

# Computationally Viable Handling of Beliefs in Arguments for Persuasion

Emmanuel Hadoux  
University College London  
London, UK  
e.hadoux@ucl.ac.uk

Anthony Hunter  
University College London  
London, UK  
anthony.hunter@ucl.ac.uk

**Abstract**—Computational models of argument are being developed to capture aspects of how persuasion is undertaken. Recent proposals suggest that in a persuasion dialogue between some agents, it is valuable for each agent to model how arguments are believed by the other agents. Beliefs in arguments can be captured by a joint belief distribution over the arguments and updated as the dialogue progresses. This information can be used by the agent to make more intelligent choices of move in the dialogue. Whilst these proposals indicate the value of modelling the beliefs of other agents, there is a question of the computational viability of using a belief distribution over all the arguments. We address this problem in this paper by presenting how probabilistic independence can be leveraged to split this joint distribution into an equivalent set of distributions of smaller size. Experiments show that updating the belief on the split distribution is more efficient than performing updates on the joint distribution.

## I. INTRODUCTION

Persuasion is an activity that involves one party trying to get another party to do some action or to believe something. It is an important and multifaceted human ability. Obviously, it is vital in occupations such as sales and politics. However, it is also necessary in many other spheres of life. Consider, for example, a doctor persuading a patient to drink less, a road safety expert persuading drivers to not text while driving, or an online safety expert persuading the users of social media sites to not reveal too much personal information.

There are various kinds of interaction surrounding persuasion including asking questions, providing information, and saying things to win favour (*e.g.*, by flattering the other agent, by making small talk or being humorous). Nonetheless, providing appropriate arguments and counterarguments is central to the success of the activity.

Within AI, there is an increasing interest in developing computational models of argument, in part for persuasion. This includes frameworks for constructing arguments and counterarguments from logical knowledgebases and formal criteria for evaluating sets of arguments to determine which are winning arguments. Works on models of dialogical argumentation, *i.e.*, argumentation undertaken by multiple agents through dialogues, are also emerging (for reviews see [1], [2]). For instance, dialogical argumentation can potentially be used for systems to persuade users to change their behaviour (*e.g.*, to eat less, to exercise more, to use less electricity or to vote) [3].

Most proposals for dialogical argumentation focus on protocols (*e.g.*, [4]–[7]). Although strategies are recognized as being important if we want artificial agents to be successful in, for example, persuasion, they have been under-developed (see [8] for a review). Furthermore, recent developments in strategies for dialogical argumentation suggest that probabilistic modelling of the other agents is important (see for example [9]–[14]).

In this paper, we are concerned with a recent proposal for probabilistic modelling of the user in dialogical argumentation for persuasion ([13], [14]). The modelling of beliefs in arguments is based on a belief distribution over the power set of the set of arguments. This knowledge is used to determine which are the optimal moves to make in the dialogue at any point. By having some knowledge of what the other agent believes, more intelligent choices can be made. After each move, the belief is updated using an update function. Whilst the approach in [13] and [14] provides a promising framework for modelling how different kinds of user represent and update their beliefs in arguments, there is a question of computational viability when handling an increasing number of arguments.

We address this question by proposing a method for decomposing a belief distribution, defined over all the arguments, into smaller belief distributions over subsets of arguments. We show that we can use the existing updating methods over these smaller belief distributions and that it is equivalent to updating the original distribution. We provide theoretical and empirical results to demonstrate the viability of our approach.

We proceed as follows: in Section II we review how we represent belief in arguments using a belief distribution. After, we show in Section III how we update the belief in arguments. In Section IV, we introduce the notion of a split distribution as an efficient way of representing a belief distribution. We then propose in Section V the concept of metagraphs, a way to obtain a split distribution. Section VI presents the application of the update methods on a split distribution instead of on the main joint distribution. In order to identify the most efficient representation, Section VII shows how we can rank different split distributions for a belief distribution. Section VIII provides empirical results to indicate the computational viability of our proposal in this paper. Finally, in Section IX we conclude and discuss future work.

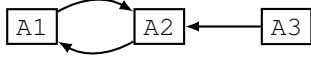


Fig. 1: Argument graph with 3 arguments

## II. ARGUMENTATION PROBLEMS AND PERSUASION

We base our paper on Dung’s abstract argumentation framework [15] where a set of arguments and a set of attacks are represented by a directed graph  $G$ . Each node denotes an argument and an arc from  $A$  to  $B$  denotes  $A$  is a counterargument for  $B$  (in other words,  $A$  attacks  $B$ ).

**Example 1** (Example of argument graph). *Consider arguments  $A_1 =$  “Patient has hypertension so prescribe diuretics”,  $A_2 =$  “Patient has hypertension so prescribe betablockers”, and  $A_3 =$  “Patient has emphysema which is a contraindication for betablockers” in the argument graph depicted in Fig. 1. Here, we assume that  $A_1$  and  $A_2$  attack each other because we should only give one treatment and so giving one precludes the other, and we assume that  $A_3$  attacks  $A_2$  because it provides a counterargument to  $A_2$ .*

We assume our dialogues concern an argument graph  $G = \langle \mathcal{A}, \mathcal{R} \rangle$  with  $\mathcal{A}$  the set of arguments and  $\mathcal{R}$  the set of attack relations. We assume that each argument is known by at least one of the agents in the dialogue, but it is not necessarily the case that each agent knows all the arguments. We use the epistemic approach to probabilistic argumentation ([16]–[19]) to model belief in arguments as defined next.

**Definition 1** (Belief distribution). *Let  $G = \langle \mathcal{A}, \mathcal{R} \rangle$  be an argument graph. A **belief distribution**  $P$  over  $\mathcal{A}$  is such that  $\sum_{X \subseteq \mathcal{A}} P(X) = 1$  and  $P(X) \in [0, 1], \forall X \subseteq \mathcal{A}$ . The **belief in an argument**  $A$  is  $P(A) = \sum_{X \subseteq \mathcal{A} \text{ s.t. } A \in X} P(X)$ .*

A belief distribution  $P$  is a belief distribution such that  $P(A), A \in \mathcal{A}$  is the belief one agent has that  $A$  will be accepted (i.e.,  $P(A) > 0.5$ ) by the agent we are modelling (herself or another agent). Note that this notion of acceptance is different from Dung’s definition in abstract argumentation frameworks.

**Example 2** (Example of a belief distribution). *Let  $\mathcal{A} = \{A, B\}$  where  $P(\{A, B\}) = 1/6$ ,  $P(\{A\}) = 2/3$ , and  $P(\{B\}) = 1/6$  is a belief distribution. Then,  $P(A) = 5/6 > 0.5$  and  $P(B) = 2/6 < 0.5$ . Therefore only  $A$  is accepted.*

An example of an application that we focus on in this paper is a system (the persuader) implemented as an app for persuading a user (the persuadee) to change behaviour in some respect (e.g., take more exercise, eat more fruits). The system can use a belief distribution as a model of the user ([13], [14]), and it can update the model at each stage of the dialogue (see Section III). To do so, it may ask the user about her belief in some arguments that it knows about, and, for the remainder of the arguments, use data about previous similar users for completing the belief.

Recent studies on argumentation in humans illustrate how such distributions can be obtained ([20]–[22]).

We refer to each subset of arguments  $X \subseteq \mathcal{A}$  as a **model**, denoted  $\text{Models}(\mathcal{A})$ . We can represent each model by a binary number by assuming some fixed but arbitrary order over the arguments in  $\mathcal{A}$ . Suppose  $(A_1, \dots, A_n)$  is the ordering, then a binary number  $d_1, \dots, d_n$  is used to denote a model  $X$  as follows:  $A_i$  is in  $X$  iff  $d_i = 1$ . For example, for the ordering  $(A, B, C)$ , the model  $\{A, C\}$  is denoted by 101. In the examples in this paper, we use the alphabet ordering over the letters as the ordering (e.g.,  $(A, B, C)$ ). The binary representation will only be used in the tables of beliefs, for ease of presentation.

We define the *satisfaction operator* where a model  $X$  satisfies an argument  $A$  (denoted  $X \models A$ ) if  $A$  is in  $X$ . Similarly,  $X$  satisfies the negation of an argument  $A$  ( $X \models \neg A$ ) if  $A$  is not in  $X$ .

For a distribution  $P$ , let the domain of  $P$  be obtained by the function  $D(P)$ , and let  $E(P)$  be the set of arguments appearing in the models in  $D(P)$ .

**Example 3** (Example 2 cont’d). *Let  $P(m_1, \dots, m_4)$  be the belief distribution in Example 2 where  $m_1 = \{A, B\} = 11$ ,  $m_2 = \{A\} = 10$ ,  $m_3 = \{B\} = 01$  and  $m_4 = \{\} = 00$ . So  $D(P) = \{\{A, B\}, \{A\}, \{B\}, \{\}\}$  and  $E(P) = \{A, B\}$ .*

Note that an argument can be seen as a random variable in the belief distribution. A random variable  $A$  concerning an argument  $A$  can hold two values:  $P(A = 1)$ , the belief of argument  $A$  being accepted and  $P(A = 0) = 1 - P(A = 1)$  for argument  $A$  not being accepted. This also means that conditional beliefs (e.g.,  $P(A | B)$ ) can be expressed in the same way. Of course, this also works on sets of arguments. For instance, model  $P(m_2 = \{A\})$  from Example 2 can be written  $P(A = 1, B = 0)$ . In this eventuality,  $E(P)$  only concerns the conditioned variables, not the conditioning ones. For instance, let  $P(A, B | \Phi)$  be a belief distribution,  $E(P) = \{A, B\}$  for all possible  $\Phi$ . This notation will only be used in the proofs for ease of explanation.

## III. UPDATE OF A BELIEF DISTRIBUTION

To update a user model (i.e., a belief distribution) during a dialogue, an update method takes an  $\alpha$  where  $\alpha$  is either an argument  $A$  or the negation of an argument  $\neg A$ , and then returns a revised belief distribution. Possibilities for this include probabilistic conditioning. However, in this paper, we review an alternative method from [13] for redistributing belief from models not satisfying  $\alpha$  to models satisfying  $\alpha$ .

**Definition 2** (Refinement function [13]). *Let  $\alpha$  be either an argument  $A$  or the negation of an argument  $\neg A$ ,  $P$  a belief distribution, and  $k \in [0, 1]$  a redistribution coefficient. A **refinement function**, denoted  $H_\alpha^k(P)$ , returns the belief distribution  $P'$  as follows where  $X \in \text{Models}(\mathcal{A})$ :*

$$P'(X) = \begin{cases} P(X) + (k \times P(h_\alpha(X))) & \text{if } X \models \alpha \\ (1 - k) \times P(X) & \text{if } X \not\models \alpha \end{cases}$$

and where  $h_\alpha(X) = X \setminus \{A\}$  when  $\alpha$  is of the form  $A$  and  $h_\alpha(X) = X \cup \{A\}$  when  $\alpha$  is of the form  $\neg A$ .

By convention,  $H_A^k(P) = P$  if  $A \notin E(P)$ .

TABLE I: Examples of Belief Redistribution

$AB$	$P$	$H_A^1(P)$	$H_{\neg A}^1(P)$	$H_A^{0.75}(P)$	$H_B^1(P)$
11	0.6	0.7	0.0	0.675	0.8
10	0.2	0.3	0.0	0.275	0.0
01	0.1	0.0	0.7	0.025	0.2
00	0.1	0.0	0.3	0.025	0.0

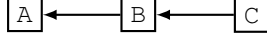


Fig. 2: Example of argument graph

Table I presents four examples of redistribution. In Definition 2,  $h_\alpha$  returns the model closest to  $X$  but with  $\alpha$  no longer satisfied. If  $k = 1$ , all the belief is transferred from the models not satisfying  $\alpha$  to models satisfying  $\alpha$ . If  $k < 1$ , then only a proportion is transferred. This gives flexibility to model update in different kinds of user.

Given a belief distribution  $P$  representing a user's beliefs at the current state of the dialogue, we want to update the model depending on the move made. For this, we introduce the notion of an update method. It generates a belief distribution  $P_i$  from  $P_{i-1}$  based on the move and the argument graph. For example, consider the system asking a user whether it believes an argument  $A$ . If the answer is "yes", the update is  $H_A^1(P_{i-1})$ , otherwise it is  $H_{\neg A}^1(P_{i-1})$ .

Below we give two further simple examples of update methods in order to illustrate the idea where we use  $H_\Phi^k(P_{i-1})$  as an abbreviation for  $H_{A_1}^k(\dots H_{A_n}^k(P_{i-1})\dots)$  when  $\Phi = \{A_1, \dots, A_n\}$ . We refer the reader to [13] for more details on update methods.

The first example is the trusting method which raises the belief in a posit  $A$  (i.e., argument  $A$  has been played), and lowers the belief in attackers and attackees. For step  $i$  in the dialogue, the **trusting method** generates  $P_i = H_{\Phi \cup \{A\}}^1(P_{i-1})$ , where  $\Phi = \{\neg C \mid (A, C) \in \mathcal{R} \text{ or } (C, A) \in \mathcal{R}\}$ .

**Example 4** (Trusting method). Consider the argument graph in Fig. 2. Let the initial belief distribution be  $P_0(011) = 0.3$ ,  $P_0(010) = 0.2$ ,  $P_0(001) = 0.3$ , and  $P_0(000) = 0.2$  where the ordering of arguments is  $(A, B, C)$ . After move  $A$  has been posited,  $P_1(101) = 0.6$ , and  $P_1(100) = 0.4$  with the trusting method.

The second example is the strict method that only allows updating of the belief in the posit when there is no attacker of the posit that is believed (i.e., for all  $(B, A) \in \mathcal{R}$ ,  $P_{i-1}(B) \leq 0.5$ ). For step  $i$  in the dialogue, the **strict method** generates  $P_i = H_{\Phi \cup \{A\}}^1(P_{i-1})$ , where  $\Phi = \{\neg C \mid (A, C) \in \mathcal{R}\}$ .

**Example 5** (Strict method). Consider the graph in Fig. 2 with the strict method. Let the initial belief distribution be  $P_0(111) = 0.2$ ,  $P_0(110) = 0.3$ ,  $P_0(011) = 0.3$ , and  $P_0(010) = 0.2$  where the ordering of arguments is  $(A, B, C)$ . After the first update  $A$ ,  $P_1(111) = 0.2$ ,  $P_1(110) = 0.3$ ,  $P_1(011) = 0.3$ , and  $P_1(010) = 0.2$ . After  $C$ ,  $P_2(101) = 0.5$ , and  $P_2(001) = 0.5$ . After the second update  $A$ ,  $P_3(101) = 1$ .

These are only some of the possible update methods. We

are assuming that the users being modelled are not necessarily rational and that one method can be more suitable than another to model a user.

#### IV. SPLITTING OF A BELIEF DISTRIBUTION

The major drawback of these update methods is that they need to go through the whole joint distribution to modify the beliefs. However, it is often unnecessary to do so when it comes to updating the belief in arguments that are unrelated to each other. Fortunately, we can take advantage of the fact that the belief in an argument may be independent of the beliefs in all the others. Using conditional independence amongst the arguments, we can divide the joint distribution into a set of smaller distributions. We refer to this set as a split. A split is created by performing marginalizations on the main joint distribution.

**Definition 3** (Marginalization). The *marginalization* of a belief distribution  $P$  with respect to arguments  $X \subseteq E(P)$  is denoted  $M(P, X)$  where  $M(P, X) = P'$  and, for all  $X' \subseteq X$ ,

$$P'(X' \mid \Phi) = \sum_{X'' \subseteq E(P) \setminus X} P(X' \cup X'' \mid \Phi)$$

where  $\Phi \subseteq \mathcal{A} \setminus E(P)$ .

We also denote  $M(P, X_1, X_2) = M(M(P, X_1), X_2)$ . Note that in the case of a joint distribution without conditioning variables,  $\Phi = \emptyset$ .

**Example 6** (Example of marginalization). Let  $P$  be defined as follows where  $E(P) = \{A, B\}$ . Let  $X = \{A\}$ . So  $M(P, X) = P'$  where  $P'(\{A\}) = P(11) + P(10)$  and  $P'(\emptyset) = P(01)$ :  $P(11) = 1/6$ ,  $P(10) = 2/3$ ,  $P(01) = 1/6$ .

**Proposition 1.** Marginalization is commutative, i.e.,  $M(P, X_1, X_2) = M(P, X_2, X_1)$ .

*Proof.* Let  $X = (X_1, \dots, X_n)$  be a set of sets of arguments such that  $X_i \subseteq \mathcal{A}$ ,  $\forall i$ . Without loss of generality, the distribution are not conditional. Moreover, let  $n = 2$ ,  $P' = M(P, X_1)$  and  $P'' = M(M(P, X_1), X_2)$ , denoted  $M(P, X_1, X_2)$ . For all  $X' \subseteq E(P'') = X_1 \cap X_2$ ,

$$\begin{aligned} P''(X') &= \sum_{X''' \subseteq E(P') \setminus X_2} P'(X' \cup X''') \\ P''(X') &= \sum_{X''' \subseteq X_1 \setminus X_2} \sum_{X'' \subseteq E(P) \setminus X_1} P(X' \cup X'' \cup X''') \end{aligned}$$

$X'''$  and  $X''$  are always disjoint, therefore,

$$\begin{aligned} P''(X') &= \sum_{X = X''' \cup X'' \subseteq X_1 \setminus X_2 \cup E(P) \setminus X_1} P(X' \cup X) \\ P''(X') &= \sum_{X \subseteq E(P) \setminus (X_1 \cap X_2)} P(X' \cup X) \end{aligned}$$

Thus,  $P'' = M(P, X_1 \cap X_2) = M(P, X_2 \cap X_1) = M(P, X_1, X_2) = M(P, X_2, X_1)$ .

Therefore, the marginalization is commutative.  $\square$

TABLE II: Clean Split Distributions

	(a)			(b)			(c)		
	abc	$P$	$P_1 \oplus P_2$	ab	$P_1$	c	$P_2$		
$m_1$	111	1/4	1/4	$m_{1,1}$	11	1/2	$m_{2,1}$	1	1/2
$m_2$	110	1/4	1/4	$m_{1,2}$	10	1/2	$m_{2,2}$	0	1/2
$m_3$	101	1/4	1/4	$m_{1,3}$	01	0			
$m_4$	100	1/4	1/4	$m_{1,4}$	00	0			

Proposition 1 allows us to deal with a sequence of marginalizations and apply them in any order while always having the same final distribution. Starting from a set of distributions or with the main joint distribution, we can apply a sequence of marginalizations in order to compute a set of smaller distributions, called a *split*.

Note that performing a marginalization on a joint distribution gives a joint distribution. However, most of the time, we deal with conditional distributions when it comes to asking the user about her beliefs about a group of arguments. Fortunately, all the operators on splits are agnostic to the distributions being conditional or not.

**Definition 4** (Split distribution). *Given a belief distribution  $P$ , a **split distribution** is a tuple of distributions  $S = (P_1, \dots, P_k)$  such that:*

- $E(P) = E(P_1) \cup \dots \cup E(P_k)$
- $\forall i, j$  s.t.  $i \neq j, E(P_i) \cap E(P_j) = \emptyset$

A split of an initial distribution separates it into smaller-sized distributions. However, the initial distribution can be computed from the split by taking the product of its subdistributions (called *components*).

This notion of split can be related to Bayesian networks in a way that they represent a joint distribution using smaller, linked, distributions. However, in this work, we manipulate the beliefs in a very different way that probabilities are handled in Bayesian networks. We moreover apply non-standard update methods on the split, *i.e.*, methods like those presented in Section III. We therefore choose not to use Bayesian networks in this work.

**Definition 5** (Combination). *Let  $P_1$  and  $P_2$  be two belief distributions. The **combined distribution** of  $P_1$  and  $P_2$  is denoted  $P_1 \oplus P_2$ . For each  $X_1 \subseteq E(P_1)$  and  $X_2 \subseteq E(P_2)$ ,  $P_1 \oplus P_2(X_1 \cup X_2) = P_1(X_1) \times P_2(X_2)$*

**Example 7** (Example of a split and combined distribution). *Consider  $P$  in the 3<sup>rd</sup> column of Table IIa, with marginals  $P_1$  in Table IIb with  $E(P) = \{A, B\}$  and  $P_2$  in Table IIc with  $E(P_2) = \{C\}$ . So  $P_1 \oplus P_2$  is in the 4<sup>th</sup> column of Table IIa where  $E(P) = \{A, B, C\}$ .*

*For instance,  $P_1 \oplus P_2(m_1) = P_1(m_{1,1}) \times P_2(m_{2,1}) = 1/2 \times 1/2 = 1/4$ . So in this example, we have  $P_1 \oplus P_2 = P$ .*

**Proposition 2.** *If  $S = (P_1, P_2)$  is a split distribution of  $P$ , (or  $(P_1, P_2, P_3)$  is a split distribution of  $P$ ), then:*

- $P_1 \oplus P_2 = P_2 \oplus P_1$ .
- $P_1 \oplus (P_2(X) \oplus P_3(X)) = (P_1 \oplus P_2(X)) \oplus P_3(X)$

TABLE III: Non-Clean Split Distribution

	(a)			(b)			(c)		
	abc	$P$	$P_1 \oplus P_2$	ab	$P_1$	c	$P_2$		
$m_1$	111	0.2	0.24	$m_{1,1}$	11	0.8	$m_{2,1}$	1	0.3
$m_2$	110	0.6	0.56	$m_{1,2}$	10	0	$m_{2,2}$	0	0.7
$m_3$	011	0.1	0.06	$m_{1,3}$	01	0.2			
$m_4$	010	0.1	0.014	$m_{1,4}$	00	0			

The proofs are straightforward given Definition 5.

**Definition 6** (Clean split). *For a split distribution  $S = (P_1, P_2, \dots, P_k)$  of  $P$ ,  $S$  is a **clean split** of  $P$  iff  $((P_1 \oplus P_2) \oplus \dots) \oplus P_k = P$ .*

**Example 8** (Example of a combined distribution for a non-clean split). *Consider  $P$  in 3<sup>rd</sup> column of Table IIIa, with the marginals  $P_1$  in Table IIIb and  $P_2$  in Table IIIc from which we obtain  $P_1 \oplus P_2$  in Table IIIa 4<sup>th</sup> column. So in this example, we have  $P_1 \oplus P_2 \neq P$ .*

The issue whether a split is clean is related to the probabilistic independence between the variables in each component of the split. The split in Table II is clean due to  $C$  being independent of  $A$  and  $B$ , denoted  $C \perp\!\!\!\perp (A, B)$ . On the other hand, in Table III,  $C$  is not independent of  $A$  and  $B$ , leading to a non-clean split. Note that this issue is unrelated to  $A$  being independent of  $B$  or not.

## V. METAGRAPHS

In order to create a split distribution from the main joint distribution, we need to make some assumptions about the dependencies between the arguments in the argument graph based on its structure. This enables us to create a metagraph from which the split distribution is derived.

We first introduce some subsidiary definitions concerning cycles, then the notion of a metagraph and finally we define the independence assumptions needed to build the metagraph.

**Definition 7** (Graph cycles). *Let  $G = \langle \mathcal{A}, \mathcal{R} \rangle$  be an argument graph. Then:*

- a set of arguments  $\{A_1, \dots, A_n\} \subseteq \mathcal{A}$  is a **cycle** in  $G$  iff  $\forall i < n, A_i$  attacks  $A_{i+1}$  and  $A_n$  attacks  $A_1$ ,
- if  $C \subseteq \mathcal{A}$  is a cycle in  $G$ , then argument  $A \in C$  is a **participant** in the cycle  $C$ ,
- the **cycles** of an argument  $A$ , denoted  $\text{Cycles}(A)$ , is the set of cycles in which the argument is a participant defined as follows:

$$\text{Cycles}(A) = \{C \subseteq \mathcal{A} \mid A \in C \text{ and } C \text{ is a cycle in } G\}$$

- a set of cycles  $\{C_1, \dots, C_k\}$  is **complete** for a graph  $G$  iff  $\forall A \in \bigcup_{i=1}^k C_i, \text{Cycles}(A) \subseteq \{C_1, \dots, C_k\}$ ,
- a set of cycles  $T$  is **minimal complete** for a graph  $G$  iff  $T$  is complete for  $G$ , and there is not some  $T' \subset T$  such that  $T'$  is complete for  $G$ .

Each argument in a graph is either in no minimal complete set of cycles, or it is in exactly one minimal complete set of cycles. The latter is shown by the following result.

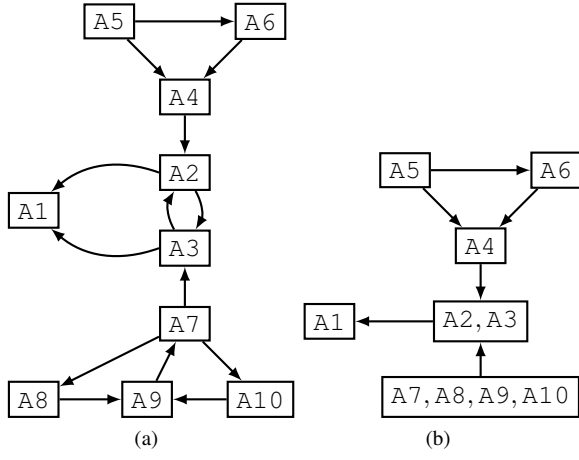


Fig. 3: Argument graph and associated metagraph

**Proposition 3.** Let  $G$  be a graph. If  $T_1$  and  $T_2$  are minimal complete cycles for a graph  $G$ , and  $T_1 \neq T_2$ , then:

$$\left( \bigcup_{C \in T_1} C \right) \cap \left( \bigcup_{C \in T_2} C \right) = \emptyset$$

**Definition 8 (Flock).** Let  $G = \langle \mathcal{A}, \mathcal{R} \rangle$  be an argument graph. A non-empty set of arguments  $F \subseteq \mathcal{A}$  is a **flock** of  $G$  iff:

- 1)  $F = \{A\}$  and  $A \in \mathcal{A}$  is not a participant in any cycle of  $G$ , or;
- 2) there is a minimal complete set of cycles  $T$  of  $G$  and  $F = \bigcup_{C \in T} C$

Let  $\text{Flocks}(G) = \{F \subseteq \mathcal{A} \mid F \text{ is a flock of } G\}$ .

**Proposition 4.** Let  $G = \langle \mathcal{A}, \mathcal{R} \rangle$  be an argument graph.  $\text{Flocks}(G)$  is a partition of  $\mathcal{A}$  (i.e.,  $\forall F_i, F_j \in \text{Flocks}(G), F_i \cap F_j = \emptyset$  and  $\bigcup_{F \in \text{Flocks}(G)} F = \mathcal{A}$ ).

Now we define the metagraph where each node denotes a flock in the original graph and each arc denotes an attack between arguments in different flocks in the original graph.

**Definition 9 (Metagraph).** A **metagraph** for a graph  $G = \langle \mathcal{A}, \mathcal{R} \rangle$ , is defined as a graph  $\text{Meta}(G) = \langle \mathcal{F}, \mathcal{H} \rangle$  where  $\mathcal{F}$  is  $\text{Flocks}(G)$ , and:

$$\mathcal{H} = \{(X_1, X_2) \in \mathcal{F} \times \mathcal{F} \mid \exists (A, B) \in \mathcal{R} \cap (X_1 \times X_2)\}$$

Fig. 3b depicts an example of metagraph.

**Proposition 5.** If  $\text{Meta}(G)$  is a metagraph of  $G$ , then  $\text{Meta}(G)$  is a directed acyclic graph.

A metagraph allows us to organize the arguments in a graph so that we can identify which arguments are involved in cycles together. We specify some intuitive independence assumptions that we use in this paper.

**Definition 10 (Metagraph assumptions).** Let  $\text{Meta}(G)$  be a metagraph of  $G$ . The **metagraph independence assumptions** are defined as follows:

- (Flock dependence)  $\forall F \in \text{Flocks}(G), \forall A \in F, A \perp\!\!\!\perp (F \setminus A)$ ,

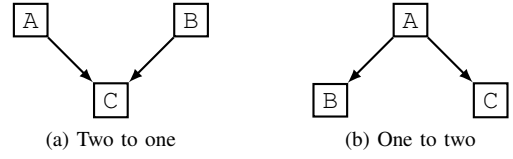


Fig. 4: Structures with three arguments

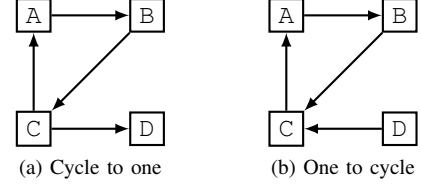


Fig. 5: Cycle structures

- (Inter-flock conditional independence)  $\forall F_1, F_2 \in \text{Flocks}(G) \times \text{Flocks}(G)$  s.t.  $(F_1, F_2) \notin \mathcal{H}$  and  $(F_2, F_1) \notin \mathcal{H}, \forall A_1 \in F_1, A_1 \perp\!\!\!\perp F_2, \forall A_2 \in F_2, A_2 \perp\!\!\!\perp F_1$

In other words, arguments in a flock are dependent. Therefore, we assume we require a joint distribution over all the arguments in a flock. Moreover arguments in different flocks are conditionally independent. Therefore, we assume that the arguments in a flock are independent of all arguments in flocks that are not connected to the flock.

An argument graph is the composition of characteristic smaller structures such as Fig. 4 and 5. Following the previous assumptions, we can create a split distribution for the whole metagraph by combining the splits we propose for each of these characteristic structures.

Fig. 4 shows the two possible three-argument sub-structures of graphs where no cycle is involved. A split associated with Fig. 4a is  $(P(A), P(B \mid A), P(C \mid A, B))$ .

Note that an arc can occur between A and B. However, this structure is only a combination of Fig. 4a and 4b.

Similarly, one split associated with Fig. 4b is  $(P(A), P(B \mid A), P(C \mid A))$ .

When cycles are involved in a split with conditional beliefs, the corresponding component of the split is only conditioned by the arguments of the cycle that are actually in relations. For instance, one split associated with the graph in Fig. 5a is  $(P(A, B, C), P(D \mid C))$ . However, when the arc is connected to a cycle, the whole cycle component of the split is conditioned by the attacking argument. A split associated with the graph in Fig. 5b is  $P_1 = (P(A, B, C \mid D), P(D))$ .

Fig. 3 synthesizes the transformation of the argument graph depicted in Fig. 3a into the metagraph of Fig. 3b. The split associated to this metagraph is  $(P(A5), P(A6 \mid A5), P(A4 \mid A5, A6), P(A2, A3 \mid A4, A7), P(A1 \mid A2, A3), P(A7, A8, A9, A10))$ .

A clean split is a form of modularity in a joint distribution. This means that we can split a distribution, perform updates on the split components, join them back together and obtain the same result as if we modified the main belief distribution in the first place. In other words, rather than updating the joint

distribution, we perform it on the selected parts of the split distribution.

## VI. UPDATE OF A SPLIT DISTRIBUTION

Let  $S = (P_1, P_2)$  be a clean split of belief distribution  $P$ . Suppose  $P_1$  has been updated to  $P'_1$ , i.e., for some  $\mathbb{A} \in \mathcal{A}$ , and some  $k \in [0, 1]$ ,  $P'_1 = H_{\mathbb{A}}^k(P)$ . We can propagate the refinement of  $P'_1$  on  $P_2$ , denoted  $P'_1 \oplus P_2$ .

**Proposition 6.** Let  $\mathbb{A} \in E(P_1)$  and  $S = (P_1, P_2)$  be a clean split of  $P$ :  $H_{\mathbb{A}}^k(P) = H_{\mathbb{A}}^k(P_1) \oplus P_2$

*Proof.* Any  $X \subseteq E(P)$  can be partitioned into two subsets  $X_1 \subseteq E(P_1)$  and  $X_2 \subseteq E(P_2)$  such as  $X_1 \cup X_2 = X$ . Therefore, building upon the refinement function in Definition 2,  $P'(X_1 \cup X_2) =$

$$\begin{cases} P(X_1 \cup X_2) + (kP(h_{\alpha}(X_1 \cup X_2))) & \text{if } X \models \alpha \\ (1-k) \times P(X_1 \cup X_2) & \text{if } X \not\models \alpha \end{cases}$$

$$\begin{cases} P_1(X_1)P_2(X_2) + (kP(h_{\alpha}(X_1 \cup X_2))) & \text{if } X \models \alpha \\ (1-k) \times P_1(X_1)P_2(X_2) & \text{if } X \not\models \alpha \end{cases}$$

$$\begin{cases} P_1(X_1)P_2(X_2) + (kP((X_1 \cup X_2) \setminus \{\mathbb{A}\})) & \text{if } X \models \mathbb{A} \\ P_1(X_1)P_2(X_2) + (kP((X_1 \cup X_2) \cup \{\mathbb{A}\})) & \text{if } X \models \neg \mathbb{A} \\ (1-k) \times P_1 \oplus P_2(X_1 \cup X_2) & \text{otherwise} \end{cases}$$

$$\begin{cases} P_1(X_1)P_2(X_2) + (kP_1(X_1 \setminus \{\mathbb{A}\})P_2(X_2)) & \text{if } X \models \mathbb{A} \\ P_1(X_1)P_2(X_2) + (kP_1(X_1 \cup \{\mathbb{A}\})P_2(X_2)) & \text{if } X \models \neg \mathbb{A} \\ (1-k) \times P_1 \oplus P_2(X_1 \cup X_2) & \text{otherwise} \end{cases}$$

$$\begin{cases} P_1(X_1)P_2(X_2) + (kP_1(h_{\alpha}(X_1))P_2(X_2)) & \text{if } X \models \alpha \\ (1-k) \times P_1 \oplus P_2(X_1 \cup X_2) & \text{if } X \not\models \alpha \end{cases}$$

$$\begin{cases} (P_1(X_1) + (kP_1(h_{\alpha}(X_1))))P_2(X_2) & \text{if } X \models \alpha \\ (1-k) \times P_1 \oplus P_2(X_1 \cup X_2) & \text{if } X \not\models \alpha \quad \square \end{cases}$$

**Corollary 1.** A clean split remains clean after an update.

*Proof.* Let  $S = (P_1, P_2)$  be a clean split of  $P$ . Let  $P' = H_{\alpha}^k(P)$  such as  $\alpha \in E(P_1)$  (and thus  $\alpha \notin E(P_2)$ ). Let  $(P'_1, P'_2)$  be a split of  $P'$  and  $P'_i = H_{\alpha}^k(P_i)$ . This split is clean if  $P'_1 \oplus P'_2 = P' = H_{\alpha}^k(P) = H_{\alpha}^k(P_1 \oplus P_2) = H_{\alpha}^k(P_1) \oplus P_2$ . As  $\alpha \notin E(P_2)$ ,  $H_{\alpha}^k(P_2) = P_2 = P'_2$ . Therefore,  $H_{\alpha}^k(P_1) \oplus P_2 = P'_1 \oplus P'_2$ . The split is still clean.  $\square$

In the light of Proposition 6 and Corollary 1, updating the split distribution is equivalent to updating the joint distribution. However, in the following sections, we show that it is more efficient to update the split distribution.

## VII. RANKING OF SPLITS

For a given joint distribution, there can exist several splits that are equivalent to this distribution. In order to compare them and choose the most efficient one, we need to assign a value to each split. We define the valuation of a split as the sum, for each argument, of the size of the component containing this argument.

**Definition 11** (Valuation of a split). Let  $S = (P_1, \dots, P_n)$  be a split of a distribution  $P$  such that  $E(P) = \mathcal{A}$ . The **value**  $x$  of  $P$  is:

$$x = \sum_{\mathbb{A} \in \mathcal{A}} \sum_{P_i \in S \text{ s.t. } \mathbb{A} \in E(P_i)} |P_i|$$

where  $|P_i|$  means the cardinality of  $D(P_i)$ .

This value is in fact proportional to the amount of time required to update each argument once in sequence using this split distribution.

**Example 9** (Example of valuation). Let  $P_1 = (P(A, B|C), P(C), P(D))$  be a split. The value assigned to this split is  $x = 2 \times 2^3 + 2^1 + 2^1 = 20$ .

This valuation method enables us to have a total order on the splits, which defines a preference relation.

**Definition 12** (Preference on splits). Let  $P$  (resp.  $P'$ ) be a split and  $v$  (resp.  $v'$ ) be its valuation. A split  $P$  is **preferred** to  $P'$ , denoted  $P \succ P'$  iff  $v < v'$ .

**Example 10** (Example of valuation and ranking). Let  $P_1 = (P(A5), P(A6 | A5), P(A4 | A5, A6), P(A2, A3 | A4, A7), P(A1 | A2, A3), P(A7, A8, A9, A10))$  be a split associated with the metagraph in Fig. 3b. Another split can be  $P_2 = (P(A1, A2, A3, A4, A5, A6 | A7), P(A7, A8, A9, A10))$ . Let  $P$  be the joint distribution.

The value  $v$  of  $P$  is  $v = 10 \times 2^{10} = 10240$ , the value  $v'$  of  $P_1$  is  $v' = 2^1 + 2^2 + 2^3 + 2 \times 2^4 + 2^3 + 2^4 = 70$  where the value  $v''$  of  $P_2$  is  $v'' = 6 \times 2^7 + 2^4 = 784$ . We then see that  $P_1 \succ P_2 \succ P$ .

Of course, the improvement in the update computation time is a tradeoff with the space taken to hold the whole split in memory. However, the difference of space is bound. Indeed, the worst case is when each variable is conditioned by all the others. For instance, with a joint distribution  $P = P(A, B, C)$ , the worst split would be  $P_2 = (P(A|B, C), P(B|A, C), P(C|A, B))$ . If the number of arguments is  $n$ , the size of the joint distribution is  $2^n$  and the upper bound size for a split is  $n \times 2^n$ . However, we have  $P_2 \not\succeq P$ , there is thus no reason to choose  $P_2$  over  $P$  as a split.

**Definition 13** (Irreducible split). Let  $P$  be a distribution and  $S = (P_1, \dots, P_k)$  be a clean split of  $P$ . We say the split is **irreducible** iff  $\nexists P_i, P'_i$  and  $P''_i$  such that  $(P'_i, P''_i)$  is a clean split of  $P_i$ .

**Proposition 7.** An irreducible clean split is **unique**.

*Proof.* Let  $P$  be a distribution of beliefs such as  $E(P) = \{A, B, C\}$  without loss of generality. Let  $S_1 = (P(A, B), P(C))$  and  $S_2 = (P(A), P(B, C))$  be two irreducible clean splits of  $P$ . Therefore,

$$P(A, B, C) = P(A, B) \times P(C) = P(A) \times P(B, C)$$

$$P(A, B | C) \times P(C) = P(A | B, C) \times P(B, C)$$

as  $C \perp\!\!\!\perp (A, B)$  in  $S_1$  and  $A \perp\!\!\!\perp (B, C)$  in  $S_2$ .

$$P(A | B, C) \times P(B | C) \times P(C) = P(A | B, C) \times P(B, C)$$

$$P(A | B, C) \times P(B) \times P(C) = P(A | B, C) \times P(B, C)$$

Finally,  $P(B) \times P(C) = P(B, C)$ .

Therefore,  $B \perp\!\!\!\perp C$  and  $S_1$  and  $S_2$  can be reduced to  $S_3 = (P(A), P(B), P(C))$ .  $\square$

TABLE IV: Computation Time for Updates in Different Graphs of 50 Arguments (in ms)

# cycles	# arcs	1 update	50 updates
2 cycles	10 arcs	2ms	107ms
	30 arcs	6ms	236ms
4 cycles	10 arcs	1ms	45ms
	30 arcs	3ms	114ms
10 cycles	10 arcs	0.03ms	1.6ms
	30 arcs	0.06ms	2.5ms

TABLE V: Mean Number of Arguments per Cycle for the Bigger and Smaller Cycles

# cycles	Mean size bigger cycle	Mean size smaller cycle
2 cycles	15 arguments	7 arguments
4 cycles	12 arguments	3 arguments
10 cycles	8 arguments	2 arguments

**Corollary 2.** *A split  $P$  is irreducible iff there is no  $P'$  such that  $P'$  is a split and  $P' \succ P$ .*

## VIII. EXPERIMENTS

We have conducted two experiments using different graphs, joint distributions and splits.<sup>1</sup> The results of the first experiment are presented in Tables IV and V. Table VI contains the results for the second experiment.

The aim of the first experiment is to show the computational efficiency of a split maximizing the number of components while minimizing the size of each component. The experiment consists of three steps performed as follows:

- 1) Starting from a set of 50 arguments, we randomly draw sets of arguments (2, 4 or 10 sets) representing cycles in an argument graph and therefore flocks in the metagraph. Increasing the number of cycles amounts to an increase in the number of components of the split.
- 2) We then randomly connect the different flocks to have an actual instance of metagraph (recall that an argument that is not involved in a cycle is its own flock).
- 3) We finally perform either 1 or 50 updates on each instance and compute the time taken to perform the update only.

The results are averaged on 1000 different metagraphs, with different number of arguments in cycles and different set of connections between the flocks. Table V shows the mean number of arguments in the smaller-sized and bigger-sized cycles for each number of cycles. We moreover average the update part on 100 runs to ensure that both big and small components are updated. Indeed, updating a bigger-sized component takes more time and should occur more often as it concerns more arguments.

Table IV shows the computation time of 1 and 50 updates on argument graphs of 50 arguments. As we can see, performing

<sup>1</sup>The code for the experiments can be found at <https://gist.github.com/EHadox/5a07463086d7dd52b50c29ae79923b3d>. A new version of the library is being developed and can be found at <https://github.com/ComputationalPersuasion/splittercell>.

TABLE VI: Computation Time for 20 Updates (in ns)

# args	Time for 20 updates
25	497ns
50	517ns
75	519ns
100	533ns

more updates takes proportionally more time. Increasing the number of connections increases the size of each component of a split due to more conditioning variables, increasing the computation time. Interestingly increasing the number of cycles (thus the number of small-sized components) does decrease the computation time.

It is very important to note that those results only concern the time taken by the update part. Combining a split into the main joint distribution after the update costs more by several orders of magnitude than updating the main joint distribution in the first place. However, this combination is only required when computing the belief of all the arguments. If we only need a subset of those arguments (*e.g.*, the goal arguments only), we do not need to combine the split into the whole joint distribution.

As a second experiment, we fix the number of cycles, the maximum number of arguments in each, the number of updates and the ratio of connections. The aim of this experiment is to show how updating a split scales with respect to the number of arguments in the graph. For each number of arguments, a partition of the set of arguments is made such that each subset contains at most 15 arguments. The graph is connected with a factor of 1/2, *i.e.*, for 50 arguments, for instance, the graph contains 25 arcs. The experiment is averaged on 1000 instances, with different sets of arcs. Likewise, we have performed 1000 sets of 20 updates for averaging. Table VI presents the results of the computation of 20 updates with different numbers of arguments. Interestingly, these results show that, when the joint distribution has been substantially split (as here where the maximum cycle size is 15 arguments), the computation time is only increased by 4% when the total number of arguments doubles.

## IX. CONCLUSION AND DISCUSSION

The results in this paper show the viability of using a belief distribution as a way to model the user in computational persuasion. By using the assumptions proposed for the metagraphs, we can split a belief distribution. Furthermore, we have shown both theoretically and empirically how we can efficiently and effectively update the smaller distributions while conserving the exact value.

This paper is motivated by the need to use probabilistic models of the user in dialogical argumentation for tasks including persuasion. Furthermore, this paper builds on a recent proposal for persuasion dialogues with probabilistic user models using [13], [14].

Further proposals for using probability theory in dialogical argumentation include: a probabilistic model of the opponent is used by an agent to select moves based on what it believes

the other agent is aware of [10]; the history of previous dialogues is used to predict the arguments that an opponent might put forward [9]; a probabilistic finite state machine can represent the possible moves that each agent can make in each state [23], generalized to POMDPs when there is uncertainty about what an opponent is aware of [12]. However, none of these use the beliefs by the persuadee in the arguments. In [11], a probabilistic model of beliefs of the persuadee is used by the persuader to choose beliefs to present, but there is no consideration of update of the model resulting from dialogue.

The results reported in this paper have been undertaken as part of a project to develop a framework for computational persuasion based on computational models of argument for use in technology for behaviour change<sup>2</sup>. In future work, we will implement and evaluate systems for behaviour change that undertake argumentation for persuasion with users. This will incorporate user modelling of belief in arguments based on the proposal in this paper.

#### ACKNOWLEDGMENT

This research was partly funded by EPSRC grant EP/N008294/1 for the Framework for Computational Persuasion project. The authors would like to thank Sylwia Polberg for her valuable comments.

#### REFERENCES

- [1] P. Besnard and A. Hunter, *Elements of Argumentation*. MIT Press, 2008.
- [2] I. Rahwan and G. Simari, *Argumentation in Artificial Intelligence*. Springer, 2009.
- [3] A. Hunter, "Opportunities for argument-centric persuasion in behaviour change," in *Proceeding of the 14th European Conference on Logics in Artificial Intelligence (JELIA)*, ser. LNCS, vol. 8761. Springer, 2014, pp. 48–61.
- [4] H. Prakken, "Coherence and flexibility in dialogue games for argumentation," *Journal of Logic and Computation*, vol. 15, no. 6, pp. 1009–1040, 2005.
- [5] —, "Formal systems for persuasion dialogue," *Knowledge Engineering Review*, vol. 21, no. 2, pp. 163–188, 2006.
- [6] X. Fan and F. Toni, "Assumption-based argumentation dialogues," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.
- [7] M. Caminada and M. Podlaszewski, "Grounded semantics as persuasion dialogue," in *Proceedings of the 4th International Conference on Computational Models of Argument (COMMA)*, 2012.
- [8] M. Thimm, "Strategic argumentation in multi-agent systems," *Kunstliche Intelligenz*, 2014.
- [9] C. Hadjiniokolis, Y. Siantos, S. Modgil, E. Black, and P. McBurney, "Opponent modelling in persuasion dialogues," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013, pp. 164–170.
- [10] T. Rienstra, M. Thimm, and N. Oren, "Opponent models with uncertainty for strategic argumentation," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.
- [11] E. Black, A. Coles, and S. Bernardini, "Automated planning of simple persuasion dialogues," in *Proceedings of the 15th International Workshop Computational Logic in Multi-agent Systems (CLIMA)*, ser. LNCS, vol. 8624. Springer, 2014, pp. 87–104.
- [12] E. Hadoux, A. Beynier, N. Maudet, P. Weng, and A. Hunter, "Optimization of probabilistic argumentation with Markov decision models," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
- [13] A. Hunter, "Modelling the persuadee in asymmetric argumentation dialogues for persuasion," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.
- [14] —, "Two dimensional uncertainty in persuadee modelling in argumentation," in *Proceedings of 22nd European Conference on Artificial Intelligence (ECAI)*, 2016.
- [15] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.
- [16] M. Thimm, "A probabilistic semantics for abstract argumentation," in *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, ser. Frontiers in Artificial Intelligence and Applications, vol. 242. IOS Press, 2012, pp. 750–755.
- [17] A. Hunter, "A probabilistic approach to modelling uncertain logical arguments," *International Journal of Approximate Reasoning*, vol. 54, no. 1, pp. 47–81, 2013.
- [18] A. Hunter and M. Thimm, "Probabilistic argumentation with incomplete information," in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 2014.
- [19] P. Baroni, M. Giacomin, and P. Vicig, "On rationality conditions for epistemic probabilities in abstract argumentation," in *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA)*, 2014.
- [20] I. Rahwan, M. I. Madakkatel, J.-F. Bonnefon, R. N. Awan, and S. Abdallah, "Behavioural experiments for assessing the abstract argumentation semantics of reinstatement," *Cognitive Science*, vol. 34, no. 8, pp. 1483–1502, 2010.
- [21] A. Rosenfeld and S. Kraus, "Providing arguments in discussions based on the prediction of human argumentative behavior," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.
- [22] F. Cerutti, N. Tintarev, and N. Oren, "Formal arguments, preferences, and natural language interfaces to humans: an empirical evaluation," in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 2014.
- [23] A. Hunter, "Probabilistic strategies in dialogical argumentation," in *Proceedings of the 8th International Conference on Scalable Uncertainty Management (SUM)*, ser. LNCS, vol. 8720. Springer, 2014, pp. 190–202.

<sup>2</sup>www.computationalpersuasion.com