IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2020, accepted February 7, 2020, date of publication February 11, 2020, date of current version February 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973169

Adaptive Exploration Strategy With Multi-Attribute Decision-Making for Reinforcement Learning

CHUNYANG HU[1] AND MENG XU[2]
[1]School of Computer Engineering, Hubei University of Arts and Science, Xiangyang 441053, China
[2]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Meng Xu (menghsu@mail.nwpu.edu.cn)

This work was supported in part by the Natural Science Foundation of Hubei Province under Grant 2013CFC026, and in part by the International Science and Technology Cooperation Program of Hubei Province under Grant 2017AHB060.

ABSTRACT Reinforcement Learning (RL) agents often encounter the bottleneck of the performance when the dilemma of exploration and exploitation arises. In this study, an adaptive exploration strategy with multi-attribute decision-making is proposed to address the trade-off problem between exploration and exploitation. Firstly, the proposed method decomposes a complex task into several sub-tasks and trains each sub-task using the same training method individually. Then, the proposed method uses a multi-attribute decision-making method to develop an action policy integrating the training results of these trained sub-tasks. There are practical advantages to improve learning performance by allowing multiple learners to learn in parallel. An adaptive exploration strategy determines the probability of exploration depending on the information entropy instead of the suffocating work of empirical tuning. Finally, transfer learning extends the applicability of the proposed method. The experiment of the robot migration, the robot confrontation, and the real wheeled mobile robot are used to demonstrate the availability and practicability of the proposed method.

INDEX TERMS Reinforcement learning, exploration and exploitation, multi-attribute decision-making, adaptive exploration, transfer learning.

I. INTRODUCTION

A. REINFORCEMENT LEARNING

Reinforcement learning allows agents to perform tasks through trial-and-error learning, which is a type of machine learning algorithm [1]. Agents gain experience by interacting with the environment constantly and ultimately acquire the optimal strategy to guide them get the greatest cumulative reward in the learning process. RL methods require agents to actively explore the unknown environment and receive feedback from the environment to one action taken [2], [3]. Agents use positive or negative feedback to acquire experience that they need to optimize the policy when they perform a task. Recently, this multi-learner parallel learning approach, such as the A3C algorithm [4], has successfully helped agents achieve beyond the human-level in some video games [5], [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.

B. THE DILEMMA OF THE EXPLORATION AND EXPLOITATION

The trade-off between exploration and exploitation has always been a dilemma without a unified solution in reinforcement learning systems [7]. The exploration strategy guides the learning agent to explore more unknown environments by collecting new experience. An appropriate method for exploration must determine the opportunity to collect new experiences and to exploit current experience so as to obtain the greatest cumulative reward [8]. Conversely, it is inappropriate for agents to exploit for a long period and even the current experience is inaccurate or inadequate. Therefore, it is crucial to develop an appropriate action policy for the exploration scheme, which will affect the convergence rate of the RL algorithm and the cumulative reward that agents can receive.

Previous researchers usually used the probabilistic exploration method to tackle this problem, such as the epsilon-greedy [9], [10]. The epsilon-greedy involves a disadvantage,

which is that the learning time is exponentially proportional to the scale of state space [11]. Meanwhile, a fixed value for ε always maintains the same probability of each action selected in the learning process, so a lot of ineffective exploration has emerged even the dilemma of local optimum. The softmax policy is another common method for exploration [12], [13]. Previous researchers proposed a method of using Boltzmann distribution and simulated annealing (SA) to tackle the conflicting requirements of exploration and exploitation [14]. Exploration guided by an intrinsic motivation has been extensively studied by scholars before [15]. Not only the intrinsic motivation, Thompson sampling, bootstrapped models [16] and parameter space exploration [17] can be used as a mechanism to guide exploration.

C. MULTI-ATTRIBUTE DECISION-MAKING METHODS

Problems of multi-attribute decision-making (MADM) are common occurrences when decision-makers are faced with multiple factors to make a wise decision [18]. These factors can be regarded as attributes and used as evaluation criteria to evaluate a scheme, and the scheme with the highest evaluation is adopted by a decision-maker as the best one, which is the process of the MADM. Multiple attributes for the decision-making result need to be ranked or sorted by taking into account many related factors that are usually measured or evaluated using either numerical values with certain units based on the prior experience [19]. The ordered weighted averaging (OWA) operator is a very common and effective method for a multi-attribute decision-making problem [20]. The learning experience gained by multiple learners serves as the prior experience for the source task, if the available actions act as the scheme set. The value functions learned by each learner contribute to the final decision-making result as an evaluation factor, that is, an effective action that the agent will take. Multi-attribute decision-making provides a solution to exploit the prior experience gathered from these sub-tasks.

D. RESEARCH GAPS

For the dilemma of exploration and exploitation, prior works provide good exploration without exploiting the particular structure of the task itself. However, agents need to learn many tasks, not just one, in which prior experience can be used to inform how exploration in new scenarios should be performed. Complex tasks often require a lot of learning time. Moreover, the classical training method does not effectively collect more experience when this training method is compared with the parallel training method of multi-learner [21], [22]. Therefore, it is an important issue for a learning agent to determine a way of exploration for new experience and exploitation using the current experience.

For the MADM, the conventional methods use the empirical aggregation operator, which has a subjective impact on the decision-making results. Conventional aggregation operator weight attributes empirically, which undoubtedly bring a lot of subjective factors to decision-making results. Moreover, evaluation values are often inaccurate because they are calculated using these unreasonable weights.

E. RESEARCH METHODS

In this study, we use three ways: designing an effective exploration strategy, expanding prior experience for exploitation and designing an appropriate action policy with the current knowledge, to fill the gaps mentioned above for the trade-off problem between exploration and exploitation.

Firstly, to achieve the first way, this study uses an adaptive exploration strategy to determine the value for ε using the information entropy [23], which encourages agents to explore more in the early stage of learning and exploit more in the later stage. Generally, for an action policy, it is appropriate to require agents to make rational use of the current learning experience. Secondly, to achieve the second way, this paper proposes an effective method that collects learning experience using a divide-and-conquer approach. The proposed method decomposes the source task into several sub-tasks and exploits the prior experience from sub-tasks that are trained by the same (Temporal Difference)TD method [24]. The method of multi-learner parallel learning expands the source of prior experience and improves the performance of learning algorithms, but the original method does not. Thirdly, to achieve the third way, this study proposed an action policy using the MADM, which regards the available actions taken by an agent as schemes, and regards the state-action value function obtained from each sub-task as attributes for these schemes. This study uses the MADM method to calculate the evaluation value for each scheme, and then the action with a maximum evaluation value is almost one that the agent will take. The average value for the standardized reward of each sub-task is calculated to act as the current reward for the source task and this average value is used to update the action value function iteratively.

To fill the gaps mentioned above for the MADM, this study presents a MADM method with a support function, which weights attribute using the visibility graph theory [25]. This method considers the influence of both the value and location relationship of attributes on the decision-making results, so the subjective factors are eliminated.

Transfer learning extends the learning model to different task scenarios and accelerates the learning process for the new task. Intermediate tasks bring more prior experience to agents to reduce the difficulty of executing the target task, which differs greatly from the source task.

F. CONTRIBUTIONS IN THIS WORK

The main contributions of this paper are as follows.

1) For the complex task, this study decomposes the source task into several sub-tasks in a divide-and-conquer approach, and then these sub-tasks are trained by the same training method respectively. This divide-and-conquer method can not only expand the source of prior experience but also accelerate the learning rate by multi-learner parallel training.

2) An action policy is developed using a MADM method and this action policy determines an action depending on the acquired knowledge. This study uses a MADM method integrating a support function to calculate the weights of attributes, which can eliminate the subjective factors of empirical methods and increase the accuracy of results for a MADM problem.
3) To tackle the dilemma of exploration and exploitation, this study first develops an adaptive exploration strategy with a MADM. The adaptive exploration strategy uses the information entropy technology to determine the threshold for the epsilon-greedy. The experimental results show that the proposed method outperforms competitors.

G. STRUCTURE OF THIS PAPER

The remainder of this paper is organized as following. Section II presents the background for the proposed methods and briefly describes the Q-learning, the softmax policy and the MADM. Section III presents a new MADM method using a support function and this method is used for developing the action policy. Section IV presents a training method for subtasks and a standardized method for rewards obtained by each learner. Section V presents an adaptive exploration strategy and an action policy using the proposed MADM method. Experiments are conducted to demonstrate the effectiveness of the proposed methods in Section VI. Section VII gives the detail of future improvement for the adaptive exploration by using transfer learning. Conclusions are drawn in the last section.

II. BACKGROUND

A. Q-LEARNING ALGORITHM

Q-learning is a model-free reinforcement learning algorithm, which is proposed by Watkins in 1989 [26]. Q-learning is commonly used because it is very simple and fast converges. The Q value is given by,

$$Q(s,a) = E\left[r|\,(s,a)\right] + \gamma \sum\nolimits_{s'} T_a^{ss'} \max Q(s',a') \tag{1}$$

The law for updating Q values using the TD error is given by,

$$Q_{t+1}(s,a) = (1-\alpha)\, Q_t(s_t,a) + \alpha\left[r_t + \gamma \max Q_t(s_{t+1},a')\right] \tag{2}$$

where α is the learning rate and the range is $(0, 1)$. The learning rate reflects the efficiency of the learning process. A round of learning process is terminated when the agent reaches the target state. The agent then returns to the initial state and starts the next round until the end of the whole learning process, so the optimal strategy is obtained.

B. SOFTMAX POLICY

Softmax policy is an action policy that is commonly used for exploration schemes to tackle the dilemma of exploration and exploitation [9]. The agent uses this method to select the action using the average reward for each action. The action a_t with the highest average reward is the best one to be selected. The simulated annealing (SA) [27] algorithm optimizes the softmax policy to control the randomness of actions.

The probability for each action is given by,

$$P_i = \frac{\exp\left(a_i\right)}{\sum_{k=0}^{K} \exp\left(a_k\right)} \tag{3}$$

where P_i represents the probability for selecting an action a_i and the total number of available actions is K.

The probability for selecting action a_i is given by,

$$P\left(a_i|s_t\right) = \frac{\exp\left(Q\left(s_t,a_i\right)/T_t\right)}{\sum_{k=1}^{K} \exp\left(Q\left(s_t,a_k\right)/T_t\right)} \tag{4}$$

where T_t is the temperature parameter. The temperature parameter for the simulated annealing algorithm is tuned using Eq.(5).

$$\begin{cases} T_0 = T_{\max} \\ T_{t+1} = \eta\left(T_t - T_{\max}\right) + T_{\min} \end{cases} \tag{5}$$

where η is the annealing factor and its range is $0 \le \eta \le 1$.

C. MULTI-ATTRIBUTE DECISION-MAKING

In general, selected schemes often have many predefined attributes, which affect the decision-making result. MADM measures each attribute and gives the evaluation value for each scheme. The aggregation operators are commonly used to find a solution for the problem of MADM, which calculates the evaluation value for each scheme effectively. Ordered weighted averaging (OWA) operator is a simple but effective information aggregation operator among all aggregation operators and it weights each attribute depending on its significance [28].

A set of original data is $(b_1, b_2, \ldots, b_m)$, which is sorted from large to small to obtain an ordered sequence $(\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_m)$. The OWA operator is given by,

$$OWA(b_1, b_2, \ldots, b_m) = \sum\nolimits_{i=1}^{m} \omega_i \hat{b}_i \tag{6}$$

where, $(\omega_1, \omega_2, \ldots, \omega_m)$ is a weight vector and $\sum_{i=1}^{m} \omega_i = 1, \omega_i \in [0, 1]$.

III. A MULTI-ATTRIBUTE DECISION-MAKING METHOD WITH A SUPPORT FUNCTION

A. SUPPORT FUNCTION

Graph technology has been used to address a type of the machine learning problem, such as the sparse feature extraction, the dimensionality reduction and so on, and has been proved to have excellent performance [29], [30]. After the values for attributes are ordered, the visibility graph theory converts the values in the data sequence into nodes in a complex network (CN) [31]. Each node in a CN corresponds to the value in the data sequence one by one. The degree of support is the degree of correlation between values in the data sequence, which is reflected by the connection relationship

between nodes in a CN. The more connections a node has, the higher the support for the value receives.

The definition of a visibility graph is as follows:

Definition 1: Two data are represented by two-tuple $(i, \hat{b}_i)$ and $(j, \hat{b}_j)$ respectively in the data sequence. If there is a correlation between two arbitrary data, then for any data $(k, \hat{b}_k)$ between the two data, it satisfies Eq.(7).

$$\hat{b}_k < \hat{b}_j + (\hat{b}_i - \hat{b}_j)\frac{j-k}{j-i}(i, \hat{b}_i) \tag{7}$$

If two values in the data sequence satisfy Eq.(7), the visibility graph theory emphasizes that the corresponding two nodes are connected in complex networks. The degree of a node is defined as the number of edges connected with other nodes in the CN. In general, the degree of a node is positively related to its importance and the support function describes the support degree for nodes. Coulomb's law emphasizes that the support between nodes needs to consider both the value for nodes and the distance between nodes [32].

Definition 2: If the values of the two nodes are D_i, D_j and the distance between the two nodes is $dis_{ij} = |D_i - D_j|$. The support function between the two nodes is given by,

$$Supp(D_iD_j) = \frac{D_iD_j}{dis_{ij}^n} \tag{8}$$

where n is a positive integer. $Supp(D_iD_j)$ denotes the support for the node D_j to the node D_i. The sum of support for the node D_i received by all nodes is given by,

$$Sum(D_i) = \sum_{\substack{j=1 \\ j\neq i}}^{n} Supp(D_i, D_j) \tag{9}$$

B. THE ORDERED WEIGHTED AVERAGING OPERATOR WITH A SUPPORT FUNCTION

A set of ordered data $(b_1, b_2, \ldots, b_n)$ from large to small is represented by $D = \{D_1, D_2, \ldots, D_n\}$. In the ordered data sequence, the number of data is n. Ordered weighted averaging with a support function (VOWA) is given by,

$$VOWA(D_1, D_2, D_3, \ldots, D_n) = \sum_{i=1}^{n} \omega_i D_i \tag{10}$$

where $\omega_i \in [0, 1]$, $\sum_{i=1}^{n} \omega_i = 1$, ω_i is the weight for the data D_i.

If there is a data sequence $(b_1, b_2, \ldots, b_n)$, and any permutation of that data sequence is represented by $(\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_n)$. Evaluation values of the VOWA operator for the two data sequences are equal, as shown in Eq.(11).

$$VOWA(\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_n) = VOWA(b_1, b_2, \ldots, b_n) \tag{11}$$

Then, the weight for the data D_i is given by,

$$\omega_i = \frac{Sum(D_i)}{\sum_{i=1}^{n} Sum(D_i)} \tag{12}$$

where the sum of the support for the node D_i is $Sum(D_i)$. All nodes that satisfy Eq.(7) are connected with the node D_i. So, evaluation value using the VOWA operator for this sequence is given by,

$$VOWA(D_1, D_2, D_3, \ldots, D_n) = \sum_{i=1}^{n} \left(\frac{Sum(D_i)}{\sum_{i=1}^{n} Sum(D_i)}\right) D_i \tag{13}$$

IV. TRAINING FOR SUB-TASKS

A. A TRAINING METHOD FOR SUB-TASKS

This paper proposes an adaptive exploration strategy with a MADM to address the dilemma of exploration and exploitation. For a type of the problem of the multi-objective decision-making, it is hard to solve the source task. So, the divide and conquer approach may be a solution for the multi-objective decision-making problem. Several trained sub-tasks can be used as modular building blocks to develop a rapid prototype for a complicated task to improve the learning performance. For the complicated task, the learning experience for related sub-tasks is assembled to induce an action policy. In this work, inspired by the divide and conquer approach, a complex task is decomposed into sub-tasks, and the learning experience of trained sub-tasks are fused to complete a complex task. A TD method [24] is used to train the sub-tasks and this training method is shown in **Algorithm 1.** For example, in robot soccer games, we define the task of a robot playing soccer as a complex task, which consists of several sub-tasks: passing the ball, taking the ball away, avoiding obstacles, shooting and so on.

B. A STANDARDIZED METHOD FOR REWARDS

These sub-tasks are trained by the same training method separately and each learner receives a different reward. The reward received by an agent for the source task needs to consider the rewards received by all sub-tasks. The standardized method deals with all rewards, which belong to different sub-tasks.

If the starting time of an episode is t_1, the reward received by sub-task k is $r_k^{t_1}$.$r_k^{t_p}$ is the reward that sub-task k receives at time t_p. The average value of rewards for all sub-tasks receives at the time t_p is μ_{t_p}. The variance of the rewards that all sub-tasks received at the time t_p is σ_{t_p}. The average and variance of the rewards for all sub-tasks are given as Eqs. (14) and (15).

$$\mu_{t_p} = \frac{\sum_{k=1}^{m} r_k^{t_p}}{m} \tag{14}$$

$$\sigma_{t_p} = \sqrt{\frac{1}{m}\sum_{k=1}^{m} (\mu_{t_p} - r_k^{t_p})^2} \tag{15}$$

where, m is the number of sub-tasks.

The standardized reward $R_k^{t_p}$ for sub-task k is given by,

$$R_k^{t_p} = \frac{r_k^{t_p} - \mu_{t_p}}{\sigma_{t_p}} \quad k = 1, 2, \ldots, m \tag{16}$$

The average value of the standardized rewards at the time t_p for each sub-task is $\hat{R}_{t_p}$, which will be given to the source

Algorithm 1 The Training for Each Sub-Task

1. **Definition**
2. s_t := Current state
3. a_t := Current action
4. $reward(t)$:= Instant reward
5. s_{t+1} := The state of the next moment
6. a_{t+1} := The action of the next moment
7. α := Learning rate for RL agent
8. γ := Discount factor
9. δ_t := The TD error for the current state and action
10. **Initialization**
11. Initialize s_t, s_{t+1}, a_t, a_{t+1}, $reward(t)$, Q table and cumulative reward $totalreward$.
12. **Repeat** (for each step)
13. Choose an initial state s_0
14. $t \leftarrow 0$; $i \leftarrow 0$;
15. **Repeat** (for each step of the episode):
16. Take action a_t(e.g.,ε-greedy strategy)
17. Observe s_{t+1}, and reward $reward(t)$
18. $totalreward+ = reward(t)$;
19. Take action a_t(e.g.,ε-greedy strategy)
20. $s_t \leftarrow s_{t+1}$
21. $$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[reward(t) + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$$
22. Update the TD error:
23. $\delta_t = \alpha[Q\_next(s_t, a_t) - Q\_current(s_t, a_t)]$
24. $t++$
25. **until** s is terminal.
26. $i++$

task, as shown in Eq.(17).

$$\hat{R}_{t_p} = \frac{\sum_{k=1}^{m} R_k^{t_p}}{m} \tag{17}$$

V. AN ADAPTIVE EXPLORATION STRATEGY WITH A MULTI-ATTRIBUTE DECISION-MAKING

A. AN ADAPTIVE EXPLORATION STRATEGY

An adaptive exploration strategy uses information entropy to achieve a more effective exploration. The proposed method achieves a state-action-dependent exploration with a certain probability. The value ε depends on the number of all available actions in the current state. At the beginning of a learning process, if a common action appears, the agent will explore more, which is indicated by the defined information entropy during this learning process. If the agent acquires sustainable experience indicated by the temporal difference error (TD error), the probability of exploration will be reduced. In this study, we use fluctuation value $|\Delta Q(s_t, a_t)|$ to represent the TD error. The TD error is given by,

$$|\Delta Q(s_t, a_t)| = |r + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)| \tag{18}$$

If the fluctuation value is large, the agent makes more exploration and vice versa. After a learning step, the probability for exploration is calculated. $A = \{a_1, a_2, \ldots, a_{N_A}\}$ is the action set and the number of actions is N_A. The number of times that the action a_i is taken at the state s_t is $N(a_i, s_t)$. The probability of taking action a_i in the state s_t is given by,

$$\begin{cases} P(a_i|s_t) = N(a_i, s_t) / \sum_{a_i \in A} N(a_i, s_t), & if\ N(a_i, s_t) \neq 0 \\ P(a_i|s_t) = 0, & if\ N(a_i, s_t) = 0 \end{cases} \tag{19}$$

The entropy $E_H(s_t)$ for the state s_t is given by,

$$\begin{cases} E_H(s_t) = -\sum_{a_i \in A} (\log P(a_i|s_t)) \cdot (P(a_i|s_t)) \\ \bar{E}_H(s_t) = E_H(s_t) / \log N_A \end{cases} \tag{20}$$

where, $\bar{E}_H(s_t)$ is the normalized entropy.$(\log 0) \times 0 = 0$.

The value for ε in the epsilon-greedy can be calculated using the $\bar{E}_H(s_t)$, which is shown in Eq.(21).

$$\varepsilon = \begin{cases} \left(1 - \bar{E}_H(s_t)\right), & otherwise \\ \left(1 - \bar{E}_H(s_t)\right)\left(1 - \exp\left(\dfrac{-|\Delta Q(s_t, a_t)|}{\sum_{a_i \in A} N(a_i, s_t)}\right)\right), & if\ \left(\dfrac{-|\Delta Q(s_t, a_t)|}{\sum_{a_i \in A} N(a_i, s_t)}\right) < \mathrm{H}_c \end{cases} \tag{21}$$

where, H_c is a constant value, and $|\Delta Q(s_t, a_t)|$ is the TD error. $\bar{E}_H(s_t)$ is a measurement of the uniformity for the available actions. The value of $\bar{E}_H(s_t)$ is maximum, if each action has been tried the same frequency at state s_t. The smaller the value of $\bar{E}_H(s_t)$, the more the agent will explore, and vice versa. $\bar{E}_H(s_t)$ gives an opportunity for an agent to try each action possible. In the early stage of a learning process, the agent will explore more and the agent performs more exploitation in the later stage.

B. AN ACTION POLICY WITH A MADM

The source task is decomposed into multiple sub-tasks, and then each sub-task is trained in the same way. The prior experience for an agent depends on the learning experience of each sub-task, which enlarges the source of prior experience. This study uses a MADM method using a support function to calculate the weight for attributes. The state-action value function for m in the current state s_t and the current action a_t is $Q^{(1)}(s_t, a_t), Q^{(2)}(s_t, a_t), \ldots, Q^{(m)}(s_t, a_t)$. The action set is defined as the scheme set that the agent takes. Attributes for the scheme is the state-action value function of the source task and the state-action value function of each sub-task. Then we use the MADM method to give each scheme an evaluation value. The agent performs the action that belongs to the maximum evaluation value according to the definition of MADM. $Q^{(p)}(s_t, a_1), Q^{(p)}(s_t, a_2), Q^{(p)}(s_t, a_3), \ldots, Q^{(p)}(s_t, a_n)$ is the state-action function for sub-task p for the state s_t and action a_t. n is the number of actions. The state-action value function of the source task for state s_t is $\{Q(s_t, a_1), Q(s_t, a_2), Q(s_t, a_3), \ldots, Q(s_t, a_n)\}$.

The decision-making matrix Q is given in (22), where (22), as shown at the bottom of this page.

The scheme set is represented as $U = \{a_1, a_2, \ldots, a_n\}$. $Y = \{Q^1(s_t, a_i), Q^2(s_t, a_i), \ldots, Q^m(s_t, a_i), Q(s_t, a_i)\}$ is the attribute for the scheme a_i. The process of calculating the evaluation value $VOWA(a_k)$ of the action a_k for the sub-task k involves the following steps.

Step 1: For the scheme a_k, attributes $V = \{Q^1(s_t, a_k), Q^2(s_t, a_k), \ldots, Q^m(s_t, a_k), Q(s_t, a_k)\}$ are ordered from large to small and the ordered data is $\{Q^{(1)}(s_t, a_k), Q^{(2)}(s_t, a_k), \ldots, Q^{(m)}(s_t, a_k), Q^{(m+1)}(s_t, a_k)\}$.

Step 2: Visibility between nodes is defined as the direct connection between nodes in complex networks. We evaluate the visibility nodes for each node in turn. For the node $Q^{(i)}(s_t, a_k)$, the set VQ for the visibility nodes is given by,

$$\begin{cases} VG(ps) : Q^{(kp)}(s_t, a_k) \le Q^{(ps)}(s_t, a_k) \\ \quad + \left(Q^{(i)}(s_t, a_k) - Q^{(ps)}(s_t, a_k)\right) \\ \quad \cdot \left(\dfrac{ps - kp}{ps - i}\right) \cdot \left(i, Q^{(i)}(s_t, a_k)\right); \forall i < kp < ps \\ VQ = \{Q^{(ps)}(s_t, a_k) | VG(p), 1 \le ps \le m + 1\} \end{cases} \tag{23}$$

where kp and ks are indexes from $(1, m)$ for any two nodes, which satisfy $kp < ks$.

Step 3: The support for the node $Q^{(i)}(s_t, a_k)$ received from its visibility nodes is given by,

$$Sum\left(Q^{(i)}(s_t, a_k)\right) = \sum_{Q^{(ps)}(s_t, a_k) \in VQ} Supp\left(Q^{(ps)}(s_t, a_k), Q^{(i)}(s_t, a_k)\right) \tag{24}$$

Step 4: The weight of each node is calculated using the support for this node and the weight vector is written as $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3, \ldots, \omega_{m+1})$. Taking ω_s as an example, we use Eq.(25) to calculate the weight.

$$\omega_s = \frac{Sum\left(Q^{(s)}(s_t, a_k)\right)}{\sum_{j=1}^{m+1} Sum\left(Q^{(j)}(s_t, a_k)\right)} \tag{25}$$

Step 5: The evaluation value for the scheme a_k is given by,

$$\begin{aligned} VOWA(a_k) &= \sum_{i=1}^{m+1} \omega_k Q^{(i)}(s_t, a_k) \\ &= \sum_{i=1}^{m+1} \left(\frac{Sum\left(Q^{(i)}(s_t, a_k)\right)}{\sum_{j=1}^{m+1} Sum\left(Q^{(j)}(s_t, a_k)\right)}\right) \\ &\quad \cdot \left(Q^{(i)}(s_t, a_k)\right) \end{aligned} \tag{26}$$

The action a_k with the highest evaluation value $VOWA(a_k)$ will be selected.

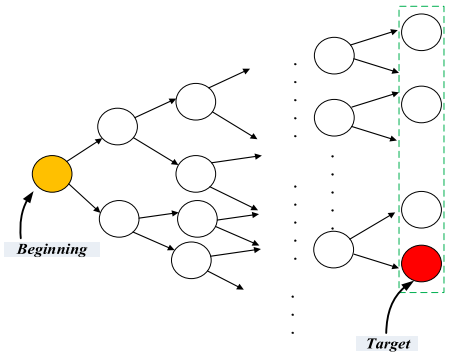


FIGURE 1. The network for robot migration.

C. THE WHOLE ALGORITHM FOR THE ADAPTIVE EXPLORATION STRATEGY WITH A MADM

This study defines a random number ran. If $ran < \varepsilon$, the agent chooses an action randomly or chooses an action using the MADM method. Compared with the classical ε -greedy strategy, the proposed adaptive exploration strategy avoids the trouble of manual tuning and increases the learning performance. The values of attributes change if the learning agent moves to the next state. So, these weights are recalculated transiently. These weights for attributes will be recalculated using the support function when choosing a new action. The adaptive exploration strategy with MADM is detailed below.

VI. EXPERIMENTS AND ANALYSIS

A. EXPERIMENT ON ROBOT MIGRATION

In this experiment, a robot migration experiment was performed to validate the efficiency of the proposed epsilon-greedy with adaptive strategy (adaptive strategy). Competitors include: epsilon-greedy (epsilon-greedy) [33] and Boltzmann probabilistic exploration (Boltzmann exploration) [34]. As shown in Fig.1, the network for robot migration satisfies the structure of the binary tree. In this study, two groups of comparative experiments with 4 layers of the network (nodes' number is 15) and 10 layers of the network (nodes' number is 1023) are set respectively. Each node in the robot migration network represents a state, and there is no transition between the same layers. The experimental parameters for robot migration are shown in Table 1.

There are 15 endpoints on the network with 4 layers. Starting from the starting position, the robot will repeatedly follow the branches to reach the lowest endpoint and will be rewarded for each endpoint. A total of 14 actions can be

$$Q = \begin{pmatrix} Q^{(1)}(s_t, a_1) & Q^{(1)}(s_t, a_2) & \cdots & Q^{(1)}(s_t, a_{n-1}) & Q^{(1)}(s_t, a_n) \\ Q^{(2)}(s_t, a_1) & Q^{(2)}(s_t, a_2) & \cdots & Q^{(2)}(s_t, a_{n-1}) & Q^{(2)}(s_t, a_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q^{(m)}(s_t, a_1) & Q^{(m)}(s_t, a_2) & \cdots & Q^{(m)}(s_t, a_{n-1}) & Q^{(m)}(s_t, a_n) \\ Q(s_t, a_1) & Q(s_t, a_2) & \cdots & Q(s_t, a_{n-1}) & Q(s_t, a_n) \end{pmatrix} \tag{22}$$

Algorithm 2 The Adaptive Exploration Strategy With MADM

1.**Definition:**
2. N:= the number of actions
3. H_c:= the constant threshold
4. $VOWA()$:= Evaluation Value that is given by MADM
5. $P(a_t|s_t)$:= the probability of action a_t at state s_t
6. $E_H(s_t)$:= the entropy at state s_t
7. $\bar{E}_H(s_t)$:= the normalized entropy at state s_t
8. **Initialization:**
9. Randomly Initialize $Q(s, a)$ for state and action $s_t a_t$
11. Initialize $\varepsilon \leftarrow 1$
12. **Repeat** (for each step)
14. Initialize the current state s_t.
15. **Repeat** (for each step of the episode):
16. Initialize a random number $ran \leftarrow rand\ (0, .., 1)$
17. **If** $ran < \varepsilon$ then
18. Randomly choose an action a_t.
19. **else**
20. $a_t \leftarrow \max\{VOWA(s_t, a_1), \ldots, VOWA(s_t, a_n)\}$
21. **End if**
22. Take action a_t, and observe rewards from each sub-task
23. Observe the next state s_{t+1}.
24. Calculate the average value $\hat{R}$ for the standard rewards using the rewards obtained from the sub-tasks.
25. $\left|\Delta Q(s_t, a_t)\right| \leftarrow \left|\hat{R} + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right|$
26. $Q(s_t, a_t) \leftarrow \alpha \Delta Q(s_t, a_t) + Q(s_t, a_t)$
27. Calculate the action probability $P(a_t|s_t)$.
28. Calculate $E_H(s_t)$ and $\bar{E}_H(s_t)$ respectively.
30. **If** $\left(\frac{-|\Delta Q(s_t,a_t)|}{\sum_{a_i \in A} N(a_i,s_t)}\right) \geq \mathrm{H_c}$ **then**
31. $\varepsilon = \left(1 - \bar{E}_H(s_t)\right)$
32. **else**
33. $\varepsilon = \left(1 - \bar{E}_H(s_t)\right) \cdot \left(1 - \exp\left(\frac{-|\Delta Q(s_t,a_t)|}{\sum_{a_i \in A} N(a_i,s_t)}\right)\right)$
34. **End if**
35. $s_t \leftarrow s_{t+1}$
36. **until** s_t is terminal.
37. **End**

selected by the robot, and the Q value for each endpoint is written as $Q_1 \sim Q_{15}$. If the robot reaches the endpoint 15, the reward is $+1000$. If the robot reaches the remaining endpoints, they are not rewarded. The experimental results give the value for $Q_1 \sim Q_{15}$.

In the migration network, the endpoint Q_7 is the only node leading to the endpoint Q_{15}, and the experimental results show that the Q value of Q_{15} will change in the experiment, so this study uses the changes of the Q value of Q_{15} to compare different methods. In this network, the robot is encouraged to choose the endpoint Q_7 because the Q_7 node is the only node leading to Q_{15}. Fig.2 shows the curve of the value for Q_{15}, where three different methods are compared. The adaptive exploration method converges at 12th

TABLE 1. Experimental parameters.

Parameter	Value
Learning rate α	0.3
Discount rate γ	0.9
Exploring rate ε	0.9
Annealing factor	0.9
Maximum temperature parameter	0.1
Minimum temperature parameter	0.01
The constant H_c	0.02

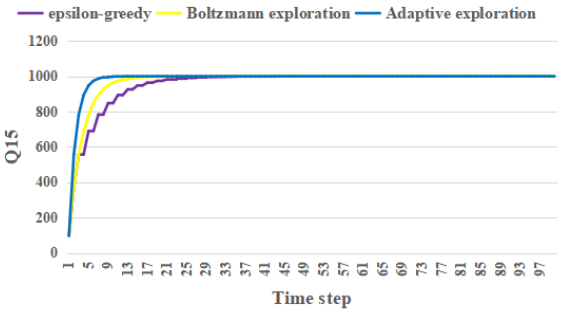


FIGURE 2. The value of Q15 obtained by these three methods.

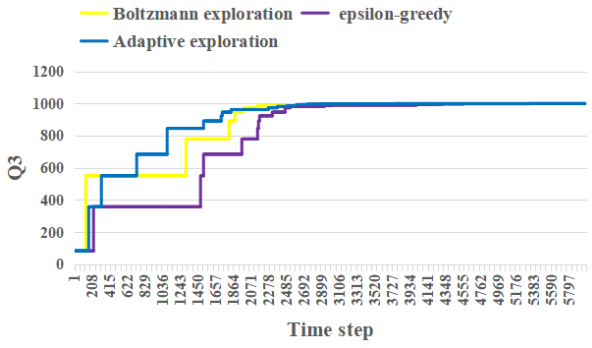


FIGURE 3. The value for Q3 obtained by these three methods.

time steps. The Boltzmann exploration method and epsilon-greedy method converge at 23th time steps and 43th time steps respectively. The experimental results show that the curve of the adaptive exploration strategy converges faster than the other two methods and the adaptive strategy can accelerate the convergence of learning algorithm.

We extend the 4-layers network to 10-layers and then analyze the experiment results. In this experiment, different strategies, the adaptive exploration strategy, epsilon-greedy strategy, and Boltzmann exploration were compared using the value of Q_3. In Fig.3, the adaptive exploration strategy is represented by the blue curve, the Boltzmann exploration policy is represented by the yellow curve and epsilon-greedy strategy is represented by the purple curve. The adaptive exploration method converges at 3746th time steps. The Boltzmann exploration and epsilon-greedy method converge at 3970th time steps and 5337th time steps respectively. Compared with other strategies, the adaptive exploration strategy

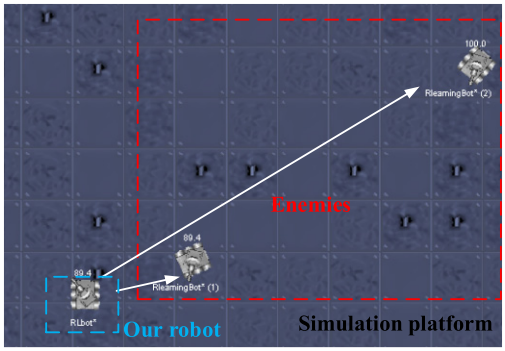


FIGURE 4. Robocode platform.

has the fastest convergence rate, in terms of the value of Q_3. Similar results are shown for a more complex task scenario.

B. EXPERIMENT ON ROBOT CONFRONTATION

Robocode [35], [36] is an open-source platform developed for the multi-robot confrontation. The introduction to the Robocode platform is shown in https://robocode.sourceforge.io/. We use several scenarios with different enemies to test the availability of the proposed method in this experiment.

In the platform, robots will take strategies to attack the enemy and get rewards, which is shown in Fig.4. Both sides have 100 health points at the beginning of each round. It indicates the end of a round if the health point of one side becomes 0. The state space, action space and the reward function for an agent are shown below.

State space: The state space includes: the absolute orientation angle, the relative direction angle, and the distance between the robots. The range of absolute direction angle is $0 \sim 360°$, which is discretized into four states and the relative direction angle is also discretized into four kinds of states. The distance between the robots is divided into 20 discrete values.

Action space: The movement of the robot consists: movement and rotation. Each robot can move to arbitrary directions with arbitrary distances in the ground at each time step. Our robot can attack their opponents with bullets of different energies. The action space includes forward, backward, clockwise rotation, and anticlockwise rotation 4 kinds of different movements.

Reward function: If a robot is hit by bullets or fires bullets hit the enemy, its health point will change. Different states have different health changing rules for the agent and the bullets have a different energy. Different reward functions are designed for different sub-tasks, respectively. Three different sub-tasks are designed for this experiment.

Sub-Task 1: Attack the enemy. The reward signal for subtask 1 is shown in Eq.(27).

$$r = \begin{cases} +3, & \text{Hit the enemy} \\ -3, & \text{Hit by the enemy} \end{cases} \tag{27}$$

Sub-Task 2: Don't get hit by the enemy. The reward signal is negative. $r = -3$ if our robot is hit by the enemy.

TABLE 2. Experimental parameters.

Parameter	Value
Learning rate α	0.1
Discount rate γ	0.9
Exploring rate ε	0.95
The initial value of Decay function	0.95
Maximum temperature parameter	0.1
Minimum temperature parameter	0.01
The constant H_c	0.025

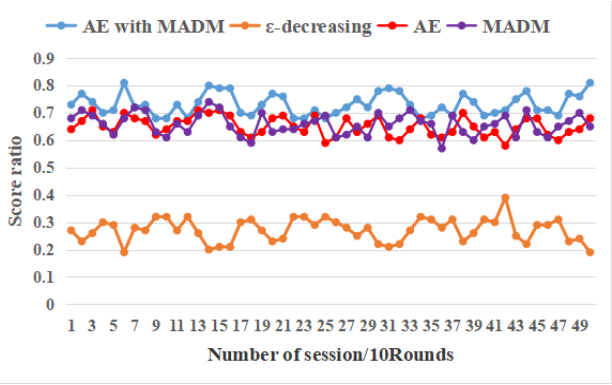


FIGURE 5. Score ratios of the ε-decreasing, the AE method, the MADM method and the proposed AE with MADM.

Sub-Task 3: Do not collide with the enemy. The reward signal is negative $r = -2$ if our robot collides with the enemy.

The adaptive exploration strategy with the MADM (AE with the MADM), the AE method, the MADM method, the ε-greedy strategy with an attenuation threshold (ε-decreasing) [4] and the Softmax-greedy method (softmax-greedy) [34] were compared in this platform to demonstrate the effectiveness of the proposed method. The experimental parameters are set, which is shown in Table 2.

Every 10 rounds are counted as a session, and the session score is the average score that is calculated using the scores collected every 10 rounds. Three sub-tasks were trained by the TD method to converge respectively before this experiment was performed. Firstly, we train a robot "ε-decreasing" using the ε-decreasing method and then use the robot using the proposed AE with the MADM method, the AE method and the MADM method to fight it for 500 rounds respectively. We define the score ratio as an indicator for these experimental results, which is calculated using the ratio of one side's score to the total score of both sides, as shown in Eq.(28).

$$Scoreratio_i = \frac{score_i}{\sum_{i=1}^{M} score_i} \tag{28}$$

where $M = 2$ is the number of individuals participating in the count.

Fig.5 shows the curve of the score ratio for the ε-decreasing method, the AE method, the MADM method and the proposed AE with MADM. The experimental results show that the proposed method has the highest score and the

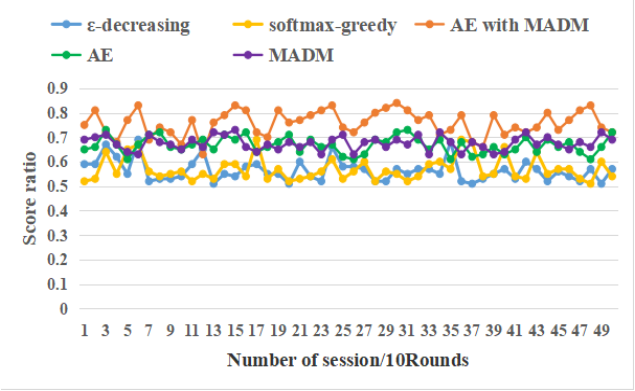


FIGURE 6. Score ratios of the ε-decreasing, the AE method, the MADM method, the proposed AE with the MADM and the softmax-greedy.

ε-decreasing method has the lowest one. The scores of the AE method and the MADM method are lower than those of the proposed method, which shows that the two methods can be effectively combined to improve learning performance.

Then, we use the robot using the AE with the MADM method, the AE method, the MADM method, the robot using the softmax-greedy method and the robot using ε-decreasing to fight 500 rounds with the robot using the ε-greedy strategy respectively.

Fig.6 shows the score ratio for these five methods. Similar to the results shown in Fig.5, the score of the AE with the MADM is higher than that of methods AE and MADM. The experimental results shown in Fig.6 further demonstrate that the combination of the AE and the MADM can improve the performance of the learning algorithm. Besides, compared with the ε-decreasing method and the softmax-greedy, the AE with the MADM method has a higher score ratio, which shows that the action policy using MADM helps the agent get higher scores in confrontation. We decompose the source task into several sub-tasks by a divide-and-conquer method to expand the source of prior experience for the agent. Meanwhile, the MADM method can help the agent select actions more effectively by using prior experience.

In order to demonstrate the low fluctuation for the three methods, a fluctuation curve of score ratio to describe the learning performance for different algorithms. The fluctuation of score ratio is described by the first-order difference of score ratio, that is, $scoreratio_{t+1} - scoreratio_t$, where the ratio of the latter score is used successively minus the ratio of the former score. As shown in Fig. 7-8, the experimental results show that the fluctuation range of the proposed method is smaller than the other two competitors by comparing the fluctuation curves of these methods, which indicates that the proposed method has low fluctuation. A lower fluctuation means that the agent chooses more appropriate actions to maintain a stable decision.

C. EXPERIMENT ON THE REAL WHEELED MOBILE ROBOT (WMR)

In order to test the practicality of the proposed method, this study executed the image-based visual servoing (IBVS)

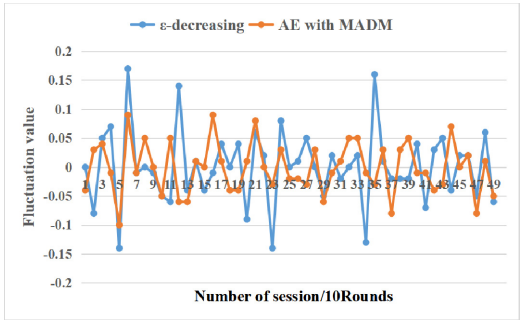


FIGURE 7. The fluctuation value of the ε-decreasing and the AE with the MADM method.

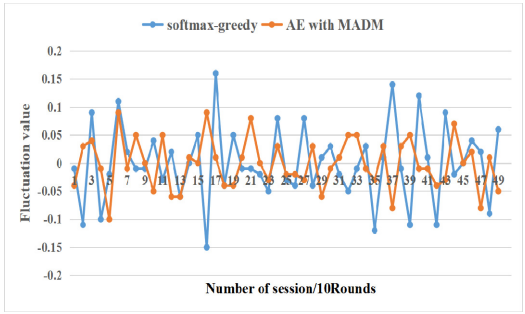


FIGURE 8. The fluctuation value of the softmax-greedy and the AE with the MADM method.

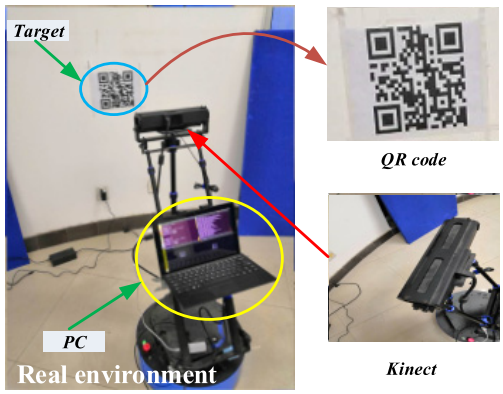


FIGURE 9. The experimental environment for the WMR.

control experiment in the real environment, based on the previous works [10], [37]. The experimental environment is shown in Fig.9. Previous work has shown that Q-learning can improve the performance of the IBVS controller by adjusting the mixture parameter for the IBVS controller [10], [37]. We carried out experiments on the basis of this work and the four vertex coordinates of the Quick Response (QR) code are used as the feature points in IBVS [10]. Firstly, we train the WMR in the simulation environment, and then the prior knowledge (the Q-table) that is learned in simulation is directly transferred to the WMR before the real-world experiment, in order to reduce the training time of real-world experiments.

Three different methods were compared: the proposed method (AE with the MADM), ε-greedy strategy

TABLE 3. Experimental parameters.

Parameter	Value
Learning rate α	0.15
Discount rate γ	0.90
Exploring rate ε	0.95
Annealing factor	0.90
Maximum temperature parameter	0.10
Minimum temperature parameter	0.01
The gain for visual servoing λ	1.5

(ε-greedy) [33] and the softmax strategy with simulated annealing algorithm (Softmax-SA) [37]. For the real-world environment, the parameters for this experiment are set as shown in Table 3. For the RL model in this experiment, based on the previous work [10], [37], we decompose the source task into three sub-tasks: not losing any feature, reaching the desired position and achieving the desired position faster.

Sub-Task 1: The reward +100 is given when the WMR reaches its target position. In other cases, however, a few negative rewards are given. The reward function is given by,

$$reward = \begin{cases} +100, & \text{Reach the target position} \\ -2, & \text{Others} \end{cases} \tag{29}$$

Sub-Task 2: If the WMR is deemed to be losing the target and is given a bad reward −100. In other cases, a few positive rewards are given. The reward function is given by,

$$reward = \begin{cases} -100, & \text{Missing the target features} \\ +2, & \text{Others} \end{cases} \tag{30}$$

Sub-Task 3: The reward is given according to the distance between the WMR and the desired position, which drives the WMR to reach the target position faster. The reward function is given by,

$$reward = -100\left(\sum_{j=1}^{N}\left|F_j^c - F_j^*\right| / N\sqrt{col^2 + row^2}\right) \tag{31}$$

where N is the number of image features used in the IBVS system. F_j^c is the current feature and F_j^* is the desired feature. col and row are the length and width of the obtained image plane, respectively.

Since the average errors for four feature points are similar, only the feature errors for two diagonal feature points are shown. Each method was tested 50 times, and the average value for the 50 tests was selected as the experimental results. The experimental results are shown in Fig.10. Fig.10 shows the convergence curve for the feature error. From the experimental results, these three methods converge at last, but the convergence rate is different. The AE with the MADM method converges at 148th time steps, the Softmax-SA method converges at 181th time steps and the ε-greedy method converges at 198th time steps. The experimental results show that compared with the other two methods, the proposed method has the fastest convergence rate. The results on the real-robot show that the adaptive exploration and the

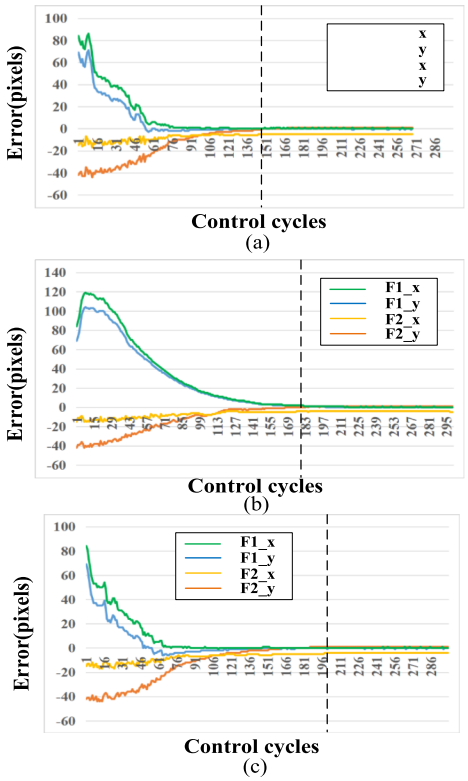


FIGURE 10. Comparison of the feature error for the three methods. (a) The AE with the MADM method. (b) The Softmax-SA method. (c) The ε-greedy method.

MADM can accelerate the convergence rate of the learning algorithm. Reducing training time means that the risk of damaging the real-robot will be reduced.

VII. A FUTURE IMPROVEMENT FOR ADAPTIVE EXPLORATION STRATEGY USING TRANSFER LEARNING

To further investigate the improvement of learning performance, this study uses the transfer learning extends the learning model to different scenarios to improve the generalization of the proposed AE with MADM. This study compares the AE with the MADM method using transfer learning (AE-MADM with TF) with the MADM method without transfer learning (AE-MADM without TF). The contrast experiment was divided into two groups. The first group used AE-MADM with TF and AE-MADM without TF to fight with two formations of tank robots that are trained by ε-greedy strategy respectively. The second group used AE-MADM with TF and AE-MADM without TF to fight with four formations of tank robots that are trained by ε-greedy strategy respectively. The robots were set in different formations to ensure the robots in the same team will not attack each other. The robot that is trained using the ε-greedy strategy and the robot that is trained using the AE-MADM method will fight for 500 rounds before the beginning of the experiment, so the robot "AE-MADM" has some prior experience. The direct transfer strategy [38] is performed if

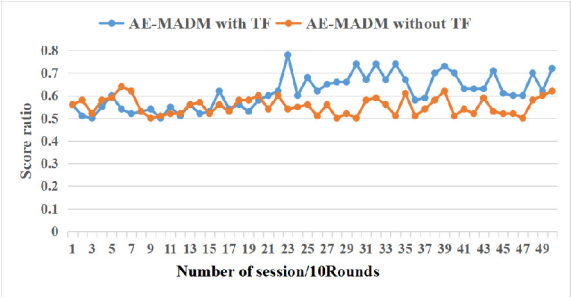


FIGURE 11. The experimental results for the first group.

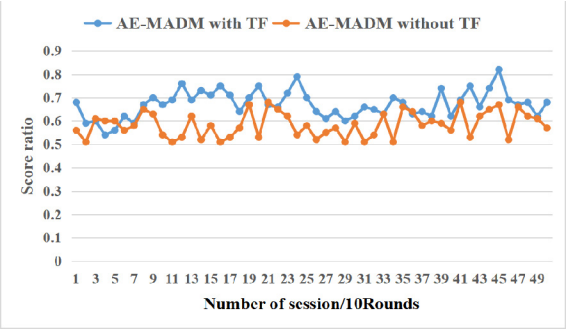


FIGURE 12. The experimental results for the second group.

agents have the same state and action space in the different tasks, which is given by,

$$\begin{cases} \rho S : S_{past} \rightarrow S_{current} \\ \rho A : A_{past} \rightarrow A_{current} \end{cases} \tag{32}$$

An intermediate task gives the agent more prior experience to achieve the target task if the target task is far away from the source task. In this experiment, the second experiment is regarded as the target task, then the first experiment can be regarded as the intermediate task.

Fig.11 and Fig.12 show the experimental results for the first and second experiments, respectively. In the first experiment, we used a formation as the enemy, which consists of two robots trained by ε-greedy strategy. The score ratio of AE-MADM with the TF method and AE-MADM without the TF method remained between 0.5 and 0.81. However, the experimental results show that the score ratio of AE-MADM with the TF method is still higher than that of AE-MADM without the TF method because agents learn from prior experience using transfer learning, so the score is higher. With the number of rounds increases, the score ratio of AE-MADM with the TF method remains around 0.7, while the score ratio of AE-MADM without the TF method remains around 0.6.

Compared with the first experiment, the second experiment is more difficult. The learning experience that is gained in the first experiment is transferred to the second experiment using the direct transfer strategy. In the first 220 rounds, the AE-MADM with the TF method does not perform better than AE-MADM without the TF method. With the increase of rounds, the former scores gradually higher than the latter, which shows that the agent can not only exploit the past experience but also gain new knowledge. The experimental results show that transfer learning extends the proposed method to more difficult task scenarios.

VIII. CONCLUSION

The dilemma of exploration and exploitation in RL systems is a challenging problem. In this work, to address the dilemma of exploration and exploitation, an adaptive exploration strategy with multi-attribute decision-making is proposed. The probability of exploration is determined by an adaptive exploration strategy, which uses the information entropy instead of the experience of manual tuning. Meanwhile, we investigate how to expand the source of prior experience from the structure of the task itself, and how to integrate these multi-source prior experience. Firstly, the source task is decomposed into several sub-tasks, and these sub-tasks are trained by the same TD method separately. Because each sub-task has a different reward function, a reward standardization method is proposed. The average value of standardized rewards for each sub-task is used as a reward for the complex task. Compared with the conventional method that the agent learns directly in the environment, the training method running several learners in parallel can accelerate the learning rate. The transfer learning method extends the proposed learning model to more difficult tasks. We executed several experiments to demonstrate the effectiveness of the proposed method. The experimental results show that the proposed method outperforms the conventional methods in terms of convergence rate and learning performance.

In the future, we will try to extend the proposed method to the RL systems in high-dimensional space and this learning method needs to be framed in the existing works, such as hierarchical RL [39] and curriculum learning [40], [41]. Since some rough prior experience exists in the previous learning tasks, advanced knowledge transfer methods and the Domain adaptation might lead to more efficient learning performance. Previous works have focused on the RL methods to address the fault diagnosis and fault tolerant control [42]–[44]. So, we will investigate the application of the proposed method to the fault diagnosis. In addition, integrating the possibility of applying multi-attribute decision-making in the deep reinforcement learning system is also worthy of study.

REFERENCES

[1] T. Mannucci, E.-J. Van Kampen, C. De Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1069–1081, Apr. 2018.

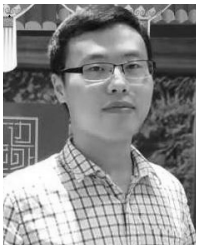
[2] H. Li, D. Liu, and D. Wang, "Manifold regularized reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 932–943, Apr. 2018.

[3] K.-S. Hwang, W.-C. Jiang, and Y.-J. Chen, "Pheromone-based planning strategies in Dyna-Q learning," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 424–435, Apr. 2017.

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," 2016, *arXiv:1602.01783*. [Online]. Available: http://arxiv.org/abs/1602.01783

[5] K. Shao, Y. Zhu, and D. Zhao, "StarCraft micromanagement with reinforcement learning and curriculum transfer learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 1, pp. 73–84, Feb. 2019.
[6] S. Haobin, X. Meng, and H. Kao-Shing, "Behavior fusion for deep reinforcement learning," *ISA Trans.*, to be published, doi: 10.1016/j.isatra.2019.08.054.
[7] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002.
[8] I. J. Sledge and C. J. Príncipe, "Balancing exploration and exploitation in reinforcement learning using a value of information criterion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Jun. 2017, pp. 2816–2820.
[9] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 5, no. 9, p. 1054, Sep. 1998.
[10] H. Shi, X. Li, and K. S. Hwang, "Decoupled visual servoing with fuzzy Q-learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 241–252, Oct. 2016.
[11] C. Craye, D. Filliat, and J.-F. Goudou, "RL-IAC: An exploration policy for online saliency learning on an autonomous mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4877–4884.
[12] K. Iwata, "Extending the peak bandwidth of parameters for softmax selection in reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1865–1877, Aug. 2017.
[13] T. Goto, N. Homma, M. Yoshizawa, and K. Abe, "A phased reinforcement learning algorithm for complex control problems," *Artif. Life Robot.*, vol. 11, no. 2, pp. 190–196, Jul. 2007.
[14] H. Shi, G. Sun, Y. Wang, and K.-S. Hwang, "Adaptive image-based visual servoing with temporary loss of the visual signal," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 1956–1965, Apr. 2019.
[15] D. Pathak, P. Agrawal, and A. A. Efros, "Curiosity-driven exploration by self-supervised prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Aug. 2017, pp. 16–17.
[16] I. Osband, C. Blundell, and A. Pritzel, "Deep exploration via bootstrapped DQN," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 4026–4034.
[17] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," 2017, *arXiv:1706.01905*. [Online]. Available: http://arxiv.org/abs/1706.01905
[18] H. Garg and R. Arora, "A nonlinear-programming methodology for multi-attribute decision-making problem with interval-valued intuitionistic fuzzy soft sets information," *Appl. Intell.*, vol. 48, no. 8, pp. 2031–2046, Aug. 2018.
[19] S.-M. Chen and Z.-C. Huang, "Multiattribute decision making based on interval-valued intuitionistic fuzzy values and linear programming methodology," *Inf. Sci.*, vol. 381, pp. 341–351, Mar. 2017.
[20] Z. S. Xu and Q. L. Da, "The ordered weighted geometric averaging operators," *Int. J. Intell. Syst.*, vol. 17, no. 7, pp. 709–716, Jul. 2002.
[21] K.-S. Hwang, Y.-J. Chen, and C.-J. Wu, "Fusion of multiple behaviors using layered reinforcement learning," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 4, pp. 999–1004, Jul. 2012.
[22] J. L. Lin, K. S. Hwang, and Y. L. Wang, "A simple scheme for formation control based on weighted behavior learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1033–1044, Oct. 2013.
[23] J. A. Núñez, P. M. Cincotta, and F. C. Wachlin, "Information entropy," *Celestial Mech. Dyn. Astron.*, vol. 64, nos. 1–2, pp. 43–53, Mar. 1996.
[24] P. Dayan, "The convergence of TD(λ) for general?" *Mach. Learn.*, vol. 8, nos. 3–4, pp. 341–362, May 1992.
[25] G. Zhu, Y. Li, and P. Wen, "Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 6, pp. 1813–1821, Nov. 2014.
[26] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.
[27] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.
[28] R. R. Yager and D. P. Filev, "Induced ordered weighted averaging operators," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 141–150, Apr. 1999.
[29] Z. Liu, J. Wang, G. Liu, and L. Zhang, "Discriminative low-rank preserving projection for dimensionality reduction," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105768.
[30] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, "Structured optimal graph based sparse feature extraction for semi-supervised learning," *Signal Process.*, vol. 170, May 2020, Art. no. 107456.
[31] Z. Wen and Y. Jia, "Predicting links based on knowledge dissemination in complex network," *Phys. A, Stat. Mech. Appl.*, vol. 471, pp. 561–568, Apr. 2017.
[32] W. J. Martin, "KELEA (kinetic energy limiting electrostatic attraction) offers an alternative explanation to existing concepts regarding wave-particle duality, cold fusion and superconductivity," *J. Mod. Phys.*, vol. 07, no. 15, pp. 1995–2007, Jan. 2016.
[33] V. Narayanan and S. Jagannathan, "Event-triggered distributed control of nonlinear interconnected systems using online reinforcement learning with exploration," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2510–2519, Sep. 2018.
[34] A. Sangari and W. Sethares, "Convergence analysis of two loss functions in soft-max regression," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1280–1288, Mar. 2016.
[35] R. Harper, "Evolving robocode tanks for Evo robocode," *Genetic Program. Evolvable Mach.*, vol. 15, no. 4, pp. 403–431, Dec. 2014.
[36] B. G. Woolley and G. L. Peterson, "Unified behavior framework for reactive robot control," *J. Intell. Robot. Syst.*, vol. 55, nos. 2–3, pp. 155–176, Jul. 2009.
[37] H. Shi, M. Xu, and K.-S. Hwang, "A fuzzy adaptive approach to decoupled visual servoing for a wheeled mobile robot," *IEEE Trans. Fuzzy Syst.*, to be published, doi: 10.1109/tfuzz.2019.2931219.
[38] W. Hao and G. Yang, "Transfer of reinforcement learning: The state of the art," *Acta Electron. Sinica*, vol. 36, no. 1, pp. 39–43, Jan. 2008.
[39] J. Morimoto and K. Doya, "Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning," *Robot. Auto. Syst.*, vol. 36, no. 1, pp. 37–51, Jul. 2001.
[40] J. Xin, Z. Cui, P. Zhao, and T. He, "Active transfer learning of matching query results across multiple sources," *Frontiers Comput. Sci.*, vol. 9, no. 4, pp. 595–607, Aug. 2015.
[41] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2216–2226, Jun. 2018.
[42] Y. Wu, B. Jiang, and N. Lu, "A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2108–2118, Oct. 2019.
[43] Y. Wu, B. Jiang, and Y. Wang, "Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains," *ISA Trans.*, to be published, doi: 10.1016/j.isatra.2019.09.020.
[44] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/tnnls.2019.2935608.

CHUNYANG HU received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. He served as a Visiting Scholar with the University of Massachusetts Lowell, in 2017. He is currently an Associate Professor with the School of Computer Engineering, Hubei University of Arts and Science, Xiangyang, China. His research interests include big data processing, machine learning, software-defined networks, and blockchain. He is a member of the China Computer Federation.



MENG XU was born in Anhui, China. He received the B.S. and M.S. degrees in computer science and technology from Northwestern Polytechnical University, China. He has won two national scholarships for postgraduate courses in China. Besides, he was awarded the Best Presentation Award from the IEEE IFuzzy 2018. He has published over ten journal articles and conference papers, including articles published on the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. His research interests include mobile robots, intelligent control, and machine learning.