Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2014, Article ID 270548, 8 pages
http://dx.doi.org/10.1155/2014/270548

Research Article

An Improved Reinforcement Learning Algorithm for Cooperative Behaviors of Mobile Robots

Yong Song,[1,2] Yibin Li,[1] Xiaoli Wang,[2] Xin Ma,[2] and Jiuhong Ruan[2]

[1] *School of Control Science and Engineering, Shandong University, Jinan 250061, China*
[2] *School of Mechanical, Electrical & Information Engineering, Shandong University at Weihai, Weihai 264209, China*

Correspondence should be addressed to Yibin Li; liyb@sdu.edu.cn

Received 27 November 2013; Accepted 20 January 2014; Published 5 March 2014

Academic Editor: Zoltan Szabo

Reinforcement learning algorithm for multirobot will become very slow when the number of robots is increasing resulting in an exponential increase of state space. A sequential Q-learning based on knowledge sharing is presented. The rule repository of robots behaviors is firstly initialized in the process of reinforcement learning. Mobile robots obtain present environmental state by sensors. Then the state will be matched to determine if the relevant behavior rule has been stored in the database. If the rule is present, an action will be chosen in accordance with the knowledge and the rules, and the matching weight will be refined. Otherwise the new rule will be appended to the database. The robots learn according to a given sequence and share the behavior database. We examine the algorithm by multirobot following-surrounding behavior, and find that the improved algorithm can effectively accelerate the convergence speed.

1. Introduction

In recent years, multirobot systems (MRSs) have received considerable attention because such systems possess some special capabilities such as more flexibility, adaptability, and efficiency in dealing with a complex task [1]. Multirobot learning is the process of acquiring new cooperative behaviors for a particular task by trial and error in the presence of other robots. The desired cooperative behaviors may emerge by local interactions among the robots which are with limited sensing capabilities. Multirobot system can perform more complex tasks via cooperation and coordination [2, 3].

Normally, multirobot learning method can be classified as collective swarm learning and intentionally cooperative learning based on the various levels of explicit communication. The collective swarm systems allow participating robots to learn swarm behaviors with only minimal explicit communication among robots [4, 5]. In these systems a large number of homogeneous mobile robots interact implicitly with each other based on the sharing environment. The robots are organized on the basis of local control laws, such as the stigmergy introduced by Garnier et al. [6]. Stigmergy is a mechanism of indirect interaction mediated by modifications of the sharing environment of agents [7]. The information coming from the local environment can guide the participating individual activity. The complex intelligent behavior emerges at the colony level from the local interactions that take place among individuals exhibiting simple behaviors. At present, the swarm behaviors are often modeled using methods inspired by biology. Along with the advent of artificial life, some self-organizing models of social animals have provided salutary inspirations [8]. Beckers et al. conducted some initiative simulations and physical experiments to interpret the nesting behavior of termites with stigmergy mechanism [9]. The method of swarm intelligence learning mainly involves ant colony algorithm and particle swarm optimization algorithm [10]. Beyond the above methods reinforcement learning and evolutionary algorithm are also the important methods in the design of collective swarm system. Givigi and Schwartz presented an evolutionary method of behavior learning for swarm robot system [11]. The chromosome was exchanged with distributed genetic algorithm to improve the robot behavior ability. Sang-Wook et al. discussed the use of evolutionary psychology in order to select a set of traits of personality that will evolve due to a learning process based on reinforcement learning [12].

The use of Game Theory is introduced in conjunction with the use of external payoffs.

Unlike collective swarm systems, robots in intentionally cooperative systems share the information of joint actions to determine the next state and rewards with each learning robot. More and more attention has been paid to the improved reinforcement learning algorithms. Kobayashi et al. proposed an objective-based reinforcement learning system for multiple autonomous mobile robots to acquire cooperative behavior. The proposed system employs profit sharing (PS) as a learning method [13]. Fernández et al. studied the issues of credit assignment and large scale state space problem in a multirobot environment and combined domain knowledge and machine learning algorithms to achieve successful cooperative behaviors [14]. Lee et al. presented an algorithm of behavior learning and online distributed evolution for cooperative behavior of a group of autonomous robots. Individual robots improve the state-action mapping through online evolution with the crossover operator based on the Q-values and their update frequencies [15]. Ahmadabadi et al. introduced some expertness measuring criteria and a new cooperative learning method called weighted strategy sharing (WSS). Each robot assigns a weight to the knowledge of learning robots and utilizes it according the amount of its teammate expertness [16]. The hybrid policy reduced the dimensions of robot state space.

In both swarm learning and cooperative learning the state space will grow exponentially in the number of team members. In order to deal with the slow convergence speed of standard learning algorithm, we propose an improved reinforcement learning algorithm based on knowledge sharing. The robots perform the learning process according to the predefined sequence. The learning robot will sense the current state at each time step. If the same state has existed in the rules repository, the learning robot will choose an action on the basis of the knowledge base and rules repository. The corresponding weight vector will be updated based on reinforcement learning. Otherwise the learning robot will choose an appropriate action according to the state transition probabilities function. The new rule will be appended to the rules repository. While in the process of reinforcement learning, the learning robot assigns a weight to each robot based on weighted strategy sharing. The behavior weight vector will be refined on the basis of the weighted sum of teammate expertness. Because each robot does not need to observe the actions of all its teammates, the improved reinforcement learning algorithm results in a significantly smaller state space than that for the standard Q-learning algorithm.

2. The Model of Multirobot Environment State

The work space is a two-dimensional environment model with the boundary. Each robot is considered an omnidirectional mobile robot with limited sensing capabilities. The robot has eight range sensors, which are labeled S_t ($t = 1, 2, \ldots, 8$). The eight sensors are arranged evenly. Accordingly, the detecting zone is evenly separated into eight sectors, starting counterclockwise from the direction of the robot's heading. Each sector is represented by an 8-bit binary code, which will be 1 when the robot or obstacle is located in the sector and 0 otherwise. This binary code represents the object distribution around the robot in the detection circle. The environmental state shown in Figure 1 can be described as state S, and $S = [0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1]$.

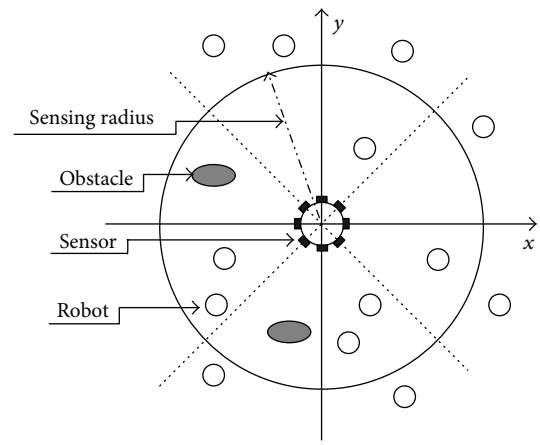


Figure 1: The state model of robot.

3. The Behaviors Database of Robots

The robots behaviors database includes the knowledge base and the rules repository. The selection rules are stored in the rules repository. Once the learning robots find the target, the position and state will be stored in the sharing knowledge base. The knowledge base is shown as follows:

$$K : \{T_i(x_i, y_i), m_i\}, \quad i = 1, 2, \ldots, n, \tag{1}$$

where T_i is the ith target, (x_i, y_i) is the position of T_i, n is the number of the target, m_i is the state of the current target, and if the target is stationary, m_i is equal to 0, otherwise 1.

The robots can choose their behaviors at each step in N directions. A group of behaviors is corresponding to a weight vector, which indicates the selection probability of every behavior. The weight vector is characterized in the following formula:

$$W_i = [w_{i1}\ \ w_{i2}\ \ \cdots\ \ w_{iN}], \tag{2}$$

where W_i is the weight vector which corresponds to the state S_i and S_i is the current state. w_{ik} is the weight corresponding to the appropriate behavior. When perceiving the environment state S_i, the robot chooses a behavior on basis of the element of weight vector W_i. The behavior selection strategy is shown by

$$P(a_k \mid S_i) = P(w_{ik}) \quad k = 1, 2, \ldots, N, \tag{3}$$

where a_k is the selected action, S_i is the current state, $P(w_{ik})$ is the probability to select an appropriate action, N is equal to 8, and w_{ik} ($k = 1, \ldots, N$) have a sum equal to 1.

The behavioral rules of robots are represented by the following formula:

$$R_i : \{\text{if } S = S_i, \text{then } P(a_k \mid S_i)\}, \tag{4}$$

where R_i is the behavioral rule corresponding to the state S_i and a_k is the selected action.

If the robots cannot perceive the target, then $S_0 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, and the robots will choose behaviors equiprobably in N directions, meaning the learning robots move randomly; that is, $W_0 = [1/N, 1/N, \ldots, 1/N]$. The equation of selection rules of behaviors is denoted as follows:

$$R_0 : \{\text{if } S = S_0, \text{then } P(a_k \mid S_i)\}. \tag{5}$$

In the course of learning the robot will perceive the current state S_i and determine if a same state has existed in the rules repository R. If existing, the corresponding weight vector will be updated based on reinforcement learning. Otherwise the new rule R_i will be appended to the rules repository R. All learning robots will share the behaviors database; that is, any robot will choose the same action to respond to the same state. The rules repository is shown as follows:

$$R = \{R_0, R_1, R_2, \ldots, R_m\} \quad m = 2^N - 1, \tag{6}$$

where R is the rules repository, R_i is the behavioral rule corresponding to the state S_i, and m is the maximum number of rules.

4. Reinforcement Learning Algorithm Based on Weighted Strategy Sharing

4.1. Reinforcement Learning Algorithm for Mobile Robots. Q-learning is a form of model-free reinforcement learning, which does not require explicit knowledge of the environment. It allows a robot to synthesize and improve behaviors through trial and error. Within the reinforcement learning framework a robot chooses an appropriate action for the current state that results in an immediate reward and attempts to maximize the long-term rewards. The algorithm converges with probability one to the optimal Q-values so long as each state-action pair is visited infinitely often and learning mate declines.

In single-agent reinforcement learning, a robot operates in accordance with a finite-discrete-time Markov Decision Process, which is formally defined as follows.

Definition 1. Finite Markov Decision Process is a four-tuple sample $\langle S, A, f, R\rangle$, where S is a finite discrete set of states, A is a finite discrete set of agent actions, $f : S \times R \to \Pi(S)$ is the state transition probability function, and $R : S \times R \to \mathbb{R}$ is a reward function.

In the process of reinforcement learning, the learning robot senses the current state and chooses an appropriate action. The environment changes its state to the succeeding state according to the state transition probabilities function. The task of robots reinforcement learning is to obtain the optical policy π^*, which makes robots acquire the maximum cumulative reward V^π beginning at every state. The cumulative value $V^\pi(s_t)$ achieved by following an arbitrary policy π from an arbitrary initial state s_t is defined as follows:

$$V^\pi(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i}, \tag{7}$$

where π is policy, r is immediate reward, and γ is discount factor.

The optimal policy π^* with the maximum cumulative rewards is shown by

$$\pi^* \equiv \arg\max_\pi V^\pi(s), \quad \forall(s). \tag{8}$$

The maximum cumulative reward by following the optimal policy from the current state is denoted as $V^*(s)$. The value of Q function is defined as the immediate reward plus the maximum cumulative reward of the succeeding state; the equation is shown as follows:

$$Q(s,a) \equiv (1-\alpha) Q(s,a) + \alpha\left(r + \gamma V^*\left(s'\right)\right), \tag{9}$$

where α is a small learning rate parameter between 0 and 1, s' is the succeeding state by performing the action a under the current state s, and $V^*(s')$ has close relation with $Q(s', a')$, as shown by

$$V^*\left(s'\right) = \max_{a'} Q\left(s', a'\right), \tag{10}$$

where a' is the optimal action under the state s'. Then the Q-values indexed on the agent's state and action at each step are updated by

$$Q(s,a) = (1-\alpha) Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q\left(s', a'\right)\right), \tag{11}$$

where $Q(s,a)$ is the value of state-action pair (s,a), α is a small learning rate, r is immediate reward, γ is discount factor, a' is the selected action according to s', and s' is the succeeding state by performing the action a under the current state s.

4.2. Weighted Strategy Sharing. Weighted strategy sharing (WSS) is an effective method of the knowledge sharing to multirobot system with communication ability [16]. It is assumed that a group of robots is learning in distinct environments. And their actions do not change the working environment of others. The learning robot calculates expertise value of participating robots. In this method, weights of each robot's expertise must be specified properly. The expertise of each robot is evaluated on the basis of the robot's location and sensor information. This strategy allows each learning robot to make a decision based on the expertise value of other robots.

The expertise is the embodiment of the knowledge and ability of the robot individuals. The methods to evaluate the robot expertise are classified into two categories. The learning system needs additional information in evaluating

the expertise by the first method. The relevant information should be continually acquired in the process of learning. The reinforcement signal was usually calculated by this category of method. The learning system only needs Q-values in evaluating the expertise without additional information or prior knowledge in the second method. The different methods of expertise evaluation should be chosen on the basis of special environment state in practice.

The cooperation of robots can be implemented at different levels. The robots can share the information of sensors, joint actions, and reinforcement signals. The knowledge and expertise of robots are all reflected in the Q-tables. So the learning policy based on sharing Q-tables can fully embody the collaboration among robots. The punishments obtained by learning robots have greater significance in the early stage of learning. The awards of robots will be more important as the cooperative behavior evolved. Therefore, the weight of punishments will decrease with the learning time. The expertise value is shown by

$$e_i = \begin{cases} \sum_{t=1}^{T_n-1} r_i(t) + r_i(T_n)\, r, & \text{if } r_i(T_n) > 0, \\ \sum_{t=1}^{T_n-1} r_i(t) + \dfrac{|r_i(T_n)|}{T_n}, & \text{if } r_i(T_n) < 0, \end{cases} \tag{12}$$

where e_i is the expertise value of the ith learning robot, $r_i(t)$ is the immediate reward of the ith robot at time-step t, and T_n denotes T_nth trial. The weight-assigning mechanism is that the learning robot only learns from more experienced robots.

The learning robot will assign different weights to other robots based on the expertise value in the process of learning. The Q-values of the learning robot will be updated based on the weighted mean of Q-values of other robots. The learning robot assigns a weight to the knowledge and expertise of other robots as follows:

$$W_{ij} = \begin{cases} \dfrac{e_j - e_i}{\sum_{k=1}^{n} e_k}, & \text{if } e_j > e_i, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where n is the number of robots, e_i is the expertise value of learning robot, e_j is the expertise value of other robots, and W_{ij} is the weight of the jth robot relative to the ith robot. Then the Q-values indexed on the robot's state and action at each step are updated by

$$Q_i(s,a) = (1-\alpha)\, Q_i(s,a) + \sum_{j=1}^{n} W_{ij} Q_j\left(s', a'\right), \tag{14}$$

where α is a small learning rate parameter between 0 and 1, s' is the succeeding state by performing the action a under the current state s, and a' is the optimal action under the state s'.

4.3. An Improved Reinforcement Learning Algorithm for Robots. The state space will grow exponentially with the increasing of the number of team members when the single agent reinforcement learning is scaled up to the multirobot domain. In order to speed up the convergence, a sequential Q-learning algorithm is proposed based on the knowledge sharing. In the sequential Q-learning algorithm, the robots learn the decision strategy one by one according to the predefined sequence. The rule repository of robots behaviors is firstly initialized in the process of reinforcement learning. Mobile robots obtain present environmental state by sensors. Then the state will be matched to determine if the relevant behavior rule has been stored in database. If the rule is present, an action will be chosen in accordance with the knowledge and the rules, and the corresponding weight will be refined. Otherwise the new rule consisted of the state and initial Q-value will be appended into the database. The robot will randomly choose an action according to the initial behavior weight and continue the learning process. While in the process of reinforcement learning, the learning robot assigns a weight to each robot based on weighted strategy sharing. The behavior weight vector will be refined on the basis of the weighted sum of teammate expertness. Figure 2 is the flow chart of improved Q-learning algorithm for mobile robots. The sequential Q-learning algorithm based on the knowledge sharing may be summarized as follows.

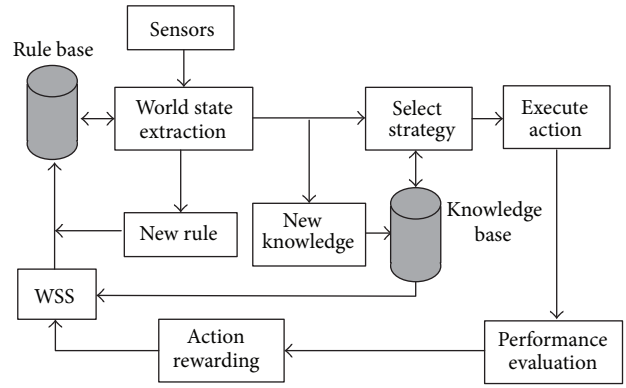


Figure 2: The flow chart of improved Q-learning algorithm for mobile robots.

(1) The rules repository is initialized to $R0$. Assume that there are n robots: $r_1, r_2, \ldots, r_n$.

(2) Repeat Step (3.1) to Step (3.5).

(3.1) Robot r_j observes current state S_i.

(3.2) If a same state has existed in the rules repository, go to Step (3.4). Otherwise, go to Step (3.3).

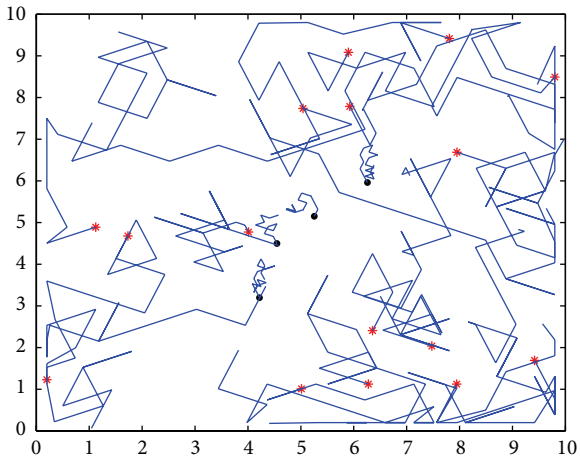
(3.3) The rules repository is updated, $R : R = [R \;\; R_i]$.

(3.4) The learning robot chooses an action according to the rule R_i and performs it. The state will be updated to new state S_i' and receive immediate reward r.

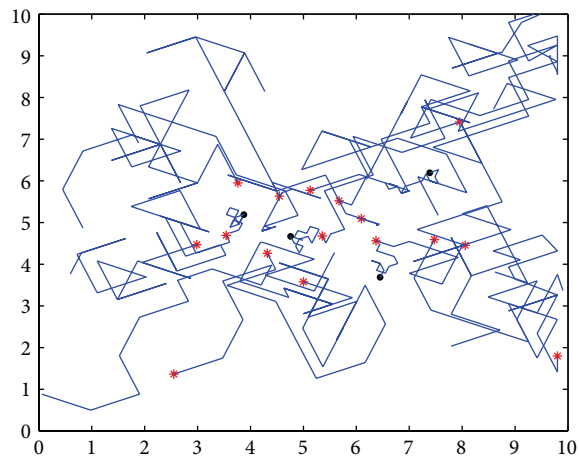
(3.5) The value of $Q_i(s,a)$ is updated as follows:

$$Q_i(s,a) = (1-\alpha)\, Q_i(s,a) + \sum_{j=1}^{n} W_{ij} Q_j\left(s', a'\right). \tag{15}$$

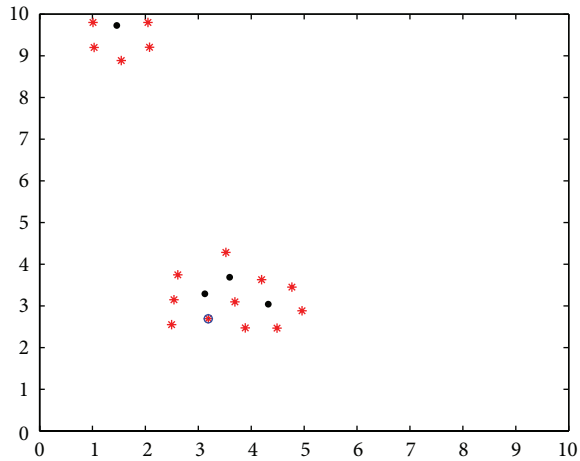
(4) If multirobot system reaches the stable state or the learning system reaches the maximum time, the learning will end. Otherwise, go to Step (2).



(a) The trajectory of predators during the first twenty trials with sequential Q-learning algorithm



(b) The trajectory of predators during the first twenty trials with the improved Q-learning algorithm



(c) The final state of following-surrounding behaviors

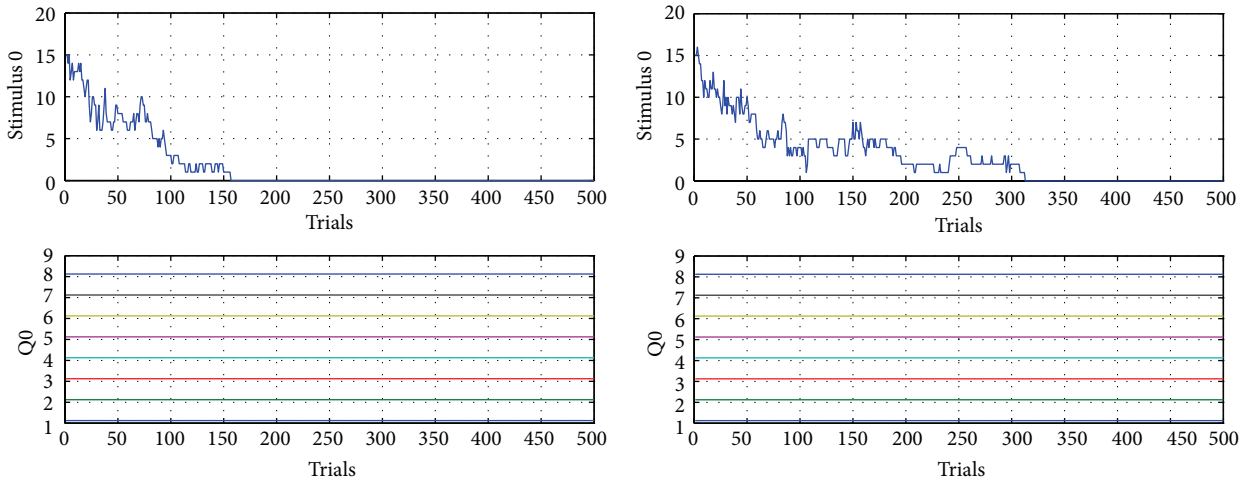
Figure 3: The evolution of the multirobot following-surrounding behavior.

5. Experimental Results

5.1. Environment Settings. To elucidate the advantage of our proposed improved Q-learning algorithm, the implementation of a cooperative pursue behavior is presented as an example. The cooperative pursue is a very challenging task of multirobot system. The performance of the task has significance for multirobot collaborative behavior. The predators in the task search the preys by the limited perception and local interaction among the robots. The predators finally surround the preys. In multirobot pursue system twenty predators and four preys are randomly distributed in a 10×10 area. The red asterisk and the black dot denote the predator and prey, respectively (see Figure 3). The task of the predators is to follow and surround the prey. The predator may perceive the current states and choose appropriate actions. The preys always strive to escape from the surrounding. If the state is $S = [1\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$, the prey will randomly choose an action in the second, fifth, and seventh sectors.

In our demonstration we set up the parameters of the learning process in the following way. The sensing radius of the predators is 1. The maximum distance of the predators is 1 for each step. The minimum distance between robots is 0.6. The reinforcement learning parameters are chosen as $\alpha = 0.3$, $\gamma = 0.95$. The explore probability ε is initialized at 0.5 and decays with the evolution of cooperative behaviors. At each learning step, the predator chooses an action according to its current state and receives reward 0.2 for actions approaching to a prey, -0.2 for actions away from a prey, and 0 otherwise.

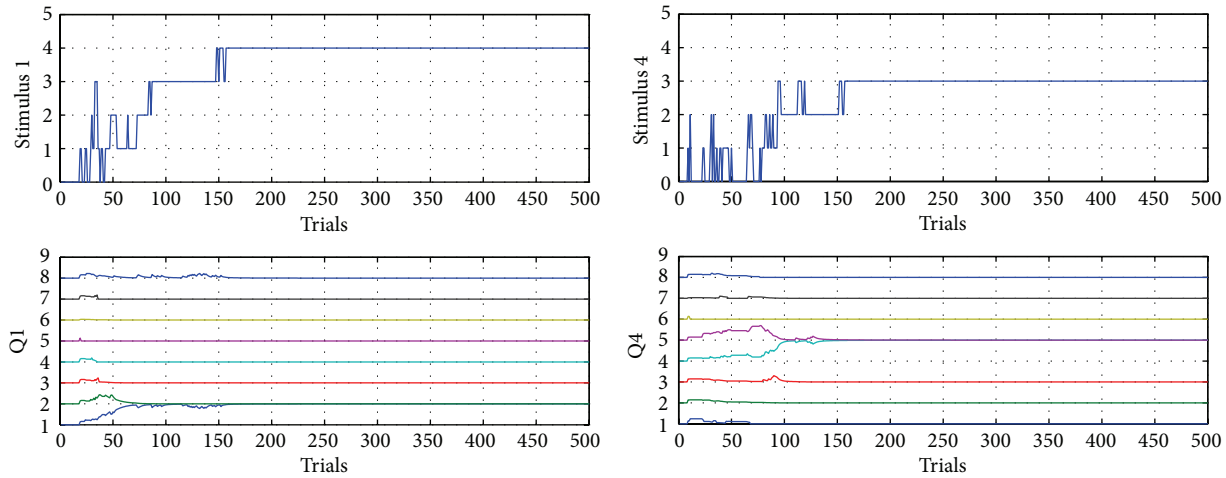
5.2. Results and Analysis. To compare the performance of the conventional Q-learning and the improved Q-learning algorithm, the experiments have been done by running each algorithm in the same work space. Figure 3 is the trajectory of predators during the first twenty trials under two learning policies. Figure 3(a) is the trajectory of predators during the first twenty trials with sequential Q-learning algorithm. The experimental results demonstrate that most of the predators search randomly in the early stage of learning because they do not find target. Only individual predators can track the target, which find the preys. Figure 3(b) shows that the predators can find preys in a short time based on knowledge sharing. The predators exhibit better aim and cooperation with



(a) The evolution of the number of the searching predators based on the improved Q-learning

(b) The evolution of the number of the searching predators based on conventional Q-learning

FIGURE 4: The evolution of the number of the searching predators.



(a) The evolution of the number of the predators which find the target in the first sector and the corresponding weight

(b) The evolution of the number of the predators which find the target in the fourth sector and the corresponding weight

FIGURE 5: The learning process of partial state-action pairs.

the improved Q-learning algorithm. Therefore, the improved Q-learning algorithm is of high efficiency in the early stage of learning based on knowledge sharing.

When the knowledge base and the rules repository are all in the initial state, the predator has not any information about the prey in the learning process. The predator will search the prey randomly. Any of the predators finds the object. All predators will follow the prey based on knowledge sharing and completely surround them. Figure 3(c) shows a final state of the following-surrounding behaviors.

Theoretically, there will be 2^N (N is the number of sensing sectors) rules in the rules repository. But the robots can only catch quite a few of these states. Most of the robots cannot find the target in the initial stage of learning. The perceiving state is Stimulus 0 = [0 0 0 0 0 0 0 0]. Most of the predators move randomly because they have nothing about the preys (see Figure 4). Once a predator detects the prey, all of them will track it on the basis of knowledge sharing. With the learning going on, more and more predators gradually find the target, and the system will reach an equilibrium state quickly. Figure 4(a) shows the evolution of the number of the searching predators based on the improved Q-learning. When the database of robots is in the initial state, the predators search randomly. The number of predators gradually diminishes until it becomes zero. If all of predators have found the preys, the system will reach an equilibrium state quickly. Figure 4(b) shows the evolution of the number of the searching predators based on conventional Q-learning. In contrast with the improved Q-learning, the conventional Q-learning converges more slowly.

In most cases predators can find the target in one sector or two. Figure 5(a) shows the evolution of the number of

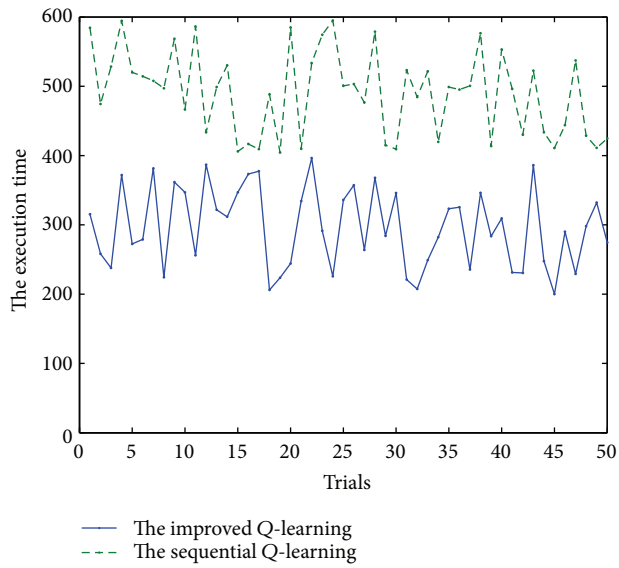


FIGURE 6: The execution time under different Q-learning algorithms.

the predators which find the targets in the first sector and the corresponding weight. Through analyzing the operation process of simulation, we can find that when the predators find the target in the first sector, they will choose an action in the first or eighth sector. Finally the predators will follow the target in the direction of the first sector. Figure 5(b) shows the evolution of the number of the predators which find the target in the fourth sector and the corresponding weight. When the predators find the target in the fourth sector, they will choose an action in the fifth or fourth sector. Finally they will follow the target in the direction of the fourth sector.

The sequential Q-learning algorithm based on the knowledge sharing can obviously reduce the complexity of the traditional Q-learning. The same rule will be updated many times in one trial of the learning. So the improved algorithm can speed up the convergence efficiently. However, the predators and preys are randomly distributed in the workspace in the initial state. And the predators will search randomly when they have no sharing knowledge. Therefore, the execution time of each cooperative task is random. To test the effectiveness and the performances of our innovative approach, each experiment has been repeated 50 times under the same condition. Figure 6 shows the execution time under different Q-learning algorithms. It is all some randomness in system that makes the execution time of predators different for each task. The simulation shows that the improved algorithm has higher efficiency. The improvement is chiefly due to the fact that predators act cooperatively based on the sharing knowledge.

6. Conclusion

When the traditional Q-learning is performed in multirobot domain, the state space will grow exponentially in the number of team members. To speed up the convergence and reduce the complexity of the traditional reinforcement learning algorithm we propose a sequential Q-learning algorithm based on knowledge sharing. The learning system will initialize the rules repository and share the knowledge base. The learning robot will perceive the current state according to the predefined sequence. If the same state has existed in the rules repository, the robot will choose an action on the basis of the knowledge base and rules repository, and the corresponding weight vector will be updated based on reinforcement learning. Otherwise the learning robot will choose an appropriate action according to the state transition probabilities function, and the new rule will be appended to the rules repository. Then the behavior weight vector will be refined on the basis of the weighted sum of teammate expertness. The sequential Q-learning algorithm based on knowledge sharing is performed in a subspace and promotes the learning efficiency. The validity of this method is tested via the simulation experiment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61105100, 61174054), The State Key Program of National Natural Science of China (no. 61233014), Independent Innovation Foundation of Shandong University (no. 2013ZRQP002), and Shandong Province Natural Science Fund (no. ZR2012FM036).

References

[1] H. K. Dong, "Self-organization of unicycle swarm robots based on a modified particle swarm framework," *International Journal of Control, Automation and Systems*, vol. 8, no. 3, pp. 622–629, 2010.

[2] Y. Wang and C. W. Desilva, "A machine-learning approach to multi-robot coordination," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 470–484, 2008.

[3] J. Liu, X. Jin, and S. Zhang, *The Model and Experiment of Multi-Agent*, Tsinghua University Press, Beijing, China, 2003.

[4] H. Hamann, *Space-Time Continuous Models of Swarm Robotic Systems*, Springer, Berlin, Germany, 2010.

[5] D. H. Kim and S. Shin, "Self-organization of decentralized swarm agents based on modified particle swarm algorithm," *Journal of Intelligent and Robotic Systems*, vol. 46, no. 2, pp. 129–149, 2006.

[6] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," *Swarm Intelligence*, vol. 1, no. 1, pp. 3–31, 2007.

[7] L. Marsh and C. Onof, "Stigmergic epistemology, stigmergic cognition," *Cognitive Systems Research*, vol. 9, no. 1-2, pp. 136–149, 2008.

[8] O. Holland and C. Melhuish, "Stigmergy, self-organization, and sorting in collective robotics," *Artificial Life*, vol. 5, no. 2, pp. 173–202, 1999.

[9] R. Beckers, O. E. Holland, and J. L. Deneubourg, "From local actions to global tasks: stigmergy and collective robotics," in *Proceedings of the Artificial Life*, pp. 181–189, MIT Press, Cambridge, UK, 1994.

[10] L. Bayindir and E. Şahin, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, no. 2, pp. 115–147, 2007.

[11] S. N. Givigi and H. M. Schwartz, "Swarms of robots based on evolutionary game theory," in *Proceedings of the 9th International Conference on Control and Applications*, pp. 1–7, ACTA Press, Montreal, Canada, June 2007.

[12] S. Sang-Wook, Y. Hyun-Chang, and S. Kwee-Bo, "Behavior learning and evolution of swarm robot system for cooperative behavior," in *Proceedings of the International Conference on Advanced Intelligent Mechatronics (AIM '09)*, pp. 673–678, IEEE/ASME, Singapore, July 2009.

[13] K. Kobayashi, K. Nakano, T. Kuremoto, and M. Obayashi, "Cooperative behavior acquisition of multiple autonomous mobile robots by an objective-based reinforcement learning system," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 777–780, IEEE, Seoul, Korea, October 2007.

[14] F. Fernández, D. Borrajo, and L. E. Parker, "A reinforcement learning algorithm in cooperative multi-robot domains," *Journal of Intelligent and Robotic Systems*, vol. 43, no. 2–4, pp. 161–174, 2005.

[15] D. W. Lee, S. W. Seo, and K. B. Sim, "Online evolution for cooperative behavior in group robot systems," *International Journal of Control, Automation and Systems*, vol. 6, no. 2, pp. 282–287, 2008.

[16] M. N. Ahmadabadi, M. Asadpour, and E. Nakano, "Cooperative Q-learning: the knowledge sharing issue," *Advanced Robotics*, vol. 15, no. 8, pp. 815–832, 2001.