Research on decision-making strategy of soccer robot based on multi-agent reinforcement learning

Xiaoli Liu

Abstract
This article studies a multi-agent reinforcement learning algorithm based on agent action prediction. In multi-agent system, the action of learning agent selection is inevitably affected by the action of other agents, so the reinforcement learning system needs to consider the joint state and joint action of multi-agent based on this. In addition, the application of this method in the cooperative strategy learning of soccer robot is studied, so that the multi-agent system can pass through the environment. To realize the division of labour and cooperation of multi-robots, the interactive learning is used to master the behaviour strategy. Combined with the characteristics of decision-making of soccer robot, this article analyses the role transformation and experience sharing of multi-agent reinforcement learning, and applies it to the local attack strategy of soccer robot, uses this algorithm to learn the action selection strategy of the main robot in the team, and uses Matlab platform for simulation verification. The experimental results prove the effectiveness of the research method, and the superiority of the proposed method is validated compared with some simple methods.

Keywords
Soccer robot, multi-agent, reinforcement learning, decision-making strategy

Date received: 16 December 2019; accepted: 20 February 2020

Topic: Robot Manipulation and Control
Topic Editor: Andrey V Savkin
Associate Editor: Bin He

Introduction
Multi-agent system (a distributed system composed of multiple independent autonomous agents, which are in the same working environment, can sense the environmental information and perform their own actions) robot soccer match is a typical multi-agent system research platform, which is also a field of artificial intelligence and robotics machine learning. At present, the subject of high challenge has received extensive attention and research. However, the process of robot soccer match is complex, dynamic and uncertain, which makes the decision-making system based on expert knowledge is lack sufficient completeness and flexibility to deal with the process of the complex, dynamic and uncertain of robot football game, however, the reinforcement learning method does not need accurate environment model and complete expert knowledge. Robots learn decision-making ability and behaviour ability through interaction with environment in competition, so it provides a new way for the research of multi-robot system.

With people’s attention to the concept of machine learning, reinforcement learning has also attracted people’s close attention. More and more methods and theories have been developed in reinforcement learning. This article uses the characteristics of decision-making of soccer robot to develop a multi-agent reinforcement learning algorithm based on agent action prediction, and applies it to the local attack strategy of soccer robot to learn the action selection strategy of the main robot in the team through simulation verification. The experimental results prove the effectiveness of the research method, and the superiority of the proposed method is validated compared with some simple methods.

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
put forward.\textsuperscript{9} Wang uses a kind of weighted linear function to analyse the game strategy of Western checkers.\textsuperscript{10} Although his method has defects, his method embodies the thought rudiments of time difference and value function approximation, which provide a thinking mode for the development of reinforcement learning in the future.\textsuperscript{11} In 1984, Chen proposed markov decision process (MDP)-based reinforcement learning model, and then reinforcement learning gradually came into public's attention.\textsuperscript{12} In the later research, Monte Carlo and dynamic programming are combined to produce the time-series difference algorithm, which is a very important algorithm, setting the basis for reinforcement learning.\textsuperscript{13} Firstly, it defines the return value and judges the state of the system based on it. Then, it makes the estimated value close to the actual value by value iteration. Wang proposed the Q-learning algorithm, which opened a new chapter for reinforcement learning.\textsuperscript{14} The Q-learning algorithm estimates the state action value function and obtains the optimal strategy step by step.\textsuperscript{15} In the study by Fang et al.,\textsuperscript{16} firstly, an agent reinforcement learning method based on zero sum game is proposed. Meola et al.\textsuperscript{17} studied the reinforcement learning method of multi-agent based on general and games and proved that the algorithm converges to the equilibrium point in a specific environment. By introducing the concept of regret value, the multi-agent learning method based on greedy strategy is studied.\textsuperscript{18} The literature extended the model, studied a multi-agent learning method based on action repetition process, discussed the convergence of the algorithm proposed a reinforcement learning method based on simulation mechanism in the application of the literature, studied the application of the method in robot motion control and discussed the application of reinforcement learning method to solve the problem.\textsuperscript{19} In this article, the distributed reinforcement learning is studied on the problems of navigation, collection and confrontation in multi-robot task, and the research method is applied in the simulation game of soccer robot.

For individual reinforcement learning, multi-agent reinforcement learning is more suitable for solving complex multi-agent problems. These new topics include the learning objectives of multi-agent, the relationship between individual return and team return, the role and influence of agents in the learning process, the acquisition of joint state and joint action and dimension disaster caused by expanded state space and action space.

However, at present, the theoretical framework of multi-agent reinforcement learning is mostly based on one-to-one game strategy, only considering the confrontation between agents, not considering the group cooperation of multi-agent, not fully analysing the change of individual state and the interaction between action selection and other agents, so as to strengthen learning in multi-agent. In terms of research, the existing work focuses on the learning of specific action implementation effect, most of which is to decompose the complex learning tasks of multi-agent, and then use individual reinforcement learning to learn each task, instead of using multi-agent reinforcement learning method itself to solve complex multi-agent problems. The main contributions of the article are as follows:

1. Probabilistic neural network (PNN) takes the joint state of multi-agent as the input of network, so it can predict the possible action of single agent.
2. At the same time of predicting possible actions, considering the actions of other agents, taking into account the idea of coordination and collaborative control, PNN is used to modify the predicted actions of the agent.
3. The action prediction unit and reinforcement learning unit are deeply integrated to improve the accuracy and accuracy of reinforcement learning.

In this article, a multi-agent reinforcement learning method based on agent behaviour is studied. PNN is used to predict the actions of agent selection, so as to obtain the combined actions of all learning agents, to extend individual reinforcement learning to multi-agent reinforcement learning and to establish the joint state space to joint action through interactive learning with environment. Finally, the application of multi-agent reinforcement learning method in robot soccer game local confrontation strategy is discussed. Experiments show that the algorithm is reliable and effective.

**Multi-agent system of soccer robot**

**The basic architecture of multi-agent system**

Agents explore the external environment through their own perceptor and will make their own adjustments to the changes of the environment, change their internal state and then perform the corresponding behaviour through the actuator to act on the external environment. The agent perceives the external environment, analyses and merges the obtained information with its internal information and then selects the behaviour under the guidance of the knowledge base and acts on the environment through the actuator. The intelligent remote management system (IRMA) system is developed by Wu.\textsuperscript{20} It is a typical cognitive structure. The main model is the belief desire intention (BDI) model. The BDI model mainly includes the following elements: belief related to the world, the goal of agent, knowledge rule base, the intention of agent to goal and belief. The advantage of agent's deliberative structure is high intelligence, but the disadvantages are that the problem solving speed is slow and it is dynamic it is difficult to apply in them. The BDI model of the single-agent in IRMA system is shown in Figure 1.

The multi-agent system is a collection of multi-agent, which is loosely coupled. It is a complex system divided into some simple, mutual communication and mutual consultation system. The idea of multi-agent system is to make the problem-solving ability of the system as a whole greater.
than the algebraic sum of the problem-solving ability of a single agent through the interaction between agents. As an agent in multi-agent system, it has two important capabilities. First of all, each agent has the general properties of the agent described above, especially autonomy. Second, each agent should communicate with each other, in a sense, it is the embodiment of sociality. The interaction among agents is not a simple data exchange, but a negotiation, cooperation or collaboration to participate in and complete certain behavior, just like the human or animal behavior in sociology, biology or ecology. Therefore, the structure of multi-agent system includes two parts: individual agent and group agent. The purpose of building a multi-agent system is to decompose a large-scale complex system into a small-scale, simple, interactive and coordinated system that is easy to manage. The main characteristics of multi-agent system include cooperation, parallelism, stability, expansibility, distribution and so on. The multi-agent system is shown in Figure 2.

The careful thinking structure based on BDI model requires each team member to maintain a stable environment model and a stable belief, so that accidental error information can be corrected and uncertain information can be recalculated, but it is not easy to establish an accurate and complete BDI model in the dynamic and real-time environment, thus affecting the correct planning and decision-making.

**Hierarchical structure of decision framework for soccer robot**

Combined with the characteristics of the deliberative structure and the reactive structure, the two structures are combined in a control framework through the layered structure, and the function blocks are constructed in layers. The main idea of layer structure is to divide the structure of agent into multiple layers according to the function of agent, and the layers can interact with each other. Hierarchical structure has the following advantages:

1. Modularize the agent. Different functions can be clearly separated and connected through defined interfaces.
2. Make the structure of the agent more compact, increase the robustness of the system and facilitate the debugging.
3. Make the complex structure into relatively simple modules, easy to implement.

We divide the agent into five layers: communication layer, action layer, visual model layer, evaluation layer and decision layer. As shown in Figure 3, at the bottom, it is reasonable to solve the action without considering the tactical interests. And the upper strategic level combines the tactical interests to further evaluate and arbitrate the candidate actions. The candidate actions in the module compete locally and then submit to the higher level arbitrators for arbitration between modules. Generally speaking, the overall structure is a hierarchical and modular multi-level evaluation arbiter.

A widely used decision framework is role-based decision tree framework. Firstly, decision tasks are divided into roles, and each role makes decision according to a different rule set. The rule set is expressed in the form of decision tree, which embodies the idea of hierarchical decision.

**Research on multi-agent reinforcement learning algorithm based on action prediction**

The generalized Markov process of multi-agent can be regarded as a random strategy, which is defined as follows: a random strategy can be represented by a tuple $(S, A^1, A^2, \ldots, A^n, f^1, \ldots, f^n, g^1, \ldots, g^n)$, where $n$ is the
number of agents in the system, $S$ is the environment state set and $A^1, A^2, \ldots, A^n$ is the action set that agent can choose. The joint action set is expressed as $A = A^1 \times \cdots \times A^n$. $f^i : S \times A \times S \rightarrow [0, 1]$ means state transition probability function; $g^i : S \times A \times S \rightarrow R$ reinforcement signal function. In multi-agent system (MAS), state transition is the result of joint action of all agents in the system. Therefore, reinforcement signal function also depends on joint action, and the mapping strategy from state to action is expanded to joint strategy $\pi$, and the reinforcement learning method of multi-agent is transformed to joint strategy $\pi$, which means mapping learning from state space to joint action space under joint strategy.

Q-learning is one of the most important reinforcement learning algorithms. Based on the above definition, the Q-learning method of two agents proposed in the literature based on general and game is extended to Q-learning of multiple agents, in which the Q-function depends on the actions performed by all agents, so the function update rules of agents at time $t$ can be expressed as

$$Q(s_t^i, a_t^i) = (1 - \alpha_t)Q_{t-1}(s_t^i, a_t^i) + \alpha_t [R_t^i(s_t^i, a_t^i) + \beta \pi^i(s_{t+1}) \cdots \pi^n(s_{t+1})Q_{t-1}(s_{t+1})]$$

(1)

$$\pi^1(s_{t+1}) \cdots \pi^n(s_{t+1})Q_{t-1}(s_{t+1})$$

$$= \sum_{a^1 \in A^1} \cdots \sum_{a^n \in A^n} P^i_t(s_{t+1}, a^1) \cdots P^n_t(s_{t+1}, a^n)Q_{t-1}(s_{t+1}, a^1 \cdots a^n)$$

(2)

where $s^i_t$ is the state variable of agent $i$, $a^i_t = \{a^1_t, a^2_t, \cdots, a^n_t\}$ represents the joint action of multi-agent. $s^i_{t+1}$ is the joint state of multi-agent at the next moment in which the state transition dependent function $s^i_{t+1} = f^i_t(s^i_t, a^i_t)$ of individual agent. The strategy of agent $i$ is represented by the probability distribution $\pi^i$ of its action set. $P^i_t(s^i_{t+1}, a^i_t)$ is the probability of agent $i$ selecting action $a^i_t$ in the joint state.

It is noted that the main difference between the above algorithm and the basic Q-learning algorithm is to redefine the function of state action pair $(s^i_t, a^i_t)$ as the function of learning agent’s state and combined action $(s^i_t, a^i_t)$. Therefore, the key problem of multi-agent reinforcement learning is how to determine the joint state and joint action of multi-agent because multi-agent is to select action at the same time. In this way, each agent cannot know what actions other agents will perform, so it is impossible to accurately determine the joint actions of multi-agent. However, for most learning problems, the actions of other agents do not happen at will, but according to a certain probability distribution of action selection strategy, the multi-agent reinforcement learning system is constructed. It is composed of action prediction unit and reinforcement learning unit, and its structure is shown in Figure 4.

**Multi-agent action prediction based on PNN**

PNN can be considered as a kind of neural network model for classification. Therefore, the action prediction unit of multi-agent reinforcement learning is used to predict the actions of other agents. Bayesian decision analysis of window estimation is realized using the structure of neural network, and the network structure is shown in Figure 5.

The method of agent action prediction is to take the joint state $s = \{s_1, s_2, \cdots, s_M\}$ of multi-agent as the input vector of network, the candidate action of agent as the decision category $\theta_1, \cdots, \theta_K, \cdots, \theta_L$ of network output, and then get the probability of action selection under the joint state through network reasoning. PNN is a four-layer forward network structure, including input layer, model layer, accumulation layer and transmission layer. The function of the output layer input layer is to transmit the input vector $s$ directly to each node mode layer of the mode layer to complete the weighted sum of the input mode vector $s$ and the weight vector $\omega_j$ of a given class, and to transmit the result to the accumulation layer after nonlinear operation. The accumulation layer accumulates the probability that the input vector belongs to the same class, and transmits the result to the decision layer. The neuron in the decision layer is a kind of competitive neuron, which receives all kinds of probability density functions from the accumulation layer.
The above PNN is completely equivalent to the Bayesian pattern classification method using the multivariate probability density function of Gaussian kernel. The calculation of the probability density function of the input vector belongs to the class is as follows.

**Figure 4.** The structure of multi-agent reinforcement learning system.

**Figure 5.** The structure action prediction based on probabilistic neural network.
In PNN, the weight of pattern layer is

\[ f_K(\tilde{s}) = \frac{1}{2\pi^M/\delta^M} \sum_{j=1}^{n_K} \exp \left[ -\frac{(\tilde{s} - \tilde{s}_{Kj})^T(\tilde{s} - \tilde{s}_{Kj})}{2\delta^2} \right] \]  

(3)

where \( M \) is the number of components of the input mode vector and \( \tilde{s}_{Kj} \) is the \( j \)-th training sample vector of class \( K \).

In PNN, the weight of pattern layer is \( \omega_j \), \( n_K \) is the number of \( k \)-class training samples and \( \delta \) is called smooth coefficient, which is used to adjust the density function.

The PNN decision-making layer uses the Bayesian decision-making criteria to judge the status of the category \( \theta \in \theta_Q \), which can be expressed as follows

\[ d(\tilde{s}) \in \theta_Q, \text{If} \quad h_Q l_Q(\tilde{s}) > h_K l_K(\tilde{s}) \]  

(4)

where \( D \) \( d(\tilde{s}) \) is the Bayesian decision of test vector, \( h_Q \) and \( h_K \) are the prior probabilities of \( \theta \in \theta_Q \) and \( \theta \in \theta_K \), respectively, \( l_Q \) is the loss that belongs to \( \theta_Q \) and is misclassified into other categories. At this point, PNN can be used to predict the actions to be selected by the agent.

In the reinforcement learning of multi-agent, the input mode vector is the joint state \( \tilde{x} \), and the number of nodes in the input layer is determined by the number of \( \tilde{x} \) components to determine the action space of the agent as the decision category. The number of candidate actions determines the number of nodes in the accumulation layer. Therefore, the problem of agent action prediction is equivalent to PNN is used to classify the input joint state vector to the action it belongs to. The action prediction unit and reinforcement learning unit are carried out at the same time in the learning process. Finally, the perfect action prediction strategy and action selection strategy mode layer nodes are grouped according to the decision category, and the weight of each node in each group represents the training sample vector.

In the process of reinforcement learning, the learning samples of each step are continuously added to the corresponding groups of the mode layer, and the number of samples belonging to each type is updated at the same time. According to formula (3), the probability \( f_K(\tilde{s}) \) can be calculated by inputting the joint state vector \( \tilde{s} \). Finally, under the joint state \( \tilde{s} \), the conditional probability of selecting the action \( a_k \) of the agent can be expressed as

\[ P(a_k|\tilde{s}) \rightarrow h_K l_K(\tilde{s}) \]  

(5)

The prior probability \( h_K \) of selecting action \( a_k \) can be estimated from the frequency of action \( a_k \) in the learning process

\[ h_K = \frac{v_k}{v} \]  

(6)

where \( v_k \) is the number of samples belonging to category \( a_k \) in the input joint state vector set selection action and \( v \) is the total number of training samples. At the same time, the conditional probability is normalized.

The loss \( l_k \) of misclassification is equivalent to the loss caused by the action \( a_k \) that should have been selected. It can be considered that \( l_k \) is related to the reinforcement signal \( r_k \) obtained by reinforcement learning

\[ l_k = \lambda r_k \]  

(7)

The above formula shows that the smaller the reward is, the greater the error classification loss is, while the function definition is related to the specific form of the enhancement signal function.

Based on this, the value updating process of table multi-agent reinforcement learning is similar to that of standard learning algorithm. The main difference is that the state of agent and joint action form the evaluation value of table. Other agent actions are predicted by the above and then updated by the methods of formulas (1) and (2), so as to realize multi-agent learning.

The whole structure PNN-based reinforcement learning is shown in Figure 6. As is shown in Figure 6, two probabilistic neural networks are applied in the whole algorithm structure, in which PNN(I) collects the environmental information transmitted by the communication channel to predict the actions of the whole multi-agent. Through formulas (1) and (2), the strategy and gradient value of Q-learning are adjusted. PNN(II) corrects the error of PNN(I) learning prediction by observing the cooperative action.
among agents. The final prediction probability is transferred to Q-learning to update and adjust the whole algorithm.

**Results and discussion**

**Simulation environment settings**

In the experiment of robot soccer, the action set of each agent includes three actions: interception, shooting and sweeping. Each action is implemented by a routine in which the sweeping action is to sweep the ball to the other half when the ball is on the edge of the court. The execution condition of each action is that the ball is in the execution range of the action of the soccer robot. The Q-learning system uses the PNN as shown in Figure 5, with eight inputs: the distance between the main agent and our goal, the distance between the main agent and the opponent’s goal, the distance between the main agent and the ball, the azimuth angle between the main agent and the ball, the distance between the main agent and the nearest teammate, the azimuth angle between the main agent and the nearest teammate, the distance between the main agent and the nearest opponent and the distance between the main agent and the nearest opponent. The output is the three basic actions mentioned above, and the number of hidden layer nodes is 24.

Based on the above research methods, the standard simulation platform of soccer robot game and the actual micro robot are used for the experiment. For the task of soccer robot game, three robots of our side and two robots of the other side are selected, and the multi-agent reinforcement learning method introduced above is used to make robot players learn the decision-making strategy. The other player uses the fixed strategy. The main task of the defensive player is to seize the ball control right of the other player. The optional actions of the player include chasing the ball and moving to the position between the robot controlling the ball and the first cooperative player. The strategy of column interception and passing is for the defensive player who is closest to the ball or who can reach the ball position as soon as possible, runtoball() move to catch the ball, and other defensive players move to the appropriate position to perform blockpass() action to block the cooperation of the other side. This article uses Matlab platform to simulate and verify the algorithm. The basic hardware configuration is i5-9400 processor, 240g solid-state hard disk.

**Parameter selection of reinforcement learning**

In the reinforcement learning process, $\alpha$ and $\beta$ are important factors that affect the convergence speed/time and accuracy of the algorithm. There are basically two ways to adjust the super parameters. One is to fix other parameters, starting from the most important parameters, one by one. The disadvantage is that the parameters may balance each other. The other is to adjust several super parameters at the same time. Through traversal or online method, if it is good to find a large area, refine the large area and continue to find. The disadvantage is that there are not too many super parameters involved in simultaneous regulation.

This article compares the accuracy analysis of the algorithm under different parameters $\beta$. As shown in Figure 7, we can see from the analysis that the $\beta$ value has no effect on the convergence time of the algorithm, but only affects the accuracy of the algorithm. When $\beta = 1.5$, the accuracy of reinforcement learning is the highest, which can reach 93%. When $\beta = 0.7$, the accuracy can reach 62.5%.

In the same way, we analysed the influence of different $\alpha$ values on the convergence speed and prediction accuracy of the algorithm. We used the fixed variable method, $\beta = 1.5$ was taken, different $\alpha$ values were chosen and analysed the different comparison strategies. The simulation results under different $\alpha$ values are shown in Figure 8.
As is shown in Figure 8, when $\alpha = 0.1$, the accuracy of the algorithm can be 73%. With the increase of a value, the accuracy first increases and then decreases. When $\alpha = 0.6$, the accuracy reaches the peak value 95%. At the same time, the convergence time of the algorithm is the fastest. Therefore, in the later simulation process, we chose $\alpha = 0.6$ and $\beta = 1.5$.

The effectiveness of multi-agent reinforcement learning

To verify the effectiveness of the learning algorithm, other strategies are used to control the ball, including random strategy, that is, our players randomly select the candidate actions with the same probability; holding strategy, that is, our players always perform the Holdball() action. Hand coded strategy: if our players have no other players within a certain distance, then hold the ball; otherwise, find out whether there is a chance to pass and wait for the opportunity to execute the pass. Figure 9 shows the change of our average plot control duration with the learning process.

At the same time, it shows the average control duration schedule of random strategy, always holding strategy and manual coding strategy. The learning results are shown in Table 1. For the basic strategies of opponents who adopt different strategies (always holding strategy, random strategy and fixed strategy), the average accuracy of agent action prediction and learning strategy can correctly execute the selected action success rate. The detailed results are shown in Table 1. For different opponents, our multi-agent can accurately perform our catching, passing and other operations, and the overall accuracy can be maintained at about 84%.

The experimental results of this article are compared with the experimental results provided in Wu. The average ball control time of our robot obtained by reinforcement learning is $10.4 \pm 0.4$ s, and the average ball control time of our robot obtained by the experimental results of this article is $11.5 \pm 0.6$ s. We analysed the factors influencing the decision-making strategy in the research object and redefine the strong strategy. The state space, action space and reinforcement signal function of learning reflect the direct interaction and restriction of multi-agent. The goal of learning is to make our players control the ball as long as possible. Therefore, the experimental results prove the effectiveness of this study.

### Conclusion

In this article, a multi-agent reinforcement learning method based on agent action prediction is studied. The actions selected by each agent are not only determined by state information but also affected by the actions performed by other agents. Therefore, the agent team cooperation is realized using the action prediction unit PNN to predict the selected actions of other agents. In addition, the proposed multi-agent reinforcement learning method is used to realize the learning of soccer robot problems. The experimental results show that the method in this article can make the soccer robot gradually master the decision-making ability and behaviour ability in the game practice without the need of expert knowledge. The method in this article can be used to solve the complex multi-agent division of labour and cooperation of energy bodies. At the same time, the simulation process of the algorithm verifies the influence of different parameters on the accuracy and convergence speed of the strategy. Therefore, in the future work, adaptive parameters can be proposed to deal with different systems, which always make the algorithm more universal.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

### ORCID iD

Xiaoli Liu https://orcid.org/0000-0002-2380-1295
References
1. Wang H, Chen X, Wu Q, et al. Integrating reinforcement learning with multi-agent techniques for adaptive service composition. *ACM Trans Auton Adapt Syst* 2017; 12(2): 1–42.
2. Wu S. A traffic motion object extraction algorithm. *Int J Bifurcat Chaos* 2015; 25(14): Article Number 1540039.
3. Jin J and Ma X. A multi-objective agent-based control approach with application in intelligent traffic signal system. *IEEE T Intell Transport Syst* 2019; 20(10): 3900–3912.
4. Shi G, He Y, Yin B, et al. Analysis of mutual couple effect of UHF RFID antenna for the internet of things environment. *IEEE Access* 2019; 7: 81451–81465.
5. Wu S, Wang M and Zou Y. Research on internet information mining based on agent algorithm. *Future Gener Comput Syst* 2018; 86: 598–602.
6. Leottau DL, Ruiz-del-Solar J and Babuška R. Decentralized reinforcement learning of robot behaviors. *Artif Intell* 2017; 1(1): 256–264.
7. Peng XB, Berseth G, Yin K, et al. DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM T Graphic* 2017; 36(4): 1–13.
8. Ke Q, Wu S, Wang M, et al. Evaluation of developer efficiency based on improved DEA model. *Wireless Pers Commun* 2018; 102(4): 3843–3849.
9. He Y, Zhang Z, Yu FR, et al. Deep reinforcement learning-based optimization for cache-enabled opportunistic interference alignment wireless networks. *IEEE Trans Veh Technol* 2017; 66(11): 10433–10445.
10. Wang X, Wang X and Qi C. Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. *J Intell Manuf* 2016; 27(2): 325–333.
11. Wu S, Wang M and Zou Y. Sewage information monitoring system based on wireless sensor. *Desalin Water Treat* 2018; 121: 73–83.
12. Chen H, Li X and Zhao F. A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks. *IEEE Sens J* 2016; 16(8): 2763–2774.
13. Wu S, Wang M and Zou Y. Bidirectional cognitive computing method supported by cloud technology. *Cogn Syst Res* 2018; 52: 615–621.
14. Wang H, Wang X, Zhang X, et al. Effective service composition using multi-agent reinforcement learning. *Knowl-Based Syst* 2015; 92(C): 151–168.
15. Wu S, Wang M, Jin M, et al. Uniform L1 stability of the inelastic Boltzmann equation with large external force for hard potentials. *Discrete Cont Dyn Syst Ser S* 2019; 12(4–5): 1005–1013.
16. Fang H, Wang C, Li S, et al. A comprehensive study on the locomotion characteristics of a metameric earthworm-like robot part B: gait analysis and experiments. *Multibody Syst Dyn* 2015; 35(2): 153–177.
17. Meola VC, Caligiore D, Sperati V, et al. Interplay of rhythmic and discrete manipulation movements during development: a policy-search reinforcement-learning robot model. *IEEE Trans Cogn Devel Syst* 2015; 8(3): 152–170.
18. Wu S. Nonlinear information data mining based on time series for fractional differential operators. *Chaos* 2019; 29: 013114.
19. Xi L, Chen J, Huang Y, et al. Smart generation control based on multi-agent reinforcement learning with the idea of the time tunnel. *Energy* 2018; 153: 977–987.
20. Wu S, Liu J and Liu L. Modeling method of internet public information data mining based on probabilistic topic model. *J Supercomput* 2019; 75: 5882–5897.