ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1736

Optimal Simple Strategies for Persuasion

Elizabeth Black and **Amanda Coles** and **Christopher Hampson**[1]

1 Introduction

Argument dialogues provide a principled way of structuring rational interactions between participants (be they human or machine), each arguing over the validity of certain claims, with each agent aiming for an outcome that achieves their dialogue goal (*e.g.*, to persuade the other participant to accept their point of view [8], or to reach agreement on an action to perform [2]). Achievement of an agent's dialogue goal typically depends on both the arguments that the agent chooses to make during the dialogue, determined by its *strategy*, and the arguments asserted by its interlocutor. The strategising agent—the *proponent*—thus has the difficult problem of having to consider not only which arguments to assert but also the possible responses of its *opponent*. This problem is compounded since the opponent may exploit knowledge inferred from those arguments asserted by the proponent to construct new arguments. Hence, the proponent must take care not to divulge information that is advantageous to its opponent.

The important challenge of how to generate strategies for such a proponent has not been widely explored [10]. Notable exceptions are the work of Hadoux *et al.* [7], which employs mixed observability Markov decision processes to generate optimal policies for the proponent to follow; the work of Rienstra *et al.* [9], which applies a variant of the minimax algorithm to determine an effective proponent strategy; and the work of Black *et al.* [3], which employs heuristic planning techniques to determine an optimal proponent strategy for a simple asymmetric persuasion setting.

We highlight two types of uncertainty in the strategic argumentation problem: uncertainty over the arguments initially know to the opponent, captured by the *opponent model*, and uncertainty over how the opponent chooses to argue, given their initial knowledge base. Both [9] and [3] deal with uncertain models of the opponents initial knowledge base, where the opponent's strategy is known (*i.e.*, optimal [9] or deterministic [3]); while [7] considers the case in which we are certain of the arguments known to the opponent but have only a stochastic model of how they will behave.

The key novelties in our approach are that we deal with both of these types of uncertainty simultaneously the former through use of an *uncertain opponent model* and the latter by generating *conformant* strategies, that is strategies that are effective regardless of the opponent strategy. Further, our work is the first to generate strategies in a setting where the opponent may exploit information obtained during the dialogue to construct arguments unknown to it at the start of the dialogue, necessitating more cautious strategies.

2 Strategic Argumentation Problem

We consider a strategic argumentation setting in which both agents exchange arguments, with the proponent aiming to convince its opponent of some topic argument. Our problem comprises the following:

- An argumentation framework $(\mathcal{A}, \rightarrow)$, comprising a set of arguments $\mathcal{A}$ and a binary *attack relation* $\rightarrow \subseteq (\mathcal{A} \times \mathcal{A})$,
- The function $Acc : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ describes the set of *acceptable* arguments $Acc(\mathcal{B})$, for each subset $\mathcal{B} \subseteq \mathcal{A}$, subject to some agreed semantics (see [5], for details).
- A proponent model $\mathcal{K}_{\mathcal{P}} \subseteq \mathcal{A}$, comprising a set of arguments available to the proponent.
- An *uncertain* opponent model $(\mathcal{E}, \mathcal{K}_{\mathcal{O}}, p)$, comprising a finite set of labels $\mathcal{E} = \{\mathcal{O}_1, \ldots, \mathcal{O}_m\}$, a function $\mathcal{K}_{\mathcal{O}} : \mathcal{E} \rightarrow 2^{\mathcal{A}}$ associating each label $\mathcal{O}_i \in \mathcal{E}$ with a possible opponent model $\mathcal{K}^i_{\mathcal{O}} \subseteq \mathcal{A}$, and a probability distribution $p : \mathcal{E} \rightarrow \mathbb{Q}$, assigning to each $\mathcal{O}_i \in \mathcal{E}$ the likelihood that the opponent initially knows $\mathcal{K}^i_{\mathcal{O}}$.
- A closure operator $\mu : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ provides a description of how each agent may derive new arguments from knowledge acquired during the dialogue. *i.e.*. $\mu(\mathcal{B}) \subseteq \mathcal{A}$ is the set of all arguments that can be derived by $\mathcal{B} \subseteq \mathcal{A}$.

A *dialogue* is a sequence of *moves* $D = [M_0, M_1 \ldots, M_n]$ in which each move $M_k \subseteq \mathcal{A}$ is a finite set of arguments. A dialogue terminates when $M_{k-1} \cup M_k = \emptyset$, and is *successful* for the proponent w.r.t. a given topic $t \in \mathcal{A}$, if $t \in Acc(M_0 \cup \cdots \cup M_n)$.

A (general) strategy for $\mathsf{Ag} \in \{\mathcal{P}, \mathcal{O}\}$ is a function $\sigma_{\mathsf{Ag}} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that $\sigma(\mathcal{B}) \subseteq \mu(\mathcal{K}_{\mathsf{Ag}} \cup \mathcal{B})$, for all $\mathcal{B} \subseteq \mathcal{A}$, that determines which move Ag should make, given the arguments asserted thus far. A *simple* strategy is a special case of strategy specified by a sequence of moves $S = [S_0, \ldots, S_n]$, with $S_k \subseteq \mathcal{K}_{\mathcal{P}}$, representing sets of arguments to be asserted in turn by the proponent, unless the opponent accepts the topic, or the sequence is exhausted, in which case we assert $\emptyset$. A pair of strategies $(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})$ generates the dialogue $D_{(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})} = [M_0, \ldots, M_n]$ given by $M_k = \sigma_{\mathsf{Ag}}(M_0 \cup \cdots \cup M_{k-1})$, where $\mathsf{Ag} = \mathcal{P}$ whenever k is even and $\mathsf{Ag} = \mathcal{O}$ whenever k is odd.

A strategy $\sigma_{\mathcal{P}}$ is effective against $\mathcal{O}_i \in \mathcal{E}$ if $D_{(\sigma_{\mathcal{P}}, \sigma_{\mathcal{O}})}$ is successful, for every possible opponent strategy $\sigma_{\mathcal{O}}$ for $\mathcal{O}_i$.

3 Persuasion Dialogues as Planning Problems

We provide a translation of a strategic argumentation problem into a propositional planning problem with (bounded) numerical variables, such that a solution to the planning problem yields an effective strategy for our proponent. Encoding the problem in the standard planning domain language PDDL2.1 [6] allows us to use an implementation of the planner POPF [4] to generate appropriate simple strategies. A key challenge in using a propositional planner to solve these problems is capturing the uncertainty about the opponent's initial beliefs. We use techniques inspired by the current state-of-the-art approach for solving conformant planning problems by compilation to classical planning [1]. The planning problem $\mathcal{P}$ is described as follows:

Variables We require the following numerical variables: (stage), $(\mathsf{probSuccess})$ and $(\mathsf{prob}(i))$, for each $\mathcal{O}_i \in \mathcal{E}$. The variable (stage)

[1] King's College London, Strand, London, UK, WC2R 2LS,
email: firstname.lastname@kcl.ac.uk
This work was partially supported by the EPSRC, grant ref. EP/M01892X/1

Action:	proponent(a) for all $a \in \mathcal{K}_\mathcal{P}$
pre	(i) canAssertP(a) (ii) (stage $= 0$) $\vee$ (stage $= 1$)
eff	(a) $\neg$canAssertP(a) (b) (stage $\leftarrow 2$) (c) **if** dialogueP($D_\mathcal{P}$), addP($a, D_\mathcal{P}, D'_\mathcal{P}$) **then** dialogueP($D'_\mathcal{P}$), $\neg$dialogueP($D_\mathcal{P}$) (d) **if** dialogueO($i, D_\mathcal{O}$) **then** temp($i, D_\mathcal{O}$), $\neg$dialogueO($i, D_\mathcal{O}$) (e) effective(i)

Figure 1: Actions in our Planning Model

regulates the order in which actions may be added to the plan, while the variables (probSuccess) and (prob(i)) are responsible for calculating the probability that a given strategy will be successful. In addition to these, we have the following propositional variables: canAssertP(a) and canAssertO(M, i, D) govern the arguments that $\mathcal{P}$ and $\mathcal{O}$ can assert, respectively, for $a \in \mathcal{K}_\mathcal{P}$, $\mathcal{O}_i \in \mathcal{E}$ and $D \subseteq \mathcal{K}_\mathcal{P}$; dialogueP($D$) and dialogueO($i, D$) describe the set of arguments that have been asserted by $\mathcal{P}$ and $\mathcal{O}_i \in \mathcal{E}$, respectively; temp($i, D$) a temporary 'storage' variable for dialogueO(i, D); successful($D_\mathcal{P}, D_\mathcal{O}$) if ($D_\mathcal{P} \cup D_\mathcal{O}$) is an acceptable set of arguments, where D_{Ag} are the arguments asserted by $\mathsf{Ag} \in \{\mathcal{P}, \mathcal{O}\}$; effective($i$) says that the strategy is effective against $\mathcal{O}_i \in \mathcal{E}$; addP($a, D, D'$) says that $D' = D \cup \{a\}$, and addO(M, D, D') says that $D' = D \cup M$.

Initial Conditions Our set of initial conditions comprises the following numerical conditions: (stage $= 0$), (probSuccess $= 0$), and (prob(i) $= p(\mathcal{O}_i)$), for all $\mathcal{O}_i \in \mathcal{E}$. We also require the following propositional initial conditions: canAssertP(a), for all $a \in \mathcal{K}_\mathcal{P}$, dialogueP($\emptyset$), and dialogueO($i, \emptyset$), for all $\mathcal{O}_i \in \mathcal{E}$, together with the following conditions that, once set, remain unaltered by the effects of any of the actions: canAssertO($M, i, D_\mathcal{P}$) iff $M \subseteq \mu(\mathcal{K}^i_\mathcal{O} \cup D_\mathcal{P})$; successful($D_\mathcal{P}, D_\mathcal{O}$) iff $t \in Acc(D_\mathcal{P} \cup D_\mathcal{O})$; addP($a, D_\mathcal{P}, D'_\mathcal{P}$) iff $D'_\mathcal{P} = D_\mathcal{P} \cup \{a\}$; addO($M, D_\mathcal{O}, D'_\mathcal{O}$) iff $D'_\mathcal{O} = D_\mathcal{O} \cup M$.

Goal Condition Our goal is a single condition (probSuccess $> \lambda$), where $\lambda \in [0, 1]$ is the required probability of success.

Actions Our set of actions comprises three types of action whose preconditions and effects are described in Figures 1–2, where all free parameters are universally quantified. For each argument $a \in \mathcal{K}_\mathcal{P}$ known to the proponent, there is a (proponent(a)) action, which emulates the act of the proponent asserting a. A single move is built up by iterated application of these actions (simulating the assertion of a set of arguments by the proponent). The opponent's move is captured by a single (opponent) action. This action simulates *all* possible responses for each possible opponent model $\mathcal{O}_i \in \mathcal{E}$, adding them to a 'pool' of possible dialogue states associated with that opponent model. Finally the action (probCount) must be applied after each (opponent) action and sums the total probability of *guaranteed* success, against each of the possible opponent models $\mathcal{O}_i \in \mathcal{E}$.

A solution to the planning problem $\mathcal{P}$, generated by a planner, is a sequence of actions that transforms the initial state into a state that satisfies the goal. Such a solution corresponds to a simple strategy that will be successful with probability $> \lambda \in [0, 1]$. To find an *optimal* simple strategy maximising λ, we first find a solution where $\lambda > 0$ and then iteratively seek solutions with strictly larger λ values.

4 Results

Owing to the lack of existing approaches for proponent strategy generation that do not depend on knowledge of the opponent's strategy, we instead benchmarked our approach against a naïve depth-first search, that exhaustively searches all simple strategies and evaluates their effectiveness with respect to the given opponent model. Our analysis showed that that for smaller examples the performance two approaches in comparable; however, for larger examples, comprising more than nine arguments, the planning approach outperforms the naïve search by several orders of magnitude.

Action:	opponent
pre	(i) (stage $= 1$)
eff	(a) (stage $\leftarrow 2$) (b) **if** dialogueP($D_\mathcal{P}$), temp($i, D_\mathcal{O}$), canAssertO($M, i, D_\mathcal{P}$), addO($M, D_\mathcal{O}, D'_\mathcal{O}$), $\neg$successful($D_\mathcal{P}, D'_\mathcal{O}$) **then** dialogueO($i, D'_\mathcal{O}$), $\neg$effective(i) (c) $\neg$temp($i, D_\mathcal{O}$)

Action:	probCount
pre	(i) (stage $= 2$)
eff	(a) (stage $\leftarrow 0$) (b) **if** effective(i) **then** (probSuccess $\leftarrow$ probSuccess + prob(i)), (prob(i) $\leftarrow 0$)

Figure 2: Actions in our Planning Model

We performed further experiments to compare to the most closely related approach by Hadoux et al. [7] on the problems described in their paper. The two approaches are not directly comparable: we do not rely on knowledge of the opponents strategy, whilst they do; they generate policies, dependent on the opponent's response, whereas we generate simple strategies. We did however make two observations: first our approach finds solutions an order of magnitude faster than theirs; and second in all of the problems they propose a simple strategy is sufficient to guarantee 100% success rate, without the need for a policy, meaning that our approach is equally strong in these particular settings. Our problems, the planner we used, and the implementation of our translation and of our naïve algorithm are all available from: `http://tinyurl.com/jfxotsg`.

REFERENCES

[1] Alexandre Albore, Hector Palacios, and Hector Geffner, 'Compiling uncertainty away in non-deterministic conformant planning.', in *European Conference on Artificial Intelligence*, pp. 465–470, (2010).

[2] Elizabeth Black and Katie Bentley, 'An empirical study of a deliberation dialogue system', in *Theory and Applications of Formal Argumentation*, 132–146, Springer LNCS 7132, (2012).

[3] Elizabeth Black, Amanda Coles, and Sara Bernardini, 'Automated planning of simple persuasion dialogues', in *Computational Logic in Multi-Agent Systems*, 87–104, Springer LNAI 8624, (2014).

[4] Amanda J. Coles, Andrew I. Coles, Maria Fox, and Derek Long, 'Forward-chaining partial-order planning', in *International Conference on Automated Planning and Scheduling*, (2010).

[5] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321 – 357, (1995).

[6] Maria Fox and Derek Long, 'PDDL2. 1: An extension to PDDL for expressing temporal planning domains.', *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).

[7] Emmanuel Hadoux, Aurélie Beynier, Nicolas Maudet, Paul Weng, and Anthony Hunter, 'Optimization of probabilistic argumentation with markov decision models', in *International Joint Conference on Artificial Intelligence*, pp. 2004–2010, (2015).

[8] Henry Prakken, 'Formal systems for persuasion dialogue', *The Knowledge Engineering Review*, **21**(02), 163–188, (2006).

[9] Tjitze Rienstra, Matthias Thimm, and Nir Oren, 'Opponent models with uncertainty for strategic argumentation', in *International Joint Conference on Artificial Intelligence*, pp. 332–338, (2013).

[10] Matthias Thimm, 'Strategic argumentation in multi-agent systems', *Künstliche Intelligenz, Special Issue on Multi-Agent Decision Making*, **28**(3), 159–168, (2014).