

Multi-agent system application in accordance with game theory in bi-directional coordination network model

ZHANG Jie^{1,*}, WANG Gang², YUE Shaohua², SONG Yafei², LIU Jiayi², and YAO Xiaoqiang²

1. College of Electronics and Information Engineering, Air Force Engineering University, Xi'an 710054, China;

2. College of Air Missile Defense, Air Force Engineering University, Xi'an 710054, China

Abstract: The multi-agent system is the optimal solution to complex intelligent problems. In accordance with the game theory, the concept of loyalty is introduced to analyze the relationship between agents' individual income and global benefits and build the logical architecture of the multi-agent system. Besides, to verify the feasibility of the method, the cyclic neural network is optimized, the bi-directional coordination network is built as the training network for deep learning, and specific training scenes are simulated as the training background. After a certain number of training iterations, the model can learn simple strategies autonomously. Also, as the training time increases, the complexity of learning strategies rises gradually. Strategies such as obstacle avoidance, firepower distribution and collaborative cover are adopted to demonstrate the achievability of the model. The model is verified to be realizable by the examples of obstacle avoidance, fire distribution and cooperative cover. Under the same resource background, the model exhibits better convergence than other deep learning training networks, and it is not easy to fall into the local endless loop. Furthermore, the ability of the learning strategy is stronger than that of the training model based on rules, which is of great practical values.

Keywords: loyalty, game theory, bi-directional coordination network, multi-agent system, learning strategy.

DOI: 10.23919/JSEE.2020.000006

1. Introduction

In the field of artificial intelligence, the conventional research object primarily focuses on the intelligent problem of individual agents. However, in the practical engineering application, agent individuals often exhibit social behaviours (e.g., information interaction and cooperation with other agent individuals). In this scenario, the concept of the multi-agent system (MAS) has been proposed and becomes one of the two research branches in distributed

artificial intelligence [1]. The MAS research is primarily split into the agent design, interactive cooperation and utility allocation between agents, and alliance training algorithm design [2,3].

For the MAS, the major problem is that there are considerable information exchanges in the system, which is not only associated with the availability of highly efficient and sophisticated message-passing mechanisms actually provided by current multi-agent platforms, but also with the selection of an appropriate communication strategy [4]. Coordination of actions and plans that should be achieved by multiple agents is one of the most arduous tasks in the multi-agent domain [5]. The component agents will sometimes compete or conflict with objectives. To achieve a common goal, agents should coordinate their plans in a way that ensures the success of each individual agent's plan. Allouche et al. [3] proposed a temporal fusion mechanism, allowing a set of agents to fuse their plans and thus formulating a global coordinated plan [6]. They define a temporal plan as a set of temporally constrained actions. The fusion of several temporal plans refers to a temporal plan, which can be executed by several agents. Furthermore, the proposed framework is employed for combat searches and rescues.

In recent years, various measurement methods in accordance with the information theory are proposed to eliminate redundant features of high dimensional data sets. However, most conventional information-based selectors ignore some features, most of which are strong as a group and weak as an individual. To deal with this problem, Sun et al. [7] introduced a cooperative game theory-based framework to assess the power of each feature. Most literature of MAS's formation focus on characteristic function games, in which the effectiveness of a coalition is not affected by how the other agents are arranged in the system. In contrast, the emphasis is placed on how the formation of one coalition can affect the action of other co-existing coalitions in the system, whereas less attention is

Manuscript received April 30, 2019.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (61503407; 61806219; 61703426; 61876189; 61703412) and the China Postdoctoral Science Foundation (2016 M602996).

paid to more general class of partition function games. In the case of the competing or conflicting objectives of each agent, Rahwan et al. [8] proposed the first computational study on coalitional games with externalities in the MAS context. They focus on the coalition structure generation (CSG) problem, which covers the process of finding an exhaustive and disjoint division of the agents into coalitions, thus the performance of the entire system is optimized.

The game theory faces the model and the solution of various game issues has undergone rapid development over half a century and has obtained several landmark achievements [9]. Economists often ask how private information is shared through markets, costly signalling, and other mechanisms. However, most information sharing is achieved through the ordinary and informal talk. Economists also hold different opinions about “cheap talk”. Some consider that the communication generally leads to efficient equilibria, while others claim that it is never credible because “talk is cheap”. In [10], the authors thought both views were wrong. They describe that some recent researches in the game theory show that people could convey private information through cheap talks. In [11], new methods that significantly impact the way were proposed to construct software for the economic game theory. They build complex and irregular games from smaller and simpler games for operators, and they show how the equilibria of these complex games can be defined recursively from the equilibria of their simpler components.

To cope with low efficiency of resource utilization and poor coordination ability among units in the current command and operation process, this paper constructs a deep learning training model based on the bi-coordinated network, and introduces the multi-agent to solve complex problems. Due to the low coupling of the multi-agent system and the self-benefit of the agent, this paper introduces loyalty based on the game theory, analyzes the game relationship between the individual income and global benefits of the multi-agent system, and establishes the algorithm model. In addition, based on the recurrent neural networks (RNNs), a network communication topology based on the bi-directional coordination network as the network communication topology of the MAS is designed, so that it is capable of autonomous learning and internal circulation, achieving the autonomous learning strategy. The goal is to increase the ability of autonomous learning strategy of the model, and enhance the convergence of the algorithm, so that the algorithm can obtain the maximum global benefit and reduce the local benefit, thus the training model has a wider adaptability. The MAS model uses the deep multi-agent enhanced learning algorithm, and constructs a vectorized actor-critic framework for learning, which improves the ability of autonomous learning strategy of the

model, and enhances the convergence of the algorithm. Compared with other training models, the model presented in this paper exhibits a better strategy learning ability and has more practical values.

The rest of this paper is organized as follows. In Section 2, the literature review is summarized. In Section 3, the design of the training algorithm based on the bi-directional coordination network algorithm is illustrated. In Section 4, simulation result and analysis of the algorithm are presented. In Section 5, the model training and the analysis of strategy learning results are described. Finally, in Section 6, conclusion and future prospects are drawn.

2. Related description

2.1 Agent design based on revenue and loyalty

To cope with low efficiency of resource utilization and poor coordination ability among units in the current command and operation process, a deep learning training model is constructed based on the bi-coordinated network, and the multi-agent is introduced to solve complex problems.

Given that the agent withdrawn from the alliance will cause the other alliance’s members to pay a certain cost and even affect the final global benefit, based on the above condition, this paper builds agent that takes the global benefit and the member agent’s interests and loyalty as parameters, that is $Agent = \{Ag, Global_Benefit, Individual_Idealincome, Actual_Income, loyalty\}$. Ag is a collection of the agent, $AG = \{Ag_1, Ag_2, Ag_3, \dots, Ag_n\}$, which is the only primary key to identify the agent. $Global_Benefit$ is the optimal global benefit of agent attribution alliance $GB = \{GB_1, GB_2, GB_3, \dots, GB_n\}$. $Individual_Idealincome$ is the ideal income of the agent after the completion of the task, $II = \{II_1, II_2, II_3, \dots, II_n\}$. II is determined by the overall ideal benefit. $Actual_Income$ is the actual individual income after the agent completes the task, $AIE = \{AIE_1, AIE_2, AIE_3, \dots, AIE_n\}$. $Loyalty$ refers to a definition that the agent participates in the alliance and the accomplishment definition after the task ends, $Ly = \{Ly_1, Ly_2, Ly_3, \dots, Ly_n\}$, where the value of loyalty is determined by GB and II at the end of the task when the union is dissolved. The formula is written as follows:

$$Ly_n = II_n - GB_n, \quad n \in 1, 2, 3, \dots, n. \quad (1)$$

Loyalty is determined by the global benefit, the actual individual income as well as the ideal income of the agent. When global benefit, the actual income and the ideal income of the agent are equal, the Ly_n of the agent will be

1. When the actual and ideal benefits of the agent are different, there will be two possibilities. One is that the agent sacrifices its own benefit to ensure the maximum global benefit, and the other is that the agent sacrifices the maximum global benefits to achieve its own benefits when performing the task. Now the loyalty is calculated by

$$Ly_n = 1 + A I e_n - I I_n, \quad n \in 1, 2, 3, \dots, n. \quad (2)$$

When Ly_n is greater than 1, it is the first case. When Ly_n is less than 1, it is the second case. The agent's loyalty is iterated over the MAS's tasks, and the loyalty mechanism is formed after many trainings. As one of the parameter criteria for participating in multi-agent alliance tasks, the calculation method is expressed as follows:

$$Ly = \sum_1^n \frac{Ly_n}{n}, \quad n \in 1, 2, 3, \dots, n. \quad (3)$$

2.2 Establishment of MAS based on improved game theory

The MAS is a system composed of multiple agents that act as intelligent entities with certain autonomy. They are ultimately coupled into the MAS which can achieve complex tasks by interacting, i.e., pursuing certain goals or achieving certain tasks [12–16]. It consists of two types of structure: centralized structure and distributed structure, covering one or more command and control subjects. The subjects only manage the subordinate member agents uniformly and participate in solving the task planning and allocation between agents, the allocation and management of shared resources, the coordination of conflicts, etc. Other members are equal, and all their actions are determined by themselves. This structure balances the advantages and disadvantages of the centralized and distributed architecture and adapts to the complex and open operating environment of the distributed MAS.

The construction of the MAS [17–20] displays the following characteristics. First, the agent in alliance may cause the loss of global benefits to maximize its own benefit. Second, there may be some loss or failure of the agent in the MAS, thus how to ensure the stability of the alliance is the key point. Third, the establishment of the MAS takes the task as the input, builds the MAS based on the task, and dismisses the alliance immediately after the completion of tasks. Furthermore, there is a hierarchy of alliance itself, i.e., there is a small alliance under the big alliance and interaction between small alliances. Furthermore, the agent may belong to more than one alliance and participate in the execution of multiple tasks in the meantime. Combined with the above characteristics, the MAS construction method is proposed in accordance with the game theory

and agent loyalty [21]. The specific construction process is shown in Fig. 1.

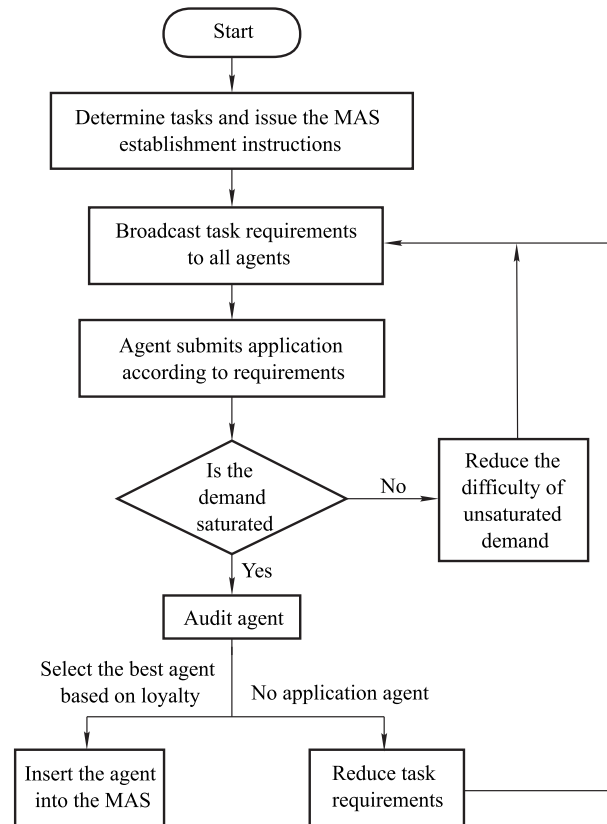


Fig. 1 Process of constructing the MAS

In the construction of the MAS, in order to avoid the situation that the task demand is too high, the MAS can be built by reducing the task demand standard or changing the task demand, the agents who are beyond the demand quantity can be sorted based on the loyalty order, and the agent with higher loyalty can be selected as the best item [22].

3. Training algorithm design based on bi-directional coordination network algorithm

Bi-directional coordination network (BiCNet) is introduced so that agents are allowed to communicate with each other before taking separate actions. On the whole, the BiCNet consists of a multi-agent actor with grouping and a critic [23]. Both are based on the bi-directional RNN. The actor receives situation sharing and locally acquired situation information, thus returning the operation of each agent to make action decisions. Accordingly, the agent can maintain its own internal state while sharing information with other cooperating agents. The bi-directional recursion mechanism is not only a means of interaction between agents, but also a local memory state. The critic takes the

state and behaviour of the actor as the input [24], returns the local Q value to each agent in each coalition and provides a specific value of the global benefit in combination with the local Q value [25].

The BiCNet [26] can be observed as expanding networks of length N (number of agents), and then the time inverse propagation is adopted to calculate the reverse gradient. The gradient is passed to individual and policy functions. They are a collection of all agents and their operations. In other words, the gradient of all agents' benefits is propagated firstly to affect each agent's actions, and then the resulting gradient is further propagated to update the parameters. Given that the policy parameters of N agents are θ and the state distribution is $\rho_{a_\theta}^\tau(s)$, $J(\theta)$ objective function is defined, and we have the following policy gradient:

$$\nabla_\theta J(\theta) = E_{s \sim \rho_{a_\theta}^\tau(s)} \left[\sum_{i=1}^N \sum_{j=1}^N \nabla_\theta a_{j,\theta}(s) \cdot \nabla a_j Q_i^{a_\theta}(s, a_\theta(s)) \right]. \quad (4)$$

To ensure the adequate mining, we increase the output noise of the actor in each step. Further consideration is given to the relevant variance. In the training of the critic, the squared loss sum is used, and the following gradient is yielded for the parameterized Q value [27]. The parameters of the critic change as follows:

$$\nabla_\xi L(\xi) = E_{s \sim \rho_{a_\theta}^\tau(s)} \left[\sum_{i=1}^N r_i(s, a_\theta(s)) + \lambda Q_i^\xi(s', a_\theta(s')) - \lambda Q_i^\xi(s, a_\theta(s)) \cdot \nabla_{\partial \xi} Q_i^\xi(s, a_\theta(s)) \right]. \quad (5)$$

Note that the gradient is also aggregated from multiple agents, as the actor does. In general, the stochastic gradient descent (SGD) algorithm is employed to optimize the actor and the critic [28]. The pseudo-code of the algorithm is presented as follows.

Algorithm 1 Pseudo-code of the algorithm

Parameter: learning rate η

Initialization: θ

Stop condition to satisfy do

Extract m data $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ and its corresponding tag $\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$ from training

Calculate the gradient:

$$g(\theta) = \frac{\partial \left(\frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}), y^{(i)}) \right)}{\partial \theta}$$

Update parameter: $\theta = \theta - \eta \times g(\theta)$

end

The theorem differs significantly from similar algorithms. Agents' dependency is incorporated in the hidden layer rather than directly embedded into the operation. In brief, the algorithm allows agents to be completely dependent on each other because the gradient of all operations can be effectively propagated across the entire network. This will help resolve any possible connection problems between multiple cooperative operations.

These parameters are shared between different agents, and the number of parameters is independent of the number of agents. Parameter sharing contributes to the compactness of the model, thus speeding up the learning process.

In the defining process of the benefit function, a time-variant global benefit is introduced based on two consecutive time steps:

$$r(s, a, b) \equiv \frac{1}{M} \sum_{j=N+1}^{N+M} R_j^t(s, a, b) - \frac{1}{N} \sum_{i=1}^N R_i^t(s, a, b). \quad (6)$$

According to the time step t , the formula gives the calculation method of the global benefits. In the global benefit $r(s, a, b)$, the formula is based on the friendly controlled agent, while the enemy's global benefits are opposite, thus the total benefits of the two camps are equal to zero.

Based on the global benefit $r(s, a, b)$, the agent jointly acts when the enemy agent takes action. The goal of the agent is to learn the strategy to achieve the maximum

global benefits, that is $E \left[\sum_{k=0}^{+\infty} \lambda^k r_t + k \right]$, where $0 \leq \lambda < 1$.

Instead, the enemy's joint strategy is to minimize the sum expected. Accordingly, we have the following minimax strategies:

$$Q^*(s, a, b) = r(s, a, b) + \lambda \max_{\theta} \min_{\phi} Q^*(s', a_\theta(s'), b_\phi(s')). \quad (7)$$

$s' = s^{t+1}$ is determined by $\tau(s, a, b)$. $Q^*(s, a, b)$ is the optimal action state function, which follows the Bergman optimization equation. The deterministic strategy $a_\theta: S \rightarrow A^N$ refers to our agent and the deterministic strategy $b_\phi: S \rightarrow B^M$ refers to the enemy. At this point, the strategy of the enemy is considered being fixed, and a specialized match model is trained. Then, the equation becomes the Markov decision processes (MDP) problem:

$$Q^*(s, a) = r(s, a) + \lambda \max_{\theta} Q^*(s', a_\theta(s')) \quad (8)$$

$$r_i^{(s,a,b)} \equiv R_i^t(s, a, b) \frac{1}{|top - K - u(i)|} \sum_{j \in top - K - u(i)} \Delta R_j^t(s, a, b) - \frac{1}{|top - K - u(i)|} \sum_{j \in top - K - u(i)} \Delta R^t(s, a, b). \quad (9)$$

Each agent person i maintains $top - K - u(i)$ and $top - K - e(i)$, and other interacting agents and enemies currently are in the top- K lists. It is replaced with (1). Agent i has N Bellman equations, where $i \in \{1, \dots, N\}$, and the same parameter θ for the strategy function:

$$Q_i^*(s, a) = r_i(s, a) + \lambda \max_{\theta} Q_i^*(s', a_{\theta}(s')). \quad (10)$$

To see this mathematically, the target single agent represented by $J_i(\theta)$ is to maximize the expected cumulative individual income r_i to $J_i(\theta) = E_{s \sim \rho_{s_d}^{\tau}} [r_i(s, a_{\theta}(s))]$ where $\rho_{a_{\theta}}^{\tau}(s)$ denotes the discount status distribution corresponding to the strategy a_{θ} under transition τ , i.e. $\rho_{a_{\theta}}^{\tau}(s) := \int_s \sum_{t=1}^{\infty} \lambda^{t-1} p_1(s)(s' = \tau_{a_{\theta}, b_{\phi}}^1(s)) ds$. It can also be taken as the fixed distribution of the MDP. Thus, the target function of the N agents represented by $J(\theta)$ can be written as follows:

$$J(\theta) = E_{s \sim \rho_{a_{\theta}}^{\tau}} \left[\sum_{i=1}^N r_i(s, a_{\theta}(s)) \right]. \quad (11)$$

Algorithm 2 Specific network design process

Use ξ and θ to initialize the actor network and evaluation network

Use ξ to initialize the target network and assess the network $\xi \rightarrow \xi'$ and $\theta \rightarrow \theta'$

Initialize the replay buffer S

$episodes = 1$

Initialize the random process U for motion exploration

Receive the initial observation state S_0

$t = 1$

For every agent, select and conduct action $a_i^t = a_{i, \theta}(s^t) + N_t$

Receive rewards $[r_i^t]_{i=1}^N$, observe the new state s^{t+1}

Store and converse $\{s^t, [a_i^t, r_i^t]_{i=1}^N, s^{t+1}\}$ in \mathbf{R}

The random batch of M transformation $\{s_m^t, [a_{m,i}^t, r_{m,i}^t]_{i=1}^N, s_m^{t+1}\}_{m=1}^M$ is sampled from \mathbf{R}

Use BiCNet to calculate the target value of each agent in each conversion:

$m = 1$

$\hat{Q}_{m,i} = r_{m,i} + \lambda Q_{m,i}^{\xi'}(s_m^{t+1}, a_{\theta'}(s_m^{t+1}))$

Calculate the critical gradient estimate by (4):

$$\Delta \xi = \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^N [\hat{Q}_{m,i} - Q_{m,i}^{\xi}(s_m, a_{\theta}(s_m))] \cdot \nabla_{\xi} Q_{m,i}^{\xi}(s_m, a_{\theta}(s_m)).$$

Calculate the actor gradient estimate by (5) and replace the Q value with a critical estimate:

$$\Delta \theta = \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^N \sum_{j=1}^N [\nabla_{a_{j, \theta}}(s_m)].$$

$$\sum_{j=1}^N \nabla_{\theta} a_j Q_{m,i}^{\xi}(s_m, a_{\theta}(s_m)).$$

The update network is based on Adam that uses the gradient estimator above to update the target network:

$$\gamma \xi + (1 - \gamma) \xi' \rightarrow \xi', \quad \gamma \theta + (1 - \theta) \theta' \rightarrow \theta'$$

4. Algorithm simulation results and analysis

4.1 Implementation comparison of SGD algorithm

Compared with other error processing algorithms, the SGD exhibits higher superiority [29], and it is difficult to yield the local optimal solution and avoid the algorithm getting into local loop. Fig. 2 below is a simulation implementation of the SGD algorithm [30] based on this model.

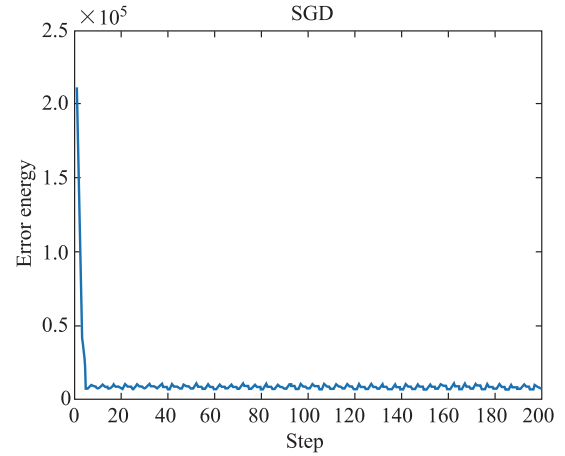


Fig. 2 SGD algorithm convergence analysis

The simulation diagram reveals that the random noise is added to the actor, resulting in a decrease in the error [31]. In the meantime, the SGD algorithm can efficiently deal with errors, and it does not fall into the local optimal solution [32].

4.2 Implementation of BiCNet correlation algorithm

4.2.1 RNNs

The parameters of the basic RNNs are set, and the simulation results are analyzed. Some simulation results are shown in Figs. 3–8.

The experimental data in Figs. 3–8 suggest that the basic RNNs algorithm is not sufficiently stable, and it is easy to fall into the local infinite loop, generating the local optimal solution; and the generation of the global optimal solution is highly accidental. Given this, the algorithm requires optimization, and the function is adjusted according to the application field [33].

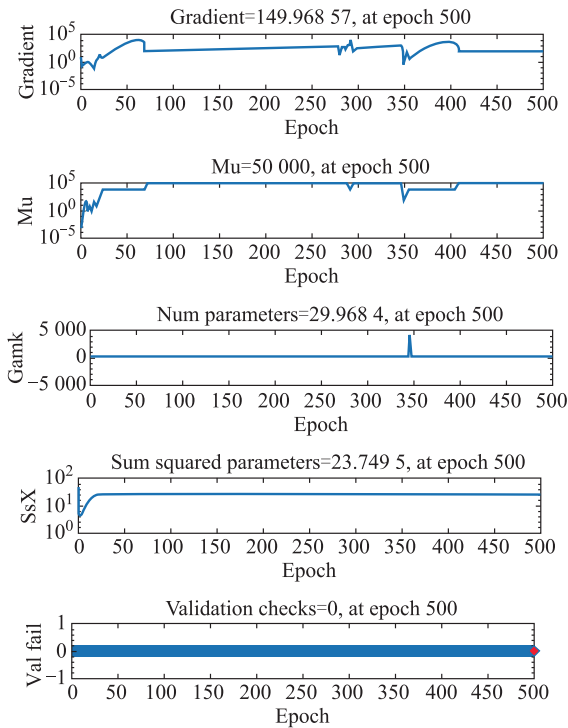


Fig. 3 Result of iteration 345 times

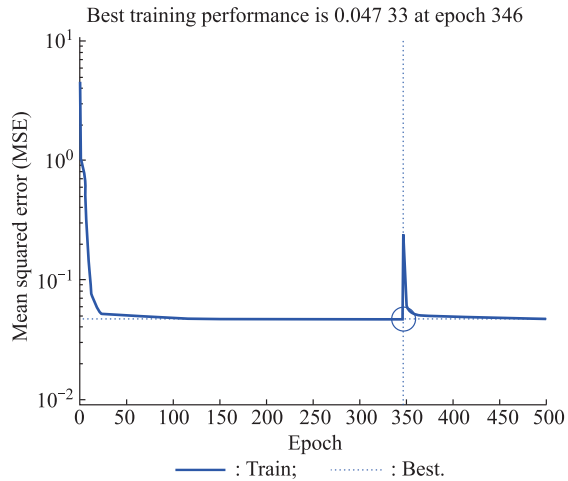


Fig. 4 Generating optimal solution by 345 iterations

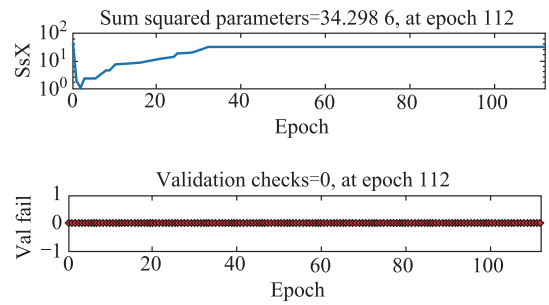
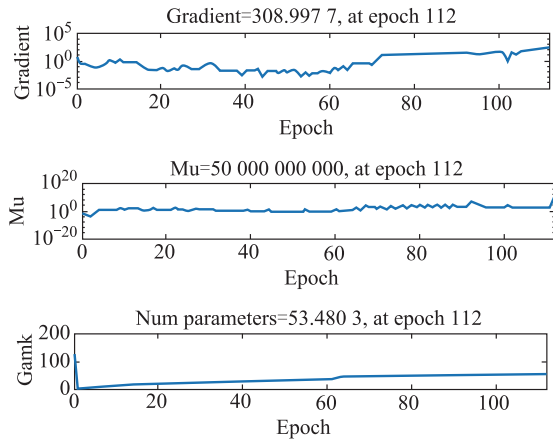


Fig. 5 Result of iteration 112 times

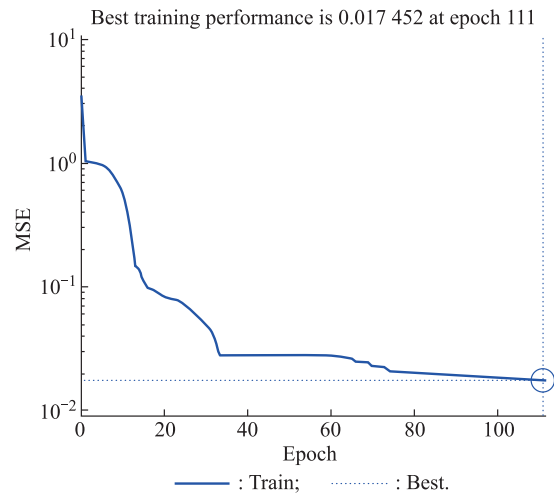


Fig. 6 Generating optimal solution by 112 iterations

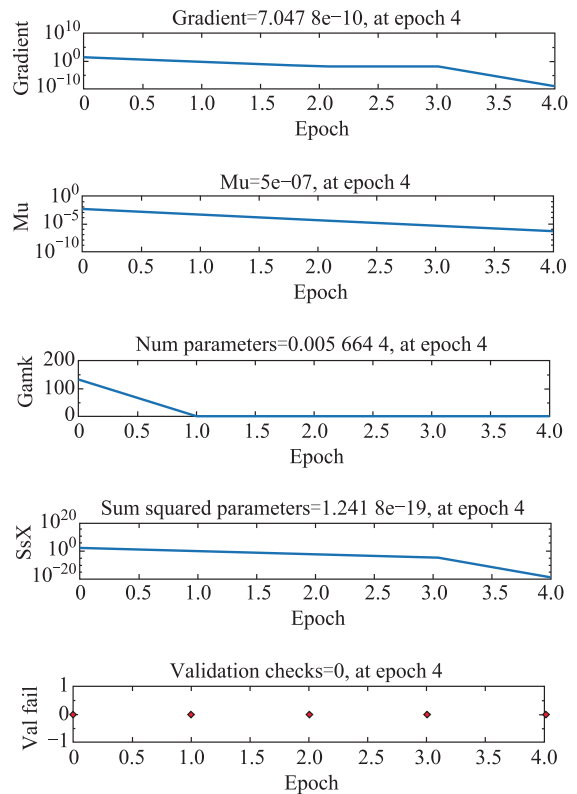


Fig. 7 Result of iteration 5 times

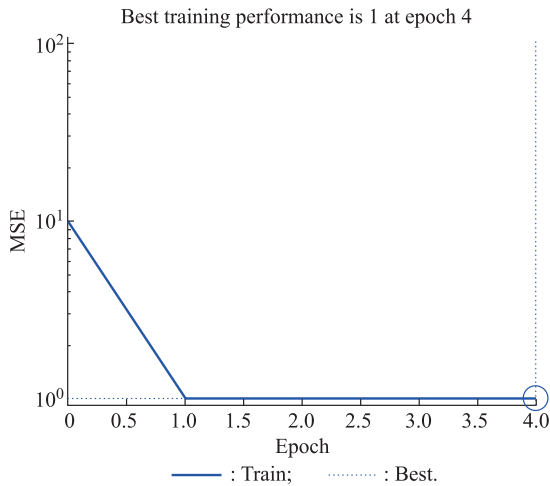


Fig. 8 Generating optimal solution by 5 iterations

4.2.2 Implementation of BiCNet algorithm simulation

There are too many iterations for the basic RNNs, and the local optimal solution occurs [34]. Besides, the algorithm is not stable that there is not only a fast-optimal solution, but also a certain number of iterations which cannot get the optimal solution, trapped in an infinite loop. Given the findings above, this paper presents a BiCNet structure based on the bi-directional RNNs. The error analysis of the structure shows that the BiCNet is stable, and the global optimal solution is yielded quickly. The result data is listed in Table 1.

Table 1 Result data of the error analysis

Sample check	True value	Predicted value	Error
Demo_1 000	201	162	0.702 594
Demo_2 000	105	79	0.679 277
Demo_3 000	105	105	0.581 262
Demo_4 000	112	112	0.393 983
Demo_5 000	153	153	0.186 213
Demo_6 000	140	140	0.168 543
Demo_7 000	109	109	0.098 862
Demo_8 000	115	115	0.081 481
Demo_9 000	167	167	0.091 533
Demo_10 000	60	60	0.060 279

In Table 1, the above results are experimentally sampled data from the 1 000th iteration to the 10 000th iteration. A test is performed every 1 000 iterations. The experiment uses a binary array for prediction. The data range is from 0 to 256. The number of hidden layers is 16.

After defining the parameter matrix, the data of BiCNet are predicted, and the error is graphically represented, as shown in Fig. 9.

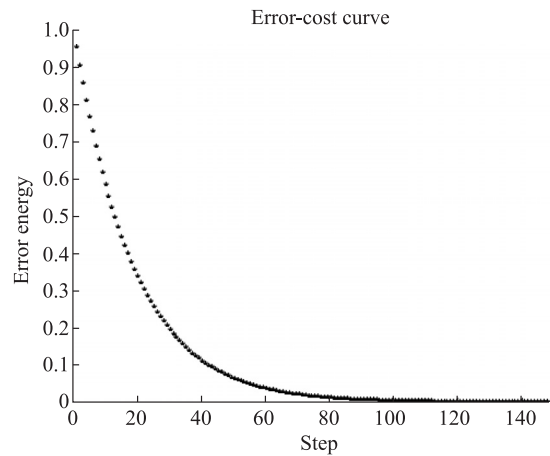


Fig. 9 Convergence analysis of BiCNet

The experimental data reveal that with the up-regulation of the number of iterations, the error decrease trend is obvious, the real value and the predicted value are inclined to be consistent, and the algorithm exhibits good convergence.

5. Model training and strategy learning result analysis

5.1 Collision-free obstacle avoidance strategy

It is observed that the scenario shown in Fig. 10 requires three fighters to destroy the radar vehicle through a green mountain obstacle. At the initial stage of learning, as shown in Fig. 11 and Fig. 12, the agent operations lack sufficient coordination. The choice of the route also complies with the choice of target points, in which the shortest line is taken as the route. The way to avoid obstacles follows the target point for fighter aircraft to avoid obstacles. When two agents approach each other, one agent will often inadvertently block the other's path.

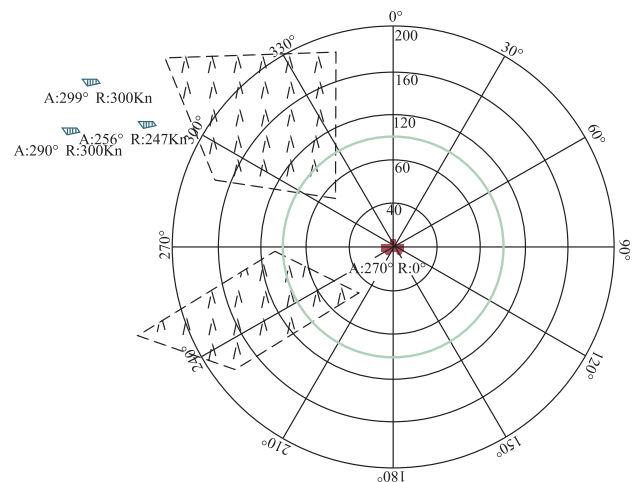


Fig. 10 Scene setting

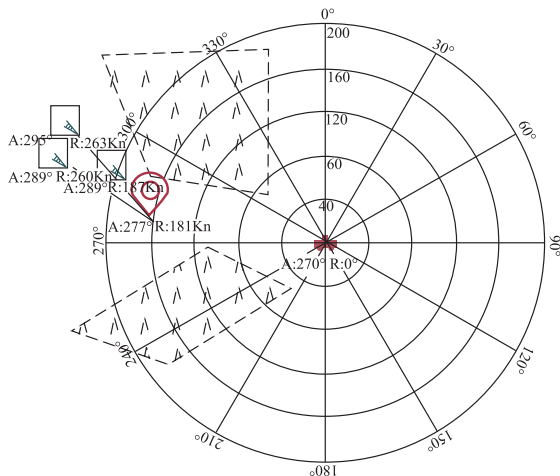


Fig. 11 Formation coordination

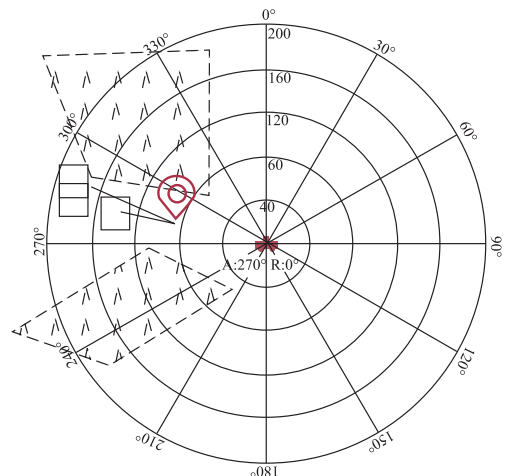


Fig. 14 Formation through the obstacle zone

This coordinated action became critical in large-scale combat. With the final effect shown in Fig. 15, as the three fighters cross the barriers in a coordinated manner and successfully destroy the target radar vehicle.

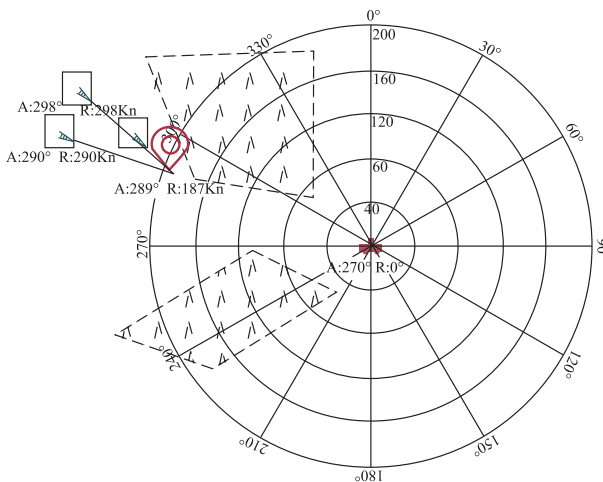


Fig. 12 Initial stage of learning

With the up-regulation of training wheels, the number of collisions decreases significantly. Finally, when the training becomes stable, the coordinated action occurs, under the same background as shown in Fig. 13 and Fig. 14.

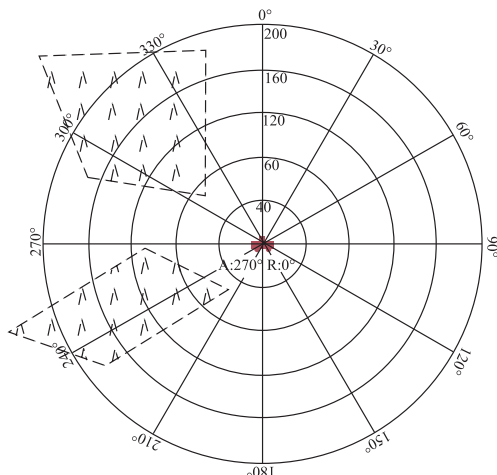


Fig. 13 Obstacle avoidance of formation

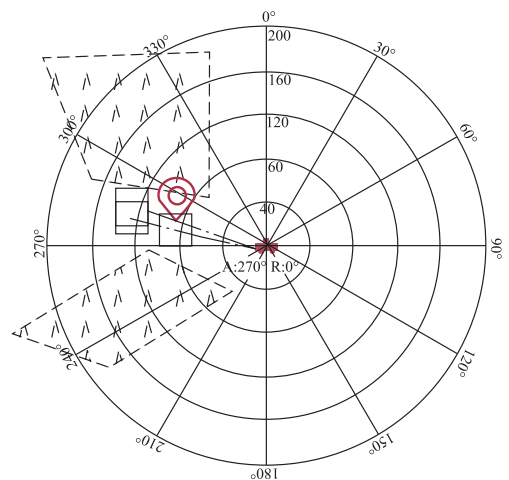


Fig. 15 Passing the obstacle zone and locking the attack target

In Table 2, the learning strategy of obstacle avoidance with collision-free [35] during the training process is beneficial to the learning of cooperative strategies in the later stage of the model. The feasibility of the strategy is verified with the flight length as the reference quantity.

Table 2 Analysis of raining results

Training times	Length of travel route /km	Number of collisions	Variation of fire coverage/%
50	460	7	28
200	490	6	28
500	350	4	27
1 000	330	3	23
2 000	280	1	15
3 000	260	0	15

5.2 Collaborative screen optimization strategy

Cover attack refers to a high-level cooperative strategy that is frequently used in the real battlefield. The cover attack means that an agent draws the enemy’s attention or attacks the enemy to attract firepower, while other agents also act to output more damage for enemy. In fact, the difficulty in conducting a cover attack is how to arrange tactics of the multi-agent in a coordinated hit and run manner [36]. As shown in Fig. 16, after 3 000 iterations, the agents learn the collaborative cover strategy. Taking the red radar and the launch vehicle as an example, after attracting firepower, a fighter aircraft is sneaked in from 90° to generate firepower. The red circle is the fire output range, and the green circle is the radar detection range. Furthermore, the launch vehicle fire is attracted by four fighters. There is a fighter in the direction of (A:75; R:116) to make a sudden attack. Through the above tactical actions, therefore, it concludes that the MAS has preliminarily mastered the cooperative shield strategy.

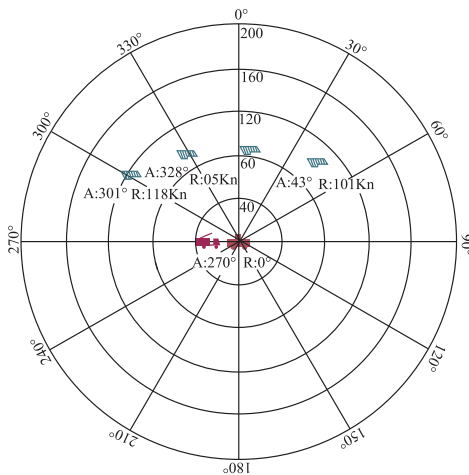


Fig. 16 Attack mode without learning strategy

As shown in Fig. 17, the initial MAS model cannot implement the strategy of covering coordination and can only fire at enemy targets through fire allocation. After 3 000 iterations, the MAS learns how to attack and fire, as shown in Fig. 17.

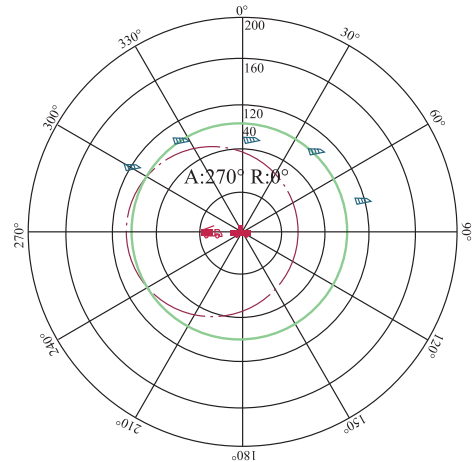


Fig. 17 Covering cooperative attack after learning strategy

The model training of the MAS is performed under the same operational background, and the parameters in the combat are analyzed based on the above strategies. With the up-regulation of the number of iterations, the optimal selection of the feint fighter sortie, surprise direction or cooperative direction is analyzed. The results are listed in Table 3.

In Table 3, the data primarily analyze the accomplishment degree of tactics from the tactical target (core), the reasonableness of the selection of the sudden attack direction, the ammunition consumption and the number of attack units required for the tactical target.

Table 3 Optimal selection of the feint fighters’ attack direction or cooperative direction

Training times	Target selection	Attack unit	Consumption of ammunition	Number of feint units	Attack direction selection	Is tactical goal accomplished	Degree of strategy completion/%
200	Radar vehicle: 1 Launch vehicle: 1	Fighter: 8	Medium-range missile: 10 Short-range missile: 0	5	No	No	0
500	Radar vehicle: 1 Launch vehicle: 1	Fighter: 8	Medium-range missile: 8 Short-range missile: 0	5	No	No	0
1 000	Radar vehicle: 1 Launch vehicle: 1	Fighter: 8	Medium-range missile: 7 Short-range missile: 0	5	No	Yes	20
2 000	Radar vehicle: 1 Launch vehicle: 1	Fighter: 7	Medium-range missile: 6 Short-range missile: 0	6	150°	Yes	60
3 000	Radar vehicle: 1 Launch vehicle: 1	Fighter: 5	Medium-range missile: 5 Short-range missile: 0	4	90°	Yes	95
5 000	Radar vehicle: 1 Launch vehicle: 1	Fighter: 5	Medium-range missile: 4 Short-range missile: 1	3	90°; 150°	Yes	98

With the rise in the number of iterations, the situation that the multi-fighter still cannot achieve the tactical target is optimized, and the MAS model gradually learns the strategy of the optimal allocation of the firepower, and then tries to cooperate with the fire optimization to a certain extent, the direction of the raid is gradually reasonable, which is gathered to the red coverage blind area.

6. Conclusions

This paper introduces a brand-new construction approach of the MAS in accordance with the game theory under the bi-directional coordination network and uses a thorough reinforcement learning algorithm of the multi-agent. The tactical action is learned by constructing a vector actor-critic framework, in which each dimension corresponds to an agent. Moreover, the tactical cooperation is achieved by the bi-directional periodic communication at the internal level. Through end-to-end learning, the BiCNet can successfully learn some effective coordination strategies. Besides, the rationality of the algorithm and the superiority and stability of the MAS are proved by the simulation experiment and engineering implementation. The research direction of the later stage is primarily to optimize the formation and path planning in the course of the battle based on obstacle avoidance as the model is employed on both sides of the battle confrontational strategies, and learn more complex strategies.

References

- [1] CASE D O. The society of mind. *Information Processing & Management*, 1988, 24(4): 499–500.
- [2] HU J P, HONG Y G. Leader-following coordination of multi-agent systems with coupling time delays. *Physica A: Statistical Mechanics and its Applications*, 2007, 374(2): 853–863.
- [3] GUPTA J K, EGOROV M, KOCHENDERFER M. Cooperative multi-agent control using deep reinforcement learning. *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, 2017: 66–83.
- [4] BÚRDALO L, TERRASA A, JULIÁN V, et al. The information flow problem in multi-agent systems. *Engineering Applications of Artificial Intelligence*, 2018, 70: 130–141.
- [5] KAYA M, ALHAJJ R. Modular fuzzy-reinforcement learning approach with internal model capabilities for multi-agent systems. *IEEE Trans. on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society*, 2004, 34(2): 1210–1223.
- [6] ALLOUCHE M K, BOUKHTOUTA A. Multi-agent coordination by temporal plan fusion: application to combat search and rescue. *Information Fusion*, 2010, 3(11): 220–232.
- [7] SUN X, LIU Y, LI J, et al. Feature evaluation and selection with cooperative game theory. *Pattern Recognition*, 2012, 8(45): 2992–3002.
- [8] RAHWAN T, MICHALAK T, WOOLDRIDGE M, et al. Any-time coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, 2012, 186: 95–122.
- [9] XU X H, WANG Y, LIU J, et al. Analysis on the achievement milestones and limitations of Game Theory. *Proc. of the Control & Decision Conference*, 2008: 1214–1219.
- [10] FARRELL J, RABIN M. Cheap talk. *Journal of Economic Perspectives*, 1996, 10(3): 103–118.
- [11] KAGEL J H, ROTH A E. *The handbook of experimental economics*. Princeton: Princeton University Press, 2016.
- [12] OISHI S, FUKUTA N. A cooperative task execution mechanism for personal assistant agents using ability ontology. *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2016: 664–667.
- [13] GITINAVARD H, MOUSAVI S M, VAHDANI B. A new multi-criteria weighting and ranking model for group decision-making analysis based on interval-valued hesitant fuzzy sets to selection problems. *Neural Computing & Applications*, 2016, 27(6): 1593–1605.
- [14] TOHMÉ F, SANDHOLM T. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic & Computation*, 2018, 9(6): 793–815.
- [15] CĂTĂLIN D, ENEA C, GUELEV D. Model-checking an alternating-time temporal logic with knowledge, imperfect information, perfect recall and communicating coalitions. *Electronic Proceedings in Theoretical Computer Science*, 2018, 25: 103–117.
- [16] LU Z H, ZHANG L, WANG L. Controllability analysis of multi-agent systems with switching topology over finite fields. *Science China (Information Sciences)*, 2019, 62(1): 1–15.
- [17] WEN G, CHEN C L P, LIU Y J, et al. Neural network-based adaptive leader-following consensus control for a class of non-linear multiagent state-delay systems. *IEEE Trans. on Cybernetics*, 2016, 47(8): 2151–2160.
- [18] KUNITO G, AIZAWA K, HATORI M. Tracking agent for communication between multiple cooperative agents. *Electronics & Communications in Japan*, 2001, 84(5): 11–20.
- [19] DOU C, YUE D, HAN Q L, et al. A multi-agent system based event-triggered hybrid control scheme for energy internet. *IEEE Access*, 2017, 99(5): 3263–3272.
- [20] LEI Z, WEI G, YAN D W, et al. Study of reconfiguration for the distribution network with distributed generations based on multi-agent alliance algorithm. *Power System Protection & Control*, 2012, 40(10): 95–101.
- [21] CICIRELLI F, GIORDANO A, NIGRO L. Efficient environment management for distributed simulation of large-scale situated multi-agent systems. *Concurrency and Computation: Practice and Experience*, 2015, 27(3): 610–632.
- [22] YU H, SHEN Z, LEUNG C, et al. A survey of multi-agent trust management systems. *IEEE Access*, 2013, 1: 35–50.
- [23] CAO Y Q, ZHANG Z, HUANG X S, et al. Multi-agent system coalition utility allocation strategy based on loyalty. *Computer Science*, 2014, 41(5): 235–238.
- [24] LOWE R. Multi-agent actor-critic for mixed cooperative-competitive environments. <https://arxiv.org/abs/1706.02275>.
- [25] POTJANS W, MORRISON A, DIESMANN M. A spiking neural network model of an actor-critic learning agent. *Neural Computation*, 2009, 21(2): 301–339.
- [26] PENG P, WEN Y, YANG Y, et al. Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games. <https://arxiv.org/abs/1703.10069v4>.
- [27] WEN G, CHEN C L P, FENG J, et al., Optimized multi-agent formation control based on identifier-actor-critic rein-

- forcement learning algorithm. *IEEE Trans. on Fuzzy Systems*, 2018, 26(5): 2719–2131.
- [28] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. <https://arxiv.org/abs/1706.02275>.
- [29] YUAN J, LAMPERSKI A. Online control basis selection by a regularized actor critic algorithm. *Proc. of the IEEE American Control Conference*, 2017: 4448–4453.
- [30] CHAUDHARI P, SOATTO S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *Proc. of the International Conference on Learning Representations*, 2018: 1–10.
- [31] DING J, ZHU L P, HU B, et al. Stochastic gradient descent based k-means algorithm on large scale data clustering. *Applied Mechanics & Materials*, 2014, 687/691: 1342–1345.
- [32] SONG S, CHAUDHURI K, SARWATE A D. Learning from data with heterogeneous noise using SGD. *Proc. of the JMLR Workshop Conference*, 2015: 894–902.
- [33] KUZBORSKI I, LAMPERT C H. Data-dependent stability of stochastic gradient descent. <https://arxiv.org/abs/1703.01678v4>.
- [34] ARJOVSKY M, SHAH A, BENGIO Y. Unitary evolution recurrent neural networks. *Proc. of the International Conference on Machine Learning*, 2015: 1120–1128.
- [35] WANG C, ZHAO X Z, WANG Y T. Research on decision making method of formation cooperative air defense based on multi-agent cooperation. *Communications and Information Processing*, 2012, 12(5): 529–538.
- [36] YAN J, GUAN X P, TAN F X. Target tracking and obstacle avoidance for multi-agent systems. *International Journal of Automation and Computing*, 2010, 7(4): 550–556.

Biographies



ZHANG Jie was born in 1995. He is a master degree candidate at the Air Force Engineering University. His research interests are combat multi-agent based on deep learning and tactical air defense and antimissile command and control system.
E-mail: afeu.zhangjie@163.com



WANG Gang was born in 1975. He received his Ph.D. degree from the Air Force Engineering University. His research interests are machine learning, information fusion and command and control system.
E-mail: sharesunny123@163.com



YUE Shaohua was born in 1968. She received her Ph.D. degree from the Air Force Engineering University. Her research interests are command information system and intelligent command and control.
E-mail: zhouguoan@sina.cn



SONG Yafei was born in 1988. He received his Ph.D. degree from the Air Force Engineering University. His research interests are pattern recognition and intelligent information processing.
E-mail: yafei_song@163.com



LIU Jiayi was born in 1996. He is a master degree candidate at the Air Force Engineering University. His research interests are air defense and anti-missile command and control system and intelligent decision-making based on reinforcement learning.
E-mail: sixandone1@163.com



YAO Xiaoqiang was born in 1985. He received his Ph.D. degree from the Air Force Engineering University. His research interests are intelligent information processing and simulation training and simulation.
E-mail: icemissile@sina.com