The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Acquisition of Stationary Behavior Based on Multiagent Enforced SubPopulations

Joji Kato[1], Shu Hosokawa[2], and Kazushi Nakano[1,2]

[1]Dept. of Mechanical Eng. and Intelligent Systems, The University of Electro-Communications, Japan
[2]Dept. of Electronic Eng. , The University of Electro-Communications, Japan
(Tel: 81-42-443-5190, Fax: 81-42-443-5183)
(kato@francis.ee.uec.ac.jp)

Abstract: This paper presents a solution for the problems of state representation as well as a variety of optimal solutions in multiagent systems. These problems cannot be solved by traditional reinforcement learning methods such as Sarsa(λ). We apply a method of Multiagent Enforced SubPopulations to the task of stationary behavior acquisition. The stationary behavior acquisition task means that the agents continue to select behavior in order to keep the stationary state. The behaviors of acquired agents are not uniquely determined in those tasks. In addition, there is a state representation problem in multiagent systems due to the complexity of the system. Furthermore, there is no example of the Multiagent Enforced SubPopulations applied to these types of tasks, in which the design policy of the fitness function is unclear. We demonstrate the validity of our proposed method through comparison with a Keepaway task in RoboCup Soccer as a stationary behavior acquisition task.

Keywords: Multiagent system, Multiagent Enforced SubPopulations, Fitness function, Keepaway

1 INTRODUCTION

The researches on multiagent systems aim to solve more complex problems with the cooperation of multi-robots. However, the multiagent systems are difficult to control agents by top-down methods which enumerate all behaviors, because it is necessary for us to consider mutual behaviors between agents. Therefore, bottom-up methods teaching with fitness in each behavior should be proposed.

There are unsupervised reinforcement learning methods, for example, Q-learning [6], Sarsa(λ) [7] as bottom-up methods. These methods achieve the desired state by selecting a behavior to maximize an expected reward which will be received in each decision. The Q-learning guarantees convergence to an optimal value in the Markov decision process, and there are many examples in its application. However, these applications need to properly divide the state to satisfy the Markov property. In general, the amount of state divisions increases exponentially because the number of states in multiagent systems increases over that of single-agent systems. There are traditional research studies about state division methods such as tile-coding, but they cannot fundamentally resolve the problems. In addition, the optimal solution is not uniquely determined in multiagent systems, and these methods do not consider about a variety of optimal solutions.

There is the Multiagent Enforced SubPopulations (ESP) method as an evolutionary neural network method which is different from Q-learning or Sarsa(λ). The Multiagent ESP method is expected to solve the problems of state representation and of a variety of solutions which cannot be solved by traditional reinforcement learning methods, due to a combined advantage of neural networks and genetic algorithms. However, the Multiagent ESP is applied only to a prey catch task scheme, and also there is no discussion about the design of fitness functions.

In this research, the Multiagent ESP apply to a Keepaway task which is a sub-task of the RoboCup Soccer Simulation League. The aim of this task is to keep a ball as long as possible against 2 "Takers" by cooperating with 3 "Keepers". The existing research uses traditional reinforcement learning methods such as Sarsa(λ). So, the methods and accuracy of state division influence learned behavior. In addition, this task continues to select behavior in order to keep a stationary state while controlling the ball. We call these behaviors as stationary behavior. It is natural that there are more than one solution in the achieved stationary behavior. Therefore it is difficult to get a unique solution. We demonstrate the validity of our proposed method by applying the Multiagent ESP to the Keepaway task. Our proposed method is better than traditional reinforcement learning methods in multiagent tasks beyond the prey catch task. Lastly, we summarize an experimental consideration on fitness function design in the case where the Multiagent ESP is applied to the acquisition of a stationary behavior as in the case of Keepaway task.

2 BACKGROUND AND RELATED WORKS

We describe Multiagent ESP method underlies bases for our proposed method, and describe Keepaway task which is an application of our proposed method.

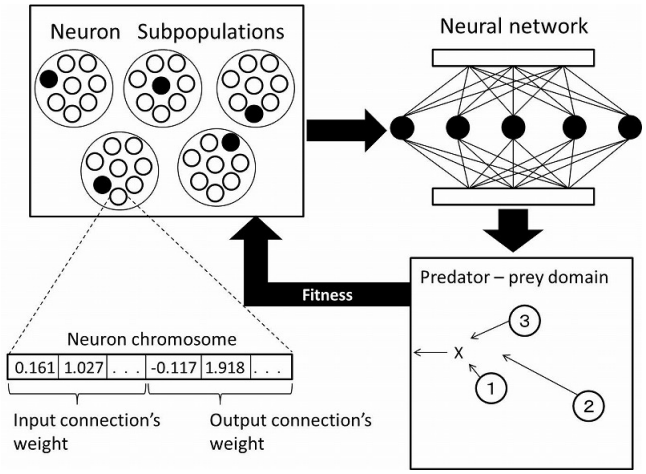

Fig. 1. ESP method applying to prey catch task

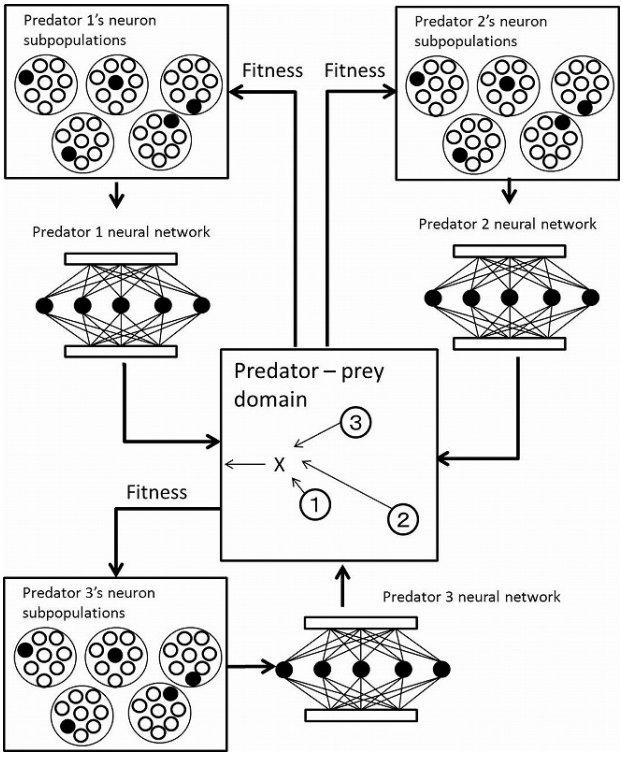

Fig. 2. Multiagent ESP method applying to prey catch task

2.1 Multiagent ESP method in prey catch task

Multiagent ESP method is a coevolution method in multiagent systems, which is proposed by C. H. Yong, R. Miikkulainen [3]. Next, Multiagent ESP method is of ESP method expanded to multiagent systems. ESP method is proposed by F. Gomes, R. Miikulaiinen [4], [5]. So, we explain ESP method and Multiagent ESP method through the prey catch task [3].

The prey catch task is a task that "Takers" chase and catch the "Prey". One task is finished when Takers catch a Prey, or time is over. The symbols ①, ②, ③ are Takers, and X is Prey in Fig. 1 showing ESP method's framework. One of Taker's policy is coordinated with learning in ESP method. The controller for learning Taker is 3 layered neural network. The input of this neural network is a difference between my current position and Prey. The output is moving direction for Taker. Hidden layer neuron has a weight of network. We define neuron subpopulations which have some different connection weights in each hidden neuron. The agent controller is constructed by random-selected neuron from neuron subpopulations in each task. If the agents catch a Prey in limited time by using this controller, it gets a constant fitness. In contrast, if the agents miss catch a Prey by using this controller, it gets more lower fitness based on average distance between Takers and a Prey. The fitness is accumulated in neuron subpopulations by repeating this task. As a result, high fitness neurons are selected by genetic algorithm, then neuron subpopulations are constructed by high fitness neuron. Lastly, we can get the controller to solve the task.

Next, we explain Multiagent ESP method which is one expansion of ESP method. All takers learn at one time to solve the task in Multiagent ESP method. Figure 2 shows a structure in which Multiagent ESP is applied to the prey catch task. The controller of each agent is of the same structure with ESP method, and is prepared with the same number as learning agents. Each controller is coordinated independently by ESP method. The calculation of fitness is same as ESP method, and there are no difference between agents.

As a property of ESP method, the input of controller can take continuous value. It is different from Q-learning or Sarsa(λ) which need state division. So, it can be expected to solve the state representation problems essentially. And also, ESP method has more than one optimal solutions in the form of neuron subpopulations after learning. As a result, it can be expected to get a class of optimal solutions in the task that the optimal solution cannot be determined.

In the existing researches which apply to prey catch task, the aim of all Takers is to go around the Prey as quickly as in order to achieve the task. This task easily determines the aim of each agent to achieve the task. And also, it is proper to hypothesize that there are a unique optimal solution. However, there are some tasks which cannot determine clearly the aim of each agent for achieving the task, and cannot determine a unique optimal solution in multiagent systems. In this research, multiagent ESP method will be applied to Keepaway task as an example of a task that cannot determine the aim of each agent and a unique optimal policy. So, we discuss about the design of fitness function.

Table 1. Agent selectable macro behaviors

Agent Role	Selectable Macro Behavior
Passer	Hold Ball, Pass Ball K_2, Pass Ball K_3
Receivers	Get Open, Go To Ball
Takers	Go To Ball, Block Pass K_2, Block Pass K_3

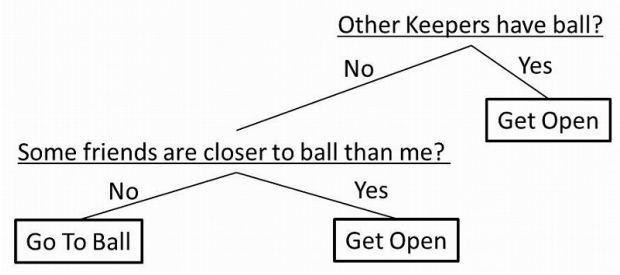

Fig. 3. Receiver's policy

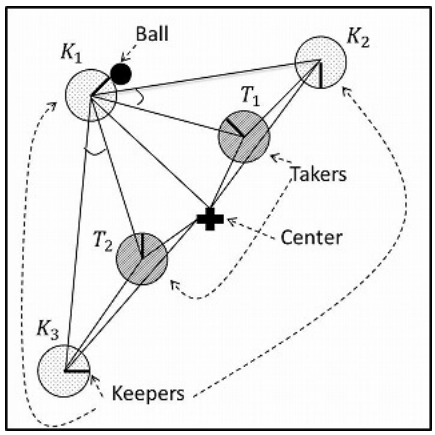

Fig. 4. State variable in Keepaway task

2.2 Keepaway task

Keepaway task is a testbed of reinforcement learning which is used in RoboCup Soccer Simulation League. This task is proposed by P. Stone, R. S. Sutton, G. Kuhlmann [1]. The aim of this task is that 3 Keepers keep a ball against 2 Takers as long as possible. Keepers are numbered K_1, K_2, K_3 in order of the distance to ball. Takers are numbered T_1, T_2 in order of the distance to K_1. And each agent have a specific ID such as K^1, K^2, K^3, T^1, T^2. K_1 is called "Passer", and he can actively control a ball. The other Keepers are called "Receiver". Task start position of Passer is top-left, and the positions of Receivers are top-right and bottom-right, and the positions of Takers are bottom-left. The task continues until fail in ball keeping. One task called one episode.

Keepers and Takers select a macro behavior once every 100 milliseconds. Here, we summarize macro behavior which each agents can select as shown in Table 1. Hold Ball behavior means to keep a ball at the current position. Pass Ball behavior means to pass a ball to a Receiver. Get Open behavior means to go to an open space. Go To Ball behavior means to go near to the ball. Block Pass behavior means to cut a pass course of the target (Keepers).

Learning objects in this task is Passer's policy. Passer is one of the Keepers who nearest to the ball, and is changed off in this task. Each Keeper has a controller learning independently, so he needs a coevolution with the others. The policies of agents without Passer are designed by human. First, Fig. 3 shows a Receiver's policy. Receivers select Get Open when friends have a ball, or friends is nearer to ball than me. They select Go To Ball in the other state. We prepare two types of Taker's policies. The first type policy is that all Takers select Go To Ball. This policy is often used in many existing researches. The second policy is that one agent selects Go To Ball, the other agent selects Block Pass. There are few researches on the second policy, which is more difficult than the first.

The lines in Fig. 4 shows a state variable which Passer can know during the episode.

Existing researches on Keepaway task uses traditional reinforcement learning method such as Sarsa(λ) [1], [2]. Sarsa(λ) algorithm needs to divide environment, but it cannot get enough state representation due to the limit of methods and accuracy of discretization in multiagent systems. The optimal solution is unique according to Sarsa(λ), but it is natural that there are more than one optimal solutions in the task which needs to continue to select stationary behavior such as this task. We think that these problems are occurred by Sarsa(λ) algorithm. In this research, Multiagent ESP method is applied to Keepaway task to improve the episode duration time.

3 MULTIAGENT ESP APPLY TO KEEPAWAY TASK

In this section, we explain how to apply the Multiagent ESP method to a Keepaway task. First, we explain configuration of our system, and next explain the design of fitness function in the Keepaway task. Finally, we explain about process of generation change in the Multiagent ESP method.

3.1 Multiagent ESP method in Keepaway task

Figure 5 shows a configuration of the Multiagent ESP method applying to the Keepaway task. Each Keeper have an independent controller, and they independently evolve according to received fitness value.

Each agent controller is in the form of three layered neural network. Its input takes 13 state values, the output does 3

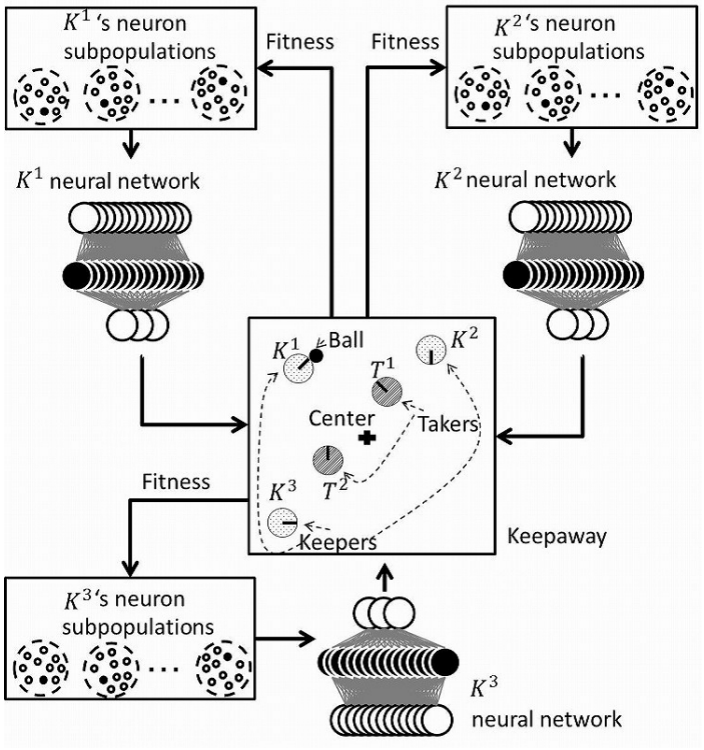

Fig. 5. Multiagent ESP method applying to Keepaway task

macro behaviors, and the number of hidden layer's neuron is 16. The calculation of activation of hidden and output neurons is made by use of the sigmoid function. Passer chooses the behavior of the highest activating neuron. The number of neuron belonging to neuron subpopulations in each agent is 100. These neurons are received fitness values calculated by the fitness function as shown in 3.2. The connection weights of controller are renewed to fit the task through the process of generation change as shown in 3.3.

3.2 Design of fitness function

In this section, we describe the design of fitness functions in the Multiagent ESP method applied to a Keepaway task. The fitness function shows how much the controller constructed by selected neurons in each episode fit to the aim of the task. Existing researches such as Sarsa(λ) give reward when Passer decides his own behavior. However, Passer's strategy can evaluate correctly at the end of task in the endless task such as the Keepaway task. The reward on the way of task is sub goal. But, it is difficult to correctly decide sub goals to achieve the aim of task. Furthermore, the research by Arai et al. shows that it is effective to combine reward and penalty to achieve the aim of task [2]. That is concluded that appropriate setting of some sub goals is effective to achieve the aim of the task. But this approach is reduced to a multipurpose optimal problem, and sometimes shows the inferior performance based on the combination of sub goals due to mutual actions.

Table 2. Sub goals and fitness functions

No.	Sub goal	Fitness
1	The purpose of making a ball to each agent as long as possible time to pass	Eq. (1)
2	The purpose of not relating with task failure	Eq. (2)
3	Our purpose to keep 5m distance from Passer to the nearest Taker	Eq. (3)

First of all, this research shows the difficulty to set some sub goals. Table 2 shows sub goals and fitness functions in own interests. Sub goal 1 is proposed by Stone et al. and has the purpose of making a ball to each agent as long as possible time to pass [1]. Sub goal 2 is proposed by Arai et al. and has the purpose of not relating with task failure [2]. Arai et al. shows the validity to combine sub goal 1 and 2 through experiments. Sub goal 3 is introduced to show the difficulty to set the sub goal by us. It is our purpose to keep 5m distance from Passer to the nearest Taker.

$$fitness = T_C - T_A \tag{1}$$

$$fitness = \begin{cases} -\frac{1}{T_E - T_A} & (\text{failure}) \\ 0 & (\text{otherwise}) \end{cases} \tag{2}$$

$$fitness = \begin{cases} \frac{dist}{5} & (0 \leq dist \leq 5) \\ \frac{4}{3} - \frac{dist}{15} & (\text{oherwise}) \end{cases} \tag{3}$$

Here, T_C is the current time, T_A is the last time for action selection, T_E is task end time, $dist$ is a distance to the nearest Takers. Figures 6, 7 show the result of preliminary experiment which shows the difficulty to set some sub goals. These experiments are made by Multiagent ESP method. Figure 6 shows the Taker's policy is Go To Ball, and Figure 7 shows the Taker's policy is Go To Ball and Pass Cut. The broken line shows the result of the combination of sub goals 1 and 2, the solid line shows the result of the combination of sub goals 2 and 3, the 1-dot dashed line shows the result of sub goal 3. Each pattern is simulated three-times. The combination of sub goals 2 and 3 shows a bad result from these figures. This fact demonstrates the difficulty in a sub goal design.

It is possible to give the rewards based on episode duration time in Multiagent ESP method, but not in Sarsa(λ) algorithm. So, our method gives fitness based on episode duration time(T_D) given by Eq. (4). Existing methods need a design of sub goals to achieve the aim of the task, but our proposed method does not need a sub goal design.

$$fitness = T_D \tag{4}$$

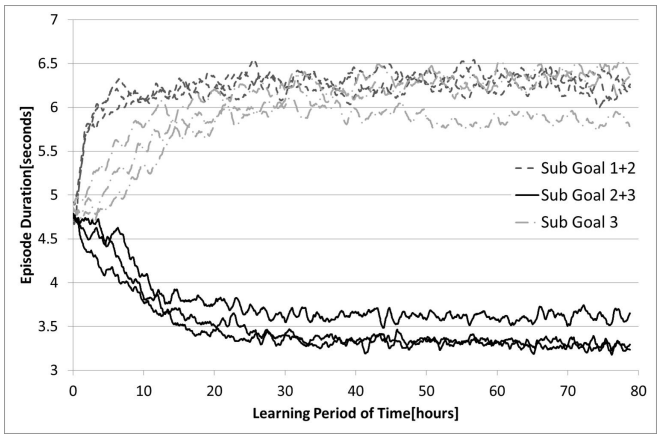

Fig. 6. Taker's policy: Go To Ball

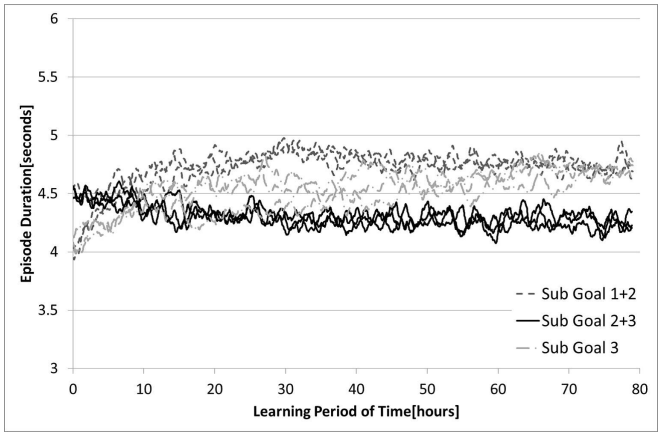

Fig. 7. Taker's policy: Go To Ball and Pass Cut

3.3 Process of generation change

In this section, we describe about the process of generation change in the Multiagent ESP method. The generation changing process is performed through three steps, which is selection, crossover, and mutation. The first selection is a process to bring down high fitness neurons to the next generation, and in our method roulette selection is used. The fitness is updated by windowing which subtract a minimum fitness value in neuron subpopulation from the fitness, and is performed by the power-law scaling which raises the fitness. The second step is one point crossover whose probability is 60 %. The third step is mutation whose probability is 10 % .

This process has to be performed just when all the neurons in neuron subpopulations have been evaluated enough. In this research, the process of generation change are performed every 1000 episodes. That is, one neuron is evaluated 10 times because there are 100 neurons in each neuron subpopulations. Finally, Fig. 8 summarized the process of generation change under these conditions.

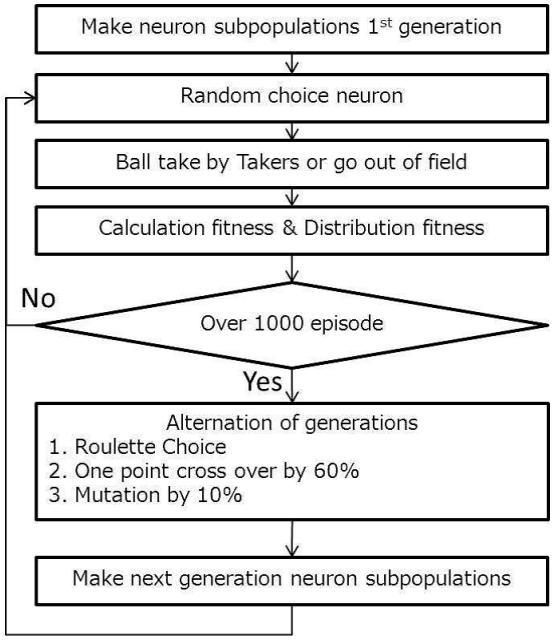

Fig. 8. Process of generation change in Multiagent ESP

4 SIMULATION RESULT

To demonstrate the validity of our proposed method, Figs. 9, 10 show the results of three-times learning of 80 hours in the Keepaway task. Our proposed method which uses the episode duration time as a fitness value is compared with the Sarsa(λ) method proposed by Stone et al. [1], and with the Multiagent ESP method which uses sub goal combination as the fitness value proposed by Arai et al. [2]. Figure 9 shows that the Taker's policy is Go To Ball. Figure 10 shows that the Taker's policy is Go To Ball and Pass Cut. The solid line shows the result of our proposed method, the 1-dot dashed line shows the result of the Sarsa(λ), the broken line shows the result of the Multiagent ESP by Arai's sub goal. The horizontal axis is the learning period of time, and the vertical axis is the episode duration time.

In both of Taker's policies, our proposed method using the episode duration time as a fitness value can give the longest episode duration time. This is because the aim of the task reflect clearly the fitness by using the Multiagent ESP. And, if Taker's policy becomes more difficult, our method can get a better solution than the existing researches by advantages of that it can take continuous state values and can have a set of the optimal solutions. We concluded that the problems in traditional reinforcement learning methods' (such as Sarsa(λ)) : the state representation problem and a variety of the optimal solutions can be solved by using our proposed Multiagent ESP method.

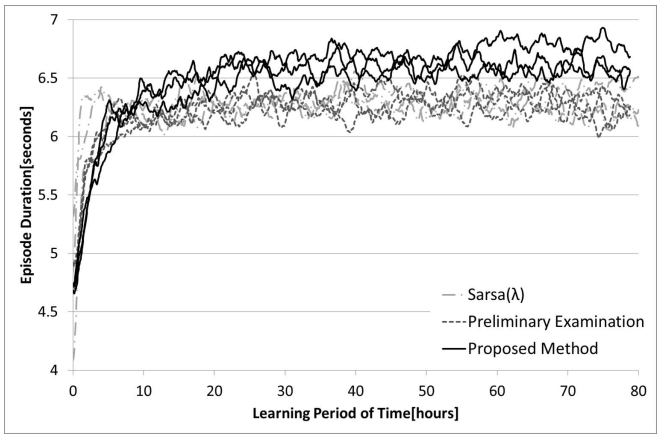

Fig. 9. Taker's policy: Go To Ball

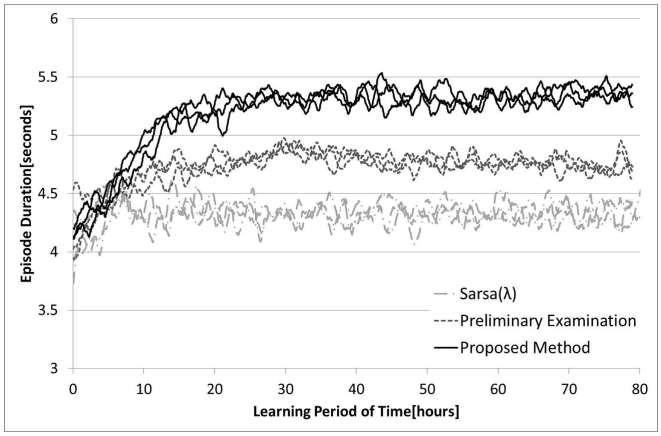

Fig. 10. Taker's policy: Go To Ball and Pass Cut

5 CONCLUSION

In this research, we proposed that the problems of state representation and a variety of optimal solutions can be solved by using our proposed Multiagent ESP method through Keepaway task. And also, we demonstrated that it produced a good learning result to calculate the fitness based on the task duration time at the end of task in order to apply the Multiagent ESP to acquisition of stationary behavior task.

REFERENCES

[1] Peter Stone, Richard S. Sutton, Gregory Kuhlmann: Reinforcement Learning for RoboCup Soccer Keepaway: International Society for Adaptive Behavior, Vol. 13, No. 3, 2005

[2] Sachiyo Arai, Nobuyuki Tanaka: Experimental Analysis of Reward Design for Continuing Task in Multiagent Domains - RoboCup Soccer Keepaway -: The Japanese Society for Artificial Intelligence, Vol. 21, No. 6, 2006

[3] Chern Han Yong and Risto Miikkulainen: Coevolution of Role-Based Cooperation in Multiagent Systems: IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, VOL. 1, NO. 3, 2009

[4] Faustino Gomez and Risto Miikkulainen: Incremental evolution of complex general behavior: Adapt. Behav., vol. 5, 1997

[5] Fasutino Gomez and Risto Miikkulainen: Solving non-Markovian control tasks with neuroevolutio: in Proc. 16th Int. Joint Conf. Artif. Intell., San Francisco, CA, 1999

[6] Watkins, C. J. C. H. and Dayan, P.: Technical Note Q-Learning: Machine Learning, Vol. 8, 1992

[7] Singh, S. P. and Sutton, R. S.: Reinforcement Learning with Replacing Eligibility Traces: Machine Learning, Vol. 22, No. 1-3, 1996