Argumentation-Based Reinforcement Learning for RoboCup Soccer Takeaway

(Extended Abstract)

Yang Gao
Imperial College London, UK
yg211@imperial.ac.uk

Francesca Toni
Imperial College London, UK
ft@imperial.ac.uk

ABSTRACT

Reinforcement Learning (RL) is widely regarded as a generic and effective technique to learn coordinated behaviours in cooperative multi-agent systems (CMAS), but it suffers from slow convergence speed due to the huge joint action space. Incorporating domain knowledge has shown to be an effective method to tackle this problem, but little research has investigated how to propose high-quality heuristics. We consider a widely used CMAS application, *RoboCup Takeaway*, and use *value-based argumentation* to extract heuristics from conflicting domain knowledge therein.

1. INTRODUCTION

Reinforcement Learning (RL), which enables agents to learn optimal behaviour by interacting with the environment, has been regarded as an effective machine learning technique to achieve coordinated behaviours in cooperative multi-agent systems (CMAS) [1, 2]. However, RL can be very slow in CMAS, mainly because of the huge joint action space which is exponential in the number of agents [1]. An effective methodology to tackle this problem is to integrate domain knowledge into RL so as to reduce the exploration time [3]. However, even though it has been reported that the effectiveness of this methodology is sensitive to the quality of the heuristics, little research has been devoted to investigate how to facilitate people to extract high-quality heuristics from conflicting domain knowledge. In this work, we consider a widely used CMAS: RoboCup Takeaway, and build a variant of *value-based argumentation frameworks* [4] to solve the conflicts within the domain knowledge so as to provide heuristics to achieve coordinated behaviours without searching the joint action space.

2. ABSTRACT AND VALUE-BASED ARGUMENTATION FRAMEWORKS

An *abstract argumentation framework* (AF) is a pair (Arg, Att) where *Arg* is a set of *arguments* and $Att \subseteq Arg \times Arg$ is a binary relation ($(A, B) \in Att$ is read 'A *attacks* B'). Suppose $S \subseteq Arg$ and $B \in Arg$. S *attacks* B iff some member of S attacks B. S is *conflict-free* iff S attacks none of its members. S *defends* B iff S attacks all arguments attacking B. Semantics of AFs are defined as sets of "rationally acceptable" arguments, known as *extensions*. For example, given some $\mathsf{F} = (Arg, Att)$, $S \subseteq Arg$ is an *admissible extension* for F iff S is conflict-free and defends all its elements; S is a *complete extension* for F iff S is conflict-free and $S = \{a|S \text{ defends } a\}$; S is the *grounded extension* for F iff S is minimally (wrt. $\subseteq$) complete for F. The (possibly empty) grounded extension is guaranteed to be unique, consisting solely of the uncontroversial arguments and being thus "sceptical".

In some contexts, the attack relation between arguments is not enough to decide what is "rationally acceptable", and the "values" promoted by arguments must be considered. *Value-based argumentation frameworks* (VAFs) [4] incorporate values and preferences over them into AFs. Their key idea is to allow for attacks to succeed or fail, depending on the relative worth of the values promoted by the competing arguments. Given a set V of values, an *audience Valpref* is a strict partial order over V (corresponding to the preferences of an agent), and an *audience-specific VAF* is a tuple $(Arg, Att, V, val, Valpref)$, where (Arg, Att) is an AF and $val : Arg \rightarrow V$ gives the values promoted by arguments. In VAF, the ordering over values, *Valpref*, is taken into account in the definition of extensions. The *simplification* of an audience-specific VAF is the AF (Arg, Att^-), where $(A, B) \in Att^-$ iff $(A, B) \in Att$ and $val(B)$ is not higher than $val(A)$ in *Valpref*. $(A, B) \in Att^-$ is read 'A *defeats* B'. Then, (acceptable) extensions of a VAF are defined as (acceptable) extensions of its simplification (Arg, Att^-). We refer to (Arg, Att^-) as the *simplified AF derived from* $(Arg, Att, V, val, Valpref)$.

3. MOTIVATING EXAMPLE: ROBOCUP TAKEAWAY GAME

The *Takeaway* game is proposed by [5]. In a N-Takeaway game, $N + 1$ ($N \in \mathbb{N}$, $N \geq 1$) hand-coded *keepers* are competing with N learning *takers* on a fixed-size field. Keepers attempt to keep possession of the ball, whereas takers attempt to win possession of the ball. Since only takers are learning in Takeaway, their learning task is to win possession of the ball as fast as possible.

Iscen and Erogul [5] proposed two *macro actions* for takers:

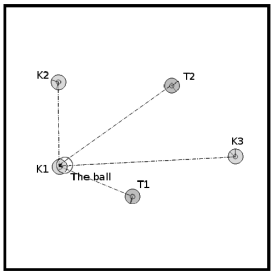
- **TackleBall()**: move directly towards the ball to tackle it
- **MarkKeeper(i)**: go to mark keeper K_i, $i \neq 1$

where K_i represents the ith closest keeper to the ball (so that K_1 is the keeper in possession of the ball). When a taker marks a keeper, the taker blocks the path between the ball and that keeper. Thus, a taker is not allowed to mark the ball holder, and in N-Takeaway, each taker can choose among $M{=}N{+}1$ actions.

The observation of each taker is represented by a *state vector*. We use the same state vector as in our previous work [6].

Consider a scenario in 2-Takeaway as illustrated in Fig 1(a). We may propose the following advice: **(a)** T_1 should tackle the ball, because it is closest to the ball; **(b)** T_2 should mark K_3, because it is closest to K_3; and **(c)** T_1 should mark K_3, because the angle between K_3 and T_1, with vertex at K_1, is smallest. Even only consid-

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*



(a)

$$\begin{tikzcd} T_1\mathbf{TK} \arrow[r, leftrightarrow] & T_1\mathbf{A}(3) \arrow[d, leftrightarrow] & \\ T_2\mathbf{C}(2) \arrow[r, leftrightarrow] & T_2\mathbf{C}(3) \arrow[r, leftrightarrow] & T_2\mathbf{A}(2) \end{tikzcd}$$

(b)

$$\begin{tikzcd} T_1\mathbf{TK} \arrow[r] & T_1\mathbf{A}(3) \arrow[d] & \\ T_2\mathbf{C}(2) \arrow[r, leftrightarrow] & T_2\mathbf{C}(3) & T_2\mathbf{A}(2) \arrow[l] \end{tikzcd}$$

(c)

Figure 1: (a) An example scenario in 2-Takeaway game, (b) its *SCAF* and (c) the simplified AF derived from its *VSCAF*.

ering these three recommendations, we can see that there exist both *internal conflicts* and *external conflicts*: item **(a)** and **(c)** internally conflict with one another because they are both recommendations for T_1, but suggest T_1 to perform different actions; item **(b)** and **(c)** externally conflict with one another because they are recommendations for different agents, but suggest them to perform the same action, which, in Takeaway, is believed wasteful in terms of efficiency. We are going to use value-based argumentation frameworks to solve these conflicts, as shown next.

4. ARGUMENTATION FRAMEWORK FOR TAKEAWAY GAMES

In line with Section 3, we give the following domain knowledge for any taker $T_i, i \in \{1, \cdots, N\}$:

1. T_i should tackle the ball if it is closest to the ball holder;
2. If the angle between T_i and a keeper, with vertex at the ball holder, is the smallest, T_i should mark this keeper;
3. If T_i is closest to a keeper, T_i should mark this keeper.

Note that this knowledge is action-based, i.e. recommending actions to agents. Given state variables (Table 2 in [6]) we "translate" the knowledge above into 3 categories of candidate arguments:

1. T_i**TK**: **TackleBall()** IF $i = \arg\min_{1 \le t \le N} dist(K_1, T_t)$

2. $T_i\mathbf{A}(j)$: **MarkKeeper(*j*)** IF $i = \arg\min_{1 \le t \le N} ang(K_j, T_t)$

3. $T_i\mathbf{C}(j)$: **MarkKeeper(*j*)** IF $i = \arg\min_{1 \le t \le N} dist(K_j, T_t)$

where $j \in \{2, \cdots, N+1\}$ for arguments referred to as $T_i\mathbf{A}(j)$ and $T_i\mathbf{C}(j)$, because K_1 cannot be marked. Overall, for a N-Takeaway game, there are $2N^2 + N$ candidate arguments[1]. We will use *Arg** to denote the set of candidate arguments, and given these candidate arguments, we build argumentation framework as follows:

DEFINITION 1. *A* Scenario-specific cooperative argumentation framework *(SCAF) is a tuple* (Arg, Att) *s.t.:*
1. *Arg*$\subseteq$*Arg** *s.t.* $A \in Arg$ *iff the premise of* A *is true in this scenario.*
2. *Att* $\subseteq$ *Arg* $\times$ *Arg s.t.* $(A, B) \in Att$ *iff*
(a) *A, B belong to the same agent but support different actions, or*
(b) *A, B belong to different agents but support the same action.*

[1]For taker T_i, $T_i\mathbf{TK}$ gives one argument and the other two categories of arguments each give N (as there are N free keepers to be marked). So there are $N \times (2 \times N + 1)$ candidate arguments in total.

Remark. To build a *SCAF*, we first select the *applicable* arguments, namely the arguments whose premises are true in this scenario. Then we build attacks between these applicable arguments to represent the conflicts between the domain knowledge. Item 2**(a)** and item 2**(b)** models the internal and external conflicts, resp..

SCAF models the conflicts within domain knowledge, but does not solve them. To this end, we incorporate *values* into *SCAF*:

DEFINITION 2. *Given a SCAF* (Arg, Att)*, a* value-based scenario specific cooperative argumentation framework *(VSCAF) is a tuple* $(Arg, Att, V, val, Valpref)$ *s.t.:*
1. *V is a set (of* values*)*
2. *val* : *Arg** $\rightarrow$ *V is a function from Arg** *to V*
3. *Valpref is a strict partial order over V*
We denote $val(A) = v$ *as* $A \mapsto v$*, and say that* A promotes v.

Given values and their preferences, a *simplified AF* can be derived from a VSCAF, as in standard VAF. We denote the new set of attacks as *Att*$^-$, and let *AF*$^-$ $=$ (Arg, Att^-) be this simplified AF derived from *VSCAF*. Then we compute the grounded extension of *AF*$^-$ to obtain the "rationally accepted" arguments, and these arguments can be used as heuristics to instruct agents.

As a concrete example, we build the *SCAF* for the scenario depicted in Fig. 1(a). First, we compute the applicability of each candidate arguments. To illustrate, we first consider argument $T_1\mathbf{TK}$. Since T_1 is closest to the ball, this argument is applicable. For the same reason, we can see that $T_2\mathbf{TK}$ is not applicable. The applicability of other candidate arguments can be decided similarly. We then build attack relationships (*Att*) between these applicable arguments. To illustrate *Att*, consider $T_1\mathbf{TK}$ and $T_1\mathbf{A}(3)$: they are both applicable for T_1 but recommend different actions, so they attack each other. Consider also $T_1\mathbf{A}(3)$ and $T_2\mathbf{C}(3)$: they are applicable for different agents but recommend the same action, so they attack each other. The resulting *SCAF* is depicted in Fig. 1(b).

Then we incorporate values into the *SCAF* to build a *VSCAF* and derive its simplified AF. We propose the following values:

1. VT: Prevent the ball being held by the keepers;
2. VA: Ensure that each pass can be quickly intercepted;
3. VC: Ensure that, after each pass, the ball can be quickly tackled.

The mapping from arguments to values (*val*) is defined as follows: $T_j\mathbf{TK} \mapsto \mathbf{VT}$, $T_j\mathbf{A}(i) \mapsto \mathbf{VA}$, $T_j\mathbf{C}(i) \mapsto \mathbf{VC}$. Further, according to our domain knowledge, let $\mathbf{VT} >_v \mathbf{VA} =_v \mathbf{VC}$. Given these values and their rankings, we can obtain the simplified AF, as illustrated in Fig 1(c). Its grounded extension is $\{T_1\mathbf{TK}, T_2\mathbf{A}(2), T_2\mathbf{C}(2)\}$, so we will recommend T_1 to tackle the ball and recommend T_2 to mark keeper K_2.

5. REFERENCES

[1] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI*, 1998.
[2] M. Ghavamzadeh, S. Mahadevan, and R. Makar. Hierarchical multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 13:197–229, 2006.
[3] S. Devlin, M. Grzes, and D. Kudenko. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems*, 14:251–278, 2011.
[4] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.
[5] A. Iscen and U. Erogul. A new perspective to the keepaway soccer: The takers (short paper). In *Proc. of AAMAS*, 2008.
[6] Y. Gao and F. Toni. Argumentation accelerated reinforcement learning for robucup keepaway-takeaway. In *TAFA*, 2013.