Neural Comput & Applic (2014) 25:263–268
DOI 10.1007/s00521-013-1503-y

REVIEW

Swarm robots reinforcement learning convergence accuracy-based learning classifier systems with gradient descent (XCS-GD)

Jie Shao · Haixia Lin · Kaibian Zhang

Received: 27 June 2013 / Accepted: 14 October 2013 / Published online: 17 November 2013

Abstract This paper presented a novel approach accuracy-based learning classifier system with gradient descent (XCS-GD) to research on swarm robots reinforcement learning convergence. XCS-GD combines covering operator and genetic algorithm. XCS-GD is responsible for adjusting precision and reducing search space according to some reward obtained from the environment, XCS-GD's innovation discovery component is responsible for discovering new better reinforcement learning rules. The experiment and simulation showed that XCS-GD approach can achieve convergence very quickly in swarm robots reinforcement learning.

Keywords Convergence · Swarm robots · Reinforcement learning · Accuracy-based learning classifier system with gradient descent (XCS-GD)

1 Introduction

Swarm robots technology has already been widely used in many fields. Urged by investigation and application, research of swarm robots reinforcement learning has been a hot field. In some task orientation applications, swarm robots reinforcement learning convergence and efficient navigation are key technologies of swarm robots navigation, and also the requirement of the rapid development of swarm robots system navigation technology [1–3].

Reinforcement learning is a challenging problem for swarm robots navigation in dynamic and stochastic environments. This paper presented a novel approach XCS-GD to improve the convergence of swarm robots reinforcement learning convergence. The novel approach is responsible for adjusting learning precision and reducing large search space according to some reward obtained from the environment and acts as an innovation discovery component that is responsible for discovering new better reinforcement learning rules.

The rest of this paper is organized as follows. Section 2, we provide a short introduction of XCS. Section 3, we describe swarm robots reinforcement learning. The experiments are designed, and the results are described in Sect. 4. Finally, conclusions and future works are given in Sect. 5.

J. Shao (✉) · H. Lin · K. Zhang
Department of Information Engineering,
Zhengzhou Chenggong University of Finance and Economics,
Zhengzhou 451200, China
e-mail: sjsyxx@163.com; sj012328@163.com

2 Accuracy-based learning classifier system

The XCS-GD classifier system is a learning classifier system (LCS) that evolves its classifier by an accuracy-based fitness approach [4–9]. Each XCS classifier contains the usual condition, action, and reward prediction parts. Swarm robots reinforcement learning convergence is inseparable from environmental information, and it is particularly important that how to obtain timely accurate environmental information through reinforcement learning. Therefore, gradient descent method mapped to learning classifier system and integrated support vector machine algorithm; a new algorithm of learning classifier system based on gradient strategy in application of swarm robots reinforcement learning convergence is proposed.

2.1 Learning classifier systems (LCS)

Learning classifier system (LCS) [10–14] is a machine learning system with close links to reinforcement learning

(RL) and genetic algorithms (GA). LCS is consisted of a population of binary rules on which a genetic algorithm altered and selected the best rules.

2.1.1 Rules and message system

Rules and message system is a special rule-based production system, which directly can interact with its dynamic environment. Using IF THEN way to express:

IF<Condition>**THEN**<ACTION>, strength

Rules and message system
Step 1: Read feedback information from the environment, codify it into message, and sent it into message list
Step 2: Match environmental information and conditions of the rules in the rule base and send the matching rules to the matching rule set
Step 3: Each matching rules generates a message and send it to message list
Step 4: Replace the original message list with the new message list
Step 5: Deal news lists via effectors, resulting in the output of rules, and messaging system
Step 6: Return Step 1

2.1.2 Credit assignment system

The task of credit assignment system is to adjust the strength of the rules, and each classifier will obtain its strength in terms of its usefulness to the whole system. The credit assignment algorithm is updated as follows:

$$B_i(t) = \rho S_i(t) f_i \tag{1}$$

where $0 < \rho \leq 1$ denotes the coefficient of biding, $S_i(t)$ is the current strength of the rule, $S_i(0) = 100$. $B_i(t)$ is the rule bidding value that participate in the bidding, f_i denotes rule efforts to participate in the bidding, $0 \leq f_i \leq 1$.

$$f_i = {}^{l}/_{L} \tag{2}$$

l denotes non-"#" numbers in condition section, L is condition section length, $L = 16$.

$$S_i(t+1) = (1 - \mathrm{Tax}_{\mathrm{life}}) S_i(t) + R_i(t) - \mathrm{Tax}_{\mathrm{bid}} * B_i(t)$$

$$\begin{cases} \mathrm{Tax}_{\mathrm{life}} = 0,\ \mathrm{Tax}_{\mathrm{bid}} = 1 & \text{for winning rules} \\ \mathrm{Tax}_{\mathrm{life}} = 0,\ \mathrm{Tax}_{\mathrm{bid}} = 0 & \text{for non-winning rules in the matching pool} \\ R_i(t) = 0 & \\ \mathrm{Tax}_{\mathrm{bid}} = 0,\ 0 \leq \mathrm{Tax}_{\mathrm{life}} \leq 1 & \text{for non-matching rules} \end{cases} \tag{3}$$

where the $\mathrm{Tax}_{\mathrm{life}}$ and $\mathrm{Tax}_{\mathrm{bid}}$ denote life tax and bid tax, respectively. $R(t)$ serves as the reward from the environment.

2.1.3 Rule discovery system

The task of rule discovery system based on GA is to select parts of rules with high strength as parents and generate new reinforcement learning rules through crossover and mutation. These learning rules displace weak rules and enter the rule set through crowding strategy to maintain excellent of the learning population for preventing the rule population from converging to local optimum.

Rule discovery system implementation process based on GA.

Step 1: Select the match highest intensity focused pair classifiers.
Step 2: Each system running for some time interval; start the genetic algorithm; using selection, crossover, and mutation, operators construct new rules.
Step 3: Newly created rule replaces rule set weakest rules and deposit rule set, in order to maintain the stability of the rule set size.
Step 4: If the entire system does not reach the set iteration value, returns Step 1.

Rules covering algorithm is to learn all the rules in the classifier condition part, and environmental information cannot be activated when matching. Covering algorithm to generate a new rule condition part and environmental information matches the rule set out the minimum intensity of the rules into rule sets.

Covering algorithm is executed as follows:

Step 1: When the rule set does not rule condition part and environmental information matches, start rule covering algorithm.
Step 2: Select the rule set minimum intensity for the classification, its condition part is set to enter the information environment, generate new rules.
Step 3: The action part of the new rule random selection.
Step 4: The strength of the new rules all the rules intensity was.
Step 5: Return Step 1.

2.2 Implementation component

The initial classifier set is randomly generated, each classifier is represented as the <state, action> pair in $|P|$. According to the environmental input, the match set $|M|$ is formed from the population $|P|$, and then action sets $|A|$ is generated by classifier's action. The final, the highest

probability action is chosen. The probability $P(a_i)$ of each action a_i is calculated by the following equation:

$$P(a_i) = \frac{\sum cl_k \in |M| a_i P_k \times F_k}{\sum cl_k \in |M| a_i F_k}. \quad (4)$$

2.3 Reinforcement component

Each classifier in the process of implementation components will obtain a reward from the environment, and the reward prediction, classifier fitness strength will be updated as follows:

$$P \leftarrow R + \gamma \max_a P(a) \quad (5)$$

where $0 \leq \gamma \leq 1$ learning rate, R serves as the reward from the environment.

$$P_j \leftarrow P_j + \beta(P - P_j) \quad (6)$$

$$\varepsilon_j \leftarrow \varepsilon_j + \beta\left(\left|P - \varepsilon_j\right| - \varepsilon_j\right) \quad (7)$$

$$k_j = \begin{cases} 1 & \text{if } \varepsilon_j \leq \varepsilon_0 \\ \alpha(\varepsilon_j/\varepsilon_0)^{-v} & \text{otherwise} \end{cases} \quad (8)$$

$$k'_j = \frac{(k_j \times \text{num}_j)}{\sum cl_k \in |A|_{-1}(k_k \times \text{num}_k)} \quad (9)$$

where $\varepsilon_0(\varepsilon_0 > 0)$ control prediction errors redundancy, $\alpha(0 < \alpha < 1)$ and $v(v > 0)$ denote constant, which control decline of accuracy k rate when ε_0 is exceeded. In action set, the absolute accuracy value k is converted to a relative accuracy value k'. XCS's fitness value is to be updated based on the relative precision of the value:

$$F_j \leftarrow F_j + \gamma * (k'_j - F_j). \quad (10)$$

3 Swarm robots reinforcement learning

3.1 Improved crossover operator based on XCS

Step 1: The policy parameter vector in XCS, the classifier is represented as <conditions, policy parameters>.
Step 2: Select $p_1 = (r_1, s_1)$ and $p_2 = (r_2, s_2)$, conditions and policy parameters are cross-evolution, $r_i = (a_1^{p_i}, a_2^{p_i}, \ldots a_l^{p_i})$ and $s_i = (\delta_1^{p_i}, \delta_2^{p_i}, \ldots \delta_l^{p_i})$ is produced.
Step 3: Each new classifier o_i produced a new strategy parameter vector $s_i = (\delta_1^{o_i}, \delta_2^{o_i} \ldots \delta_l^{o_i})$, $\delta_j^{o_i}$ is random parameters, $\delta_j^{o_i} \in [c_{\min}^i - I_i\alpha, c_{\max}^i + I_i\alpha]$, and $c_{\min}^i = \min(\delta_i^{p1}, \delta_i^{p2})$, $c_{\max}^i = \max(\delta_i^{p1}, \delta_i^{p2})$, $I_i = (c_{\min}^i, c_{\max}^i)$.

3.2 Fitness function design

We think robot fitness function as the main elements of reinforcement learning that robot can avoid obstacles in dynamic or static. So we design the fitness function by whether warm robots is within safety limits.

3.2.1 Safe fitness function

We considered all obstacles as particles; each obstacle has a safe radius; if the distance between robot and obstacle is greater than the safety radius, then it is considered to be safe; if it is less than the safety radius, then that is not safe. Relationship between the radius R and distance d as follows:

$$\text{fit1} = \begin{cases} 0 & d \geq R \\ -1 & d \leq R \end{cases} \quad (11)$$

where $d = \sqrt{(x_o - x_r)^2 + (y_o - y_r)^2}$, obstacle coordinates (x_o, y_o), robot current coordinates (x_r, y_r). fit1 value indicates the robot particle, and obstacles are in a safe distance, and then its fitness is 0; if the robot particle and obstacles are not in a safe distance, then the fitness is -1.

3.2.2 The fitness function for robot strength

$$\text{fit2} = S_i(t+1) \quad (12)$$

3.2.3 Probability success rating fitness function of the swarm robots system

The swarm robots system does not reach the target position before, individual robot to move within the region, at the moment, the robot to the target position prediction own pre-close to the desired speed and azimuth of the target is calculated, and the results posted via the wireless network and to other robot upgrade for leadership robot. Other robots calculated according to the received information on its own position and movement trend, and also calculate the target and other robots and their relative position, the case according to the other robots, the speed and azimuth of the own pre-close to the ideal target for final level programs, and leading robot using weighted probability of success the first establish the position estimation equation as follows:

$$\begin{cases} \left|(x_i + v_i t\cos\theta_i) - (x_g + v_g t\cos\theta_g)\right| = \varepsilon_{i,x} \\ \left|(y_i + v_i t\sin\theta_i) - (y_g + v_g t\sin\theta_g)\right| = \varepsilon_{i,y} \\ \varepsilon_{i,x}^2 + \varepsilon_{i,y}^2 = \overline{\sum l_0^2} \end{cases} \quad (13)$$

where x_i, y_i, v_i, θ_i is, respectively, the robot position, expected speed, and azimuth; x_g, y_g, v_g, θ_g is, respectively, the robot the target position, speed, and direction; $\Sigma'0$ is The target robot mobile security radius; $\varepsilon_{i,x}$ and $\varepsilon_{i,y}$ are the deviation of the two-dimensional component.

Leadership robot x_{leader}, y_{leader}, v_{leader}, θ_{leader} into the above and obtained t_{leader} on behalf of the following formula This system feasibility probability as follows:

$$p_{\text{system}} = p(\varepsilon_{i,x}^2 + \varepsilon_{i,y}^2) \leq \overline{\sum {}_0^{r2}} | t_i \leq t_{\text{leader}} \tag{14}$$

$$\text{fit3} = \begin{cases} 1, & P_{\text{system}} \geq 0.65 \\ 0, & P_{\text{system}} < 0.65 \end{cases} \tag{15}$$

where $P_{\text{system}} \geq 0.65$, indicating that the overall planning of the swarm robots system is feasible; $P_{\text{system}} < 0.65$, indicating that the overall planning of the swarm robots system is not feasible. Taking these factors, the integration fitness function of swarm robots reinforcement learning as follows:

$$f = (1 + \text{fit1}) * \text{fit2} * \text{fit3} * F_j. \tag{16}$$

3.3 Convergence strategy of swarm robots path planning

The XCS classifier system is an LCS that evolves its classifier by an accuracy-based fitness approach. Each XCS classifier contains the usual condition, action, and reward prediction parts. Complementary, XCS contains a prediction error estimate and a fitness estimate, which represents the relative accuracy of a classifier.

XCS can find a set of learning rules through interaction with the environment. These rules can be used to guide the robot collision avoidance and can provide real-time, dynamic feedback for the robot. The warm robots can autonomously learn optimal convergence strategy.

Step 1: The optimal learning strategies as the XCS initial rule set, and randomly perform one of the learning strategies.
Step 2: Condition part of the message in the message list is compared with the current regulations, matched regulations are put into the matching rule set.
Step 3: Selecting the appropriate classifier from the matching rules, and sent these rules to the effectors to guide the robot to generate the corresponding planning action, and to determine the convergence effect based on the feedback value.
Step 4: Repeat Step 2–Step 3 until the matching rule set is empty or rule discovery mechanism is triggered.
Step 5: The rule discovery system using genetic algorithms and rules covering algorithm to construct the new rules.
Step 6: Merge the new rules, so that it can be summed up the previous two samples. The new learning of individual rules is sent to the public rule set for the other robot XCS to generate new rules to share.
Step 7: If swarm robots does not meet the convergence effect, return to Step 2.

4 Experiments and simulation

Suppose all swarm robots experiments are done in a rectangular area, O1–O3 are three dynamic obstacles, the rest is static obstacles in the region. Figure 1 shows the swarm robots reinforcement learning in a static environment. Figure 2 shows swarm robots reinforcement learning in a dynamic environment.

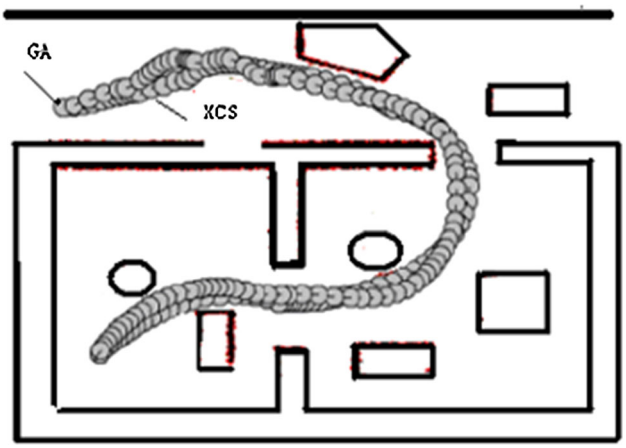

Fig. 1 Swarm robots path planning in a static environment based on XCS

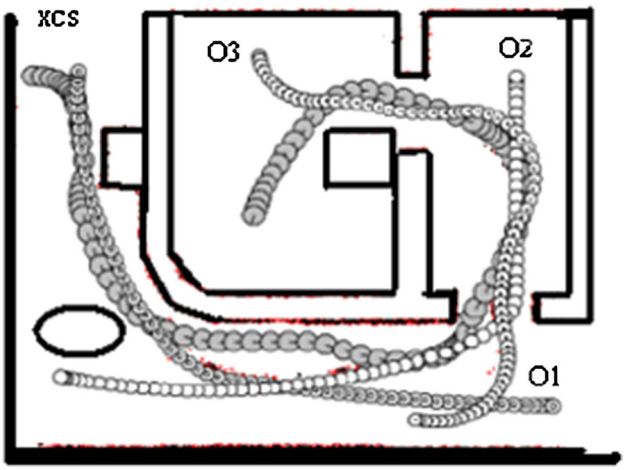

Fig. 2 Swarm robots path planning in a dynamic environment based on XCS

Under normal circumstances, the performance of a reinforcement learning algorithm needs two aspects to determine, one is convergence of algorithm and the other is convergence speed. From Figs. 3 and 4, we can see algorithm convergence trend after a certain number of learning. $R(t)$ is the volatility in the initial process of learning. After a certain number of learning, $R(t)$, the oscillation amplitude is very small, we can consider the approximation to the convergence. Figure 3 is the convergence of traditional reinforcement learning. Figure 4 is reinforcement learning based on XCS; whether it is the robot individual reinforcement learning or swarm robots learning, all robot can receive the desired learning convergence curve in the short period of time.

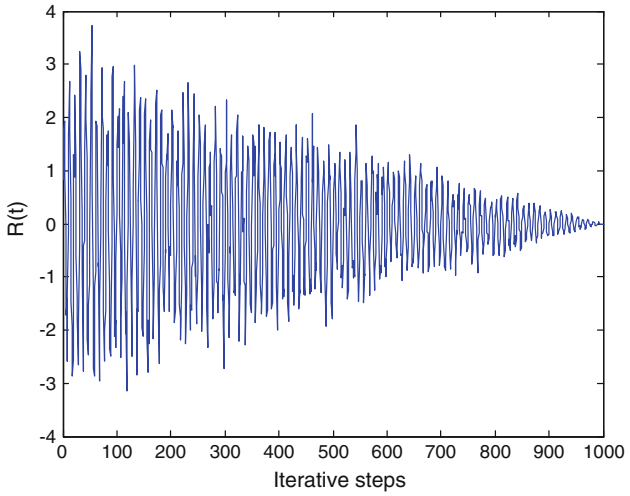

Fig. 3 Convergence steps based on GA

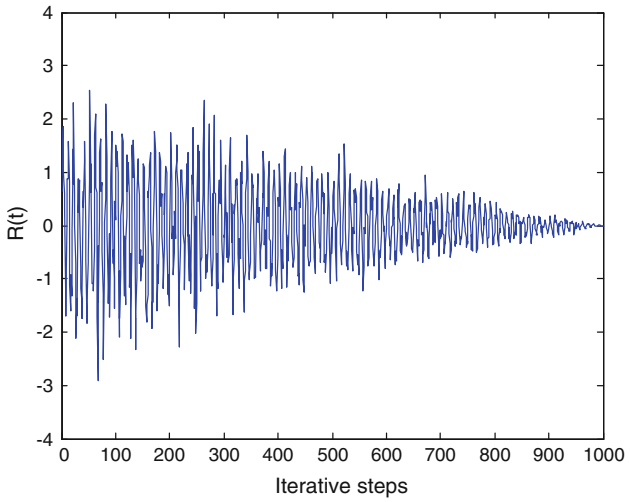

Fig. 4 Convergence steps based on XCS

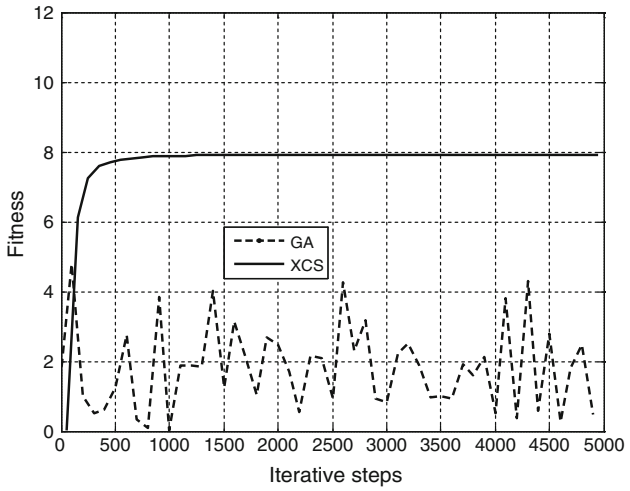

Fig. 5 The convergence iteration curves of the dynamic environment based on the XCS

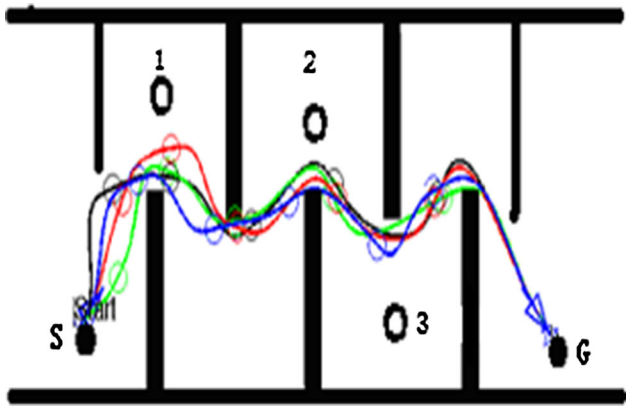

Fig. 6 Multi-robot trajectory in the U-shaped environment

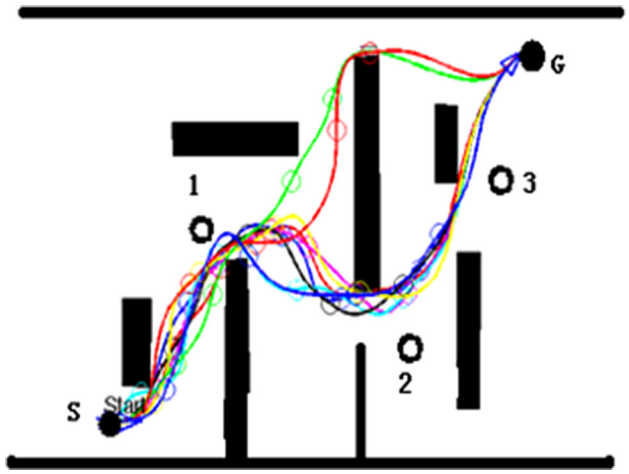

Fig. 7 Multi-channel robot trajectories in narrow channel environment

Figure 5 is the convergence curves of the dynamic environment based on the XCS, and swarm robots have made a better learning convergence. Simulation curve is based on GA algorithm for the swarm robots in unknown dynamic narrow environments, and swarm robots cannot achieve convergence.

Figure 6 shows multi-robots in the narrow context of the U-shaped trajectory. Dynamic obstacles (No. 1–3) were in their narrow U-shaped environment for up and down movement, four were successfully reach the ultimate goal of the robot point G point.

Figure 7 is the multi-robot trajectory in narrow channel environment. Dynamic obstacles (No. 1–3) were down as a straight line in their environment, all involved in the planning of the robots have achieved satisfactory simulation curve. Presented in this paper is XCS-GD-based integration algorithm, because all XCS-GDs have strong ability to forecast returns, all the stability of the robot can achieve a satisfactory convergence effect for multi-robot trajectory in the narrow dynamic environment.

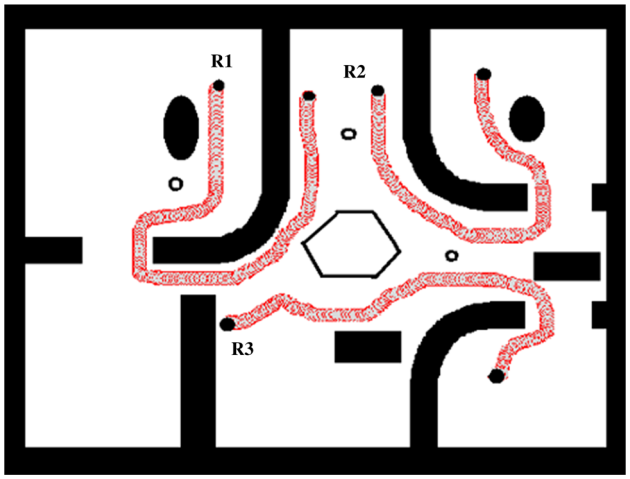

Fig. 8 Trajectories of multi-robot in multi-convex and concave narrow environment

Figure 8 is the trajectory of the multi-robot systems with multi-punch obstacles in narrow environment. Although there are several easy to fall into local minimum risk, the robots (R1, R2, R3) can still be very stable to reach their goals.

5 Conclusions and future works

This paper presented a novel approach to solve swarm robot reinforcement learning convergence. The advantages of XCS-GD are its accuracy-based representation that can easily reduce learning space and improve online learning ability. Simulation indicated that XCS-GD used in the swarm robot's reinforcement learning convergence is effective, and swarm robots can achieve stably convergence very quickly.

Acknowledgments The authors would like to thank the anonymous reviewers and the editor for their helpful comments and suggestions. This work is partially supported by 2013 Henan College "professional comprehensive reform pilot" project and 2012 Education Department of Henan Science and Technology Research Key Project.

References

1. Shao J, Yang J, Wan M, Huang C (2010) Research on convergence of multi-robot path planning based on learning classifier system. J Comput Res Dev 47(5):948–955
2. Lan T, Liu S (2007) Research on multi-robot robot system inspired by biological swarm intelligence. Robot 29(3):298–304
3. Shao J, Yang J (2009) Research on convergence of robot path planning based on LCS. In: Proceedings of Chinese conference on pattern recognition, Oct 22–24, Nanjing, China, pp 271–276
4. Dixon PW, Corne DW, Oates MJ (2002) A preliminary investigation of modified XCS as a generic data mining tool. Adv Learn Classif Syst 2321:133–150
5. Gemeinder M, Gerke M (2003) GA-based path planning for mobile robot systems employing an active search algorithm. Appl Soft Comput 3:149–158
6. Baneamoon SM, Abdul Salam R, Talib AZH (2007) Learning process enhancement for robot behaviors. Int J Intell Technol 2(3):172–177
7. Bull L, Studley M, Bagnall A, Whittley I (2007) Learning classifier system ensembles with rule-sharing. IEEE Trans Evol Comput 11(4):496–502
8. Baneamoon SM, Salam RA (2008) Applying steady state in genetic algorithm for robot behaviors. In: 2008 International conference on electronic design. IEEE, Piscataway, NJ, pp 930–934
9. Bull L (2003) A simple accuracy-based learning classifier system. University of the West of England, Bristol
10. Wang Y, Huber M, Papudesi VN, Cook DJ (2003) User-guided reinforcement learning of robot assistive tasks for an intelligent environment. Proc IEEE/RJS Int Conf Intell Robots Syst 1:424–429
11. Bull L, Kovacs T (2005) Foundations of learning classifier systems: an introduction. Found Learn Classif Syst 183:1–17
12. Musilek P, Li S, Wyard-Scot L (2005) Enhanced learning classifier system for robot navigation. In: IROS 2005, IEEE/RSJ International conference on intelligent robots and systems, Alberta, Canada, 2–6 Aug 2005, pp 3390–3395
13. Bull L, Sha'Aban J, Tomlinson A, Addison JD, Heydecker BG (2004) Towards distributed adaptive control for road traffic junction signals using learning classifier systems. In: Bull L (ed) Applications of learning classifier systems. Springer, Berlin, pp 276–299
14. Bay SJ (1995) Learning classifier systems for single and multiple mobile robots in unstructured environments. Mobile Robots X, Philadelphia, PA, pp 88–99