

Optimization of Probabilistic Argumentation with Markov Decision Models

Emmanuel Hadoux, Aurélie Beynier, Nicolas Maudet

Sorbonne Universités, UPMC Univ Paris 06 /
CNRS, UMR 7606, LIP6, F-75005, Paris, France

Paul Weng

SYSU-CMU Joint Institute of Engineering, Guangzhou, China
SYSU-CMU Shunde International Joint Research Institute, Shunde, China

Anthony Hunter

Department of Computer Science
University College London, London, UK

Abstract

One prominent way to deal with conflicting viewpoints among agents is to conduct an argumentative debate: by exchanging arguments, agents can seek to persuade each other. In this paper we investigate the problem, for an agent, of optimizing a sequence of moves to be put forward in a debate, against an opponent assumed to behave stochastically, and equipped with an unknown initial belief state. Despite the prohibitive number of states induced by a naive mapping to Markov models, we show that exploiting several features of such interaction settings allows for optimal resolution in practice, in particular: (1) as debates take place in a public space (or common ground), they can readily be modelled as *Mixed Observability Markov Decision Processes*, (2) as argumentation problems are highly structured, one can design optimization techniques to prune the initial instance. We report on the experimental evaluation of these techniques.

1 Introduction

Argumentation is by essence a dialectical process, which involves different parties exchanging pieces of information. In a persuasion dialogue, agents have conflicting goals and try to convince each other, but more generally, agents can have various goals when engaged in an argumentative interaction. Dialectics have been investigated in formal argumentation, but mostly as a mean to provide a proof-theoretical counterpart to argumentation semantics [Modgil and Caminada, 2009], leaving no room for proper strategies. When autonomous agents do interact, however, they will typically fail to have winning strategies or act fully rationally, sometimes hide arguments that they know to be true, etc.

In a recent survey, Thimm [2014] provide an overview of the state-of-the-art about strategic argumentation in multi-

agent systems. A key problem is designing for an agent strategies of argumentation (*i.e.*, which arguments to put forward in the course of the dialogue).

As described in Thimm and Garcia’s classification [2010], a key element to consider is the *awareness* of agents. Two extremes of the spectrum are when agents are *fully ignorant*, *i.e.*, they just know their own arguments; or *omniscient*, *i.e.*, they know arguments (and strategies) that opponents have at their disposal. In the former case the agent will typically have to rely on heuristic approaches (*e.g.*, [Kontarinis *et al.*, 2014]). While this may prove efficient in practice, it is in general very difficult to offer any guarantee on the outcome. In the case of omniscient agents, one can use game-theoretic approaches, like backward induction. However, the strong required assumptions are problematic.

Of course, one can opt for an arguably more realistic, intermediate, modelling. In Rienstra *et al.*’s work [2013] for instance, a setting with an uncertain opponent model is proposed. In Hadjinikolis *et al.*’s work [2013], the opponent modelling is updated through the information exchanged during the dialogue. In this paper we also take such an intermediate approach. We follow Hunter’s recent proposition [2014] and suppose that the behaviour of agents is stochastic. Specifically, we assume that, given a certain state of the debate, it is known, probabilistically, how the opponent may react. These probabilities may have been obtained by expert knowledge, or by observation of previous interactions with the same agent (or at least, type of agent), *e.g.*, a vendor may be able to predict from past interactions the possible counter-arguments that could be put forward by a skeptical consumer.

In particular, our approach does *not* assume that the opponent will play optimally, and does not in general suppose knowledge of the initial state of the opponent. This stands in sharp contrast with approaches optimizing against a supposedly optimal agent, which can rely on backward induction or similar techniques. We will see that it is possible to obtain optimal policies in such a setting, despite this uncertainty which induces a huge potential state space. Our objective is to ex-

plore to what extent optimal resolution is feasible.

The remainder of this paper is organized as follows. Sections 2 and 3 provide the necessary background on argumentation problems with probabilistic strategies (APS). Section 4 presents different ways to account for APS with Markov models, concluding on a mapping from APS to mixed observability MDPs. Section 5 investigates different optimization techniques, which as observed in Section 6, may have dramatic effects on the practical efficiency of resolution.

2 Dialogical argumentation problem

A dialogical argumentation problem is a turn-based game between two agents. Dialogue games have been largely studied to characterize argumentation systems [Prakken, 2006]. A dialogical argumentation problem is defined by a set \mathcal{A} of arguments and a set of attacks $\mathcal{E} = \{e(x, y) \text{ s.t. } (x, y) \in \mathcal{A}^2\}$ where $e(x, y)$ means x attacks y . During her turn, an agent can fire a rule to add arguments to the debate, to attack present arguments or to revise her knowledge.

We use a state-based model to represent the execution state of the model [Black and Hunter, 2012] and a logic-based formulation of states. For convenience, for a predicate p and a set $X \subseteq \mathcal{A}$, $p(X)$ denotes the conjunction of this predicate applied on each element of the set if the result is unambiguous, e.g., $p(X) = \bigwedge_{x \in X} p(x)$, and $p\{X\}$ represents the set $\{p(x) | x \in X\}$. Thus, $2^{p\{X\}}$ is the set of all subsets of $\{p(x) | x \in X\}$.

A public state in space $\mathcal{P} = 2^{a\{\mathcal{A}\}} \times 2^{\mathcal{E}}$ gathers used arguments and attacks, where $a(x)$ means argument x has been put forward by some agent. The public state can be observed by both agents. Each agent i also maintains an internal private state in space $\mathcal{S}_i = 2^{h_i\{\mathcal{A}\}}$ representing the arguments she knows, where $h_i(x)$ means argument x is (privately) known by agent i . This internal state is only observable by the agent herself and may evolve through the debate if the agent revises her knowledge.

We define an accepted argument following Dung's dialectical semantics in its grounded form [Dung, 1995], i.e., an accepted argument is either not attacked or defended. Note that other semantics can be applied to define the acceptance of an argument [Dung, 1995].

The possible moves an agent i can make are defined by a set of rules r_j 's of the form $r_j : \text{prem}_j \Rightarrow \text{act}_j$. Premise prem_j is a conjunction of a , h_i and e predicates (or their negations) applied on one or more arguments. A rule can only be fired by an agent i if its premise is fulfilled. Act act_j is a set of modifications on predicates of the public space and private state of agent i . The possible modifications are denoted:

$\boxplus(p)/\boxminus(p)$ to add/remove p to/from the public space, where p is either $a(x)$ or $e(x, y)$ for $(x, y) \in \mathcal{A}^2$.

$\oplus(h_i(x))/\ominus(h_i(x))$ to add/remove predicate $h_i(x)$ to/from the private state, for $x \in \mathcal{A}$ and agent i .

Note that the agents are *focused*, i.e., they cannot decide not to play if at least one rule can be fired. The rules can be seen as a protocol for the agent, defining her possible behavior in each state of the problem.

3 Probabilistic modelling of a dialogue

In order to improve the modelling of the opponent behavior in dialogical argumentation, Hunter [2014] has recently proposed to introduce probabilities while representing argumentation moves in a framework which we called *Argumentation Problems with Probabilistic Strategies* (APS).

For a set X , $\text{Pr}(X)$ denotes the set of probability distributions over X and $\Pi = [\pi_1/x_1, \pi_2/x_2, \dots, \pi_n/x_n]$ denotes an element of $\text{Pr}(X)$, where the probability for Π of getting $x_j \in X$ is π_j . A probabilistic rule r in an APS is then defined as: $r : \text{prem} \Rightarrow \text{Pr}(\text{Acts})$ where Acts denotes the set of all possible acts. Distinct acts, i.e., set of modifications, are then possible when applying a probabilistic rule. We denote r_j^i the j -th rule of agent i and $r_{j,k}^i$ is the k -th act of rule r_j^i , i.e., $r_{j,k}^i = \text{act}_k$ if $r_j^i = [\pi_1/\text{act}_1, \pi_2/\text{act}_2, \dots, \pi_n/\text{act}_n]$. For a given rule r , we denote $\text{prem}(r)$ (resp. $\text{acts}(r)$) the premise (resp. the set of acts of positive probability) of r .

This representation allows for specifying probabilistic argumentation protocols. While this framework is descriptive, it does not tackle the issue of optimizing the sequence of moves of the agents. In this paper, we propose to optimize the argumentation strategy of one agent facing a probabilistic opponent playing by the probabilistic rules of the APS.

To characterize the possible desired argumentation outcomes, each agent i has a goal state g_i which is a conjunction of $g(x)$ or $g(\neg x)$ where each x is an argument and $g(x)$ (resp. $g(\neg x)$) means that x is (resp. is not) accepted in the public state. Although the agents are considered as selfish, individual goals might not be antagonistic. Indeed, in some cases, the public state may satisfy both goals. In those situations, both agents are then considered as winners. In order to model realistic argumentation games, the goal of an agent is assumed to be private information and cannot be observed by the other agent. An agent that optimizes her moves does so with this limited knowledge about the opponent.

An APS is characterized by the tuple $\langle \mathcal{A}, \mathcal{E}, \mathcal{S}_1, \mathcal{S}_2, g_1, g_2, \mathcal{P}, \mathcal{R}_1, \mathcal{R}_2 \rangle$ with:

- \mathcal{A}, \mathcal{E} and \mathcal{P} , as previously defined,
- \mathcal{S}_i the internal states of agent i ,
- g_i the goal of agent i ,
- $\mathcal{R}_i = \{r : \text{prem} \Rightarrow \text{Pr}(\text{Acts})\} \cup \{\emptyset \Rightarrow \emptyset\}$, the set of probabilistic rules.

The empty rule $\emptyset \Rightarrow \emptyset$ permits to skip the turn of an agent having no rule that can be fired this turn. This rule is fired if and only if no other rule can be. Note there are $|\mathcal{S}_i| = 2^{|\mathcal{A}|}$ possible private states, and $3^{|\mathcal{A}|}$ possible goal states.

Example 1. Consider a concrete dialogical argumentation problem. A famous debate in the gamer community is whether e-sport is a sport. The arguments are as follows: (a) e-sport is a sport, (b) e-sport requires focusing and generates tiredness, (c) not all sports are physical, (d) sports not referenced by IOC exist, (e) chess is a sport, (f) e-sport is not a physical activity, (g) e-sport is not referenced by IOC, (h) working requires focusing and generates tiredness but is not a sport.

Assume that agent 1 wants to persuade that e-sport is a sport. This example can be formalized by an APS as follows:

- $\mathcal{A} = \{a, b, c, d, e, f, g, h\}$
- $\mathcal{E} = \{e(f, a), e(g, a), e(b, f), e(c, f), e(h, b), e(g, c), e(d, g), e(e, g)\}$
- $g_1 = g(a)$

Assume the following rules formalize the agents' behaviors:

- $\mathcal{R}_1 = \{h_1(a) \Rightarrow \boxplus a(a),$
 $h_1(b) \wedge a(f) \wedge h_1(c) \wedge e(b, f) \wedge e(c, f) \Rightarrow$
 $0.5 / \boxplus a(b) \wedge \boxplus e(b, f) \vee 0.5 / \boxplus a(c) \wedge \boxplus e(c, f),$
 $h_1(d) \wedge a(g) \wedge h_1(e) \wedge e(d, g) \wedge e(e, g) \Rightarrow$
 $0.8 / \boxplus a(e) \wedge \boxplus e(e, g) \vee 0.2 / \boxplus a(d) \wedge \boxplus e(d, g)\}$
- $\mathcal{R}_2 = \{h_2(h) \wedge a(b) \wedge e(h, b) \Rightarrow \boxplus a(h) \wedge \boxplus e(h, b),$
 $h_2(g) \wedge a(c) \wedge e(g, c) \Rightarrow \boxplus a(g) \wedge \boxplus e(g, c),$
 $a(a) \wedge h_2(f) \wedge h_2(g) \wedge e(f, a) \Rightarrow$
 $0.8 / \boxplus a(f) \wedge \boxplus e(f, a) \vee 0.2 / \boxplus a(g) \wedge \boxplus e(g, a)\}$

g_2 is unknown to agent 1. There are $3^{|\mathcal{A}|} = 6561$ possible goal states. The sizes of the state spaces are: $|\mathcal{S}_1| = |\mathcal{S}_2| = 256$, $|\mathcal{P}| = 65536$.

The initial state $(s_1, p, s_2) \in \mathcal{S}_1 \times \mathcal{P} \times \mathcal{S}_2$ of this problem is assumed to be: $(\{h_1(a, b, c, d, e)\}, \{\}, \{h_2(f, g, h)\})$.

From Example 1, we can build the graph of arguments and attacks of Figure 1. Each argument is represented by a vertex and each edge formalizes an attack. Bold face arguments are used by agent 1 while the others are used by the opponent.

All the possible sequences of states can be represented as a *Probabilistic Finite State Machine* (PFSM) [Hunter, 2014]. For instance, starting from the initial state given in Example 1, the sequence of rules $(r_{1,1}^1, r_{3,2}^2, r_{3,1}^1)$, alternatively for agent 1 and agent 2, leads the environment to the state $\{a(a), a(g), e(g, a), a(e), e(e, g)\}$. This is a winning state for agent 1 as $a(a)$ is true, is attacked but also defended. $a(a)$ is therefore accepted.

In order to compute an optimal policy for agent 1, one can use dynamic programming methods on the PFSM in order to backtrack the policy from the winning state, but this requires to know the internal state of the opponent. Indeed, in order to know which rules the opponent is able to fire we need to either know the internal state or build a PFSM for each possible internal state. In order to leverage this assumption we propose to use Markov models to represent and solve the problem.

4 From APS to MOMDPs

An APS allows for describing the argumentation protocols and the probabilistic behavior of an opponent. In this section, we show that the problem of optimizing the sequence of

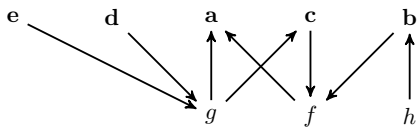


Figure 1: Graph of arguments of Example 1

moves for one agent (against an opponent assumed to behave stochastically, and equipped with an unknown initial belief state) can be formalized as a Mixed Observable Markov Decision Process (MOMDP) defined from the APS.

4.1 Markov Models

Markov Decision Processes (MDP) provide a general mathematical framework to model sequential decision making problems under uncertainty [Bellman, 1957]. An MDP is characterized by the tuple $\langle S, A, T, R \rangle$ with:

- S and A , the sets of states and actions,
- $T : S \times A \rightarrow \text{Pr}(S)$ the transition function specifying the probability of transitioning to a state s' while executing an action a from s ,
- $R : S \times A \rightarrow \mathbb{R}$ the reward function formalizing the preferences on the states and actions.

When the states are no longer observable but the decision-maker has partial information about the state of the system, one can rely on the *Partially Observable Markov Decision Process* (POMDP) model [Puterman, 1994]. In this model, after each action, instead of receiving the new state of the problem, the agent receives an observation about this state. A POMDP is characterized by the tuple $\langle S, A, T, R, O, \Omega \rangle$ with O and Ω being respectively the set of observations and the observation function $S \times A \rightarrow \text{Pr}(O)$.

The strategy of a POMDP can be represented compactly as a *policy graph*, which is a deterministic finite automaton to follow depending on the observation received after each decision step. An optimal policy maximizes the expected discounted sum of rewards.

Although POMDPs describe a powerful mathematical framework, they suffer from a high computational complexity [Papadimitriou and Tsitsiklis, 1987]. In various settings, some components of the state are fully observable while the rest of the state is partially observable. *Mixed Observability Markov Decision Processes* (MOMDP) [Ong et al., 2010] have been proposed to account for such problems. MOMDPs exploit the mixed-observability property thus leading to a higher computational efficiency.

An MOMDP, characterized by the tuple $\langle S_v, S_h, A, O_v, O_h, T, \Omega, R \rangle$ is a structured POMDP $\langle S, A, T, R, O, \Omega \rangle$ where $S = S_v \times S_h$ and $O = O_v \times O_h$.

4.2 Conversion of an APS to an MOMDP

In this paper, we argue that the decision problem of an agent in an APS can be formalized as an MOMDP. We adopt the point of view of agent 1 in the argumentation problem. At each decision step, the agent must decide for the best argumentation move while anticipating the opponent moves and the possible future states of the debate. The next state of the debate is uncertain since the opponent's behavior is non-deterministic. Moreover, the agent observes her own private state and the public state but does not observe the opponent's private state. This setting complies with the definition of states and observations in MOMDPs.

In order to optimize the argumentation strategy of agent 1, we transform the APS into an MOMDP. As the purpose

of this work is to optimize agent 1’s decisions, it is obvious that she has to choose which alternative of a rule to apply instead of being guided by a probability distribution. The possible (deterministic) actions of agent 1 are then defined by splitting each alternative of the rules of agent 1 defined in the original APS, into separate deterministic rules (or actions). The resulting MOMDP is defined as follows:

- $S_v = \mathcal{S}_1 \times \mathcal{P}$, $S_h = \mathcal{S}_2$,
- $A = \{\text{prem}(r) \Rightarrow m \mid r \in \mathcal{R}_1 \text{ and } m \in \text{acts}(r)\}$. This set is obtained by decomposing each act m of positive probability of each probabilistic rule r in \mathcal{R}_1 .
- $\Omega(\langle s_v, s_h \rangle, a, \langle s'_v \rangle) = 1$ if $s_v = s'_v$, otherwise 0,
- T, O_v, O_h and R are defined as below.

To specify the transition function on states, we first need to introduce the notion of *application set*.

Definition 1. Let $C_s(\mathcal{R}_i)$ be the set of rules of \mathcal{R}_i that can be fired in state s . The application set $F_r(m, s)$ is the set of predicates resulting from the application of act m of a rule r on s . If r cannot be fired in s , $F_r(m, s) = s$.

In the following, we add a subscript to the hidden part of the state to distinguish between hidden information of agent 1 and hidden information of agent 2.

Example 2. (Example 1 continued) Let $s = \{a(\mathbf{b}), h_2(\mathbf{h}), h_2(\mathbf{g})\}$, therefore, $C_s(\mathcal{R}_2) = \{r_1^2\}$ with r_1^2 being the first rule of \mathcal{R}_2 . Let m_1 and m_2 be respectively the modifications of r_1^2 and r_2^2 (with r_1^2 and $r_2^2 \in \mathcal{R}_2$). $F_{r_1^2}(m_1, s) = \{a(\mathbf{b}), a(\mathbf{h}), e(\mathbf{h}, \mathbf{b}), h_2(\mathbf{h}), h_2(\mathbf{g})\}$ as $r_1^2 \in C_s(\mathcal{R}_2)$ and $F_{r_2^2}(m_2, s) = s$ as $r_2^2 \notin C_s(\mathcal{R}_2)$.

Let $r : p \Rightarrow m$ be a rule/action in A , $s' = F_r(m, s)$ the state resulting from the application of m on state s , $r' \in C_{s'}(\mathcal{R}_2)$ a rule in the set of rules of agent 2 that can be fired in s' such that $r' : p' \Rightarrow [\pi_1/m_1, \dots, \pi_n/m_n]$ and $F_{r'}(m_i, s') = s'_i$. The function T can then be defined as $T(s, r, s'_i) = \pi_i$.

Note that the rule fired by agent 2 at each step is merged into the transition function. In fact, we focus on the first agent’s strategy and the probabilistic behavior of the second agent is formalized by the transition probabilities.

In the MOMDP formalization of an APS, O_v and O_h can be omitted. Indeed, there is no observation on the hidden part that is not already in the visible part of the state. Hence $O_v = S_v$ and $O_h = \emptyset$.

The reward function is defined as follows: each action that does not reach a goal state needs to have a strictly negative value (i.e., a positive cost). If the goal is reached, the reward needs to be positive. That way, the policy favors shorter argument sequences reaching the goal. The notion of goal can be extended to account for partially reached goals. For instance, if the goal of the agent is to have $g(\mathbf{a})$ and $g(\mathbf{b})$ but, only $g(\mathbf{a})$ is reached, a part of the reward could be obtained. More generally, if using another semantic for the acceptance of the arguments, the reward can be modulated depending on the value of the accepted arguments in the goal.

After conversion, Example 1 yields an MOMDP whose sets have the following sizes: $|S_v| = 256 * 65536 =$

$16\,777\,216 = |O_v|, |S_h| = 256, |A| = 5$. In the corresponding POMDP, the size of the set of states is $|S| = |S_v| \times |S_h| = 4\,294\,967\,296$. This fully justifies the use of MOMDPs over POMDPs when the problem fits the MOMDP framework as it drastically increases solving performances [Ong *et al.*, 2010].

5 Optimizing the APS

In order to improve the scalability of argumentation problems that can be formalized and solved, we propose several optimization schemes reducing the size of the generated MOMDP. A subtlety occurs because these optimizations may depend upon each other, and it may be useful to apply them several times.

We say that we reach a *minimal* model when no further reduction of the model is possible by application of these techniques. Now this raises an obvious question: as optimizations may influence each other, we may well reach *different* minimal models, depending on the sequence of application chosen.

In this section we provide several guarantees in this respect: (i) we show uniqueness of the minimal model under the iterated application of three schemes, (ii) we show that for the last scheme, uniqueness of the model requires some mild conditions to hold. As a corollary, the optimal policy is preserved throughout these optimizations. In any case, the returned solutions are robust (i.e., they can only improve if the context turns out to be more favorable).

[Irr.] Pruning irrelevant arguments. The first optimization consists in removing the arguments of each agent that are neither modified and never used as premises (“Irrelevant arguments”). This optimization is applied separately on the public and private states. An argument can thus be irrelevant in the description of the private state but can be relevant in the public state. We refer to an internal (resp. public) argument to denote the argument in the private (resp. public) state.

Example 3. In Example 1, we can, for instance, remove the internal argument f from the private state of agent 1. Applying this optimization on the example removes 3 arguments from the private state of agent 1.

Note that, if part of the goal turns out to be an irrelevant argument, this optimization could modify the goal. But this is a degenerate case: when the irrelevant argument is not compatible with the goal state, the outcome of the debate is known a priori (the agent loses the debate anyway), thus we do not consider these cases. Otherwise, the argument is removed from the goal state.

[Enth.] Inferring attacks. The second optimization considers the set of attacks. Let y be a public argument ($a(y)$), if $e(x, y)$ exists and $\boxplus e(x, y) \Rightarrow \boxplus a(x)$ (i.e., each time $e(x, y)$ is added, $a(x)$ also is), as the set of attacks is fully observable, we can infer attacks from the sequence of arguments put forward in the public space and thus remove the attacks from the rules and the states. In fact $e(x, y)$ is no longer used and the semantic of $\boxplus a(x)$ becomes “add argument a and attack y if it is present”.

Example 4. In Example 1, this optimization removes the 8 attacks from the problem definition.

[Irr(s_0 .) Pruning arguments wrt. initial state. For this optimization, we exploit the knowledge about the initial state s_0 . As a result, this optimization requires to regenerate the MOMDP if the initial state changes. This optimization consists of two steps: 1) for each predicate $p \in s_0$ that is not later modified, update the set of rules by removing all the rules that are not compatible with p and then remove p from the premises of the remaining rules 2) remove all rules of the opponent that can never be fired after an action of agent 1:

1. $\forall i, \forall p \in s_0$ s.t. $\exists r \in \mathcal{R}_i$ s.t. $p \in \text{prem}(r)$ and $\nexists r' \in \mathcal{R}_i$ s.t. $p \in \text{acts}(r')$:
 - (a) $\mathcal{R}_i \leftarrow \{r \in \mathcal{R}_i \mid \neg p \notin \text{prem}(r)\}$
 - (b) $\forall r \in \mathcal{R}_i, \text{prem}(r) \leftarrow \text{prem}(r) \setminus p$
2. Let S' be the set of states resulting from the execution of an action of agent 1, i.e., states $s' = F_r(m, s), \forall s \in \mathcal{S}_1 \times \mathcal{P} \times \mathcal{S}_2, \forall r \in C_s(\mathcal{R}_1), \forall m \in \text{acts}(r), \forall r' \in \mathcal{R}_2$ if $r' \notin C_{s'}(\mathcal{R}_2) \forall s' \in S'$ then, $\mathcal{R}_2 \leftarrow \mathcal{R}_2 \setminus \{r'\}$

Note that this optimization is an extension of the optimization on irrelevant arguments. Indeed, after being replaced by their initial value in premises, the arguments become unused and are thus removed.

Example 5. In Example 1, this removes the 5 internal arguments of agent 1.

Note that, this optimization cannot be done for the opponent since her initial internal state is unknown.

Proposition 1. Applying **Irr**, **Ent**, and **Irr(s_0)** does not affect the optimal policy and (a) the optimized model is unique, minimal for those three optimization schemes and independent of the order in which they are applied (as long as they are applied until reaching a stable model).

Proof. (**Irr**.) If an internal or public argument is never used in premises, no rule depends on this argument and no argument can attack it. Moreover, an argument is never modified if and only if it never appears in an act of a rule. Such an argument thus keeps its initial value. We deduce that an Irrelevant argument does not influence action choices and it cannot be added or removed to the state of the debate. This argument is then not relevant to the decision problem and it can be safely removed from the description of the APS (and thus also in the MOMDP).

(**Ent**.) Under this assumption, representing the attacks does not give more information about the current state and can then be removed.

(**Irr(s_0 .)**) For the first part of the optimization, the proof is the same as the one of **Irr**: after replacing the predicate by their value. For the second part of the optimization, a rule is removed if and only if it can never be fired. It will thus never correspond to a possible argumentation action and removing the rule does not modify computed strategies.

(a) **Ent**. is the only optimization on attacks, it thus does not conflict with others and can be placed anywhere in the sequence of optimizations. The optimal sequence for the other

two is (**Irr(s_0), Irr**)+. Indeed, the other way around, **Irr(s_0)** may remove rules making some arguments suitable for **Irr** and involve making another cycle. The order-independency application of the optimization schemes implies the unicity and minimalism of the model. \square

[Dom.] Pruning dominated arguments. Optimizations can be pushed further by using the graph of attacks. Note that unattacked arguments are leaves of the graph. We start by defining the notion of *dominance*.

Definition 2. If an argument is attacked by any unattacked argument, it is dominated.

Since dominated arguments cannot belong to an optimal strategy, the optimization scheme consists in pruning dominated arguments.

Example 6. In our example, we can see that argument **b** is dominated by argument **h**.

This optimization scheme assumes that agent 2 will necessarily fire a rule consisting in adding an argument defeating the dominated argument. Note that this is irrespective of the opponent being an optimal player or not. However, this would not hold if (i) the opponent does not know all her rules, (ii) the debate length is limited (in which case it may make sense to put forward an argument because the attacking argument may lie outside of the debate) and (iii) the opponent cannot play all her arguments.

Proposition 2. If (a) the opponent knows all her rules, (b) can play all her arguments and (c) the debate length is infinite then, applying **Dom**. does not affect the optimal policy.

Proof. If the argument is *truly* dominated, the action adding a dominated argument can be in the optimal policy if and only if no attacking argument can be played. Otherwise, it adds an extraneous step and thus minimizes the reward as we want the shortest sequence. If the argument is *falsely* dominated (i.e. this argument is finally not attacked and thus may be in the optimal policy), it means the opponent cannot put the attacking arguments forward and thus that assumptions (a), (b) and/or (c) do not hold. \square

Nonetheless, applying **Irr** or **Irr(s_0)** may modify the graph of attacks: some unattacked arguments of the opponent can be removed and dominated arguments may appear to be non-dominated. In Example 1, if the opponent cannot play argument **h**, **b** is no longer dominated and it must not be pruned.

Proposition 3. If all dominated arguments always remain dominated after applying **Irr** or **Irr(s_0)**, the optimized model is unique, minimal and independent of the order in which the optimization schemes are applied (as long as they are applied until reaching a stable model).

Proof. In such a case, it means **Dom**. can be anywhere in the sequence of application. As it does not interfere with either **Irr**. or **Irr(s_0)**, Proposition 1 still holds. \square

Otherwise, **Irr** and **Irr(s_0)** must be applied before **Dom** in order to keep only dominated arguments.

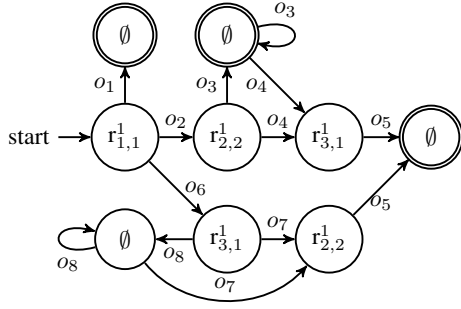


Figure 2: Policy graph for Example 1

6 Experiments

Even if the transformation of an argumentation problem to an MOMDP exploits observable information to reduce the high dimensionality of the problem, it can still lead to a huge state space. It may thus be impossible to use exact solving methods. We ran experiments to test the scalability of the approach proposed in previous sections. We developed a library (github.com/EHadoux/optimizer) to automatically transform an APS into a MOMDP and we applied the previously described optimizations on the problem. Since the exact algorithm MO-IP [Araya-López *et al.*, 2010] was unable to compute a solution in a reasonable amount of time (a few ten of hours), we used MO-SARSOP [Ong *et al.*, 2010], with the implementation of the APPL library [NUS, 2014].

On the problem of Ex.1, after solving the generated MOMDP, the policy graph of agent 1 is shown in Figure 2. The observations of agent 1 are :

$$\begin{aligned}
 o_1 &= \{a(\mathbf{a})\}, o_2 = \{a(\mathbf{a}), a(\mathbf{f})\}, o_3 = \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{f})\} \\
 o_4 &= \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{f}), a(\mathbf{g})\} \\
 o_5 &= \{a(\mathbf{a}), a(\mathbf{c}), a(\mathbf{e}), a(\mathbf{f}), a(\mathbf{g})\}, o_6 = \{a(\mathbf{a}), a(\mathbf{g})\} \\
 o_7 &= \{a(\mathbf{a}), a(\mathbf{e}), a(\mathbf{f}), a(\mathbf{g})\}, o_8 = \{a(\mathbf{a}), a(\mathbf{e}), a(\mathbf{g})\}
 \end{aligned}$$

To follow this policy, start on the first node, apply the rule and move in the graph depending on the observation received. From the point of view of agent 1, accepting states (double circled) are final states of the debate. The agent has no more actions to execute unless the other agent adds or removes a predicate that changes the state. Note that the second node of the top row is an accepting state from which the agent can transition. Indeed, receiving observation o_3 can have two meanings: either the opponent has not played $a(\mathbf{g})$ yet or she will never be able to. From that, the decision-maker can consider waiting for the opponent to play or not. Of course, this policy takes into account the ability for the opponent to apply a rule she has already applied before. Below we consider another example where some predicates can be removed from the state, unlike in Ex.1.

Example 7. This example contains three arguments \mathbf{a} , \mathbf{b} , \mathbf{c} and a special argument \mathbf{s} meaning agent 1 surrenders and thus loses the debate immediately. Rules are:

- $\mathcal{R}_1 = \{h_1(\mathbf{a}) \wedge a(\mathbf{b}) \Rightarrow [1.0 / \boxplus a(\mathbf{a}) \boxplus \boxplus e(\mathbf{a}, \mathbf{b}) \wedge \boxplus e(\mathbf{b}, \mathbf{a})] \\ a(\mathbf{c}) \Rightarrow [1.0 / \boxplus a(\mathbf{s})]\}$
- $\mathcal{R}_2 = \{h_2(\mathbf{b}) \wedge h_2(\mathbf{c}) \Rightarrow [0.9 / \boxplus e(\mathbf{b}, \mathbf{a}) \wedge \boxplus e(\mathbf{a}, \mathbf{b}), \\ 0.1 / \boxplus a(\mathbf{c}) \wedge \boxplus e(\mathbf{c}, \mathbf{a})]\}$

	None	Irr.	Enth.	Dom.	Irr(s_0).	All
Ex 1	—	—	—	—	—	0.56
Ex 7	3.3	0.3	0.3	0.4	0	0
Dv.	—	—	—	—	—	32
6	1313	22	43	7	2.4	0.9
7	—	180	392	16	20	6.7
8	—	—	—	—	319	45
9	—	—	—	—	—	—

Table 1: Computation time (in seconds)

The initial state is $(\{h_1(\mathbf{a})\}, \{\}, \{h_2(\mathbf{b}), h_2(\mathbf{c})\})$, $g_1 = g(\mathbf{a})$.

Figure 3 shows the optimal policy graph for Example 7. The observations of agent 1 are as follows:

$$\begin{aligned}
 o_1 &= \{a(\mathbf{a}), e(\mathbf{b}, \mathbf{a})\}, o_2 = \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b}), a(\mathbf{c}), e(\mathbf{c}, \mathbf{a})\} \\
 o_3 &= \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b}), a(\mathbf{c}), e(\mathbf{c}, \mathbf{a}), a(\mathbf{s})\}, o_4 = \{a(\mathbf{a}), e(\mathbf{a}, \mathbf{b})\}
 \end{aligned}$$

Finally, we investigated the influence of each optimization on the computation time. Table 1 reports computation times required to solve the problems while applying different sets of optimizations before solving the problem with MO-SARSOP. We considered Ex.1, Ex.7 and a slightly modified version (in order to fit it in our framework) of Dvorak (Dv.) problem taken from [DBAI group, 2013]. A dash in the table means that the computation of the optimal policy took more than 30 min. and 0 means that the time is less than 0.01 secs.

We can see that for Ex.1 only the fully optimized problem can be solved in a reasonable amount of time. In order to study how the method scales, we also generated instances built on bipartite argumentative graphs (but not trees) with an increasing number of arguments evenly split among the two agents. In Table 1, Line n (where $n = 6, \dots, 9$) shows the time needed to solve problems with n arguments.

7 Conclusion and discussion

In this paper we explored the following research question: can we find –and to what extent– the optimal policy of an agent facing an opponent playing stochastically in an argumentative dialogue. We first showed that one can take advantage of the fact that arguments are exchanged through a public space, making MOMDP a suitable model. Next we exploited the fact that the domain of argumentation is highly structured: different schemes can be designed to minimize the obtained model, while preserving the optimality of the policy. Our experimental findings are balanced: on one hand we show the effectiveness of these optimization schemes, which make several examples solvable in practice. On the other hand optimal resolution remains extremely costly with these models,

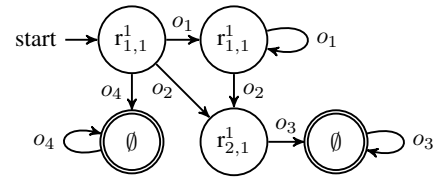


Figure 3: Policy graph for Example 7

and it seems at the moment very unlikely to handle instances involving more than a dozen of arguments. We believe this provides valuable insights as to what can be done in practice when designing argumentative agents. Future work involves conducting comprehensive experiments, possibly with new optimizations. One possible room for improvement is to use knowledge of the goal. Indeed by representing the goals of the opponent in a belief function, we can update it using the observation at each step, and eventually learn the adversary's goal, in order to avoid them.

Acknowledgments

Funded by the French National Research Agency under grant ANR-10-BLAN-0215 and grant AMANDE ANR-13-BS02-0004.

References

- [Araya-López *et al.*, 2010] M. Araya-López, V. Thomas, O. Buffet, and F. Charpillet. A closer look at MOMDPs. In *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 2010.
- [Bellman, 1957] Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6, 1957.
- [Black and Hunter, 2012] Elizabeth Black and Anthony Hunter. *Executable logic for dialogical argumentation*, pages 15–20. *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2012.
- [DBAI group, 2013] DBAI group. [dbai.tuwien.ac.at /research/project/argumentation/dynpartix/](http://dbai.tuwien.ac.at/research/project/argumentation/dynpartix/), 2013.
- [Dung, 1995] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.
- [Hadjinikolis *et al.*, 2013] Christos Hadjinikolis, Yiannis Siantos, Sanjay Modgil, Elizabeth Black, and Peter McBurney. Opponent modelling in persuasion dialogues. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.
- [Hunter, 2014] A. Hunter. Probabilistic strategies in dialogical argumentation. In *International Conference on Scalable Uncertainty Management (SUM'14) LNCS volume 8720*, 2014.
- [Kontarinis *et al.*, 2014] Dionysios Kontarinis, Elise Bonzon, Nicolas Maudet, and Pavlos Moraitis. Empirical evaluation of strategies for multiparty argumentative debates. In *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings*, pages 105–122, 2014.
- [Modgil and Caminada, 2009] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–132. Springer, 2009.
- [NUS, 2014] National University of Singapore NUS. <http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl>, 2014.
- [Ong *et al.*, 2010] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee. Planning under uncertainty for robotic tasks with mixed observability. In *The International Journal of Robotics Research*, 2010.
- [Papadimitriou and Tsitsiklis, 1987] Christos Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, August 1987.
- [Prakken, 2006] Henry Prakken. Formal systems for persuasion dialogue. *Knowledge Eng. Review*, 21(2):163–188, 2006.
- [Puterman, 1994] M. L. Puterman. *Markov Decision Processes: discrete stochastic dynamic programming*. John Wiley & Sons., 1994.
- [Rienstra *et al.*, 2013] Tjitze Rienstra, Matthias Thimm, and Nir Oren. Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, August 2013.
- [Thimm and Garcia, 2010] Matthias Thimm and Alejandro J. Garcia. Classification and Strategical Issues of Argumentation Games on Structured Argumentation Frameworks. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems 2010 (AAMAS'10)*, May 2010.
- [Thimm, 2014] Matthias Thimm. Strategic argumentation in multi-agent systems. *Künstliche Intelligenz, Special Issue on Multi-Agent Decision Making*, 28(3):159–168, June 2014.