Taking Turns in General Sum Markov Games

(Extended Abstract)

Peter Vrancx*
Computational Modeling Lab
Vrije Universiteit Brussel
Brussels, Belgium
pvrancx@vub.ac.be

Katja Verbeeck
Information Technology
KaHo St. Lieven (KULeuven)
Ghent, Belgium
katja.verbeeck@kahosl.be

Ann Nowé
Computational Modeling Lab
Vrije Universiteit Brussel
Brussels, Belgium
ann.nowe@vub.ac.be

ABSTRACT

This paper provides a novel approach to multi-agent coordination in general sum Markov games. Contrary to what is common in multi-agent learning, our approach does not focus on reaching a particular equilibrium between agent policies. Instead, it learns a basis set of special joint agent policies, over which it can randomize to build different solutions.

The main idea is to tackle a Markov game by decomposing it into a set of multi-agent common interest problems; each reflecting one agent's preferences in the system. With only a minimum of coordination, simple reinforcement learning agents using Parameterised Learning Automata are able to solve this set of common interest problems in parallel. As a result, a team of simple learning agents becomes able to switch play between desired joint policies rather than mixing individual policies.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms

Keywords

agent coordination, learning agents, Markov games, reinforcement learning

1. INTRODUCTION

A large part of the multi-agent learning literature focuses on finding a Nash equilibrium between agents' policies [5]. However, while a Nash equilibrium represents a local optimum, it does not necessarily represent a desirable solution for the problem at hand. This is clearly demonstrated by the famous prisoner's dilemma game, where the unique Nash equilibrium does not represent a desirable outcome, since both agents can simultaneously do better.

In this paper we are interested in agents learning realistic patterns of behavior. One such pattern, which occurs naturally in human interactions is turn taking. When faced with conflicting interests, humans can often compromise by agreeing to take turns to select each participant's desired result. This compromise frequently leads to outcomes that are more desirable than those reached by a population of agents selfishly optimizing their individual rewards. In [3] the problem of learning to take turns is studied in repeated games. The goal of this paper is to show how simple reinforcement learning agents are able to learn interesting patterns of play, like turn taking, in general sum Markov Games.

We present a new multi-agent coordination approach for learning patterns of desirable joint agent policies. To do so, we depart from the idea of jointly learning an equilibrium in the full Markov game. Instead, our main idea is to tackle a Markov game by decomposing it into a set of multi-agent common interest problems called Multi-agent Markov Decision Processes (MMDPs); each reflecting one agent's preferences in the system. Reinforcement learning agents using Parameterised Learning Automata [2] (PLA) are able to solve this set of MMDPs in parallel. As a result, a team of simple learning agents can switch play between these desired joint policies. A trusted third party is used to communicate simple coordination signals to regulate the switches. In case all agents can fully trust their opponent players, this third party is even unnecessary.

2. MARKOV GAME DECOMPOSITION

The main idea behind our algorithm is to split the Markov game into a number of common interest problems, one for each agent. These problems are then solved in parallel, allowing agents to switch between different joint policies, in order to satisfy different agents' preferences. The agents learn the preferred outcome for each participant in the game. We develop a system in which agents alternate between optimising different agent goals in order to satisfy all agents in the system. This implies that we let agents switch between playing different joint policies.

To implement this mechanism, a single recurrent state in the system is select as the *switch state*. Play is then divided in episodes, with a single episode comprising the time-steps between 2 subsequent visits to the switch state. Each episode a different objective to be optimised can be selected, with the different objectives here corresponding to

*funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders.

Cite as: Taking Turns in General Sum Markov Games (Extended Abstract), Peter Vrancx, Katja Verbeeck and Ann Nowé, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1439-1440

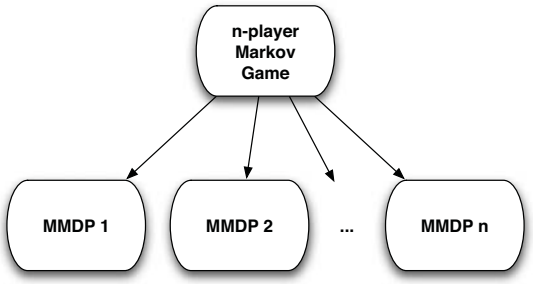

Figure 1: Markov game decomposition

the reward function of different agents.

Thus agents do not only learn their own policy, they also learn the preferences of the other agents. To do this, agents associate multiple learning automata (LA) with each state, one for each agent in the system. One automaton is used to learn their own policy, while the others are used to learn the preferences of the other agents. When they start a new episode, focussing on the next agent's goal, they use and update the automata corresponding to that goal in each state until the episode ends. At the end of the episode the agents all receive the reward corresponding to agent of the past episode.

This system effectively transforms the Markov game into a set of MMDPs. Each MMDP represents the problem of finding the joint policy that maximises the reward for a single agent in the system. By switching between different automata to learn different agents' preferences, the agents are actually solving each of the MMDPs in an interleaved manner, using the algorithm described in the previous section. Using the PLA update system, the agents will find the optimal joint policy in each MMDP, which corresponds to the joint policy maximizing the reward for the corresponding agent. Thus when the automata have converged, the agents continuously alternate between the stationary joint policies that are optimal for the different agents in the system.

By implementing a coordination mechanism, agents also correlate their policy choices. This means that agents are not limited to the product distributions, given by individually mixing policies. Instead, agents using this system to coordinate their policy switches, can play only desired joint policies, rather than the entire cross product of their individual policy sets. This allows them to only reach desirable outcomes of the game, in this case the joint policies that maximise the reward for one of the participating agents.

3. COMBINING JOINT POLICIES

Using the switching mechanism described above we can learn the joint policies that maximise each agent's individual payoff. One additional requirement to implement this system is a mechanism to decide which MMDP will be played next. This mechanism determines how the different joint policies learnt in the set of MMDPs are combined into a single solution and consequently how much each agent's goal is optimised.

In our setting we implement this switching mechanism using a separate dispatcher agent. This agent is separate from the other agents and does not participate in the actual learning problem. Instead this agent coordinates all other agents and determines the reward to optimise next. In this way the actual learning agents do not need information on the actions and rewards of others or even the fact that other agents are present in the system. Whenever the system reaches the switch state, the current episode ends and the dispatcher becomes active. The dispatcher then collects the total rewards up to the current time step for each agent and sends each agent in the system 2 pieces of information: a feedback for the last episode and the index for the next problem to be played. The learning agents themselves do not need to know whose reward they are optimising, they can simply use the index to select the corresponding automata during the next episode.

Currently, we focus on an *egalitarian* selection mechanism. This means we try to maximise the minimum of the players' rewards and the dispatcher will always choose to optimize the payoff of the worst performing agent, i.e. the agent with the lowest average reward over time for the entire running time. In this way we can resolve dilemma's resulting from agents having different preferences for the game outcomes, by letting them take turns to play their optimum outcome. This allows each agent to achieve their desired objective at least some of the time and assures that no agent will always be stuck with its minimum payoff. In [4] we demonstrate these results on a set of motivational examples.

4. DISCUSSION

In this paper we introduced a multi-agent learning algorithm which allows agents to switch between stationary policies in order to equalize the reward division among the agent population. In the present system we rely on a dispatcher agent to select the objective to play and the correlate agents' policy switches. If we assume that all agents are cooperative and willing to sacrifice some payoff in order to equalize the rewards in the population, this functionality could also be embedded in the agents, either by letting agents communicate or allowing each agent to observe all rewards . In systems where agents cannot be trusted or are not willing to cooperate, methods from computational mechanism design could be used to ensure that agenst' selfish interests are aligned with the global system utility. Another possible approach is considered in [1], where the other agents can choose to punish uncooperative agents.

5. REFERENCES

[1] S. de Jong and K. Tuyls. Learning to cooperate in public-goods interactions. In *EUMAS 2008*, 2008.

[2] M. Thathachar and V. Phansalkar. Learning the global maximum with parameterized learning automata. *Neural Networks, IEEE Transactions on*, 6(2):398–406, 1995.

[3] P. Vanderschraaf and B. Skyrms. Learning to take turns. *Erkenntnis*, 59:311–347(37), November 2003.

[4] P. Vrancx. *Decentralised reinforcement learning in Markov games.* PhD thesis, Computational Modeling Lab, Vrije Universiteit Brussel, 2010.

[5] P. Vrancx, K. Verbeeck, and A. Nowe. Decentralized learning in markov games. *IEEE Transactions on Systems, Man and Cybernetics (Part B: Cybernetics)*, 38(4):976–81, 2008.