Heterogeneous Agent Cooperative Planning Based on Q-Learning

Chenfeng Gan, Wei Liu, Ning Wang, and Xingyu Ye

***Abstract*—In this paper, we present a model to achieve the collaboration of heterogeneous agent in the open-dynamic environment. This model simulates a disaster rescue scenario, defines the environment, action space, reward function and action selection strategy with Q-learning algorithm. Heterogeneous rescue agent is used to assist agent in the scene. Experiments based on the python environment prove that the heterogeneous agent collaboration method can effectively complete the collaboration in unknown environment, and it has better performance than the homogeneous method.**

***Index Terms*—Multiagent-system, collaboration, heterogeneous, reinforcement learning.**

I. Introduction

In multi-agent systems (MAS) [1], in order to achieve a global goal, agents cooperate, share knowledge and resources with others. agents during cooperation can be divided into two kinds, homogeneous agents and heterogeneous agents. Homogeneous agents have the same behavioral capacity and organizational structure, such as mutual cooperation in robot football match [2]. Heterogeneous agents differ in structure and function. For example, the interaction between distributed observer and controller in autonomous formation control [3]. The structure of the agent depends on complexity of the problem in the whole agent cooperation field, such as the problem of assigning software processes (or tasks) to hardware processors in a distributed robot environment [4], formalizing the problem of assigning task. This model combines typical constraints in robot applications.

Traditional agent cooperative problem has produced some classical methods, such as blackboard model, contract network model, relation web model and so on. In the blackboard model [5] and the commitment model, through a Shared problem-solving-space, experts or intelligent agents can access the information on the blackboard, and experts learn from experience to solve problems, the main defects of this method are high consumption of time and space and low efficiency. The contract network [6] model uses the communication mechanism to negotiate the tasks assigning process, it can accomplish the assignment of tasks through the bidding process between nodes, it's main defects are frequent communication, waste of system resources and small application scope. Traditional methods normally have the following shortcomings:

a) Under the circumstance of model-based cooperation, the new agents will change the system and cannot guarantee the normal progress of cooperation.

b) The change of environment may lead to the failure of agent cooperation.

c) Difficult to dynamically allocate the agent's resource occupation.

Then the optimized heterogeneous agent cooperative commitment model generates commitment protocols dynamically according to the current environment and selects the optimal protocol according to the maximum utility value [7]. However, the generation of agent commitment mostly stays in the manual preset and creation stage [8], [9], which still has the following defects:

a) In the open system, the applicability of the protocol on the new agent cannot be guaranteed.

b) The change of environment may invalidate the protocol.

c) The agents cannot cope with the resource redistribution when the priority changes.

At present, the main problem of heterogeneous agent research focus on the dynamic cooperation under unknown environment [10]. Reinforcement learning (RL) method has the characteristics of dynamic learning in an unknown environment [11]. The algorithms allow the agent to simulate the collaboration process under a certain communication mechanism, and determines the optimal scheme by using the accumulation of reward values. Research on agent collaboration based on reinforcement Learning includes the multi-crawler system of the Internet [12], research on Q-learning algorithm's multi-agent planning [13] and multi-agent collaboration research on the optimized evolutionary algorithm [14], etc.

In order to explore the cooperation of heterogeneous agents, this paper proposes a heterogeneous agent cooperation model based on Q-learning reinforcement learning method, so as to solve the problem that the traditional heterogeneous agent cooperation is not effective due to the unknown environment and changes in the environment. In this paper, firstly we put forward the agent collaboration requirements based on real scenes, highlight the heterogeneity of agents, and establish the planning model of heterogeneous agent reinforcement learning. Then, we take the heterogeneous multi-agent collaborative target-searching as an empirical case, the experiment proves that, compared with traditional multi-agent reinforcement learning, the collaborative planning of heterogeneous agents under reinforcement learning is suitable for unknown and variable environments. Finally, the influence of agent scale on the robustness of the model was explored through multiple experiments by increasing the number of agents.

Manuscript received September 9, 2020; revised November 22, 2020.

Chenfeng Gan, Wei Liu, Ning Wang, and Xingyu Ye are with Wuhan Institute of Technology, Wuhan, China (e-mail: 728384289@qq.com, liuwei@wit.edu.cn, 1757674599@qq.com, 849413957@qq.com).

Reinforcement learning, also known as reinforcement learning, is a machine learning method that takes environmental feedback as input and adapts to the environment, differs from supervision and semi-supervised learning [15], reinforcement learning doesn't set any results or output standard, it focuses on the performance. Generally speaking, reinforcement learning has four elements: environment, reward, action and state. Based on this, an reinforcement learning model is established to obtain an optimal policy for a specific problem, so as to maximize the reward obtained under this strategy. In fact, policy is a combination of a series of actions. Common reinforcement learning methods include Dijkstra algorithm [16], A* algorithm [17], genetic algorithm [18], particle swarm optimization, ant colony optimization, and so on.

Q-learning reinforcement learning method is a typical unsupervised machine learning method of reinforcement learning, It has the advantage of eliminating the need for data sets and labeling processes. At present, there are many researches on planning using Q-learning. For example, q-learning and image data set are combined to automatically identify security items [19], and q-learning is used to identify various kinds of interference in the field of radar anti-jamming to improve radar threat identification ability, etc. However, the current study is generally for single agent or homogeneous agents, such as the use of single agent reinforcement learning for business division routing [20], single robot visual path planning [21] in deep scene, as well as ship avoidance of multiple ships in the same state [22], etc. In this paper, Q-learning is combined with heterogeneous agents to achieve cooperation.

II. Heterogeneous Agent Cooperative Planning with Reinforcement Learning

Collaboration of homogeneous agents has limited ability to solve complex tasks, so cooperation of heterogeneous agents is worth studying. Establish a model based on reinforcement learning to achieve the cooperation of heterogeneous agents, and explore the model's ability to deal with problems in an unknown and dynamic environment.

A. The Establishment of Collaboration Model

In the post-disaster rescue scenario, the heterogeneous agent collaborative rescue task is proposed. Heterogeneous agent-based cooperative decision-model based on Q-learning (HADQ) was established (Fig. 1).

In the Python environment, import the matrix and function module NumPy, the graphical development interface tkinter module. Build an intensive learning planning model for post-disaster rescue, draw a scene and add terrain, generate multiple targets in the scene. The location information of these targets is random, corresponding to the unknown environment. Set up rescue heterogeneous agent and search from the starting point, the action process is processed by Q-learning, Rescue agent interacts with the environment, choosing actions, and makes rewards or punishments according to the quality of the actions, the purpose of the learning process is to maximize positive feedback. Reinforcement learning process can be seen as discrete state Markov process planning (Markov decision process, MDP) [22], MDP is based on Markov stochastic processes and can also be represented by a tuple (S, A, P, R, γ).

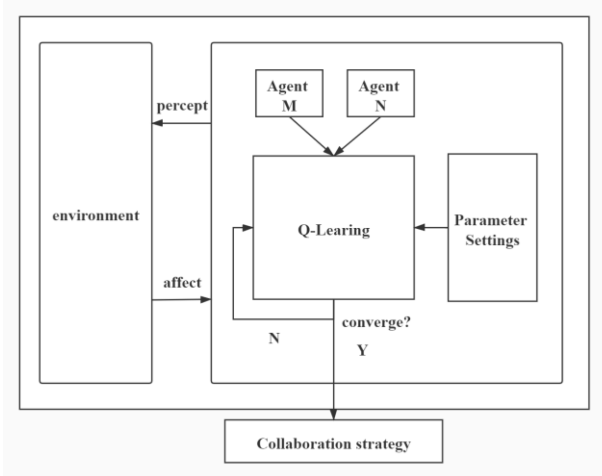

Fig. 1. Heterogeneous agent collaboration programming model.

B. The Model Definition

In the simulation, *S* is the limited statue space after the disaster, that is, where agents can be in the search and rescue area; *A* is the action set of the rescue agent, such as stop or move around; *P* stands for probability, generally, $P_a(s, s') = P(s'/s, a)$), which represents the conditional probability of the rescue agent arriving at State *S′* by action A under state S; *R* represents the reward function, which means the incentive (positive or negative) obtained by the agent during the state transition; $\gamma \in (0, 1)$ is the damping factor of excitation, and the excitation at the next moment is attenuated according to it.

The model follows four elements of reinforcement learning [23]:

1) Environment model

The assisting scene of heterogeneous agents includes agents and targets, accessible areas and impassable obstacles, starting and ending points, the setting of the geographic information does not need specific rules, and the scene is unknown to the agent, whose collaborative planning is completed by reinforcement learning.

Definition of Heterogeneous agents: *H* and *M* are heterogeneous rescue agents, carrying out the removal task for the targets in the scene. Their heterogeneous nature is manifested in the different removal efficiency for different targets. We set *h* and *m* as two kinds of targets. The rescue agents can distinguish the target by the parameter setting in the algorithm.

2) Action space definition

The action space of an agent is the set *A* of all actions it can take in all states. In this model, the agent moves in the discrete space. The action space consists of four movements: up, down, left and right.

$$A = \{up, down, left, right\} \tag{1}$$

3) Reward function

The reward function defines the quality of the agent's search action, increases the probability that the agent will take rewarded actions, and conversely reduces the probability that the agent will take punished actions.

Reward function is a very important link in reinforcement learning, which ensures that the planning converges to the optimal direction. The design of reward function should consider the following two aspects:

a) Sufficient penalty should be given to the agent when it meets boundary and blocking, so that agents can learn to avoid impassable directions in the unknown environment. When blocking happens, the agents get punished, defined as follows:

$$R_b = -50;$$

b) For the heterogeneous rescue agents, we set the following rules:

There are heterogeneous agents *H* and *M*, at the meantime, target *h*, *m* and *x*. The heterogeneity of the rescue agents is that *H* can completely rescue *h*, while can only provide partial help to *m*, and vice versa. agent *H* gets a reward of 10 when it finds *h* in the scene and removes *h*, when *H* gets to *m*, a reward of 5 is also obtained but m still exists. Otherwise, the same is true for *M*. The target *x* can be removed only when two agents find it. Through the continuous learning, we expect to get the optimal rescue scheme.

At the same time, we need to consider the collision between the action agents, and penalize the collision of agent during the learning process, so as to ensure the simulation reasonable.

The expression of reward function *R*:

$$R(H)=\begin{cases} 10 & When\ H\ found\ target\ h,\ clear\ h\ from\ the\ scene \\ 5 & When\ H\ found\ target\ m \\ 100 & When\ H\ gets\ to\ the\ end \\ \text{-}10 & When\ H\ meets\ M \\ \text{-}1 & other \end{cases} \tag{2}$$

$$R(H)=\begin{cases} 10 & When\ M\ found\ target\ m,\ clear\ m\ from\ the\ scene \\ 5 & When\ M\ found\ target\ h \\ 100 & When\ M\ gets\ to\ the\ end \\ \text{-}10 & When\ M\ meets\ H \\ \text{-}1 & other \end{cases} \tag{3}$$

In order to reduce the impact of randomness in the early search, we set a negative reward for his actions before it encounters other objects, this prevents the agent from staying still in the early searching.

4) Value function

In order to learn the optimal strategy, we use the value function. There are two types of value function in reinforcement learning:

State value function: When agent adopts strategy π, the cumulative return *R* follows a distribution, and the expected *E* of the cumulative return at state *s* is the state value function:

$$V_\pi(s) = E_\pi[R_t|\ s_t = s] \tag{4}$$

Status-behavior value function: it represents the expectation of cumulative return when agent adopts strategy π and takes action *a* in state *s*.

$$q_\pi(s,a) = E_\pi[R_t|\ s_t = s, a_t = a] \tag{5}$$

The difference between them is that the state value function represents the expectation obtained by using different actions in the current state. The state-behavior function represents the expectation of reaching different states after performing actions in the current state.

C. Generation of a Collaborative Plan

The Q-learning process of heterogeneous agents is as Fig. 2. We use Bellman equation for recursion:

$$Q_\pi(s_t,a_t) = E\left[r + \gamma E\left[Q_\pi(s_{t+1},a_{t+1})\right]\right] \tag{6}$$

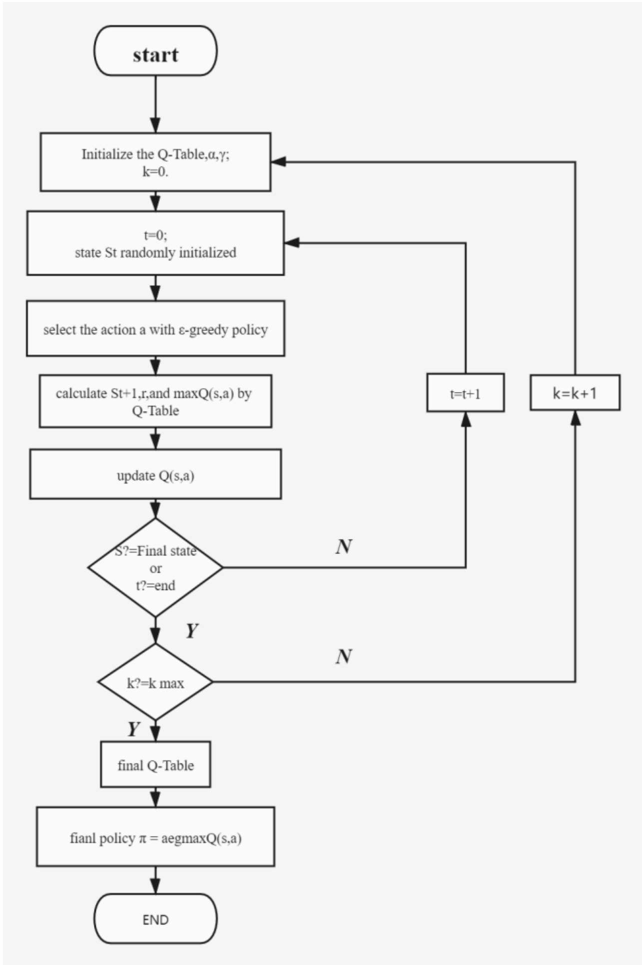

Fig. 2. Q-learning algorithm flow.

The formula is based on the state Markov property, which means that the next state is only relevant to the current state. In the agent iteration process, the value of *Q* table will be constantly updated:

$$Q(s,a) = r + \gamma(max(Q(s',a'))) \tag{7}$$

In the formula, *r* stands for the reward after taking action *a* under the current state *S*. The bigger the value is, the more likely it is to refer to the previous learning experience. *Max (Q (s', a '))* is the previous maximum *Q* value by far;

Experiment parameter setting: α, learning rate: the higher the learning rate, the less the training effect retained in the past, set as 0.01 in the experiment; Discount factor γ, the bigger the discount factor, the more inclined it is to the previous experience, in the experiment, it was set as 0.9. ε Greed index, the bigger the value ε, the better the training in the early stage, the smaller the accuracy in the later stage, set to 0.1 in the experiment.

We adopt greedy strategy to randomly search. Before each action is taken, a random number is compared with ε. If the number exceeds, the action with the highest reward is selected from the Q table. Otherwise, take a random action. It makes sure that the search is global:

$$\pi(\mathrm{a} \mid \mathrm{s})\begin{cases} \arg\max\limits_{\mathrm{a} \in A} Q(a) & random > \varepsilon \\ & random \leqslant \varepsilon \end{cases} \quad (8)$$

In the scene, the rescue heterogeneous agents cooperate with each other. After iteration, when all the rescue agents finish the task, an optimal action sequence is generated, that is, the optimal planning, as shown in Fig. 3.

The strategy of agent, written as π *(s, a)*, describes a behavior mode. It is a function that returns the probability of taking an action in a certain state Therefore, for a given state:

$$\Sigma_a \pi(s,a) = 1 \quad (9)$$

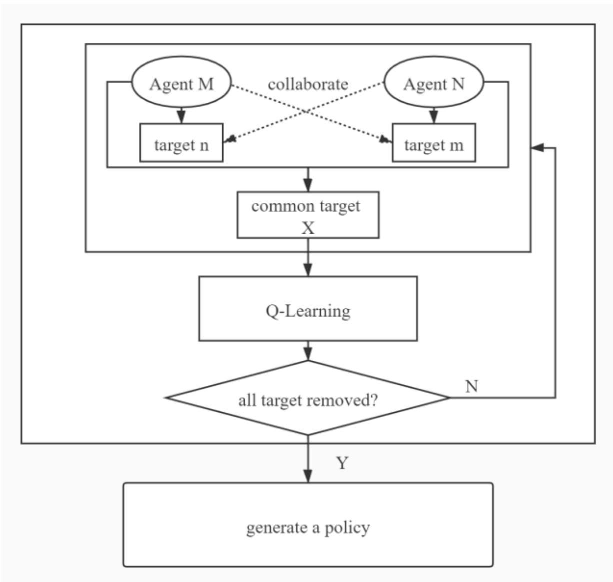

Fig. 3. Heterogeneous agent collaboration planning process.

The final strategy in this model is the optimal state-action mapping obtained by the heterogeneous agent after iterative learning by completing the rescue task. The algorithm is as follows:

Algorithm: Heterogeneous agent Q-Learning

a) *Initialize s1 s2*
b) *Repeat(for each episode):*
c) *Choosea1froms1usingpolicyderivedfrom Q1(e.g.,ε-greedy)*
d) *Choosea2froms2usingpolicyderivedfrom Q2(e.g.,ε-greedy)*
e) *Takeactiona1,observer1,s1′*
f) *Takeactiona2,observer2,s2′*
g) *Q1 (s1,a1)←Q1 (s1,a1)+α[r1+γmaxa'Q1 (s1′,a1′)−Q1 (s1,a1)]*
h) *Q2 (s2,a2)←Q2 (s2,a2)+α[r2+γmaxa'Q2 (s2′,a2′)−Q2 (s2,a2)]*
i) *s1←s1′*
j) *s2←s2′*
k) *Until s1 and s1are both terminal*

HADQ is an extension of the Q-learning algorithm model. Compared with the single-agent obstacle avoidance path planning, it improves the simulation of the model to heterogeneous multi-agents and has better expansibility.

III. Simulation Experiment and Result Analysis

In order to explore the feasibility and efficiency of heterogeneous agent collaboration, we will set collaboration experiment and comparison experiment, and further gradient experiment to discuss the processing capacity range.

In the scenario, the road blocking part is added to help the simulation. In the scenario, the starting point coordinate of the rescue agent, the iteration speed and the target coordinate are set. See Fig. 4 for an example of simulation.

Fig. 4. Disaster scenario simulation.

IV. Example of Simulation Experiment Scenario

In the simulation, we use squares to represent two kinds of agents. The yellow and green squares are agent *H* and agent *M*, respectively. The triangles with different color represent the different targets. The black circle represents the agent's common target, and the black block part is the obstacle in the scene. The blue circle represents the exit position.

A. The Simulation Experiments

Three simulation experiments are set up:

Experiment 1: Homogeneous agent cooperation experiment. We set that the agents can rescue all targets with a reward of 7.

Experiment 2: Heterogeneous agent cooperation experiment. We set heterogeneous agents and targets into two kinds. When the agent reaches the target he is mainly responsible for, it will give a higher reward and the target disappears from the map. If it reaches other targets, it will gain a lower reward and the target needs further rescue. This rule is determined by the heterogeneous of the rescue agent, and the average reward of each agent in the scene is guaranteed to be the same as that of the experiment 1. Set the corresponding reward as follows Table I:

TABLE I: THE DEFINITION OF THE AGENT

	Target m	Target n	Target X
mobile agent M	Reward 10 remove m	Reward 4	Remove when M and N both arrive
mobile agent N	Reward 4	Reward 10 remove n	

Experiment 3: To find out processing capacity range of HADQ, adding the number of *h* and *m*. The quotient of the total reward against the number of heterogeneous targets is used as the evaluation criterion. The target number setting of experiment 3 is shown in the Fig. 5, and the targets were successively increased from 1 to 4 pairs:

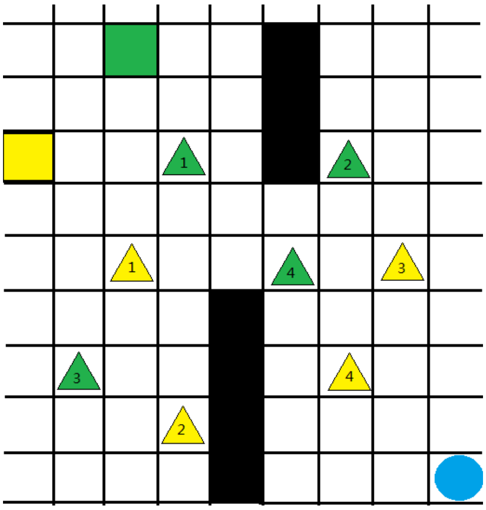

Fig. 5. Screenshot of the gradient contrast experiment.

B. Evaluation and Analysis.

In experiments 1 and 2, we found that the convergence speed of the algorithm was not good at the beginning due to the randomness of the early search. We added conditions to encourage agents to approach the end point, punish agents when search away from the end point, and apply this method to all experiments; We use *dis* to represent the distance of agent from the exit position. *agent []* and *circle []* respectively represent the coordinates of agent and exit position. The optimization is as follows:

$$dis = \sqrt{(Agent[x]-circle[x])^2 - (Agent[y]-circle[y])^2}$$
$$\begin{cases} dis_t > dis_{t+1}, reward = a(a>0) \\ dis_t < dis_{t+1}, reward = -a(a>0) \end{cases} \tag{10}$$

In the formula, *agent []* and *circle []* represent the rescue agent and the exit position, *x*, *y*, respectively, on behalf of its horizontal ordinate and vertical coordinate, dis_t and dis_{t+1} represents the distance of the action agent at time *t* and *t+1*, *a* is a reward value.

Record the data of three experiments, and draw a chart as shown in Fig. 6. *Award1* and *Award2* represent the reward accumulation of experiment 1 and experiment 2. We can see from the accumulation trend of total reward values that homogeneous collaboration accumulates rewards faster in the early stage, but after a period of iteration, when the collaboration scheme is convergent and mature, the total reward value of experiment 2 is higher, indicating that the collaborative planning of heterogeneous agents has better performance.

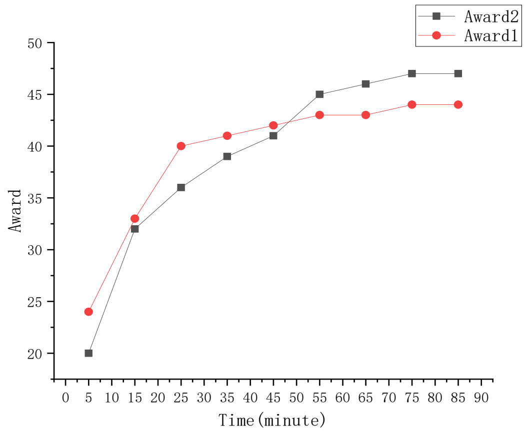

Fig. 6. Data statistics of experiment 1 and experiment 2.

TABLE II: THE RESULTS OF EXPERIMENT 3

Number of targets	2	3	4	5	6	7	8
Reward_all/target_number	14	17	19	18	20	16	15

According to experiment 3 (Table II), in the case of two heterogeneous agents, the impact of the number of targets on the average collaboration return is calculated. The recorded data is plotted in Fig. 7, which shows that the two heterogeneous agents have the highest efficiency when dealing with two types of targets with a total of about 6 targets.

In summary, after a period of early learning, the overall return of heterogeneous agent scheme will obviously exceed that of homogeneous agent. When heterogeneous agents work together, the whole has better global learning efficiency and higher total rewards. The pertinence of the heterogeneous agent itself finally brings higher joint returns on the premise of reinforcement learning. However, the heterogeneous agent model has limited processing capacity for a large number of targets.

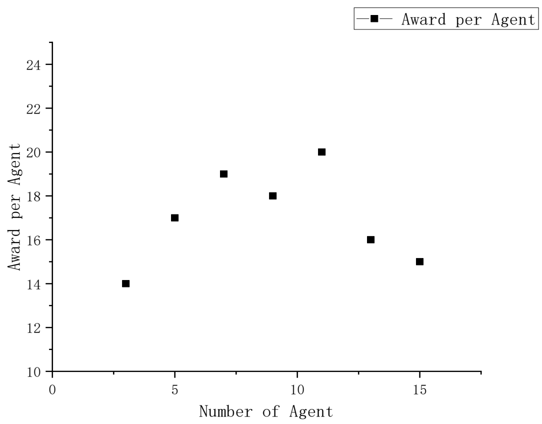

Fig. 7. Statistics of gradient experiment results.

V. CONCLUSION

In the heterogeneous agent collaboration based on Q-Learning, agents gradually get familiar with the environment in the continuous iteration, and after contacting the target, the joint planning of multiple agents is different from that of homogeneous agents due to the difference of reward value. There are still some aspects where improvements can be made:

1) In the early stage of learning, the search of rescue agent is in the global search because it has not received the reward of the target, and the randomness is very strong in the early stage. In this case, the overall calculation time is unsatisfactory. It's needed to guide agents in the early stage and pay attention to the quality of search results.
2) However, it has not been able to highlight the complex cooperative nature among heterogeneous agents, such as some targets need to be rescued by different agents under a certain order.
3) By contrast experiments, the optimal number of targets that a fixed number of heterogeneous agents can handle is explored. but a relatively common standard has not been summarized, which requires further research.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Chenfeng Gan conceptualized the research idea and finalized the paper; Wei Liu analyzed the data; Xingyu Ye & Ning Wang reviewed the article and supervision of whole research.

Acknowledgment

This project supported by Hubei Province Technology Innovation Major Project (No.2019AAA045), Hubei natural science foundation project (No.2019CFB172).

References

[1] W. V. D. Hoek and M. Wooldridge, "Model checking cooperation, knowledge and time — A case study," *Research in Economics*, vol. 57, no. 3, pp. 235-265,2003.

[2] T. Hang and M. Fang, "Layered model based multi-agent system collaboration mechanism and its application in RoboCup 3D," *Journal of Anqing Normal University (Natural Science Edition)*, no. 23, p. 75, 2017.

[3] N. Han and K. Zhang, "The distributed formation control of heterogeneous multi-agent systems visa output regulation," *Journal of Longdong University*, vol. 29, no. 1, 2018.

[4] J. Cano, D. R. White, A. Bordallo *et al*., "Solving the task variant allocation problem in distributed robotics," *Autonomous Robots*, vol. 42, no. 7, pp. 1477-1495,2018.

[5] M. Li, W. Liu, and Y. Zhang, "Mulit-agent dynamic task allocation based on improved contract net protocol," *Journal of Shandong University (Engineering Science)*, vol. 46, no. 2, pp. 51-63, 2016.

[6] J. Liu and Y. Zhang, "A collaborative task decision-making method based on blackboard framework," *Fire Control & Command Control*, vol. 42, no. 11, pp. 43-48, ,2017.

[7] J. Wang, W. Liu, K. Wu *et al*., "Progress in heterogeneous agent collaboration," *Journal of Wuhan Institute of Technology*, vol. 39, no. 4, pp. 378-386, 2017.

[8] W. Liu and J. Wang, "GCC based agents collaboration mediation: A case study of medical waste AGV transportation," in *Proc. 2018 the International Conference on Software and Services Engineering*, 2018, pp. 151-156.

[9] J. Wang, W. Liu, and S. Li, "Agent collaboration in intelligent parking lot systems: Dynamic generation of commitment protocol," in *Proc. 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), IEEE Computer Society*, 2017, pp. 437-444.

[10] O. V. Darintsev, B. S. Yudintsev, A. Y. Alekseev *et al*., "Methods of a Heterogeneous multi-agent robotic system group control," *Procedia Computer Ence*, vol. 150, pp. 687-694, 2019.

[11] Y. Gao, S. Chen, and X. Lu, "Research on reinforcement learning technology: A review," *ACTA AUTOMATICA SINICA*, vol. 30, no. 1, pp. 86-100, 2004.

[12] Z. Feng, "Research on cooperation and competition mechanism among agents," Dissertation, Xihua University, Chengdu, 2015.

[13] U. Ümit, S. G. Mehmet, and E. Bostanci, *A Q-Learning-Based Approach for Simple and Multi-agent Systems*, *Multi-agent Systems - Strategies and Applications*, 2020.

[14] Y. Fan and L. Li, "Multi agent system coordination model based on optimized evolutionary algorithms," *Bulletin of Science and Technology*, no. 11, pp. 207-210, 2015.

[15] S. Haykin, *Neural Networks and Machine Learning*, China Machine Press, 2011.

[16] Y. Xu, S. Wang, and X. Wang, "Research on urban traffic congestion based on Dijkstra improved algorithm," *Geospatial Information*, vol. 17, no. 11, pp. 80-82, 2019.

[17] W. Xu, Q. Zhang, Y. Zou *et al.*, "Research on transportation path planning for logistics UAV based on improved A* Algorithm," *Journal of East China Jiaotong University*, vol. 36, no. 6, pp. 39-46, 2019

[18] Y. Guo, G. Goncalves, and H. Tienté, "Retraction note to: A multi-agent based self-adaptive genetic algorithm for the long-term car pooling problem," *Journal of Mathematical Modelling and Algorithms in Operations Research*, vol. 14, no. 4, p. 481, 2015.

[19] J. Ding, S. Chen, and G. Lu, "X-ray security inspection method using active vision based on Q-learning algorithm," *Journal of Computer Applications*, vol. 38, no. 12, pp. 3414-3418, 2018.

[20] Z. Ji, J. Lan, Y. Jiang *et al*., "Q-learning based business differentiating routing mechanism in SDN architecture," *Chinese Journal of Network and Information Security*, vol. 4, no. 9, pp. 17-22, 2018.

[21] K. Wang, X. Bu, R. Li *et al.*, "Path planning for robots based on deep reinforcement learning by depth constraint," *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 46, no. 12, pp. 77-82, 2018.

[22] Z. Zhao and J. Wang, "Ship automatic anti-collision path simulations based on reinforcement learning in different encounter situations," *Science Technology and Engineering*, vol. 18, no. 18, pp. 218-223, 2018.

[23] B. H. Abed-Alguni, S. K. Chalup, F. A. Henskens *et al*., "A multi-agent cooperative reinforcement learning model using a hierarchy of consultants, tutors and workers," *Vietnam Journal of Computer Science*, vol. 2, no. 4, pp. 213-226, 2015.

[24] W. Du and S. Ding, "Overview on multi-agent reinforcement learning," *Computer Science*, vol. 46, no. 8, pp. 1-8, 2019.

Chenfeng Gan was born in Hubei, China in 1994. He received the B.E. degree from Wuhan Institute of Technology, Wuhan, China, where he is currently pursuing the M.E. degree. His research interests are verification of self-adaptive system.

Wei Liu received her Ph.D. in the State Key Lab of Software Engineering of Wuhan University in 2008 and B.Sc in Computer School of Wuhan University in 2003. Her research interests include semantic computing, intelligent software engineering and service computing. Currently she serves as a vice professor in School of Computer Science and Engineering at Wuhan Institute of Technology and senior researcher at Hubei Province Key Laboratory of Intelligent Robot.

Ning Wang was born in Henan, China in 1997. She received the B.E. degree from Wuhan Institute of Technology, Wuhan, China, where she is currently pursuing the M.E. degree. Her research interests are natural language processing and deep learning.

Xingyu Ye was born in Zhejiang, China in 1996. He received the B.E. degree from Wuhan Institute of Technology, Wuhan, China, where he is currently pursuing the M.E. degree. His research interests are verification of self-adaptive system.