Bare Demo of IEEEtran.cls for Conferences

Michael Shell
School of Electrical and
Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678-2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—Hereby we present methods and algorithms we used in order to compete in the Domination Game. Constructing effective methods and strategies for our agents has been a quite challenging task, given a multi-agent partially observable environment. We will discuss how fundamental problems were dealt with, such as the representation of states and decision making, but we will also explain how different mechanics we implemented for the agents movement and shooting accuracy improved their performance. Finally, we will present a learning method we have implemented as a test case, comparing it to predefined strategies and analyze the results.**

I. Introduction

Multi agent reinforcement learning has been and still is an active area of research of Artificial Intelligence. Algorithms and methods describing multi agent systems are applied in several domains, such as robotic teams, distributed control, data mining, etc.[1]. In extension to that, partially observable multi agent systems have also been targeted by researchers as a realistic and very challenging category of problems. The Domination Game (DG) can be described as a multi agent partially observable system. Each team competing in the DG has to be fully coordinated and constantly communicating given the fact that the agents have a limited range of sight and do not have full knowledge of all the worlds objectives at all times[1]. Hence, in order to behave optimally, a teams agents need to share knowledge, in order to tackle partial observability, and learn optimal strategies by deciding joint actions. Although effective algorithms for optimal decision making in partially observable multi agent systems already exist, implementing such in the DG is not an easy task. In order to implement such algorithms, we have to solve a set of problems that compose the DG.

- How are states represented?
- Which are the possible actions?
- Which is the optimal strategy?

In order to represent a state the game is in, we could use numerous features. First of all, we have to define whether each agent will keep notice of his own state or will all the agents states compose one higher state that describes the game at that particular timestep? Secondarily, we have to define which features will compose each state. Combining the agents possible position, all his foes positions, the state of the control points and ammo points, would give us a huge state space which is computationally very expensive to rely on. As a consequence, we have to think of ways to reduce the state space enough to describe it in an efficient way. The possible action set also has to be defined in a smart way. To start with, we have to define points of interest on the map which we consider strategically important, and define as actions, movements towards these points of interest. However, depending on the strategy used, the agent might have to hunt down foes in order to temporarily remove them from the game so, this has to be implemented separately. Finally, defining an optimal strategy is also a minor problem that has to be solved. Do we favor strategies that dominate control points, or is dominating the ammo spots the way to win? Both strategies had to be tested and maybe combined in order to achieve optimal results.

II. Related Work

Numerous algorithms have been implemented regarding multi agent learning problems, such as Nash-equilibrium-based methods, Q-Learning and portfolio algorithms[2]. Some methods like Q-Learning were designed for single agent learning and then adapted to multi agent environments.

Paper [2] compares applied, promising and newly proposed algorithms and applies them on several classical games, such as the climbing game and the prisoner's dilemma. Results show that algorithms applied like UCB and M3 outperform well-known learning methods like Q-Learning and Minimax.

Paper [3] discusses how multiple agents choose strategies for optimal joint actions in a cooperative environment such as the prisoner's dilemma game. Different approaches and strategies are compared, emphasizing on the tendency of the agents to cooperate after a certain number of games, in order to achieve maximum rewards. In the case of DG, a team's agents have to be fully cooperative in a sense that, if they were to act based on individual reward functions, they might end up chasing the same control points and thus, losing the game.

Furthermore, paper [4] discusses algorithms applied onto a game very similar to DG, called the "tracking agents". Although the authors discuss planning methods, the similarity to DG is that a single "target" agent is trying to escape from

[1] We name world objectives, the existence of ammo and the position of all enemy agents.

”tracker” agents in a partially observable environment (formalized as a ”multi agent pursuit scenario”). The tree search algorithms implemented (such as depth-limited Minimax) are trying to maximize cooperation between the trackers using 3 different heuristic methods.

Although numerous algorithms can be found to work efficiently on a partially observable multi agent scenario, the most challenging task is to successfully reduce the state space. Since the DG is not considered a ”classical” game such as the prisoner’s dilemma, it is up to us to try and adapt such learning algorithms on it. The algorithm we tried is proven to work efficiently on such scenarios, given that the representation of states is precise and computationally feasible.

Acknowledgment

The authors would like to thank...

References

[1] L. Busoniu, R. Babuska and B. De Schutter, *A Comprehensive Survey of Multiagent Reinforcement Learning*, IEEE Transactions on systems, MAN, and Cybernetics, Part C: Applications and Reviews, vol. 38, No. 2, March 2008.

[2] B. Bouzy, M. Metivier, *Multi-agent Learning Experiments on Repeated Matrix Games*, Universite de Paris.

[3] F. Schweitzer and R. Mach, *Distribution of Strategies in a Spatial Multi-Agent Game*, ETH Zurich.

[4] E. Raboin, D. Nau,U. Kuter, S. K. Gupta and P. Svec, *Strategy Generation in Multi-Agent Imperfect-Information Pursuit Games*, University of Maryland.