Multi-agent Reinforcement learning for Planning and Scheduling Multiple Goals

Sachiyo Arai Katia Sycara Terry R. Payne
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213 USA
E-Mail: {sachiyo, katia, terryp}@cs.cmu.edu

1. Introduction

Recently, reinforcement learning has been proposed as an effective method for knowledge acquisition of the multiagent systems. However, most researches on multiagent system applying a reinforcement learning algorithm focus on the method to reduce complexity due to the existence of multiple agents[4] and goals[8]. Though these pre-defined structures succeeded in putting down the undesirable effect due to the existence of multiple agents, they would also suppress the desirable emergence of cooperative behaviors in the multiagent domain. We show that the potential cooperative properties among the agent are emerged by means of Profit-sharing[2][3] which is robust in the non-MDPs.

2. Extended Pursuit Game

This paper uses an extended Pursuit Game where there exist multiple preys and multiple hunters as shown in Figure1(a). Each hunter is assumed to be a learning agent, whereas the prey does not learn and moves randomly in the environment which consists of triangular cells to reduce the size of the state space where three hunters are required to capture a prey. A hunter can know the location of a prey only when the prey is in the hunter's sight which is limited as shown in Figure1(b). The sight of hunter is decomposed into 15 different areas and each area represents its status in terms of {vacancy, existence of the hunter, existence of the prey}(*Note: other hunters and preys are distinguishable from each other.*) The final goal of the agents is to capture all the preys in the environment. Under these conditions, the hunters need not only to find the path to the prey but also to decide each target prey which should be common to the hunters. As far as finding a path to the prey, the hunters must come close to the target prey. On the other hand, deciding which prey to target for capture requires additional cooperation to form consensus on the sequence of capturing the preys. Therefore, we need to take *perceptual aliasing problem* and the *agents' concurrent learning* [5][1] into consideration.

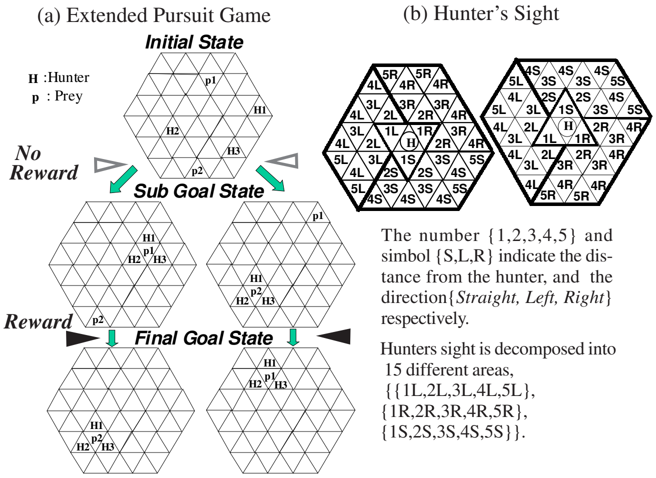

Figure 1. Pursuit Game of Multiple Preys

3. Profit-sharing Approach

The most important difference between Profit-sharing approach and the DP-based reinforcement learning algorithms, such as Q-learning[7] and Temporal Difference[6], is that Profit-sharing does not use *One-step backup* and not need eligibility trace to treat a delayed reward. Therefore, it is robust against the problems due to the existence of multiple agents, such as *concurrent learning*and *perceptual aliasing*. In addition, it can save a required memory-space because it does not need to keep eligibilities and whole state-spaces which the agent experienced.

First, when a hunter observes current state o_t, it checks its lookuptable to search the matched state as o_t and gets its action set $A_t = \{left, right, straight.stay\}$, which consists of available actions at time t. The action is selected by the roulette selection, soft-greedy method, in which the selection rate of the action is in proportion to its current weight. This selection method makes hunter behave under the stochastic policy and explore its strategy. After hunter outputs the selected action a_t, it checks if a reward is given or not. If there is no reward after an action a_t , the hunter stores the state-action pair (o_t, a_t) into its episodic-memory as a *rule*, and continues the same cycles until getting the reward R. We call the period from the start to the getting R, an *episode*.

Second, when the hunter got the reward R, it reinforces rules which are stored in the episodic-memory according to a credit assignment function $f(R, t)$ which satisfies the "Rationality Theorem[3]". For example, the geometrically decreasing function $f = R(1/(L-1))^{T-t}$ (T: time at goal, t=0: time at initial state) is satisfied this Theorem[1]. The gist of this Theorem is that the reward should not be given to an ineffective action which makes agent move in a loop path more than to the effective action which makes the agent move straight.

4. Experiments and Discussion

Environments 3 Hunters ,3 Preys	Required Number of Steps Av.(S.D) After **5,000** Episodes	After **50,000** Episodes	After **500,000** Episodes	After **1,000,000** Episodes
To capture 3 preys Without Global	762.9(186.4)	155.1(7.2)	84.5(3.4)	76.2(1.8)
To capture 3 preys With Global	136.4(32.6)	69.3(7.8)	59.6(7.1)	59.4(6.4)
To capture 1 prey With Global	49.5(9.8)	28.6(4.3)	24.7(3.7)	22.1(2.6)
*C.f. To capture one Prey in the H3P1**	35.6(7.6)	22.3(3.9)	19.3(3.2)	17.0(1.8)

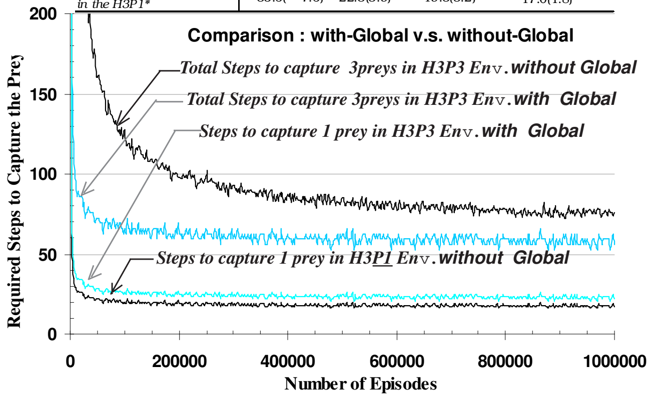

Figure 2. With-Global v.s. Without-Global

We use function $f(R, t) = R(0.3)^{T-t}$ to assign a reward(R) to each state-action pair of the episodic-memory. In each experimental condition, hunters learn 1,000,000 episodes as a trial, the lookup table of each hunter is reset after each trial, and iterated 10 trials to evaluate the average and standard deviation.

To evaluate performance of the hunters without global knowledge, we compared with the baseline condition in which a single global-agent schedules the ordering of prey capture. In this case, the global-agent is given the information about the location of all the preys and hunters, then selects a target prey by $Prey_j = \arg\min_{j \in preys} \sum_i distance(Prey_j, Hunter_i)$. Then, all hunters focus on the target prey, which the global-agent decided to capture, and neglect the other preys. In this case, a hunter ignores the other preys although they could be in its sight. After capturing the 1st prey, the global-agent decides the next target and hunters repeat the same procedure until capturing whole preys.

Figure2 shows the learning curves of the required steps to capture the 3 preys and 1 prey, labeled H3P3 and H3P1 respectively. The x-axis indicates the number of episodes and the y-axis indicates the average of required steps in 10 trials. The *with-global-agent* condition shows more effective performance than *without-global-agent* to capture whole preys because the hunters' target is always consistent among them. In the *with-global-agent* method, the state space size of each hunter's is constant $(((Hunters - 1) + 1))^{15}$), regardless of number of preys. And also the acquired policy of capturing the 1st prey could reuse to capture the second and third prey. However, what we notice here is that the required steps to capture the *1 prey* in the H3P3-with-global is larger than that in the H3P1-without-global condition. This fact implies that hunters in the H3P3-with-global seem to be thrown into a kind of perceptual aliasing and to be compelled them to move unnatural way because they are concealed non-target prey from their sights. And in with-global method, the hunters could not pursue multiple preys opportunisticly which is realized in the without-global-method.

References

[1] Arai, S.; Miyazaki, K.; and Kobayashi, S. 1997. Generating Cooperative Behavior by Multi-Agent Reinforcement Learning. In *Proceedings of 6th European Workshop on Learning Robots*, 111–120.

[2] Grefenstette, J. 1988. Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms. *Machine Learning* 3:225–245.

[3] Miyazaki, K.; Yamamura, M.; and Kobayashi, S. 1994. On the Rationality of Profit Sharing in Reinforcement Learning. In *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, 285–288.

[4] Ono, N.; Fukumoto, K., and Ikeda, O. 1997. Collective Behavior by Modular Reinforcement Learning Animats *Proceedings of the 4th International Conference on simulation of Adaptive Behavior*, 618–624.

[5] Sen, S., and Sekaran, M. 1995. Multiagent Coordination with Learning Classifier Systems. In Weiss, G., and Sen, S., eds., *Adaption and Learning in Multi-agent systems*. Berlin, Heidelberg:Springer Verlag. 218–233.

[6] Sutton, R. 1988. Learning to Predict by the Methods of Temporal Differences. *Machine Learning* 3:9–44.

[7] Watkins, C., and Dayan, P. 1992. Technical note: Q-learning. *Machine Learning* 8:55–68.

[8] Whitehead, S. D. 1993. Learning Multiple Goal Behavior via Task Decomposition and Dynamic Policy Merging. In J.H.Connell, *Robot Learning*, Kluwer Academic Press, 45-78.

[1]common ratio is decided by $1/(L-1)$ (L: the number of available actions at each time step).