Parallel Reinforcement Learning Systems Including Exploration-Oriented Agents

Takeshi Tateyama¹, Seiichi Kawata² and Yoshiki Shimomura¹

¹Faculty of System Design, Tokyo Metropolitan University
1-1, Minami-Ohsawa, Hachioji-Shi, Tokyo, Japan
²Advanced Institute of Industrial Technology
1-10-40, Higashi-Ooi, Shinagawa-Ku, Tokyo, Japan
email: tateyama-takeshi@c.metro-u.ac.jp

Abstract - We propose a new strategy for parallel reinforcement learning; using this strategy, the optimal value function and policy can be constructed more quickly than by using traditional strategies. We define two types of agents: the main agent and the exploration-oriented agent. The main agent selects actions mainly for exploitation, and the exploration-oriented agent concentrates on exploration using the $k$-certainty exploration method. These agents learn in the same environment in parallel and update the shared value function alternately. By using this strategy, the construction of the optimal value function is expected, and the optimal actions can be selected by the main agents quickly. The experimental results of the $n$-armed bandit problems showed the availability of our method.

I Introduction

Reinforcement learning[1] is known as a valuable machine learning method to construct optimal policies in unknown environments. However, one of the main problems in the reinforcement learning method is that it requires numerous trials to construct optimal policies in large or complex environments. In order to overcome this problem, parallel reinforcement learning methods have been investigated by several researchers(e.g. [4][2][5]). These methods are very efficient in increasing the speed of reinforcement learning by sharing the experiences of multiple agents. In these methods, several agents learn in a common environment in parallel to accomplish a common task. The agents update the shared value function or combine the value functions of all agents periodically. Because the parallel learning system can obtain more information per unit time than the single agent learning system, it can expect to realize the optimal policy more quickly. However, several problems remain to solved. One of them is that it is difficult to realize a balance between exploration and exploitation simultaneously with single agent learning systems. In fact, it is difficult for the designer to find the best learning parameters in order to realize a good balance between the optimality and speed of the learning.

In this paper, we propose a new strategy for parallel reinforcement learning; using this strategy, the optimal value function and policy can be constructed more quickly than by using traditional strategies. In our method, we consider the division of roles between the agents for the learning process. We define two types of agents: the main agent and the exploration-oriented agent. The main agent mainly selects actions for exploitation. In contrast, the exploration-oriented agent concentrates on exploration using the $k$-certainty exploration method[3] to create the optimal value function quickly. This strategy can realize a good balance between the optimality and the speed of the learning and facilitates the selection of the learning parameters. In this paper, the validity of our method is shown using the $n$-armed bandit problems[1][2].

II Parallel Reinforcement Learning

The parallel reinforcement learning method is a very efficient technique to increase the speed of reinforcement learning by sharing the experiences of multiple agents. In this method, several agents learn in a common environment in parallel in order to accomplish a common task. These agents update shared value function or combine the value function of all agents’ periodically to share the experiences of the agents. Because the parallel learning system can obtain more information per unit time than the single agent learning system, it can expect to realize the optimal policy more quickly. Several researchers are working on the parallel reinforce-
ment learning systems. In the methods proposed by Tan, the agents use the same decision policy or average their individual policies periodically[4]. He demonstrated the effectiveness of these techniques by applying them to multi-agent hunter games. Kretchmar has proposed an useful algorithm for combining the experiences of the agents using weighted average[2]. He used the 10-armed bandit problems as the learning tasks to show the availability of his method. Mori and Yamana have proposed the computing system for updating the shared value function asynchronously using parallel processor computer systems[5]. The advantage of their system is that the overhead of communication between the processors is small. In their paper, the validity of their system is shown using the simulations of the maze problem and the mountain car task.

### III Parallel Reinforcement Learning Systems Including Exploration-Oriented Agents

The parallel reinforcement learning methods introduced in the previous section can increase the speed of learning. However, several problems remain to be solved. Similar to the case with a single agent system, one of the problems is that it is difficult to realize balance between exploration and exploitation to find the optimal policy quickly. In particular, if the agents select actions using a shared value function and the value function is biased such that it executes actions that are not optimal, all the agents execute these actions with a high probability and the policy cannot converges on an optimal solution, although the learning speed is increased. It is very difficult for the designer to find the best learning parameters in order to realize a good balance between the optimality and the speed of the learning. In order to overcome this problem, we propose a new strategy for parallel reinforcement learning. In our method, we consider the division of roles between the agents for the learning process. We define two types of agents: the main agent and the exploration-oriented agent. The main agent mainly selects actions for exploitation. The algorithm used for action selection is a complete greedy strategy or a stochastic choice method similar to the greedy strategy(for example, the $\epsilon$ - greedy method with a very small $\epsilon$). In contrast, the exploration-oriented agent concentrates on exploration to create an optimal value function. In this paper, we use the $k$-certainty exploration method[3] as the algorithm for the action selection of the exploration-oriented agent. The $k$-certainty exploration method realizes very efficient exploration by selecting the actions whose frequencies of execution are low due to their priority. Further details of this algorithm can be found elsewhere[3]. The outline of our method is shown in Figure 1. In the learning phase, the main agents and the exploration-oriented agents learn and update the shared value function in parallel. In this phase, the shared value function approaches the optimal function due to the exploration-oriented agents. In addition, the policies of the main agents are improved, although they mostly execute only the greedy actions. Subsequently, if required, the strategy is changed from the learning phase to the exploitation phase when the main agents show good performances. In this phase, all the agents select greedy actions using the shared optimal value function constructed in the learning phase. This strategy maximizes the sum of the rewards received by all the agents over the long run.

### IV Simulation

#### A n-Armed Bandit Problem

The $n$-armed bandit problems are often used in experiments to evaluate reinforcement learning algorithms(e.g.,
In the problem, an agent plays a slot machine with \( n \) levers. At each time step (one play), the agent selects an action to pull the lever \( l_a(1 \leq a \leq n) \). The agent receives a reward depending on the selected action. The reward is computed using a normal probability distribution with mean \( Q^*(a) \) and variance 1. The \( Q^*(a) \) is computed using a normal distribution with mean 0 and variance 1 in advance. However, the agent does not know the \( Q^* \) and the variance. The objective of the agent is to obtain an optimal policy to maximize the expected total reward through trial and error.

In our multi-agent experiment, \( N \) slot machines whose \( Q^* \) and variances for computing rewards have the same values are generated for \( N \) agents. At each time step \( t \), each agent \( \text{Agent}_i \) selects an action \( a \) to play the slot machine \( \text{Slot}_i(1 \leq i \leq N) \). After receiving reward \( r_{it}^* \), each agent updates the shared Q-table using the following equation[1]:

\[
Q_{t+1}(a) = Q_t(a) + \frac{1}{k_a + 1}(r_{it}^* - Q_t(a))
\]  \hspace{1cm} (1)

where \( k_a \) is the sum of the number of executions of an action \( a \) for all agents. After updating the shared Q-table, the \( k_a \) is updated using the following equation:

\[
k_a \leftarrow k_a + 1
\]

where \( k_a \) is the sum of the number of executions of an action \( a \) for all agents. After updating the shared Q-table, the \( k_a \) is updated using the following equation:

\[
k_a \leftarrow k_a + 1
\]

In fact, the shared Q-table is updated \( N \) times each time. If the agents can realize the optimal shared Q-table (viz. \( Q \) approaches \( Q^* \)) quickly, they can realize the optimal policy quickly as well.

B Experimental Results

In this section, we evaluate our multi-agent learning strategy using 10-armed bandit tasks (\( n = 10 \)). In one 10-armed bandit task, each agent plays the slot machine 1000 times (in fact, action selection is performed 1000). Each learning system using different learning strategies performs the task 2000 times using 2000 randomly generated slot machines to collect the evaluation data. We evaluate each learning system by comparing the graphs of the average rewards and the average error of the shared Q-table. The error of the shared Q-table at time \( t \), \( \text{Error}(Q_t) \), is computed using the following equation:

\[
\text{Error}(Q_t) = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(Q^*(a_i) - Q_t(a_i))^2}
\]

First, we test the standard strategy of action selection for parallel reinforcement learning in order to evaluate our new strategy. In this strategy, all the agents select actions using the same algorithm that is tuned by a designer in order to realize a “balance” between exploitation and exploration. The most common methods are the \( \epsilon \)-greedy[1] method and softmax action selection using Boltzmann distribution[1]. In this experiment, we use the \( \epsilon \)-greedy method (\( \epsilon = 0.1 \)) as the action selection algorithm for all agents. In this case, the agents select the greedy action with probability 0.9 and select the random action with probability 0.1.

We performed some experiments in which we varied the number of agents \((N = 1, 2, 5, 10)\). Figure 2 shows the results of the agent whose ID is 1. These results indicate that the increase in the number of agents accelerates the learning speed, which is similar to the experimental results of the other authors’. The purpose of our study is to realize better learning performances than these systems by using a new learning strategy.

Next, we evaluate our learning strategy by comparing it to the standard strategy described above. First, we use one \( \epsilon \)-greedy agent (\( \epsilon = 0.1 \)) and \( N - 1 \) exploration-oriented agents that select actions by using the \( k \)-certainty exploration methods. The results of the main agent (\( \epsilon \)-greedy agent) are displayed in Figures 3, 4, and 5. These results show that our learning strategy realizes a better performance than the standard strategy. These graphs show that the shared Q-tables of our method can approach the true Q-table \( Q^* \) more quickly than the stan-
Figure 3: The results of the system including the exploitation-oriented agents ($\epsilon$ of the main agents is 0.1, $N = 2$)

Figure 4: The results of the system including the exploration-oriented agents ($N = 5$)

Figure 5: The results of the system including the exploration-oriented agents ($N = 10$)

In this paper, we proposed a new strategy for parallel reinforcement learning: using this strategy, the optimal value function and policy can be constructed more quickly than by using traditional strategies. By including the exploration-oriented agents in parallel reinforcement learning systems, our method can realize a good balance between the optimality and the speed of the learning and facilitate the selection of the learning parameters. The validity of our method was shown using difficult parameter tuning.

V Conclusion
the $n$-armed bandit problems. Although these problems are very simple, it is considered that our strategy can be applicable to more complex tasks; this may be realized by using the standard reinforcement learning algorithm (e.g. Q-learning [6]) to update the shared value function.

Figure 6: The results of the systems including greedy agents ($\epsilon = 0$) and exploration-oriented agents.

References


