Improving Search Efficiency in the Action Space of an Instance-Based Reinforcement Learning Technique for Multi-robot Systems

Toshiyuki Yasuda and Kazuhiro Ohkura

Graduate School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, Hiroshima 739-8527, Japan
{yasu, kohkura}@hiroshima-u.ac.jp
http://www.ohk.hiroshima-u.ac.jp

Abstract. We have developed a new reinforcement learning technique called Bayesian-discrimination-function-based reinforcement learning (BRL). BRL is unique, in that it not only learns in the predefined state and action spaces, but also simultaneously changes their segmentation. BRL has proven to be more effective than other standard RL algorithms in dealing with multi-robot system (MRS) problems, where the learning environment is naturally dynamic. This paper introduces an extended form of BRL that improves its learning efficiency. Instead of generating a random action when a robot encounters an unknown situation, the extended BRL generates an action calculated by a linear interpolation among the rules with high similarity to the current sensory input. In both physical experiments and computer simulations, the extended BRL showed higher search efficiency than the standard BRL.

Key words: Multi-robot System, Reinforcement Learning, Autonomous Specialisation, Action Search

1 Introduction

This paper introduces a robust instance-based reinforcement learning (RL) approach for controlling autonomous multi-robot systems (MRS). Although RL has proven to be an effective approach for behaviour acquisition in an autonomous robot, it generates quite sensitive results for segmentation of the state and action spaces. This problem can have severe results as the system becomes more complex. When segmentation is inappropriate, RL often fails. Even if RL obtains a successful result, the achieved behaviour might not be sufficiently robust. In traditional RL, human designers segment the space using implicit knowledge based on their personal experience, because there are no guidelines for segmenting the space.

Two main approaches for overcoming this problem and learning in a continuous space have been discussed. One applies function-approximation techniques such as artificial neural networks to the Q-function. Sutton [1] used CMAC and

F. Almeida e Costa et al. (Eds.): ECAL 2007, LNAI 4648, pp. 325–334, 2007.

Morimoto and Doya [2] used Gaussian softmax basis functions for function approximation. Lin represented the Q-function using multi-layer neural networks called *Q-net* [3]. However, these techniques have the inherent difficulty that a human designer must properly design their neural networks before executing RL. Another method is adaptive segmentation of the continuous state space according to the robots' experiences. Asada *et al.* proposed a state clustering method based on the Mahalanobis distance [4]. Takahashi *et al.* used the nearest-neighbour method [5]. However, these methods generally require large learning costs for tasks such as continuously updating data classifications every time new data arrives.

Our research group proposed an instance-based RL method called the continuous space classifier generator (CSCG), which proves to be effective for behaviour acquisition [6]. We also developed a second instance-based RL method called Bayesian-discrimination-function-based reinforcement learning (BRL) [7]. Our preliminary experiments proved that BRL affords far better performance than CSCG.

This paper introduces an extension of BRL that accelerates learning speed. Our focal point for the extension is the process of action searching. The standard BRL has a rule-producing function. In a standard BRL, a robot performs a random action and stores an input-output pair as a new rule when it encounters a new situation. This random action sometimes produces one novel situation after another, resulting in unstable behaviour. To overcome this problem, we added a function that performs an action based on acquired experience.

The remainder of this paper is organised as follows: Section 2 introduces the target problem; Section 3 explains our design concept and the controller details. Section 4 presents the results of our experiments. Section 5 contains our conclusions.

2 Task: Cooperative Carrying Problem

Our target problem is a simple MRS composed of three autonomous robots, as shown in Fig. 1. This problem is called the *cooperative carrying problem* (CCP), and involves requiring the MRS to carry a triangular board from the start to the goal. A robot is connected to the different corners of the load so that it can rotate freely. A potentiometer measures the angle between the load and the robot's direction θ. A robot can perceive the potentiometer measurements of the other robots, as well as its own. All three robots have the same specifications-each robot has two distance sensors d and three light sensors l. The greater d / l becomes, the nearer the distance to an obstacle or a light source. Each robot has two motors for rotating two omnidirectional wheels. A wheel provides powered drive in the direction it is pointing and passive coasting in an orthogonal direction at the same time.

The difficulties in this task can be summarised as follows:

- The robots have to cooperate with each other to move around.
- They begin with no predefined behaviour rule sets or roles.

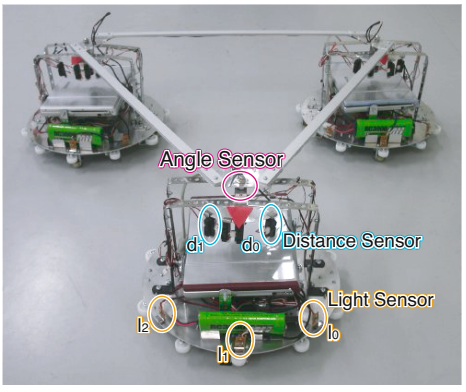


Fig. 1. Cooperative carrying problem

- They have no explicit communication functions.
- They cannot perceive the other robots through the distance sensors because the sensors do not have sufficient range.
- Each robot can perceive the goal (the location of the light source) only when the light is within the range of its light sensors.
- Passive coasting of the omnidirectional wheels brings a dynamic and uncertain state transition.

3 APPROACH

3.1 BRL: RL in Continuous Learning Space

Our approach, called BRL, updates the classifications only when such an update is required. In BRL, the state space is covered by multivariate normal distributions, each of which represents a rule cluster, C_i. A set of production rules is defined by Bayesian discrimination. This method can assign an input, $\boldsymbol{x}$, to the cluster, C_i, which has the largest posterior probability, $\max \Pr(C_i|\boldsymbol{x})$. Here, $\Pr(C_i|\boldsymbol{x})$ indicates the probability calculated by Bayes' formula that a cluster C_i holds the observed input $\boldsymbol{x}$. Therefore, using this technique, a robot can select the rule most similar to the current sensory input. In this RL, production rules are associated with clusters segmented by Bayes boundaries. Each rule contains a state vector $\boldsymbol{v}$, an action vector $\boldsymbol{a}$, a utility u, and parameters for calculating the posterior probability, *i.e.* a prior probability f, a covariance matrix $\boldsymbol{\Sigma}$ and a sample set Φ.

The learning procedure is as follows:

(1) A robot perceives the current sensory input $\boldsymbol{x}$.
(2) Using Bayesian discrimination, the robot selects the most similar rule from a rule set. If a rule is selected, the robot executes the corresponding action $\boldsymbol{a}$, otherwise, it performs a random action.

(3) The robot transfers to the next state and receives a reward r.
(4) All rule utilities are updated according to r. Rules with a utility below a certain threshold are removed.
(5) When the robot performs a random action, the robot produces a new rule combining the current sensory input and the executed action. This executed new rule is memorised in the rule table.
(6) If the robot receives no penalty, an internal estimation technique updates the parameters of all rules. Otherwise, the robot updates only the parameters of the selected rule.
(7) Go to (1).

Action Selection and Rule Production. In BRL, a rule in the rule set is selected to minimise g, *i.e.* the risk of misclassification of the current input. We obtain g based on the posterior probability $\Pr(C_i|\boldsymbol{x})$. $\Pr(C_i|\boldsymbol{x})$ is calculated as an indicator of classification for each cluster by Bayes' Theorem:

$$\Pr(C_i|\boldsymbol{x}) = \frac{\Pr(C_i)\Pr(\boldsymbol{x}|C_i)}{\Pr(\boldsymbol{x})}. \tag{1}$$

A rule cluster of i-th rule, C_i, is represented by a $\boldsymbol{v}_i$-centred Gaussian with covariance $\boldsymbol{\Sigma}_i$. The probability density function of the i-th rule's cluster is therefore represented by

$$\begin{aligned}\Pr(C_i|\boldsymbol{x}) = &\frac{1}{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \\ &\cdot \exp\left\{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{v}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{v}_i)\right\}.\end{aligned} \tag{2}$$

A robot requires g_i instead of calculating $\Pr(C_i|\boldsymbol{x})$[1], because no one can correctly estimate $\Pr(\boldsymbol{x})$ in Eq.(1). A robot must select a rule using only the numerator. The value of g_i is calculated as

$$\begin{aligned}g_i &= -\log(f_i \cdot \Pr(\boldsymbol{x}|C_i) \\ &= \frac{1}{2}(\boldsymbol{x}-\boldsymbol{v}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{v}_i) \\ &\quad -\log\left\{\frac{1}{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\right\} - \log f_i,\end{aligned} \tag{3}$$

where f_i is synonymous with $\Pr(C_i)$.

After calculating g for all rules, the winner rl_w is selected as that with the minimal value of g_i. As mentioned in the learning procedure in Sec. 3.1, the action in rl_w is performed if g_w is lower than a threshold $g_{th} = -\log(f_0 \cdot P_{th})$, where f_0 and P_{th} are predefined constants. Otherwise, a random action is performed.

[1] The higher $\Pr(C_i|\boldsymbol{x})$ becomes, the lower g_i becomes.

3.2 Extended BRL

Basic Concept. We have some RL approaches that provide learning in continuous action spaces. An actor-critic algorithm built with neural networks has a continuous learning space and modifies actions adaptively [8]. This algorithm modifies policies based on TD-error at every time step. The REINFORCE algorithm theoretically also needs immediate reward [9]. These approaches are not useful for tasks such as the navigation problem shown in Sec. 2, because the robot gets a reward only when it reaches the goal. BRL, however, proves to be robust against a delayed reward.

In the standard BRL, a robot performs a random search in its action space, and these random actions can produce unstable behaviour. Therefore, reducing the chance of random actions may accelerate behaviour acquisition and provide more robust behaviour. Instead of performing a random action, BRL needs a function that determines action based on acquired knowledge.

BRL with an Adaptive Action Generator. To accelerate learning, in this paper, we introduce an extended BRL by modifying the learning procedure, Step (2) in Sec. 3.1. In this extension, instead of a random action, the robot performs a knowledge-based action when it encounters a new environment. To do this, we set a new threshold, $P'_{th}(< P_{th})$, and provide three cases for rule selection in Step (2) as follows:

- $g_w < g_{th}$: The robot selects the rule with g_w and executes its corresponding action $\boldsymbol{a}_w$.
- $g_{th} \leq g_w < g'_{th}$: The robot executes an action with parameters determined based on rl_w and other rules with misclassification risks within this range as follows:

$$\boldsymbol{a}' = \sum_{l=1}^{n_r} \left(\frac{u_l}{\sum_{k=1}^{n_r} u_k} \cdot \boldsymbol{a}_l \right) + N(0, \sigma), \tag{4}$$

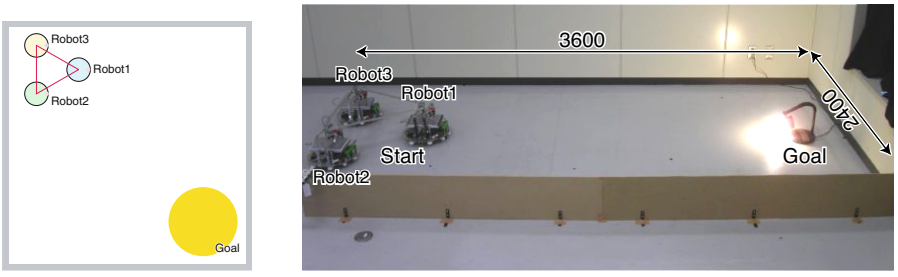
  where n_r is the number of referred rules, and $N(0, \sigma)$ is a zero-centred Gaussian noise with variance σ. This action is regarded as an interpolation of previously-acquired knowledge.
- $g'_{th} \leq g_w$: The robot generates a random action.

In this rule selection, the first and third cases are the same as the standard BRL.

4 Experiments

4.1 Settings

Figure 2 shows the general view of the experimental environments for simulation and physical experiments. In the simulation runs, the field is a square surrounded by a wall. The robots are situated in a 3.6-meter-long and 2.4-meter-wide pathway. The task for the MRS is to move from the start to the goal (light source).



(a) Simulation (b) Real Robots

Fig. 2. Experimental Environment

All robots get a positive reward when one of them reaches the goal ($l_0 > thr_{goal} \vee l_1 > thr_{goal} \vee l_2 > thr_{goal}$). A robot gets a negative reward when it collides with a wall ($d_0^i > thr_d \vee d_1^i > thr_d$). We represent a unit of time as a *step*. A *step* is a sequence that allows the three robots to get their own input information, make decisions by themselves, and execute their actions independently. When the MRS reaches the goal, or when it cannot reach the goal within 200 steps in simulations and 100 steps in physical experiments, it is put back to the start. This time span is called an *episode*.

The settings of the learning mechanisms are as follows.

Prediction Mechanism (NN) Our previous work [7], verified BRL as a successful approach to CCP, with a reformation such that the state space was constructed with sensory information and predictions of the movements of the other robots in the next time step, to decrease the learning problem dynamics.

The prediction mechanism attached is a three-layered feed-forward neural network that performs back propagation. The input is a short history of sensory information, $I = \{ \cos\theta_{t-2}^i, \sin\theta_{t-2}^i, \cos\psi_{t-2}^i, \sin\psi_{t-2}^i, \cos\theta_{t-1}^i, \sin\theta_{t-1}^i, \cos\psi_{t-1}^i, \sin\psi_{t-1}^i, \cos\theta_t^i, \sin\theta_t^i, \cos\psi_t^i, \sin\psi_t^i\}$, where $\psi_t^i = (\theta_t^j + \theta_t^k)/2 \quad (i \neq j \neq k)$. The output is a prediction of the posture of the other robots at the next time step $O = \{\cos\psi_{t+1}^i, \sin\psi_{t+1}^i\}$. The hidden layer has eight nodes.

Behavior Learning Mechanism (BRL) The input is $\boldsymbol{x} = \{ \cos\theta_t^i, \sin\theta_t^i, \cos\psi_{t+1}^i, \sin\psi_{t+1}^i, d_0^i, d_1^i, l_0^i, l_1^i, l_2^i \}$. The output is $\boldsymbol{a} = \{m_{rud}^i, m_{th}^i\}$, where m_{rud}^i and m_{th}^i are the motor commands for the rudder and the throttle respectively. σ in Eq.(4) is 0.05. For the standard BRL, $P_{th} = \{0.012, 0.01\}$. For the extended BRL, $P_{th} = 0.012$ and $P'_{th} = 0.01$. The other parameters are the same as the recommended values in our journal [7].

4.2 Result: Simulations

Figure 3 shows the averages and the deviations of steps that the MRS takes by the end of each episode. In the early stages, the MRS requires a lot of trial

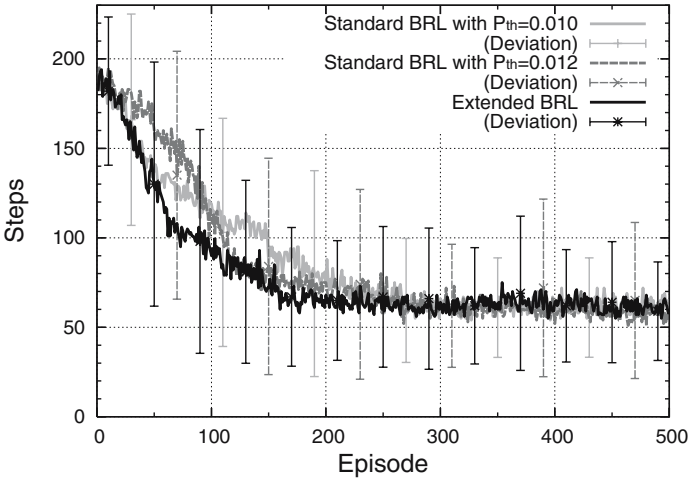


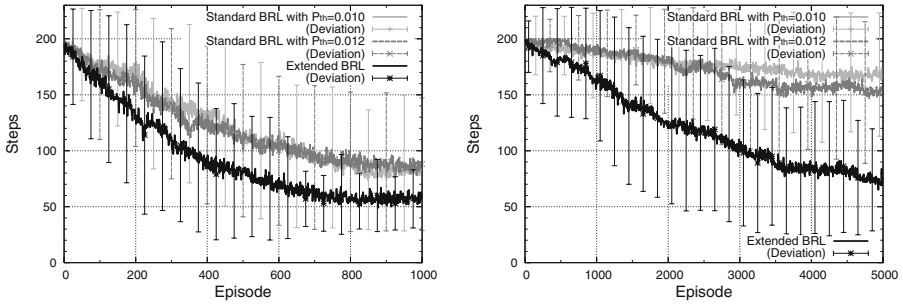
Fig. 3. Mean Learning History for 50 Simulations of Three Robots

and error and takes many steps to finish the episode. After such a trial and error process, the behaviour of MRS becomes more stable and it takes fewer steps. An MRS with the standard BRL stably achieves the task within nearly constant steps after the 250th episode, and the extended BRL accomplishes this in 200 episodes. This means that, in terms of learning speed, the extended BRL outperforms the standard one.

For the 50 independent runs, the MRS achieved different globally stable behaviour. However, we found a common point that robots always achieved cooperative behaviour by developing team play organised by a leader, a sub-leader and a follower. This implies that acquiring cooperative behaviour always involved autonomous specialisation. The extended BRL displayed higher adaptability, and yielded autonomous specialisation faster than the standard BRL.

Discussion. There is no significant difference in results in the learning performance of the BRLs for a three-robot CCP; therefore, we tested four- and five-robot CCP performance for more dynamic and complicated problems. The four robots use a square load, and the five robots have a pentagonal load. In these CCPs, ψ is the average of the angles between two neighbouring robots and the load. The other controller settings are the same as those for the three-robot CCP.

Figure 4 shows the average and the deviations of steps an MRS takes by the end of each episode. As the number of robots increases, we can find that the extended BRL provides increasingly better results than the standard BRL, although it requires more episodes before obtaining stable behaviour. The extended BRL has a function for coordinating behaviour as well as reducing the number of random actions that can result in unstable behaviour. These results



(a) Four Robots (b) Five Robots

Fig. 4. Mean Learning History for 50 Simulations

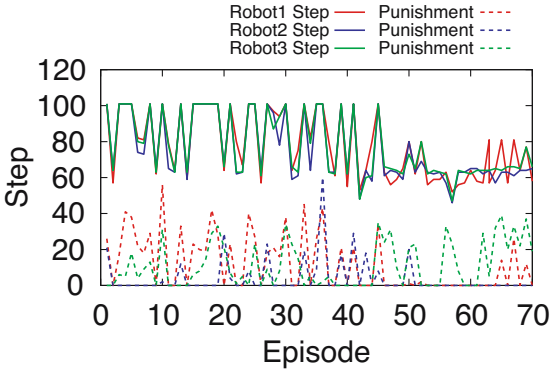
show that the extended BRL has a higher learning ability and is less dependent on the number of robots in the MRS. This implies that the extended BRL might have more scalability, which is one of the advantages of MRS over single-robot systems.

Although parameters that are more refined might provide better performance, parameter tuning is outside the scope, because BRL is designed for acquiring reasonable behaviour as quickly as possible, rather than optimal behaviour. In other words, the focal point of our MRS controller is not optimality but versatility. In fact, we obtain similar experimental results through experiments with an arm-type MRS similar to that in [6] using the same parameter settings.

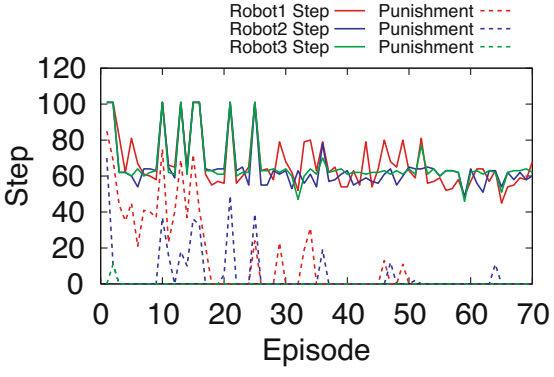
4.3 Result: Physical Experiments

We conducted five independent experimental runs for each BRL. The standard BRL provided two successful results and the extended BRL provided four. Fig. 5 illustrates the best results of the physical experiments. These figures illustrate the number of steps and punishments in each episode. Comparing these results shows that the extended BRL requires fewer episodes to learn behaviour. The other successful results of the extended BRL show better performance than the best result of the standard BRL. The behaviour of the extended BRL is also more stable than that of the standard, because the MRS with the standard BRL gets several punishments after learning goal-reaching behaviour.

Figure 6 shows an example of the behaviour of the extended BRL. In the early stages, robots have no knowledge and function by trial and error. During this process, robots often collide with a wall and become immovable (Fig. 6(a)). Then, some robots reach the goal and develop appropriate input-output mappings (Fig. 6(b)). Observing the acquired behaviour and investigating rule parameters, we found that the robots developed cooperative behaviour, based on autonomous specialisation.



(a) Standard BRL



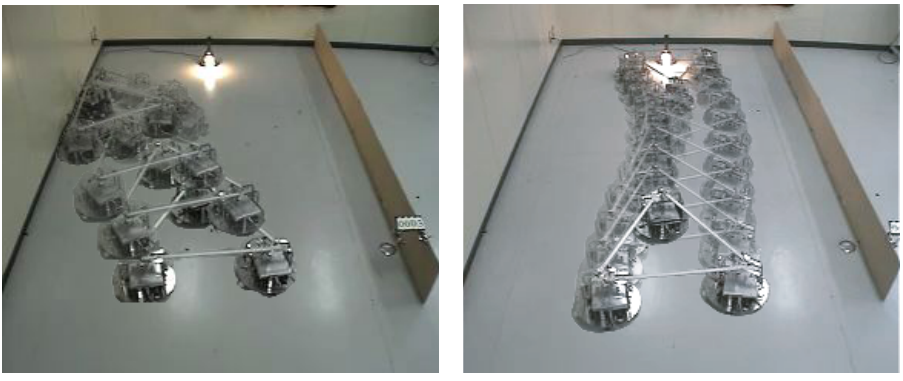
(b) Extended BRL

Fig. 5. Learning History: Physical Experiment

5 Conclusions

We investigated the RL approach for the behaviour acquisition of autonomous MRS. Our proposed RL technique, BRL, has a mechanism for autonomous segmentation of the continuous learning space, and proved effective for MRS through the emergence of autonomous specialisation. For accelerated learning, we proposed an extension of BRL with a function to generate interpolated actions based on previously acquired rules. Results of the simulations and physical experiments showed that the MRS with an extended BRL did learn behaviour faster than that with the standard BRL.

In the future, we plan to investigate the robustness and re-learning ability in a changing environment. We also plan to increase the number of sensors and adopt other expensive sensors such as an omnidirectional camera that will allow a robot to incorporate a variety of information, and thereby acquire more sophisticated cooperative behaviour in more complex environments.



(a) In the Early Stage (b) After Successful Learning

Fig. 6. An Example of Acquired Behaviour: Extended BRL

References

1. Sutton, R.S.: Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. In: Advances in Neural Information Processing Systems, vol. 8, pp. 1038–1044. MIT Press, Cambridge (1996)
2. Morimoto, J., Doya, K.: Acquisition of Stand-Up Behavior by a Real Robot using Hierarchical Reinforcement Learning for Motion Learning: Learning "Stand Up" Trajectories. In: Proc. of International Conference on Machine Learning, pp. 623–630 (2000)
3. Lin, L.J.: Scaling Up Reinforcement Learning for Robot Control. In: Proc. of the 10th International Conference on Machine Learning, pp. 182–189 (1993)
4. Asada, M., Noda, S., Hosoda, K.: Action-Based Sensor Space Categorization for Robot Learning. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1502–1509. IEEE, Los Alamitos (1996)
5. Takahashi, Y., Asada, M., Hosoda, K.: Reasonable Performance in Less Learning Time by Real Robot Based on Incremental State Space Segmentation. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1502–1524. IEEE, Los Alamitos (1996)
6. Svinin, M., Kojima, F., Katada, Y., Ueda, K.: Initial Experiments on Reinforcement Learning Control of Cooperative Manipulations. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 416–422. IEEE, Los Alamitos (2000)
7. Yasuda, T., Ohkura, K.: Autonomous Role Assignment in Homogeneous Multi-Robot Systems. Journal of Robotics and Mechatronics 17(5), 596–604 (2005)
8. Doya, K.: Reinforcement Learning in Continuous Time and Space. Neural Computation 12, 219–245 (2000)
9. Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning 8, 229–256 (1992)