Coaching a Simulated Soccer Team by Opponent Model Recognition

Patrick Riley
Computer Science Dept
Carnegie Mellon University
Pittsburgh PA, 15213-3891
pfr@cs.cmu.edu

Manuela Veloso
Computer Science Dept
Carnegie Mellon University
Pittsburgh PA, 15213-3891
mmv@cs.cmu.edu

ABSTRACT

In multiagent domains with adversarial and cooperative agents, team agents should be adaptive to the current environment and opponent. We introduce an online method to provide the agents with team plans that a "coach" agent generates in response to the specific opponents. The coach agent is equipped with a number of pre-defined opponent models. The coach is then able to quickly select between different models online by using a naive Bayes style algorithm, making the planning adaptive to the current adversary. The coach uses a Simple Temporal Network to represent team plans as coordinated movements among the multiple agents and it searches for an opponent-dependent plan for its teammates. This plan is then communicated to the agents, who execute the plan in a distributed fashion. The system is fully implemented in a simulated robotic soccer domain.

1. INTRODUCTION

Multiagent domains can include team and adversarial agents. One of the main challenges of such domains is the coordination and response of teammates to the adversarial agents. This paper overviews our work to address this problem. We use the concrete simulated robotic soccer platform, which is a rich multiagent environment, including fully distributed team agents in two different teams of up to eleven agents. Our work is driven by the goal of significantly improving the performance of teams of agents through their adaptation and effective response to different adversarial teams.

This paper reports our work to gather and respond to the opponents' behaviors throughout the game. We specifically focus on responding effectively after the game is stopped, in what are known as setplay situations. Several preset setplay plans have been introduced which indeed provide great opportunities to position the teammates strategically and have been shown to impact the performance of a team. [7] In this work, we contribute *adaptive* setplays which change and improve throughout a game as a function of and in response to the opponent team's behavior.

We use our coach agent that compiles the necessary overall view of how the opponent team behaves. The coach communicates to its teammates a team plan which is executed and monitored in a fully distributed manner. The coach agent is equipped with a number of pre-defined opponent models. These models are probabilistic representations of predicted opponent movements. The models can then be matched to observed movements of the opponent agents. The whole process is as follows:

- When the game is stopped, e.g., due to an out-of-bound call, the coach rapidly takes advantage of the short available time to create a team setplay plan. The plan is generated through a hill-climbing search in plan space using an evaluation function that embeds the predictions of the opponent model perceived to be the most likely during the game. The plan generation, in addition to the recognition of the opponents' model, notably uses the model *to predict* the opponent agents' behaviors.
- In addition to this plan creation algorithm per se, one additional challenge of this research was the selection and use of an appropriate plan representation. The coach uses a Simple Temporal Network representation that effectively captures the temporal dependencies between the plan steps, and, most importantly, explicitly records bounds on the expected execution times of the actions.
- The setplay plans, as generated and delivered by the coach to the teammates, notably include the necessary information for the agents to execute and monitor the team plan in a fully distributed manner.
- The coach also observes the execution of the plan generated in order to update the selection of an appropriate model for the current opponent.

2. THE ENVIRONMENT

The system is fully implemented in the Soccer Server System[6] as used in RoboCup[3]. The Soccer Server System

[0]This work was largely undertaken at and supported by Draper Laboratory during the summer of 2000, while Patrick Riley was working there. This research was sponsored in part by the United States Air Force under Cooperative Agreement No. F30602-98-2-0135 and a National Science Foundation Graduate Research Fellowship. The views and conclusions contained in this document are those only of the authors.

AGENTS'01, May 28-June 1, 2001, Montréal, Quebec, Canada.

is a real-time, server-client system which simulates soccer between distributed agents. Clients communicate using a standard network protocol with well-defined actions. The server keeps track of the current state of the world, executes the actions which the clients request, and periodically sends each agent noisy, incomplete information about the world. Agents receive noisy information about the direction and distance of objects on the field (the ball, players, goals, etc.); information is provided only for objects in the field of vision of the agent.

3. PLANNING AND OPPONENT MODELS

A plan is represented as a Simple Temporal Network, as introduced in [1]. A Simple Temporal Network (or STN for short) is a directed graph that represents the temporal constraints between events. Each node in the graph represents an event, and each edge represents a temporal constraint between the events. The authors are not aware of any other work where STNs are used as a multi-agent plan representation. However, there are several features of STNs which make them appealing for multi-agent plans.

- The networks naturally represent the parallelism of agents' actions and the temporal constraints express some basic needed coordination between the agents.
- The temporal constraints can be used to detect failures in execution.
- The networks can help the agents solve the Monitoring Selectivity Problem[2] since it contains information about which events are important for each agent's execution path.

Our execution algorithm extends the dispatching execution algorithm by Muscettola *et al*[5], to the multi-agent case. We use the dispatching algorithm (unchanged) to maintain information about time bounds for executions of events. The agents monitor relevant nodes for completion, monitor and respect temporal constraints, and work towards executing the next node required of them. Some nodes also have constraints on their execution. For example, the agent must believe that a pass will succeed before beginning it.

In short, we use the STN plan to track two separate questions: "What should each agent be doing at each time step?" and "What have all the other agents done up to this time step?" In most STN applications, only the first question is addressed; the executor is assumed to be the only agent. Morris and Muscettola[4] have considered allowing there to be some uncertainty in the temporal execution of some events. However, they are more concerned with guaranteeing that execution will be successful, while we are more interested in the optimistic possibility that there is still some valid execution in the given temporal constraints. Also, they do not address the multi-agent aspects which are important to this work.

The plan creation process is divided into two steps. First, a path is planned for the ball, then an STN is generated from that path with roles for the agents and the necessary temporal constraints specified. A model of the opponents is used to probabilisticly predict the movements of the opponent. The path planning problem then has constraints of straight-line segments and dynamic, probabilistic obstacles. We use a hillclimbing approach with an evaluation function that uses the following weighted factors: whether the last segment is a pass or clear, the ball's final location, the length of the path, and the average and maximum pass danger. The pass danger is defined in terms of a the probabilistic locations of the opponents.

Conceptually, we want an opponent model to represent how an opponent plays defense during setplays. In order to handle the uncertainty in the environment and to allow models to represent more information than just a simple predicted location, we use probabilistic models. Given the recent history of the ball's movement (from the start of the set play for example) and the player's initial location, the model gives, for each player, a probability distribution over locations on the field. Given a set of opponent models, we use a naive Bayes classifier to determine which opponent model best matches the opponent.

4. CONCLUSION AND FUTURE WORK

In the games at RoboCup2000, it was evident that our team benefited from adaptive setplays. Our system created a variety of setplay plans in adaptation to completely unknown opponent teams. More targeted empirical comparison on the impact of the technique is challenging but it is part of our ongoing work.

During execution, the agents do not take advantage of opportunities which may occur. For example if an agent ends up with a good shot on the goal, but the plan is for it to pass, it will pass the ball anyway. Storing alternate plans and intelligently adding monitors for these plans as in [8] could make the plan execution usefully opportunistic.

Also, the Simple Temporal Network plan representation could benefit from more explicit synchronization in terms of the observations expected for particular nodes.

In this work, the opponent models were written by hand, but the algorithms here could work with models which are learned by observation of opponents. We plan to continue to explore the area of generating and using opponent models for fast recognition of behaviors in order to further exhibit adaptive, intelligent responses in our agents.

5. REFERENCES

[1] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[2] G. A. Kaminka and M. Tambe. Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, 12:105–147, 2000.

[3] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada. The robocup synthetic agent challenge. In *IJCAI-97*, pages 24–49, San Francisco, CA, 1997.

[4] P. Morris and N. Muscettola. Execution of temporal plans with uncertainty. In *AAAI-2000*, pages 491–496. AAAI Press/The MIT Press, 2000.

[5] N. Muscettola, P. Morris, and I. Tsamardinos. Reformulating temporal plans for efficient execution. In *KR-98*, 1998.

[6] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[7] P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In Veloso, Pagello, and Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, pages 35–48. Springer, Berlin, 2000.

[8] M. Veloso, M. Pollack, and M. Cox. Rationale-based monitoring for planning in dynamic environments. In *AIPS-98*, Pittsburgh, PA, 1998.