defend t_1 against a, the other completions do not work). So, $c_{power}^T(cc) = 0.16 + 0.64 = 0.8$. Similarly, $c_{power}^T(cc') = 0.2$, with $cc' = \{c_3\}$. If we consider other targets, we have* e.g. *$c_{power}^{\{t_2\}}(cc) = 1$, $c_{power}^{\{t_1\}}(cc) = 0.8$ or $c_{power}^{\{t_1\}}(cc') = 0.8$.*

We identify the best configurations w.r.t. controlling power:

Definition 6. *The most powerful control configurations for a PCAF w.r.t. a target T are*

$$mpcc^T(PCAF) = \{cc \mid \text{argmax}(c_{power}^T(cc))\}$$

The controlling power of a control configuration *cc* is the probability that it protects (according to the chosen semantics) the target. Thus, the set of most powerful control configurations are those with the highest probability to protect the target. From the definition above, if the PCAF is controllable (in the sense of [13]), the most powerful control configurations are the valid control configurations of the PCAF, with a controlling power of 1. Now, we identify a way to simplify the search for control configurations:

Theorem 1. *Given $PCAF = \langle F, C, U, p_1, p_2, p_3 \rangle$, $\forall cc \subseteq A_C$, $\exists cc'$ such that*

- *$c_{power}^T(cc') \geq c_{power}^T(cc)$;*

- *cc' is conflict free;*
- *cc' is conflict free with T;*
- *cc' is always in the same extension as T;*
- $c^{T}_{power}(cc') = c^{T\cup cc'}_{power}(cc')$.

The intuition behind Theorem 1 is that removing from a control configuration the arguments that are in conflict with the target and also mutually conflicting, can only increase the controlling power. We end up with a control configuration that is conflict free and conflict free with the target. Thus control arguments and target are always in the same extension.

4.3 Maximising Supporting Power

Once we have chosen a control configuration cc that is conflict-free, and conflict-free with the target, we know that the target will be in the same extension(s) as cc with a probability equal to the controlling power of cc. But other arguments may be present in these extensions, with a possibly different probability. And this is what we are willing to measure with the notion of supporting power.

Definition 7. *Given $PCAF = \langle F, CU, p_1, p_2, p_3\rangle$, a control configuration $cc \subseteq A_C$, a semantics σ, and a target $T \subseteq A_F$, we consider $Cont^{T}_{cc}(PCAF)$ the set of all the root completions that are controlled by cc w.r.t. T. For each $AF_r \in Cont^{T}_{cc}(PCAF)$, X_r denotes the set of σ-extensions containing T and cc. We have $\forall x_r \in X_r,\ x_r = T \cup cc \cup A_r$ where $A_r \subseteq A_U \cup A_F$.*

The supporting power of an argument $a_i \in A_U \cup A_F$, according to a control configuration cc and a target T is:

$$sp^{T}_{cc}(a_i) = \frac{\sum_{AF_r \in Cont^{T}_{cc}(PCAF)} P(AF_r) \times \sum_{x_r \in X_r} f_{x_r}(a_i)}{\sum_{AF_r \in Cont^{T}_{cc}(PCAF)} P(AF_r) \times |X_r|}$$

where $f_{x_r}(a_i) = 1$ if $a_i \in x_r$ and $f_{x_r}(a_i) = 0$ otherwise.

The supporting power of an argument a_i according to a given valid control configuration cc and a target T measures the probability of a_i to belong to the same extensions as T. Trivially, the supporting power of any element of the target is 1 and if an argument a_i has a supporting power of 1, it can be added to the target without changing the controlling power of the corresponding control configuration.

Let us now formalize the method used to optimally control a PCAF when the elements of the target do not have the same relative importance.

Definition 8. *Given $\geq$ a total pre-order on T, we define classically $t_i > t_j$ iff $t_i \geq t_j$ and $t_j \ngeq t_i$ and $t_i \approx t_j$ iff $t_i \geq t_j$ and $t_j \geq t_i$. Given such a binary relation, we can split T in $\{T_i \mid 0 \leq i \leq n\}$ such that $\bigcup_{i=0}^{n} T_i = T, \forall a \in T_{i-1}, \forall b \in T_i, a > b$, and $\forall a, b \in T_i, a \approx b$.*

The preferred control configurations pcc^T are defined by:

(1) $CE_0 = mpcc^{T_0}(CAF)$
(2) $CE_i = \{cc \in CE_{i-1} | \sum_{t_i \in T_i} sp^{T_0}_{cc}(t_i)$ *is maximum*$\}$
(3) $pcc^T = CE_n$

Here we keep the best control configurations for the set of preferred target elements, and we refine our choice by maximizing the sum of supporting power recursively on the subsequent target sets.

Example 4. *We have seen at Example 3 that $cc = \{c_2\}$ has a controlling power of 0.8 w.r.t. $T = \{t_1, t_2\}$. If the agent absolutely needs to control t_1 and t_2 at the same time, this is the best control configuration. Now, suppose that the agent can express relative priorities between t_1 and t_2: $t_1 > t_2$ (e.g. the former corresponds to pedestrian safety, while the latter corresponds to avoiding damages to the car). Then, the search for the best configuration consists in optimizing the probability to control $T_0 = \{t_1\}$, and if several configurations are possible at this first step, then the agent chooses one that optimizes the probability to control $T_0 \cup T_1$, where $T_1 = \{t_2\}$. In our example, $cc' = \{c_1\}$ and $cc'' = \{c_3\}$ both have a controlling power of 1 w.r.t. $T_0 = \{t_1\}$. Both have a controlling power of 0.2 w.r.t. $T_0 \cup T_1$.*

In this section, we have proposed two alternatives to define the best control configurations w.r.t. the probabilistic information available in the PCAF: either these are the configurations that have the highest probability to control the whole target, or these are the ones that have the highest probability to control the most important elements of the target. The above possibilities may be very important in multi-issues negotiation dialogues. For example when there is no control configuration that allows for all the arguments of the target, supporting respectively several issues belonging to the negotiation object (e.g. the price of the product, the means of payment, the delivery date, etc.) to get accepted in the opponent's theory, it might be useful to find a control configuration that can control the argument(s) of the target that support the most important/preferred issue(s).

5 COMPUTATIONAL METHOD

In [25] the authors propose an algorithm based on Monte Carlo simulations in order to calculate the probability of a subset of arguments to be accepted according to a given semantic. This is directly related to determining the acceptance of arguments in classical AFs. In our case, the problem is different. We are looking for the control configurations that maximize the probability of a subset of fixed arguments, called target, to get accepted, which is related to the notion of extension enforcement [6]. This is the first time that this kind of reasoning process is applied to probabilistic argumentation. In the meantime we want to measure the frequency of presence of other arguments together with the target and the control configuration in the extensions. We propose a Monte Carlo algorithm to approximate all these metrics at once. Algorithm 1 describes the method. For this purpose, we introduce the notation $Ext_{cc}(AF_r)$, that corresponds to the set of extensions of the AF made of the root completion AF_r and the control arguments (and attacks) from the configuration cc.

The map $cont\langle cc, occ\rangle$ is used to count the number of root completions that are controlled by the configuration cc. On line 2, we initialize $recorder\langle cc, argscounter\rangle$, that is used to compute how many times different arguments appear in the extensions. *argscounter* is also a mapping structure, where each argument a is associated to an integer. We suppose that when a configuration cc is encountered for the first time, the pair $\langle cc, 0\rangle$ is added to the map *count*. Similarly, in the map *recorder*, cc is initially associated with an *argscounter* where each argument is mapped to 0. These lines are omitted from the algorithm to keep it simple. These maps are used to compute (respectively) the controlling power and the supporting

Algorithm 1 Monte Carlo simulation for approaching mpcc

Require: $PCAF = \langle A, C, U, p_1, p_2, p_3 \rangle$, $T \subseteq A_F$

1: Initialize map $cont\langle cc, occ \rangle$
2: Initialize map $recorder\langle cc, argscounter \rangle$
3: **for** $i = 0$ to N **do**
4:   Generate AF_r randomly
5:   **for all** $cc \in CC^T(AF_r)$ **do**
6:     *occ++* //Increment *occ* associated to *cc*
7:     **for all** $ext \in Ext_{cc}(AF_r)$ **do**
8:       **for all** $a \in ext$ **do**
9:         *counter++* //Increment counter associated with a in *argscounter*
10:       **end for**
11:     **end for**
12:   **end for**
13: **end for**
14: **return** $\{\langle cc, occ \rangle \mid occ \text{ is maximal}\}$, *recorder*

power of control configurations. The controlling power of a control configuration *cc* is given by (occ/N), where *occ* is the number associated to *cc* in the map *cont*. The supporting power of any argument according to *cc* is given by $recorder(cc)$. So if we want to use the preferred control configurations instead of most powerful control configuration, we run the above algorithm with T_0 instead of T and apply the recursive definition of preferred control configuration.

Monte Carlo is an approximation method and in the above algorithm we have set a fixed number of simulations N. We know that if $N \to \infty$ we will eliminate the approximation. So we could work with a confidence interval and a given error.

Indeed the algorithm returns on top of the best control configuration *cc* a value: $max(occ)$. The metric (mcp^T will be abbreviated in *mcp*) $mcp = max(occ)/N$ (N is the number of simulations) is an average calculation of independent identically distributed random variables. The central limit theorem states that the law of *mcp* tends towards a normal law and the confidence interval is given by:

$$mcp \pm z_{1-\alpha/2}\sqrt{\frac{mcp(1-mcp)}{N}}$$

where $z_{1-\alpha/2}$ the $1-\alpha/2$ percentile of the normal distribution.

If we want to ensure *mcp* is in this confidence interval with a probability of 99% we must choose $z_{1-\alpha/2} = 2.58$.

If we want to fix the width of the confidence interval to $mcp \pm \epsilon$ we must choose $N \geq \frac{mcp(1-mcp)}{\epsilon^2} * z^2_{1-\alpha/2}$.

We can easily adapt the above algorithm to work with a confidence interval and an error threshold rather than with a fixed number of simulations. But we must be careful since *mcp* has value 0 at first and then can take value 1 in case the CAF is controllable. Therefore taking this approach will always stop after one simulation. We will prefer the interval approach by [1], used by [25]. In that case, we replace *mcp* by $mcp' = \frac{mcp + z^2_{1-\alpha/2}/2}{N + z^2_{1-\alpha/2}}$.

We need now a method to generate random root completion AF_r (line 4 of the algorithm). The method is given below:[1]

- Start with an empty *AF*
- $\forall a_i \in A_F$ add a_i in AF
- Generate $t \in]0,1[$ randomly
- $\forall a_i \in A_U$, if $p_1(a_i) \geq t$ then add a_i to AF
- $\forall (a_i, a_j) \in \rightarrow$ s.t. $a_i \in AF$ and $a_j \in AF$, add (a_i, a_j) to AF
- Generate $t \in]0,1[$ randomly
- $\forall (a_i, a_j) \in \dashrightarrow$ s.t. $a_i \in AF$, $a_j \in AF$ and $p_2(a_i, a_j) \geq t$, add (a_i, a_j) to AF
- Generate $t_1, t_2 \in]0,1[$ randomly with $t_1 + t_2 \leq 1$
- $\forall (a_i, a_j) \in \rightleftarrows$ s.t. $a_i \in AF$ and $a_j \in AF$,
  - $o_1 = 1 \iff p_3((a_i, a_j))[0] \geq t_1$
  - $o_2 = 1 \iff p_3((a_i, a_j))[1] \geq t_2$
  - $o_3 = 1 \iff 1 - p_3((a_i, a_j))[0] - p_3((a_i, a_j))[1] \geq 1 - t_1 - t_2$
- Choose randomly one option between those having value 1 and apply the corresponding *AF* transformation (add direction in one way for o_1, the other for o_2 or both for o_3).

For the last step, it is clear that at least one option has value 1. Indeed, we can transform the last option in: $o_3 = 1$ iff $p_3((a_i, a_j))[0] + p_3((a_i, a_j))[1] \leq t_1 + t_2$. Therefore if $o_1 \neq 1$ and $o_2 \neq 1$, then o_3 has value 1. It is not straightforward to generate (t_1, t_2). These are independent random variables, but t_1 and t_2 are linked by a constraint $t_1 + t_2 \leq 1$ Therefore if we generate a value of t_1 s.t. $0 < t_1 < 1$, we can only choose a value for $t_2 \in]0, 1 - t_1[$. Since we use uniform distributions for the generation of t_i, we see that using this method, we have a bias toward t_2 and the full probability distribution will not be explored. A solution is to choose randomly what variable will be simulated first, then simulate the second (in the allowed interval). This way, over a large number of simulations, we will generate couples (t_1, t_2) that cover the entire set of possible values. We now have a method to generate root completions respecting an underlying probability distribution. This algorithm is linear and its complexity is in $O(|A_U| + |A_F| + |\dashrightarrow| + |\rightleftarrows| + |\rightarrow|)$.

6 EXPERIMENTAL EVALUATION

6.1 Experimental Protocol

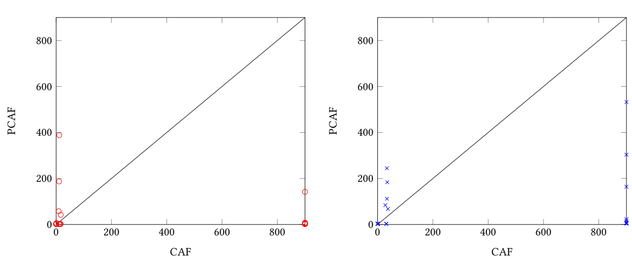
We have generated CAFs following two models of graph theory: Barabási-Albert [2] and Kleinberg [17]. Regarding the choice of Kleinberg and Barabasi-Albert, let us mention that similar graphs are commonly used in experimental studies in the field of abstract argumentation. For instance, the ICCMA competition [10, 18, 34] used Barabasi-Albert graphs for evaluating algorithms dedicated to Dung's semantics. Graphs satisfying small-world properties were also used (namely Watts-Strogatz, while we used Kleinberg). In the absence of benchmarks coming from concrete applications, this kind of "structured" benchmark seem more relevant than "purely random" graphs. For the first model, we have two sets of CAFs, denoted BA50 and BA60, with respectively 50 and 60 arguments, and a density $15-20\%$. The Kleinberg model takes as parameter an integer n, and yields graphs with n^2 nodes. We have generated CAFs with $n \in \{3, 4, 5, 7\}$, which gives the benchmark sets K9, K16, K25 and K49 (with, respectively, 9, 16, 25 and 49 arguments). In each CAF in these benchmark sets, the proportions of fixed and uncertain arguments is between 40 and 45%, while there are between 10 and 20% of control arguments. Then, the proportion of attacks that belong to $\rightarrow$ or $\dashrightarrow$ is 40%, while there are 20% of attacks in $\rightleftarrows$. Each of the benchmark sets BA50, BA60, K9, K16, K25 and K49 contains 30 CAFs. The process described above yields

[1] Recall that we focus on the case of independence of probabilities.

"classical" CAFs. In order to compare the efficiency of classical controllability and probabilistic CAFs, we generate PCAFs that are equivalent to the CAFs: each argument in A_U or attack in $\dashrightarrow$ has a probability $\frac{1}{2}$ of existing, while attacks in $\rightleftarrows$ have a probability $\frac{1}{3}$ of existing in one direction, or the other direction, or both at the same time. This makes a total of 180 CAF and 180 PCAF instances. For these CAFs and PCAFs, we compute the credulous controllability under the stable semantics for a target made of one argument $t \in A_F$, respectively using the QBF-based approach from [13], and the Monte Carlo simulation from Section 5. Let us mention that this problem is Σ_3^P-complete [27], which emphasizes the difficulty to solve it with complete algorithms, and thus justifies the need of approximate algorithms like ours. The experiments were made on a machine running on Ubuntu 20.04, with a Core i7 CPU (2.40GHz) and 8GB of RAM, with a timeout of 900 seconds. The timeout of 900 seconds was chosen since reasoning with CAFs/PCAFs is arguably harder (from a computational point of view) than classical reasoning with AFs. So we chose a timeout equivalent to 1.5 times the timeout of 600 seconds chosen at each edition of the ICCMA competition.

6.2 Results

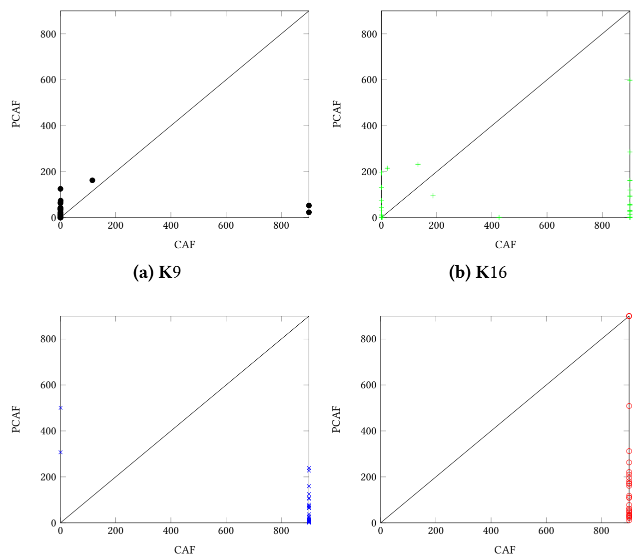
Figure 4 compares the runtime for the Barabási-Albert instances. Each point represents an instance, such that the value on the x-axis is the runtime for the classical CAF approach while the value on the y-axis is the runtime for the probabilistic approach. Thus, each point below the diagonal line represents an instance that is solved faster with the PCAF approach than with the classical CAF approach. We observe that most of the instances are in this situation. Also, only the classical CAF approach faces some timeout, including some instances that are solved almost instantly by the PCAF approach. Figure 5a compares the runtime for the K9



(a) **BA**50 (b) **BA**60

Figure 4: Runtime comparison for Barabási-Albert Instances

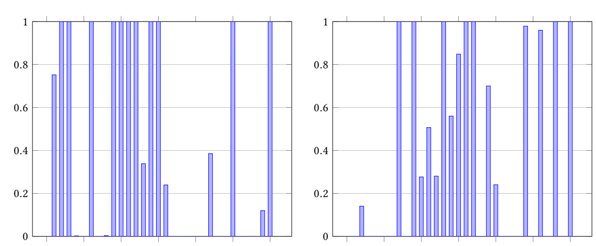
benchmark set. Clearly, for such small instances, the classical CAF approach outperforms the probabilistic approach. In the majority of these small instances the solutions with classical CAFs have been obtained almost instantly while those with PCAFs needed more time. The comparison between both approaches is less obvious for slightly larger instances, K16, as shown in Figure 5b. Indeed, the 15/30 instances were almost instantly solved with the classical CAF approach by outperforming the PCAFs one, but for the other 15/30 instances the classical CAF approach reached the timeout while the PCAFs one has obtained a solution. On larger instances, the added value of our PCAFs approach is more obvious. For K49 (Figure 5d), all the instances reached the timeout with the classical CAF approach, while have been solved (except one) with the PCAFs one. Moreover the majority of them in a relatively short time. The situation is almost the same for K25 (Figure 5c); only two instances, were surprisingly easily solved with the CAF approach and took hundreds of seconds with the PCAF approach.



(a) **K**9 (b) **K**16 (c) **K**25 (d) **K**49

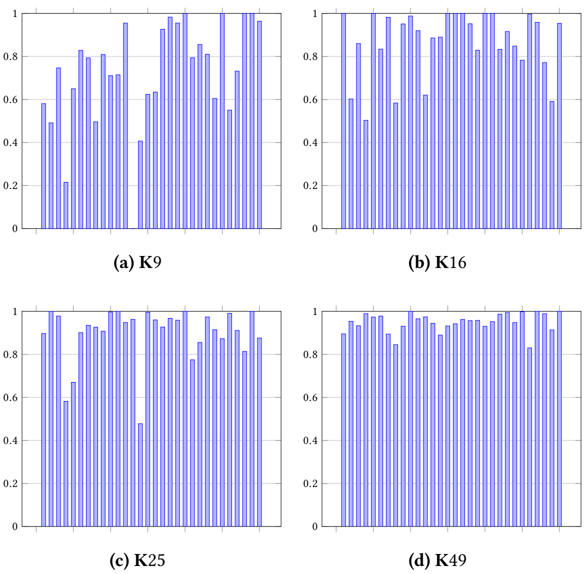
Figure 5: Runtime Comparison for Kleinberg Instances

Then we look at the results regarding controllability and controlling power. Figure 6a and 6b show the controlling power for the Barabási-Albert instances solved with our PCAFs approach. All the corresponding CAFs of these instances, either they were not controllable with the classical CAFs approach or this approach reached the timeout. However, we observe that for most of these instances we have a high controlling power *i.e.* up to 1, which means our PCAFs approach allows to accept the target with a high probability, even when the classical CAFs approach cannot. The results for the Kleinberg instances (Figure 7) are similar.



(a) **BA**50 (b) **BA**60

Figure 6: Controlling Power for Barabási-Albert Instances



(a) K9 (b) K16 (c) K25 (d) K49

Figure 7: Controlling Power for the Kleinberg Instances

7 RELATED WORK AND CONCLUSIONS

Let us briefly discuss some existing work on probabilistic argumentation. Among the different approaches, the closest one is the so-called *constellation approach*, where probabilities attached to arguments/attacks are interpreted as the probability that these elements actually appear in the system. This is the intuition behind the probabilities that we use in PCAFs. In [25], the authors are basing their work on a regular argumentation framework, and consider that both, arguments and attacks are uncertain. In their study, they make the assumption of independence of probabilities. There is no concept similar to controllability that is considered in this work; the problem studied in this paper consists in finding the probability that a given set of arguments is accepted. [20] considers a global probability distribution, removing the independence assumption, but he only considers attacks as being uncertain. And again, the goal is to compute the probability that a given set of arguments is acceptable, not to modify the system in order to get some (set of) argument(s) accepted. The *epistemic approach* for probabilistic argumentation [21, 22, 33] attaches to arguments probabilities that express a degree of belief. Contrary to the constellation approach (and our work), it does not express the probability that the argument appears in the system, but the probability (according to agent) that the argument is acceptable. [19] proposes a probability distribution over models of the arguments representation language, which can then be used to give a probability distribution over arguments that are constructed using classical logic. To that way there is a simple and clear relationship between belief in the premises of an argument, and the belief in the argument. [24] are proposing a Bayesian approach to argument-based reasoning for statistically estimating the existence of an attack between arguments. In [31, 32], probabilities are not attached to arguments or attacks, but to labellings. The authors assume an empirical probability distribution over a set of observed labellings, such that these observed labellings are drawn from a probability distribution of a probabilistic argumentation framework. Based on the observation of labellings the goal is to learn attacks between arguments. The work proposed in [30] concerns structured (*i.e.* rule-based) argumentation frameworks. The authors propose a labelling oriented framework that they call probabilistic labellings. This framework covers uncertainty on inclusion of argumentative pieces as well as uncertainty regarding acceptance of arguments or statements, even in the case all the argumentative pieces are included in the reasoning activity. Finally, let us mention the existing work on "classical" CAFs. Besides the original paper on CAFs [13] and the following work by the same authors on negotiation [12], a detailed complexity study of controllability and implementations of QBF-based and SAT-based CEGAR algorithms have been described in [27]. These works keep the original definition of controllability, which means that there are CAFs that cannot be controlled. On the opposite, [26] defines a weaker form of controllability, based on *some* completion instead of *each* completion. Somehow, this defines credulous reasoning over the set of completions, while classical controllability is skeptical. All these related papers consider qualitative uncertainty, contrary to our approach that takes into account probabilistic information. Our new method allows to identify control configurations that have the highest probability of success, so it is a good compromise between the original controllability that may be a goal hard to reach in some real world situations and the weaker controllability by [26].

In this paper we have proposed an original use of probabilities in abstract argumentation as it is the first time that probabilities are used in the context of argumentation dynamics. More precisely, inspired by the use of CAFs in negotiation dialogues [12], we identified some interesting issues that are arising such as how to choose the most probable completion under some (time or other) constraints or how to decide which subset of the target to satisfy when the whole target cannot be satisfied, and we have presented Probabilistic CAFs (PCAFs) for improving the controllability of classical CAFs. By associating (independent) probabilities with the uncertain part (arguments and attacks) of the framework, we defined the notion of controlling power that represents the probability that a control configuration reaches the target and we have proposed a computational method based on Monte Carlo. Experiments show that our new approach outperforms in most cases the QBF solving method for classical CAFs (w.r.t. runtime) and also allows to identify control configurations with a high probability of success, even when the CAF cannot be controlled with classical techniques. While the new approach is more efficient (regarding runtime) in most cases, we observe that the classical CAFs are faster for small size Kleinberg instances. As future work, we will perform more experiments and analysis in order to understand the reasons this is happening knowing that PCAFs outperform CAFs when the size starts increasing. However intuitively we could say that exact reasoning (CAFs) is relevant for easy instances (the small ones) while approximate reasoning (PCAFs) is relevant for hard instances (the larger ones).

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their helpful comments. They also thank their student Imene Sai for her work on the experimental part of the paper.

REFERENCES

[1] Alan Agresti and Brent A. Coull. 1998. Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician* 52, 2 (1998), 119–126. https://doi.org/10.1080/00031305.1998.10480550

[2] Réka Albert and Albert-László Barabási. 2001. Statistical mechanics of complex networks. *CoRR* cond-mat/0106096 (2001).

[3] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. 2007. A unified and general framework for argumentation-based negotiation. In *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*. 967–974.

[4] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. 2008. Making Decisions through Preference-Based Argumentation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, Gerhard Brewka and Jérôme Lang (Eds.). AAAI Press, 113–123. http://www.aaai.org/Library/KR/2008/kr08-012.php

[5] Leila Amgoud and Henri Prade. 2009. Using arguments for making and explaining decisions. *Artif. Intell.* 173, 3-4 (2009), 413–436.

[6] Ringo Baumann and Gerhard Brewka. 2010. Expanding Argumentation Frameworks: Enforcing and Monotonicity Results. In *Proceedings of the 3rd International Conference on Computational Models of Argument (COMMA'10)*. 75–86.

[7] Dorothea Baumeister, Daniel Neugebauer, Jörg Rothe, and Hilmar Schadrack. 2018. Verification in incomplete argumentation frameworks. *Artif. Intell.* 264 (2018), 1–26.

[8] Trevor J. M. Bench-Capon, Henry Prakken, and Giovanni Sartor. 2009. Argumentation in Legal Reasoning. In *Argumentation in Artificial Intelligence*. 363–382.

[9] Philippe Besnard and Anthony Hunter. 2008. *Elements of Argumentation.* MIT Press.

[10] Stefano Bistarelli, Lars Kotthoff, Francesco Santini, and Carlo Taticchi. 2020. A First Overview of ICCMA'19. In *Proceedings of the Workshop on Advances In Argumentation In Artificial Intelligence 2020 co-located with the 19th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2020), Online, November 25-26, 2020.* 90–102.

[11] Sylvie Coste-Marquis, Caroline Devred, Sébastien Konieczny, Marie-Christine Lagasquie-Schiex, and Pierre Marquis. 2007. On the merging of Dung's argumentation systems. *Artif. Intell.* 171, 10-15 (2007), 730–753.

[12] Yannis Dimopoulos, Jean-Guy Mailly, and Pavlos Moraitis. 2019. Argumentation-based Negotiation with Incomplete Opponent Profiles. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'19)*. 1252–1260.

[13] Yannis Dimopoulos, Jean-Guy Mailly, and Pavlos Moraitis. 2018. Control Argumentation Frameworks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*. 4678–4685.

[14] Yannis Dimopoulos and Pavlos Moraitis. 2014. Advances in Argumentation-based Negotiation. In *Negotiation and Argumentation in Multi-Agent Systems: Fundamentals, Theories, Systems and Applications*. 82–125.

[15] Sylvie Doutre and Jean-Guy Mailly. 2018. Constraints and changes: A survey of abstract argumentation dynamics. *Argument & Computation* 9, 3 (2018), 223–248.

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

[17] David A. Easley and Jon M. Kleinberg. 2010. *Networks, Crowds, and Markets - Reasoning About a Highly Connected World.* Cambridge University Press.

[18] Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. 2020. Design and results of the Second International Competition on Computational Models of Argumentation. *Artif. Intell.* 279 (2020).

[19] Anthony Hunter. 2013. A probabilistic approach to modelling uncertain logical arguments. *Int. J. Approx. Reason.* 54, 1 (2013), 47–81.

[20] Anthony Hunter. 2014. Probabilistic qualification of attack in abstract argumentation. *Int. J. Approx. Reasoning* 55, 2 (2014), 607–638.

[21] Anthony Hunter and Matthias Thimm. 2016. On Partial Information and Contradictions in Probabilistic Abstract Argumentation. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*. 53–62.

[22] Anthony Hunter and Matthias Thimm. 2017. Probabilistic Reasoning with Abstract Argumentation Frameworks. *J. Artif. Intell. Res.* 59 (2017), 565–611.

[23] Antonis Kakas and Pavlos Moraitis. 2003. Argumentation Based Decision Making for Autonomous Agents. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*. 883–890.

[24] Hiroyuki Kido and Keishi Okamoto. 2017. A Bayesian Approach to Argument-Based Reasoning for Attack Estimation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI'17)*, Carles Sierra (Ed.). 249–255.

[25] Hengfei Li, Nir Oren, and Timothy J. Norman. 2011. Probabilistic Argumentation Frameworks. In *Proceedings of the First International Workshop on Theory and Applications of Formal Argumentation (TAFA'11)*. 1–16.

[26] Jean-Guy Mailly. 2020. Possible Controllability of Control Argumentation Frameworks. In *Proceedings of the Eighth Interntational Conference on Computational Models of Argument (COMMA'20)*. 283–294.

[27] Andreas Niskanen, Daniel Neugebauer, and Matti Järvisalo. 2020. Controllability of Control Argumentation Frameworks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*. 1855–1861.

[28] Henry Prakken. 2009. Models of Persuasion Dialogue. In *Argumentation in Artificial Intelligence*, Guillermo Ricardo Simari and Iyad Rahwan (Eds.). Springer, 281–300. https://doi.org/10.1007/978-0-387-98197-0_14

[29] Tjitze Rienstra, Matthias Thimm, and Nir Oren. 2013. Opponent Models with Uncertainty for Strategic Argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*. 332–338.

[30] Régis Riveret, Pietro Baroni, Yang Gao, Guido Governatori, Antonino Rotolo, and Giovanni Sartor. 2018. A labelling framework for probabilistic argumentation. *Ann. Math. Artif. Intell.* 83, 1 (2018), 21–71. https://doi.org/10.1007/s10472-018-9574-1

[31] Régis Riveret and Guido Governatori. 2016. On Learning Attacks in Probabilistic Abstract Argumentation. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. 653–661.

[32] Régis Riveret, Dimitrios Korkinof, Moez Draief, and Jeremy V. Pitt. 2015. Probabilistic abstract argumentation: an investigation with Boltzmann machines. *Argument & Computation* 6, 2 (2015), 178–218.

[33] Matthias Thimm. 2012. A Probabilistic Semantics for abstract Argumentation. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*. IOS Press, 750–755.

[34] Matthias Thimm and Serena Villata. 2017. The first international competition on computational models of argumentation: Results and analysis. *Artif. Intell.* 252 (2017), 267–294.

[35] Johannes Peter Wallner, Andreas Niskanen, and Matti Järvisalo. 2017. Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation. *J. Artif. Intell. Res.* 60 (2017), 1–40.