OxBlue2008(2D) Team Description

Jie Ma and Stephen Cameron

Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK
{jie.ma,stephen.cameron}@comlab.ox.ac.uk
http://www.comlab.ox.ac.uk

Abstract. OxBlue2008(2D) is a robot football team for RoboCup 2D simulation. In this paper, the decision structure of our team will be presented, and a novel method called PSP (Policy Search Planning) is also proposed. Policy Search is used to find an optimal policy for selecting plans from a *plan pool*; it extends an existing gradient-search method (GPOMDP) to a MAS domain. We demonstrate how PSP can be used in RoboCup Simulation, and our experimental results reveal robustness, adaptivity, and outperformance over other methods.

1 Introduction

OxBlue2008(2D) is a robot football team for RoboCup 2D simulation and it derives from Apollo(2D). Jie was the team leader of Apollo and lead the team won the first place of China RoboCup in 2004 and 2nd place of US RoboCup Open as well as 7th place of World RoboCup in 2005.

The main purpose of our development of OxBlue2008 is to verified and promote our research on Multi-agent Systems (MAS). In particular, as cooperative skills essentially differentiate MASs from single-agent intelligence, we are interested in applying Machine Learning methods to yield cooperative behaviours in MAS scenarios.

In §2, a layered decision architecture that is used in OxBlue2008 is presented in . Under such a structure, our novel method PSP (Policy Search Planning) and its applications in RoboCup are briefly introduced in §3. It is followed by the conclusion and future work in §5.

2 Decision Architecture

In order to reduce the learning space of cooperative skills, most of today's MASs tend to adopt *vertical layered architectures* [1, 2]. This architecture is also adopted in OxBlue2008. In RoboCup soccer, pure reaction is required on some occasions, such as in a corner kick situation, where most agents don't need to make complicated decisions but to move to stationary positions. In other words, before world models are fully generated, actions will be directly sent. In more sophisticated decisions, however, such as stopping the ball from losing it, then

although world models have been created, emergency actions will be directly sent to the executor without comparing different skills in the arbitrator. This is in-between decisions. In most cases, decisions are pure deliberation - local issues such as interception and dribbling can be solved in the individual skill module, while global problems including formation and team strategies can be dealt with by advanced methods such as planning. Our decision structure is given in Figure 1.

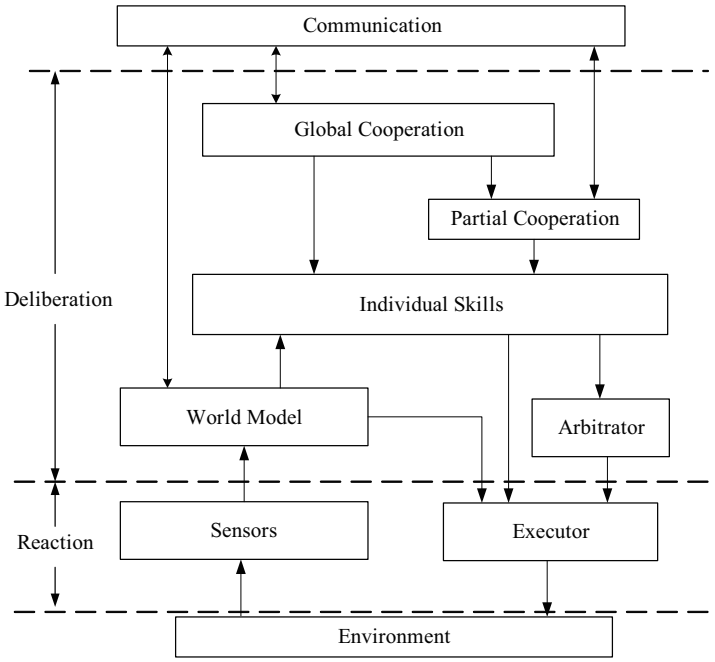

Fig. 1. Decision Architecture of OxBlue2008

3 Policy Search Planning (PSP)

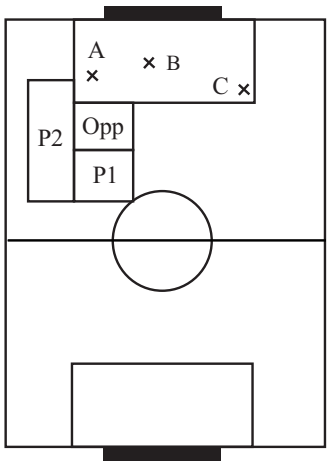

Fig. 2. A Planning scenario in RoboCup

In complex MASs, particularly in a system with hybrid individual architectures, planning plays a different role compared with that in traditional domains. In a simplified single-agent system, planning is used to directly find a goal. In dynamic MASs, however, the goal is usually difficult to achieve, or sometimes it is difficult to describe the goal. In addition, the traditional action effects will lose their original meaning: environmental state can also be changed by other agents at the same time, or sometimes it continually varies even without any actions. For example, consider a scenario

from RoboCup as shown in Figure 2: $P1$ and $P2$ are two team members with $P1$ controlling ball, *Opp* is an opponent, and they are all located in different areas. A traditional planner might construct a plan in which $P2$ dashes to point A and then $P1$ passes the ball. However, in this situation, points B and C are also potential target points for $P2$. Even from a human's perspective it is difficult to say which plan is better before fully knowing the opponent's strategies. Therefore, in multi-agent systems planning tends to be regarded as a "tutor" to increase cooperative behaviours so as to improve overall system performance. Expert knowledge can be embodied in such planning, without which agents mainly execute individual skills.

We propose a novel method called Policy Search Planning (PSP) for POMDPs, which is essentially a centralised planner for distributed actions. In the example of Figure 2, PSP can try to find the most appropriate policy for selecting a plan even without the opponent's model. Specifically, it can represent a number of complex cooperative tactics in the form of plans. Plans are shared by all the agents in advance, and policy search is used to find the optimal policy in choosing these plans. As a plan is not designed to find the goal directly but to define cooperative knowledge, the style of it is not very critical. One possible presentation, a PDDL-like planner, is shown in Figure 3.

Compared with original PDDL, *:goal* will not be included as PSP aims not to achieve it directly, and *:effect* is not needed unless it is used for parameters in policy search. The concept of *stage* is introduced, which makes complex cooperation possible, whereby if and only if the success condition of the current stage is met a planner moves to the next stage; and *role mapping* formulae are introduced to find the most appropriate agents to implement actions

```
(define (PLAN_NAME)
   (:plan_precondition CONDITION_FORMULA)
   ((:agentnumber INTEGER(N))
    (ROLE_MAPING_FORMULA(1))
    (ROLE_MAPING_FORMULA(2))
     ...
    (ROLE_MAPING_FORMULA(N)))
   ((:stagenumber INTEGER(M))
    ((:stage_1_precondition   CONDITION_FORMULA)
     (:stage_1_success        CONDITION_FORMULA)
     (:stage_1_failure        CONDITION_FORMULA)
     (:stage_1_else           CONDITION_FORMULA)
     (:action1                ACTION_FORMULA)
     (:action2                ACTION_FORMULA)...)
    ((:stage_2_precondition   CONDITION_FORMULA)
     ...)   ...
    ((:stage_M_precondition   CONDITION_FORMULA)
     ...))
  [(:effect EFFECT_FORMULA)])
```

Fig. 3. A PDDL-like Plan Structure in PSP

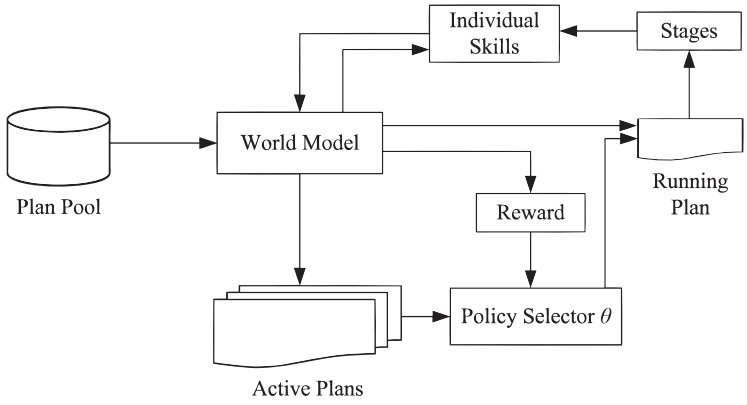

Fig. 4. Learning Process in PSP algorithm

In the PSP algorithm, a plan is actually a cooperative strategy. We can define plenty of offline plans to establish a *plan pool*, which is essentially an expert knowledge database. If the external state satisfies the precondition of a plan, the plan will be called an *active plan*. At time t, if there is only one active plan, it will be marked as the *running plan* and actions will be executed stage by stage. However, along with the growth of the plan pool, multiple active plans may appear at the same time.

Previous solutions [3, 4] chose a plan randomly, which is clearly a decision without intelligence. Q-learning is apparently a wiser approach, but unfortunately Q-learning is difficult to adopt in generalised decision architectures because all the plans cannot guarantee activation.

In this paper we employ another reinforcement learning method, policy search, to overcome this difficulty. The learning framework is illustrated in Figure 4. Due to the space, we are unable to extend the details of PSP method, but we've submitted a separated paper for RoboCup symposium 2008.

4 Experiments in RoboCup

RoboCup simulation is a suitable application to evaluate PSP algorithm because of the following three reasons: first, human have knowledge on football and need to apply it to robots; second, intelligent cooperation is urged and third, there exist no universal policy and so adaptive and adversarial strategies are required.

In order to show universality, plans are presented in a RoboCup coach language $CLang$ [5]. Due to the space, I am unable to demonstrate the structure of a plan in the form of $CLang$.

In our experiments, we created two opponent teams $OppA$ and $OppB$ which tend to defend from side and centre respectively. All the plans have three features, which are the extents to which a plan will change the ball or player positions towards the left sideline, right sideline, and the opponent goal respectively.

Our experiments consist of three parts. In our first experiment, 30 plans were defined in the plan pool, and agents play against $OppA$. In order to verify the robustness of PSP, an additional 15 plans were defined in our second experiment

under the legacy policy from the first experiment. *OppB* is also used to test adaptivity in our third experiment. Our experimental results reveal robustness, adaptivity, and outperformance over other methods; and the experimental details are included in our submitted paper.

5 Conclusion and Future Work

We introduced the decision architecture of OxBlue2008. Under such a layered structure, we proposed a novel method called PSP in a generalised POMDP scenario, in which a large selection of cooperative skills can be presented in a plan pool; and policy search is used to find the optimal policy to select among these plans.

We briefly demonstrated why and how PSP can be used in RoboCup 2D Simulation, and experimental results show explicit robustness, adaptivity and outperformance over non-learning planning and non-planning methods. In our more general (unquantified) experience with PSP it appears able to find solutions for problems that cannot be solved in a sensible timescale using earlier methods.

PSP is our first attempt to learn the optimal cooperation pattern amongst multiple agents. Our future directions are two-fold. Under RoboCup we are planning to define more plans and more features for PSP in our OxBlue 2D and 3D teams to further verify the robustness of our method. As to a generalised POMDP, we are keen to explore how the different architectures of a planner can effect learning quality, and to verify its generality in other MAS applications.

References

[1] Perraju, T.S.: Multi agent architectures for high assurance systems. In: American Control Conference. Volume 5., San Diego, CA, USA (1999) 3154–3157

[2] Stone, P., Veloso, M.: Layered learning and flexible teamwork in robocup simulation agents. In: RoboCup-99: Robot Soccer World Cup III. (2000) 65–72

[3] Obst, O.: Using a planner for coordination of multiagent team behavior. Programming Multi-Agent Systems **3862/2006** (2006) 90–100

[4] Obst, O., Boedecker, J.: Flexible coordination of multiagent team behavior using htn planning. In: RoboCup 2005: Robot Soccer World Cup IX. (2006) 521–528

[5] Chen, M., Dorer, K., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y., Yin, X.: Robocup soccer server for soccer server version 7.07 and later (August 2002)