IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 3, 2020, accepted June 4, 2020, date of publication June 9, 2020, date of current version June 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001143

An Experience Aggregative Reinforcement Learning With Multi-Attribute Decision-Making for Obstacle Avoidance of Wheeled Mobile Robot

CHUNYANG HU[1], BIN NING[1], MENG XU[2], AND QIONG GU[1]
[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 authors: Bin Ning (ningbin2000@163.com) and Meng Xu (menghsu@mail.nwpu.edu.cn)

This work was supported in part by the Ministry of Education Science and Technology Development Center, University, Industry, Education, and Research Innovation Fund-New Generation Information Technology Innovation under Project 2018A02028, in part by the Natural Science Foundation of Hubei Province under Grant 2013CFC026, in part by the International Science and Technology Cooperation Program of Hubei Province under Grant 2017AHB060, and in part by the Key Programs of Science and Technology Funds of the Xiangyang city under Grant 2020ABA00778.

ABSTRACT A variety of reinforcement learning (RL) methods are developed to achieve the motion control for the robotic systems, which has been a hot issue. However, the performance of the conventional RL methods often encounters a bottleneck, because the robots have difficulty in choosing an appropriate action in the control task due to the exploration-exploitation dilemma. To address this problem and improve the learning performance, this work introduces an experience aggregative reinforcement learning method with a Multi-Attribute Decision-Making (MADM) to achieve the real-time obstacle avoidance of wheeled mobile robot (WMR). The proposed method employs an experience aggregation method to cluster experiential samples and it can achieve more effective experience storage. Moreover, to achieve the effective action selection using the prior experience, an action selection policy based on a Multi-Attribute Decision-Making is proposed. Inspired by the hierarchical decision-making, this work decomposes the original obstacle avoidance task into two sub-tasks using a divide-and-conquer approach. Each sub-task is trained individually by a double Q-learning using a simple reward function. Each sub-task learns an action policy, which enables the sub-task to selects an appropriate action to achieve a single goal. The standardized rewards of sub-tasks are calculated when fusing these sub-tasks to eliminate differences in rewards for sub-tasks. Then, the proposed method integrates the prior experience of three trained sub-tasks via an action policy based on a MADM to complete the source task. Simulation results show that the proposed method outperforms competitors.

INDEX TERMS Reinforcement learning, experience aggregation, multi-attribute decision-making, obstacle avoidance, wheeled mobile robot.

I. INTRODUCTION

A. WHEELED MOBILE ROBOTS

Wheeled mobile robots are widely used in the complex tasks for civil or industrial occasions due to its flexibility and the capacity of faster motion than traditional industrial robots [1], [2]. Obstacle avoidance is an important issue for the practical use of wheeled mobile robots [3], [4]. Obstacle avoidance behavior allows a WMR to move between its current and target positions without any collision within the task environment, an effective motion planning method is a treatment. An obstacle avoidance scheme generates a collision-free path to achieve a defined robotic task, such as navigation.

B. REINFORCEMENT LEARNING FOR ROBOTICS

Reinforcement learning is a type of machine learning method for intelligent robots, which allows the robots to automatically discover the environment via a trial-and-error approach [5]. In the learning process, the robot constantly interacts with the task environment to get the optimal policy that maximizes

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

the long-term reward. Receiving feedback from the environment is a key approach to acquire learning experience that optimizes the strategy when performing a task. The strategy, also known as the action policy, chooses an appropriate action based on the current state perceived by the learning agent.

Based on the Markov feature of the physical environment, the definition of reinforcement learning can be framed in an infinite Markov Decision Process (MDP). In the RL domain, three means are commonly used to handle complex tasks. A dynamic programming approach is developed to address the task by driving the robot to learn the relevant MDP model [6]. These methods are called model-based approaches. Another approach is the model-free method, such as the Q-learning, which uses the value function or policy function to generate the actions. The last approach is the Dyna method, which combines the advantages of the former two methods [7].

C. THE EXPLORATION-EXPLOITATION DILEMMA IN OBSTACLE AVOIDANCE TASK

It is well known that the RL based robotic systems inevitably encounter the inherent exploration-exploitation dilemma of reinforcement learning [8], [9]. Robots explore the unknown environment via an exploration strategy that guides it to collect new learning experiences. To get the greatest discounted cumulative reward, an appropriate method is needed to determine an action that balances the opportunity to exploit prior experience and to collect brand-new experience samples. It is inefficient that the robots explore for a long period and even the experience samples are not well collected. Moreover, it is crucial to improve the storage efficiency of the experience samples in the high-dimensional spaces. Experience aggregation is a good solution [10]. Especially, high computational complexity is a choking problem when robots utilizing experience samples in a continuous large-scale space.

Previous researchers working in the obstacle avoidance tasks utilize the probabilistic exploration method to balance the exploration-exploitation, such as the epsilon-greedy [11]. A robot is guided by the epsilon-greedy strategy to finish the exploration with the fixed probability of ε and to achieve the exploitation with the prior experience in other $1-\varepsilon$. However, the training time for the epsilon-greedy strategy is proportional to the scale of state space and action space [12], [13]. Another common method for exploration-exploitation is the Boltzmann exploration strategy [14]–[16]. The Boltzmann exploration strategy guides a robot to select an action with a probability depending on the value function and a temperature function restrains the confusion of action selection. With the increase of the temperature, the uniformity of the selection of action is also increasing. Conversely, when the temperature is low, the action with the maximum value function is almost one selected.

Recently, exploration guided by an intrinsic motivation for RL based robot task has extensively investigated by researchers [17]. Jiahui Zhang et.al proposed an exploration strategy using curiosity to achieve obstacle avoidance and end-to-end navigation in an unfamiliar environment [18]. The intrinsic motivation strategy does not require accurate sensors to construct a map of the environment. In fact, not only the intrinsic motivation, Thompson sampling, and parameter space exploration have been used as a mechanism by scholars to achieve obstacle avoidance [19], [20]. An effective trade-off between exploration-exploitation is an important way to improve the control performance of robots.

D. MULTI-ATTRIBUTE DECISION-MAKING

It is a common issue in real life that decision-makers make a wise decision when faced with multiple factors and it is a case of multi-attribute decision-making [21]. These factors can be regarded as evaluation criteria to evaluate a scheme and the decision-makers usually adopt the best scheme with the highest evaluation. Considering the numerical values of the related factors, the decision-makers need to rank multiple attributes to get a decision-making result using the prior experience [22]. For the multi-attribute decision-making problem, a common and effective method is the ordered weighted averaging (OWA) operator [23]. Previous works employ a MADM method to improve the performance of reinforcement learning systems [24]. In practice, the learned policy of multiple learners serves as the prior experience and the action space acts as the scheme set. A potential solution provided by the MADM for the obstacle avoidance tasks to use the prior experience gathered from multiple learners.

E. RESEARCH GAPS

In the obstacle avoidance tasks, for the dilemma of exploration and exploitation, previous works develop a variety of exploration strategies without considering the structure of a task [25], [26]. For a specific task, conventional methods show weakness in the efficiency of collecting learning experience compared with the parallel training method with multi-learners, such as Asynchronous Advantage Actor-Critic method [27], [28]. Prior experience can be utilized to inform the robot to conduct effective exploitation that selects an appropriate action in scenarios. Therefore, it is important to investigate how to determine an approach to the exploration-exploitation dilemma, in terms of the efficiency of the collection and utilization of prior knowledge.

In the process of exploration and exploitation, for the challenge of storing the collected experience samples, conventional methods are inefficient in high-dimensional space. Even, most previous works have not focused on how to store prior experience efficiently, so it is very difficult to utilize a large number of prior experiences. Meanwhile, efficient experience storage can also improve the efficiency of exploitation.

F. RESEARCH METHODS

In this study, to fill the gaps mentioned above for exploration-exploitation in obstacle avoidance tasks, three ways are used as follows. Expanding prior experience for exploitation using a divide-conquer approach, storing the prior experience using

experience aggregation, and developing an action policy utilizing the prior experience of trained sub-tasks.

Firstly, to achieve the first way, this study introduces an effective approach to exploration and this method collects experience samples in a divide-conquer way. The original task is decomposed into several sub-tasks and these sub-tasks are trained by a double Q-learning [29]. The source of prior experience is expanded by multi-learner parallel learning, but the original method does not. Secondly, an experience aggregation method is employed to store the learning experience instead of the tabular method. The experience aggregation method classifies similar experience samples and simultaneously builds a decision-tree structure [30] to store the experience of these sub-tasks. Thirdly, to achieve the third way, this study developed an action policy using the MADM and experience aggregation, which regards the action space as the scheme set and regards the value functions from trained sub-tasks as attributes for these schemes. The MADM method is utilized to compute the evaluation value for each scheme, instantaneously. Then, the robot selects the action corresponding to a maximum evaluation value. In the fusion process of these sub-tasks, the standardization method for rewards of sub-tasks is proposed and the average value for standardized rewards is used to act as the current reward for the original task.

G. CONTRIBUTIONS IN THIS WORK

The main contributions are as follows.

1) For the original obstacle avoidance task, a divide-and-conquer approach is employed to decompose the source task into several sub-tasks. Then, a double Q-learning is used to train these sub-tasks respectively. Multi-learner parallel training can expand the source of prior experience for exploration.
2) An experience aggregation method is developed to store prior experience using a decision-tree. This method can cluster experiential samples depending on the similarity.
3) To address the exploration-exploitation in the obstacle avoidance task, this work proposes an action policy with experience aggregation and a MADM. This action policy determines an action depending on the learned experience of these sub-tasks.

H. ARTICLE STRUCTURE

The organization of this work is as follows. Section II presents the background and describes the classical Q-learning, the epsilon-greedy strategy, and Multi-Attribute Decision-Making. Section III presents a reinforcement learning model for obstacle avoidance and a training method for sub-tasks using a double Q-learning. Section IV presents an experience aggregation method and an action policy with Multi-Attribute Decision-Making. A standardized method for rewards obtained by each learner is introduced in this section. Section V presents the whole framework for the proposed method. Experiments are conducted to demonstrate the effectiveness of the proposed methods in Section VI. The experiment includes three different simulations. Conclusions are drawn in the last section.

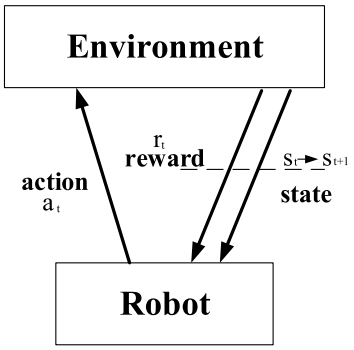

FIGURE 1. The framework for reinforcement learning.

II. BACKGROUND

A. Q-LEARNING

Reinforcement learning is suitable for a type of robot control problem in the statistical and control field, and it can learn how to complete a complex behavior by the feedback via repeated attempts [31]. An MDP quaternions model is combined with the Bellman equation to calculate the value function $V(s_t)$ to achieve the optimization strategy for the discrete control problem. The framework for reinforcement learning is shown in **Fig.1.**

In Fig.1, the learning agent selects an action a_t in the state s_t, and the current state is transited to the state s_{t+1}. Then, a reward r_t is got. The updating law for value function is given by,

$$V(s_t) = (1-\alpha)\, V(s_t) + \alpha\, (r_t + \gamma V(s_{t+1}))\,. \tag{1}$$

where α is the learning rate, $\alpha \in (0, 1]$, and γ is a discount factor.

Q-Learning is one of the most popular reinforcement learning algorithms [32]. Q-Learning uses state-action pair as the component of the value function. The updating law for Q-learning is given by,

$$Q(s_t, a_t) = (1-\alpha)\, Q(s_t, a_t) + \alpha\, [r_t + \gamma \max Q(s_{t+1}, a_{t+1})]\,. \tag{2}$$

B. ORDERED WEIGHTED AVERAGING OPERATOR

In real life, a decision scheme often involves multiple attributes, and these attributes directly affect the quality of decision results. Therefore, the evaluation values for schemes are calculated via an aggregation operator derived from the multi-attribute decision-making theory and the reasonable measure of attributes improves the scientificity and effectiveness of a decision-making result. Ordered weighted averaging (OWA) operator determines the weights for each attribute according to their importance, which is an effective and easy to develop among all aggregation operators.

$C = (c_1, c_2, \ldots, c_n)$ is a set of original data and $\hat{C} = (\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n)$ is the ordered sequence of sequence C sorted

from large to small. Eq.(3) gives the definition for the OWA operator.

$$O^{OWA}(c_1, c_2, \ldots, c_n) = \sum_{j=1}^{n} \hat{c}_j \omega_j \tag{3}$$

where the weight vector for each attribute is $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$ and it satisfies $\sum_{j=1}^{n} \omega_j = 1, \omega_j \in [0, 1]$.

C. THE EPSILON-GREEDY STRATEGY

Exploration gathers the learning experience from the environment and Exploitation selects the action with the most reward. The RL agent often encounters the exploration-exploitation dilemma. A renowned strategy in RL systems is epsilon-greedy, which ensures convergence to an optimal policy in the limit. For the epsilon-greedy strategy, the robot chooses available actions greedily with probability 1-ε and chooses actions randomly with probability ε. The action policy using epsilon-greedy is shown in Eq.(4).

$$a_t = \begin{cases} \text{Select action with MaxQ}, & \text{random}() \geq \varepsilon \\ \text{Randomly select action}, & \text{random}() < \varepsilon \end{cases} \tag{4}$$

where $\varepsilon \in (0, 1)$ is a fixed value which is determined with probability distribution in the actual task.

III. THE TRAINING METHOD FOR SUB-TASKS USING THE DOUBLE Q-LEARNING

In the training process, sub-tasks are trained by a double Q-learning separately. A double Q-learning is used to tackle the problem of traditional over-estimation in reinforcement learning systems involving the maximization operation and it is often suggested that this algorithm outperforms the traditional RL methods [33]. For each sub-task using a double Q-learning, an alternative double estimator is used to calculate the estimate for the maximum value function and a limit double Q-learning can converge to the optimal policy.

A. OBSTACLE DETECTION

Firstly, this work introduces a method to achieve the obstacle detection. To avoid obstacles, we need to calculate the position of obstacles in the field of the RGB-D camera's field of view. The WMR chooses an appropriate action according to the perceived position of obstacles to avoid obstacles. In this work, the bottom center point of the obstacle is selected as the feature point for the obstacle and a WMR perceives a time-varying horizontal position of the feature point if the WMR continues to move. In Fig.2, point A is the center of the camera and the distance between the RGB-D camera and the obstacle is represented by AB. β_1 is the robot's field of vision starting from the horizontal line AB. The point C is the feature point for the obstacle. The angle between line AB and line AC is β. Then, we need to calculate the ratio δ of point C to the height of the image.

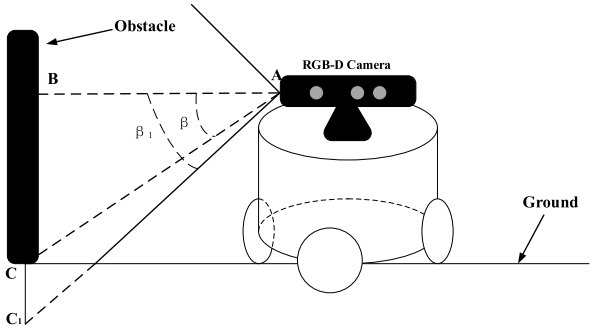

FIGURE 2. The positional relation for a WMR and the obstacle.

According to the geometric relationship, the ratio δ can be calculated by Eq.(5).

$$\delta = \frac{1}{2}\left(\frac{AB\tan(\beta_1)+AB\tan(\beta)}{AB\tan(\beta_1)}\right) = \frac{1}{2}\left(\frac{\tan(\beta)}{\tan(\beta_1)}+1\right) \tag{5}$$

where the ratio δ is determined in an actual environment.

B. THE RL MODEL FOR OBSTACLE AVOIDANCE OF WHEELED MOBILE ROBOT

In the work, a double Q-learning stores two Q functions. State space and action space for the obstacle avoidance task are shown in below.

State space includes three parts. The first is the distance between the MWR and the target, which is discretized into N_D states. The second is the position of the feature for an obstacle in the image, which is discretized into N_O states. The third is the angle between the moving direction of the WMR and the line connecting the obstacle and the WMR, which is discretized into N_A states.

Action space includes five actions, which are forward, backward, stop, left forward, and right forward.

We focus on two factors to define the reward function, the distance D from the WMR to the target and the angle Rad between the moving direction of the WMR and the line connecting the obstacle and the WMR. Three sub-tasks are designed for the obstacle avoidance task, which is reaching the target position faster, not hitting any obstacles, and not losing the target. Three simple reward functions are defined for this task.

Sub-Task 1: Reaching the target position faster. The reward signal for sub-task 1 is shown in Eq.(6).

$$r = -\left(k \times D + \sum_{i=1}^{N} \frac{|n-p| \times C_n}{p} - C_{noise}\right) \tag{6}$$

where n is the number of sensors. C_n is the output of the sensor n. The sensor noise is C_{noise}. p and k are constant, which are determined by the actual situation.

Sub-Task 2: Not hitting any obstacles. The reward signal for sub-task 2 is shown in Eq.(7), as shown at the bottom of the next page, where f is a constant.

Sub-Task 3: Not losing the target. The reward signal for sub-task 3 is shown in Eq.(8).

$$r = \begin{cases} +2, & \textit{if} \text{ don't lose the target} \\ \text{-10}, & \text{if lose the target} \end{cases} \tag{8}$$

The greatest negative reward is given to the WMR if the WMR loses the target. Conversely, a certain positive reward is given to the agent to avoid too sparse rewards. If the WMR loses the target or hits the obstacle, this episode is terminated and the next episode begins. If the WMR reaches the target position, this episode is terminated and the next episode begins.

C. THE TRAINING ALGORITHM FOR EACH SUB-TASK

It is hard to tackle a multi-objective decision-making problem directly. A divide and conquer approach is a solution to address this type of problems including obstacle avoidance [34]. Inspired by the idea of hierarchical decision-making, this work decomposes the source task into several sub-tasks [35]. For a source obstacle avoidance task, the prior experience of the trained sub-tasks is assembled to form an ensemble policy, which guides the agent to select an action for this complicated task. Each sub-task is trained by a double Q-learning.

We set two estimators, which are $Q_1(s_t, a_t)$ and $Q_2(s_t, a_t)$. In the learning process, we use $Q_1(s_t, a_t)$ to select an action using the maximization operation $\arg\max_a Q_1(s_t, a)$ and use $Q_2(s_t, a_t)$ to evaluate the value function $Q_2(s_{t+1}, \arg\max_a Q_1(s_t, a))$ for the selected action. Then, in the next step, we switch roles for the two Q-value functions. Only one value function is updated in each step. The updating law for Q-value functions by Double Q-learning is given by,

$$\begin{cases} Q_1(s_t, a_t) = (1-\alpha)\, Q_1(s_t, a_t) \\ \quad +\alpha\left[r_t + \gamma \max Q_2(s_{t+1}, \arg\max_a Q_1(s_t, a))\right] \\ Q_2(s_t, a_t) = (1-\alpha)\, Q_2(s_t, a_t) \\ \quad +\alpha\left[r_t + \gamma \max Q_1(s_{t+1}, \arg\max_a Q_2(s_t, a))\right] \end{cases} \tag{9}$$

Two estimators, $Q_1(s_t, a_t)$ and $Q_2(s_t, a_t)$, are treated completely symmetrically. The training algorithm for each sub-task using the double Q-learning is given by Algorithm 1.

Algorithm 1 Training Algorithm Using Double-Q Learning

Definition
s_t := Current state
a_t := Current action
s_{t+1} := The next state
r_t := Current reward
$\alpha_{:=}$ Learning rate
γ := Discount factor
Initialization
Initialize Q table $Q_1(s_t, a_t)$ and $Q_2(s_t, a_t)$ arbitrarily
  Repeat (for each step)
    Initial the current state s_0
    $t \leftarrow 0$;
  Repeat(for each step of the episode):

  Select an action a_t using policy derived from $Q_1(s_t, a_t)$ and $Q_2(s_t, a_t)$ (e.g.,epsilon-greedy)
  Take action a_t and observe s_{t+1}, and reward r_t.
  Updating the Q-value functions with a certain probability ξ.
  If *rand* $< \xi$:

$$Q_1(s_t, a_t) = (1-\alpha)\, Q_1(s_t, a_t) + \alpha\left[r_t + \gamma \max Q_2(s_{t+1}, \arg\max_a Q_1(s_t, a))\right];$$

  Else:

$$Q_2(s_t, a_t) = (1-\alpha)\, Q_2(s_t, a_t) + \alpha\left[r_t + \gamma \max Q_1(s_{t+1}, \arg\max_a Q_2(s_t, a))\right];$$

    $s_t \leftarrow s_{t+1}$
    t++
  until S_t is terminal.

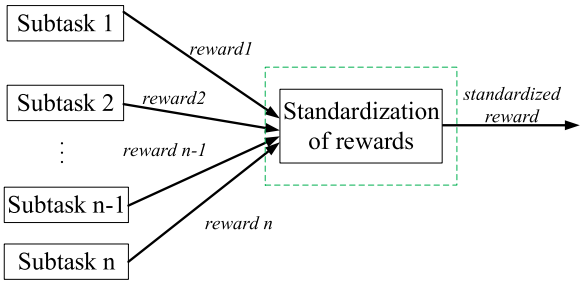

FIGURE 3. Standardization of rewards.

IV. THE ACTION POLICY WITH THE EXPERIENCE AGGREGATION AND A MULTI-ATTRIBUTE DECISION-MAKING

A. THE STANDARDIZED REWARDS FOR EACH SUB-TASK

Each reward function for the sub-task gives the learning agent using the fusion method a different reward and the standardized rewards are calculated by each learning agent, as shown in Fig.3. The standardization of rewards is necessary because different sub-task receives a different reward.

The number of sub-tasks is m. An episode starts on time t_1. The sub-task k receives a reward r_a^k at time t_a. μ_a is the average value for the rewards received by all sub-tasks on time t_a. σ_a is the variance for these rewards at that time. The

$$r = \begin{cases} -\left(f \times Rad + \sum_{i=1}^{N} \frac{|n-p| \times C_n}{p} - C_{noise}\right), & \textit{if} \text{ don't hit the obstacle} \\ \text{-100}, & \text{if hit the obstacle} \end{cases} \tag{7}$$

average value and the variance for these rewards are given by,

$$\begin{cases} \mu_a = \sum_{k=1}^{m} r_a^k / m \\ \sigma_a = \sqrt{\frac{1}{m} \times \sum_{k=1}^{m} (\frac{\sum_{k=1}^{m} r_a^k}{m} - r_a^k)^2} \end{cases} \tag{10}$$

Eq.(11) gives the standardized reward Re_a^k of sub-task k.

$$\begin{aligned} \mathrm{Re}_a^k &= \frac{r_a^k - \mu_a}{\sigma_a} \\ &= \frac{r_a^k - \sum_{k=1}^{m} r_a^k / m}{\sqrt{\frac{1}{m} \times \sum_{k=1}^{m} (\frac{\sum_{k=1}^{m} r_a^k}{m} - r_a^k)^2}}, \quad k = 1, 2, \ldots, m \end{aligned} \tag{11}$$

At time t_a, the average value for these standardized rewards $\hat{\mathrm{Re}}_a$ is given by,

$$\hat{\mathrm{Re}}_a = \sum_{k=1}^{m} \mathrm{Re}_a^k / m \tag{12}$$

This average value is given to the source task.

B. EXPERIENCE AGGREGATION IN THE EXPLORATION PROCESS

In the process of exploration, the prior experience gained by a mobile robot needs to be stored effectively and an effective storage method can also improve the efficiency of prior experience. For tasks completed by divide and conquer, the prior learning experience mainly collected from these trained sub-tasks. This paper investigates the storage method for the prior experience collected by sub-tasks and proposes an experience aggregation (EA) method using a decision-tree structure. The decision-tree is used to store the experience samples to improve search efficiency. This aggregation method can improve the exploitation efficiency of the prior experience of these trained sub-tasks for the agent of the source task.

The learning experience of sub-tasks is represented by quadruples $(s_t, a_t, Q_1(s_t, a_t), Q_2(s_t, a_t))$. In the process of experience aggregation, the most similar samples are merged into similar nodes. At the beginning of a learning process, there is only one node, the root node, to represent the experience aggregation model. The new learning experience is clustered according to the similarity between the experience samples, and prior experience will be marked by clustering. In each iteration, a new experience sample is compared with the root node. If the current state is similar, the similarity of the actions is compared. If the actions are similar, the Q-values corresponding to the similar state and action are aggregated to the root node. Otherwise, it is placed in a new child node. When a sub-task is trained, the Q-values $Q_1(s_i, a_i), Q_2(s_i, a_i)$ of the first experience sample $E_i(s_i, a_i, Q_1(s_i, a_i), Q_2(s_i, a_i))$ forms a root node in the decision-tree. The initial number of nodes is 1. The next sample $E_j(s_j, a_j, Q_1(s_j, a_j), Q_2(s_j, a_j))$ is compared with the existing node using the similarity to determine whether the sample is stored. State similarity and action similarity are two parts of similarity calculation. The Jaccard similarity is used to estimate the similarity between two experience samples.

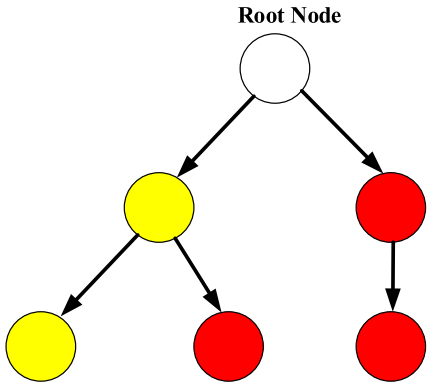

FIGURE 4. Structure for experience aggregation.

After each cluster, the center of each cluster is represented by the average value of the samples in this cluster.

The process of the experience aggregation method is shown in Fig.4. The structure of the decision tree is a binary tree and each node represents a cluster. The first sample is treated as the root node of the model. For the next sample, similarity calculation is performed for the state similarity between the root node and the new sample. If the similarity between the states is not satisfied, the sample will enter the yellow node and perform similarity calculation with the sample node, and so on. If the similarity between the states meets the conditions, the similarity of actions is calculated. If the condition of the similarity of actions is not satisfied, the sample will enter the red node and then the similarity with the red node is calculated, and so on. If the similarity condition is satisfied, the sample and the root node are the same clusters, and the sample is stored in the root node. After storing the experience samples, we use s_t, a_t as indexes for each process of getting the experience samples of sub-tasks. This process uses a step-by-step traversal method. By comparing the clustering centers, if the similarity between the index and the clustering center is less than a certain threshold, then we will search the appropriate samples from the clustering.

C. ACTION SELECTION USING A MULTI-ATTRIBUTE DECISION-MAKING

We decompose the source task into several sub-tasks, and each sub-task is trained in the same way. The learning experience of each sub-task is stored by an experience aggregation model. Prior experience of several sub-tasks will expand the source of prior experience for the source task. At time t, the Q-values for the current state s_t corresponding to each sub-task is regarded as attributes and the available actions of the learning agent are considered as the schemes. Then, we use a multi-attribute decision-making approach to develop an action policy depending on the attributes and schemes. The Q-values for sub-task u corresponding to the state s_t and action a_t is $Q^{(u)}(s_t, a_t)$. Action policy employs a multi-attribute decision-making method to calculate the evaluation value of each action, and the action with the maximum

evaluation value is selected. The number of actions is n and the action space is $A = \{a_1, a_2, \ldots, a_{n-1}, a_n\}$. For the current state s_t, the decision-making using Q-values of trained sub-tasks is shown in (13), as shown at the bottom of this page, where the attributes for the scheme a_j is $\{Q^{(1)}(s_t, a_j), Q^{(2)}(s_t, a_j)$ $, \ldots, Q^{(m-1)}(s_t, a_j), Q^{(m)}(s_t, a_j), Q(s_t, a_j)\}$. The evaluation value $Eq\left(a_j\right)$ of the scheme using a multi-attribute decision-making method is calculated as follows.

Step 1: Attributes for the scheme a_j are ordered from large to small to get a set of ordered data $\{Q^{[1]}(s_t, a_j), Q^{[2]}(s_t, a_j), \ldots, Q^{[m]}(s_t, a_j), Q^{[m+1]}(s_t, a_j)\}$.

Step 2: According to the visibility theory, we calculate the visibility attributes for each attribute, as shown in Eq.(14). Eq.(14) gives the visibility attributes $\{Q^{[k]}\left(s_t, a_j\right) | Vis\left(j\right), 1 \leq j \leq m{+}1\}$ for an attribute $Q^{[k]}\left(s_t, a_j\right)$. For any index f and k from the interval $(1, m)$, it satisfies $f < k$.

$$\begin{aligned} Vis(k) : & Q^{[f]}\left(s_t, a_j\right) \\ & \leq \left(\frac{k-f}{k-i}\right) \times \left(Q^{[i]}\left(s_t, a_j\right) - Q^{[f]}\left(s_t, a_k\right)\right) \\ & \quad \times \left(i, Q^{[i]}\left(s_t, a_j\right)\right) + Q^{[k]}\left(s_t, a_j\right); \quad \forall i < f < k \end{aligned} \tag{14}$$

Step 3: Eq.(15) gives the weight ω_k of the attribute $Q^{[k]}\left(s_t, a_j\right)$ using its visibility attributes. The final vector for weights is $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3, \ldots, \omega_m, \omega_{m+1})$.

$$\begin{aligned} \omega_k &= \frac{\sum_{Q^{[f]}(s_t,a_j) \in Vis} S\left(Q^{[f]}\left(s_t, a_j\right), Q^{[k]}\left(s_t, a_j\right)\right)}{\sum_{p=1}^{m+1} \sum_{Q^{[f]}(s_t,a_j) \in Vis} S\left(Q^{[f]}\left(s_t, a_j\right), Q^{[p]}\left(s_t, a_j\right)\right)} \\ &= \frac{\sum_{Q^{[f]}(s_t,a_j) \in Vis} \left(\frac{Q^{[f]}(s_t,a_j) \times Q^{[k]}(s_t,a_j)}{|Q^{[f]}(s_t,a_j) - Q^{[k]}(s_t,a_j)|^2}\right)}{\sum_{p=1}^{m+1} \sum_{Q^{[f]}(s_t,a_j) \in Vis} \left(\frac{Q^{[f]}(s_t,a_j) \times Q^{[p]}(s_t,a_j)}{|Q^{[f]}(s_t,a_j) - Q^{[p]}(s_t,a_j)|^2}\right)} \end{aligned} \tag{15}$$

Step 4: Eq.(16) gives the evaluation value of the action a_j.

$$\begin{aligned} & Eq\left(a_j\right) \\ & = \sum_{k=1}^{m+1} Q^{[k]}\left(s_t, a_j\right) \times \omega_j \\ & = \sum_{i=k}^{m+1} \left(Q^{[k]}\left(s_t, a_j\right)\right) \\ & \quad x \times \left(\frac{\sum_{Q^{[f]}(s_t,a_j) \in Vis} S\left(Q^{[f]}\left(s_t, a_j\right), Q^{[k]}\left(s_t, a_j\right)\right)}{\sum_{p=1}^{m+1} \sum_{Q^{[f]}(s_t,a_j) \in Vis} S\left(Q^{[f]}\left(s_t, a_j\right), Q^{[p]}\left(s_t, a_j\right)\right)}\right) \end{aligned} \tag{16}$$

Finally, the learning agent of the source task selects the action a_j with maximum evaluation value $Eq\left(a_j\right)$. The procedure for the proposed action policy with a multi-attribute decision-making method is given by Algorithm 2.

V. THE FRAMEWORK OF THE PROPOSED METHOD FOR THE OBSTACLE AVOIDANCE OF WHEELED MOBILE ROBOT

A. THE WHOLE FRAMEWORK FOR THE PROPOSED METHOD IS SHOWN BELOW

The framework for the proposed method is shown in Fig.5. The original obstacle avoidance task is divided into three sub-tasks: Reaching the target position faster, Not hitting any obstacles, and Not losing the target. The three sub-tasks have different reward functions. Each sub-task is trained to convergence by the Double-Q learning algorithm. The learning experience of each sub-task is stored in each experience aggregation (EA) model. The learning agent of the source task uses an experience aggregation model to get the prior experience, that is, the Q values of the trained sub-tasks. Then, an action policy with a multi-attribute decision-making method uses prior experience to select the current action. The environment gives different rewards to different sub-tasks. A method of reward standardization standardizes the rewards of these sub-tasks. The average value of standardized rewards is the final reward for the learning agent of the source task.

VI. EXPERIMENTS AND ANALYSIS

A. EXPERIMENT ON THE OBSTACLE AVOIDANCE FOR THE MAZE TASK

In the experimental section, three simulations are used to demonstrate the effectiveness of the proposed method. In the first simulation, the robot explores the unknown environment to learn an optimal policy to guide it from the starting position to the target position without bumping into any obstacle. An excellent action policy can guide the robot to reach the target position faster. The competitors include ε-greedy strategy (ε-greedy) [12], the ε-greedy strategy with an attenuation threshold (ε-decreasing) [36], [37] and softmax strategy (softmax) [32]. The task scenario is a map of square tiles that is m$\times$ n. In the maze environment, the gray squares represent obstacles. The action includes up, down, left, and right. In this experiment, the maze is set to 30$\times$30 square tiles and includes 100 square tiles of obstacles.

In each episode, the learning agent collects the experience sample from the task environment and learns a policy to reach the target position. The immediate reward is determined

$$D^Q = \begin{pmatrix} Q^{(1)}(s_t, a_1) & Q^{(2)}(s_t, a_1) & \cdots & Q^{(m)}(s_t, a_1) & Q(s_t, a_1) \\ Q^{(1)}(s_t, a_2) & Q^{(2)}(s_t, a_2) & \cdots & Q^{(m)}(s_t, a_2) & Q(s_t, a_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q^{(1)}(s_t, a_{n-1}) & Q^{(2)}(s_t, a_{n-1}) & \cdots & Q^{(m)}(s_t, a_{n-1}) & Q(s_t, a_{n-1}) \\ Q^{(1)}(s_t, a_n) & Q^{(2)}(s_t, a_n) & \cdots & Q^{(m)}(s_t, a_n) & Q(s_t, a_n) \end{pmatrix} \tag{13}$$

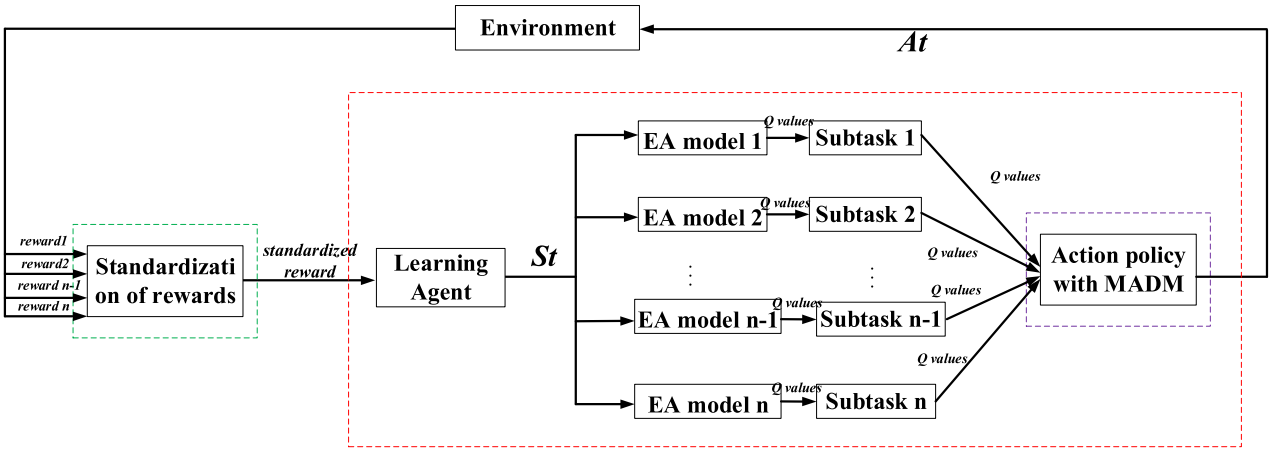

FIGURE 5. The whole framework for the proposed method.

TABLE 1. Experimental parameters.

Parameter	Value
Learning rate α	0.1
Discount rate γ	0.95
Exploring rate ε	0.95
Annealing factor	0.95
Maximum temperature parameter	0.1
Minimum temperature parameter	0.01
The constant f	0.1

by the Euler distance between the current position and the target position. If the next Euler distance is less than the current one, the learning agent will be rewarded +1. Otherwise, the learning agent will be given a reward -1. If the learning agent bumps an obstacle, the maximum punishment −100 will be given, and the robot will stay in the current position. If the learning agent reaches the target position, the maximum reward + 100 will be given. If the learning time for each episode exceeds 2000, this episode is terminated. The maximum number of episodes is 200. The experimental parameters for the first experiment are shown in Table 1.

Two different subtasks are set, reaching the target position faster, and not hitting any obstacles. The experimental results of four different methods are shown in Fig.6. Fig.6 shows the trajectories of four different methods. The proposed method can reach the target position more quickly with the shortest path. However, the motion paths for the other three methods are longer. Fig.7 and fig.8 show the number of steps of each episode and rewards for the four different methods. In Fig.7, all four methods converge after 200 training. The ε-greedy method has the largest fluctuation and the longest time step to reach the target position, after 200 training. Meanwhile, the proposed method has the minimum fluctuation and the minimum time step required for each episode, about 80 steps. The proposed method can make the robot reach the target position faster and collide with obstacles as little as possible. In Fig.8, we set the cumulative reward of each episode as the ordinate and the number of episodes as the abscissa.

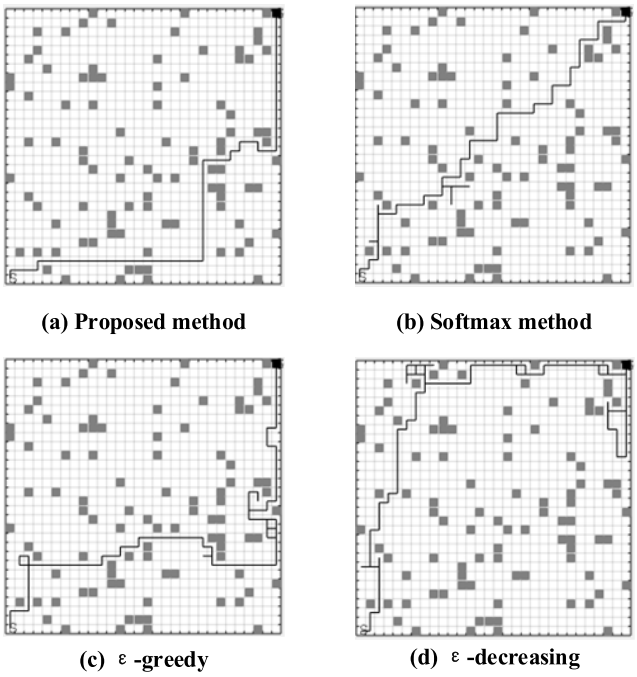

FIGURE 6. The trajectories for the four methods in task I.

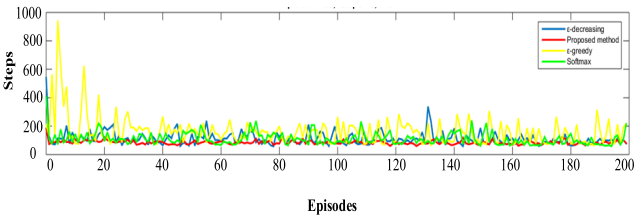

FIGURE 7. The number of steps of each episode for the four methods.

The proposed method obtained the highest cumulative reward compared with the other three methods. The cumulative reward of each episode for the proposed method is about 155, after 200 episodes. Similar to the results shown in Fig.7, the curve of the reward for the proposed method has the smallest fluctuation.

Algorithm 2 The Proposed Action Policy Using Double Q

1. **Definition:**
2. m := The number of sub-tasks
3. n := The number of actions
4. $\hat{\mathrm{Re}}_a$: = The average value for standardized rewards
5. $Eq()$: = Evaluation value for each action
6. α: = The learning rate
7. ε := The threshold value
8. ξ := Another threshold value
9. γ: = The discount value
10. **Initialization:**
11. Initialize $Q_1(s_t, a_t)$, $Q_2(s_t, a_t)$ randomly
12. Initialize $\varepsilon, \xi \leftarrow 0.9$
13. **Repeat** (for each step)
14. Initialize state s_t.
15. **Repeat**(for each step of the episode):
16. Give a random number $ran \leftarrow rand\ (0, .., 1)$
17. **If** $ran < \varepsilon$ then
18. Choose action a_t randomly.
19. **else**
20. Get
$$\hat{Q}(s_t, a_i) = \frac{1}{2}(Q_1(s_t, a_i) + Q_2(s_t, a_i)), \quad i = 1, 2, \ldots, n$$
21. $$Max\{\mathrm{Eq}(\hat{Q}, a_1), \mathrm{Eq}(\hat{Q}, a_2), \ldots, \mathrm{Eq}(\hat{Q}, a_{n-1}), \mathrm{Eq}(\hat{Q}, a_n)\} \rightarrow a_t$$
22. **End if**
23. Perform action a_t, and receive rewards from sub-tasks
24. Get the next state s_{t+1}.
25. Get the standard reward for each sub-task.
26. Get the average value $\hat{\mathrm{Re}}_a$ for the standard rewards.
27. **If** $ran < \xi$
28. Get $\Delta Q_1(s_t, a_t) = \hat{\mathrm{Re}}_a + \gamma \max Q_2(s_{t+1}, a_{t+1}) - Q_1(s_t, a_t)$
29. $Q_1(s_t, a_t) = Q_1(s_t, a_t) + \alpha \Delta Q_1(s_t, a_t)$
30. **Else**
31. Get $\Delta Q_2(s_t, a_t) = \hat{\mathrm{Re}}_a + \gamma \max Q_1(s_{t+1}, a_{t+1}) - Q_2(s_t, a_t)$
32. $Q_2(s_t, a_t) = Q_2(s_t, a_t) + \alpha \Delta Q_2(s_t, a_t)$
33. $s_t \leftarrow s_{t+1}$
34. **Until** s_t is terminal.
35. **End**

The proposed method extends the source of prior experience using a multi learner parallel learning method, and an experience aggregation model can store prior experience more efficiently. The MADM method can guide the learning agent to make more wise action decisions in the current environment. In order to further test the effectiveness of the algorithm, we expand the task scenario model to 39×36. Similar to the scenario of task 1, a larger scale scenario will bring more difficulties for the agent to complete the task. The same four methods were tested in this scenario. The maximum number of episodes is set to 200. When the robot reaches the target position, the robot will be awarded + 10. Otherwise, the robot is rewarded – 1. The trajectories of the four different methods are shown in Fig.9. From the experimental results, it can be seen that the trajectory of the proposed method is the smallest. The other three methods have a relatively long trajectory, among which the ε-greedy method has the longest trajectory and the most fluctuation.

In Fig.10, the experimental results show that the proposed method has the least fluctuation and the least time step to reach the target, about 86 time steps. Compared with the other three methods, the proposed method can reach the target position in less time. The relatively small fluctuation of the training curve means that the agent can avoid bad actions to reduce inefficient exploration. The multi-attribute decision-making method can help the agent choose effective action for each step. Meanwhile, in Fig.11, the proposed method

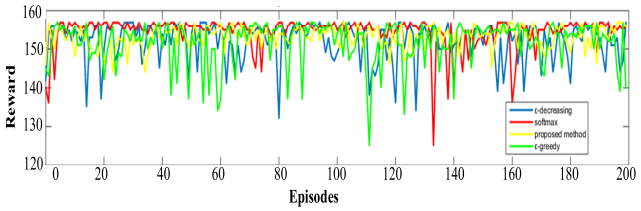

FIGURE 8. The reward of each episode for the four methods in task I.

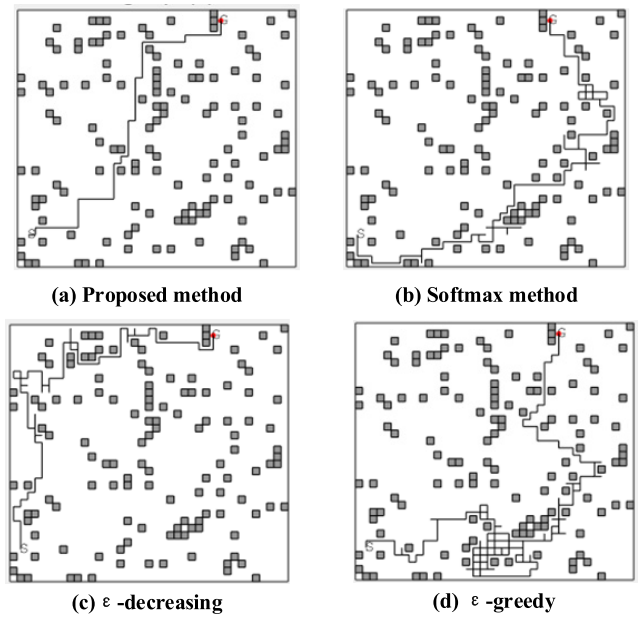

FIGURE 9. The trajectories for the four methods in task II.

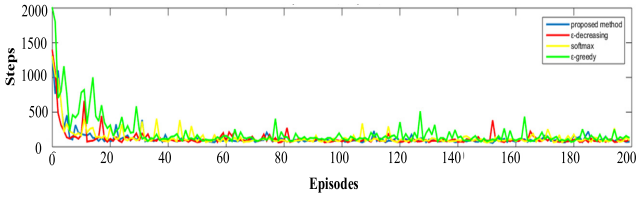

FIGURE 10. The number of steps of each episode for the four methods in task II.

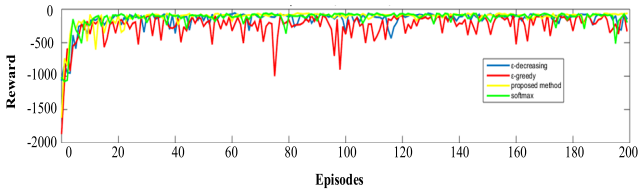

FIGURE 11. The reward of each episode for the four methods in task II.

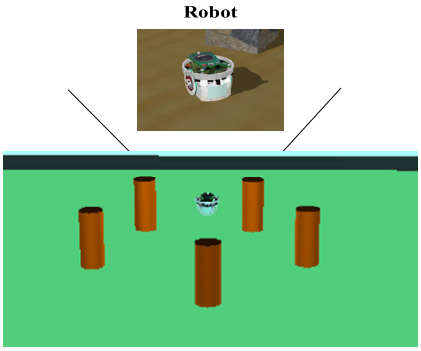

FIGURE 12. The experimental platform for obstacle collisio.

TABLE 2. Experimental parameters.

Parameter	Value
Learning rate α	0.1
Discount rate γ	0.95
Exploring rate ε	0.95
Annealing factor	0.95
Maximum temperature parameter	0.1
Minimum temperature parameter	0.01
The number of sensors	8

received the highest reward, about -71. Effective experience collection and action selection will help the learning agent get higher rewards.

B. EXPERIMENT ON WEBOTS SIMULATOR

We designed the experimental environment on the Webots simulator [38], [39], as shown in Fig.12. As shown in the experimental scenario, the initial and target positions are set. There are several obstacles between the initial position and the target position. These obstacles will hinder the robot from reaching the target position. We find an optimal path through experiments, which can ensure the maximum reward and the minimum loss. The proposed RL method for obstacle avoidance is used to solve this task. Meanwhile, two methods are used as the competitors, which are ε-greedy strategy (ε-greedy) and softmax strategy (softmax).

For this experiment, we will evaluate the experiment results from the following three aspects: average reward, the time to reach the target, and the retention time for obstacle collision. As explained before, we have compared the performance of three renowned policies. The experimental parameters are shown in Table 2.

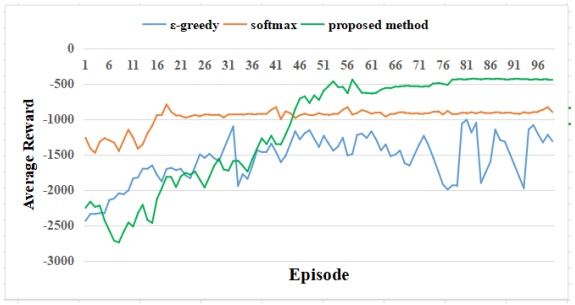

FIGURE 13. The curve of the average rewar.

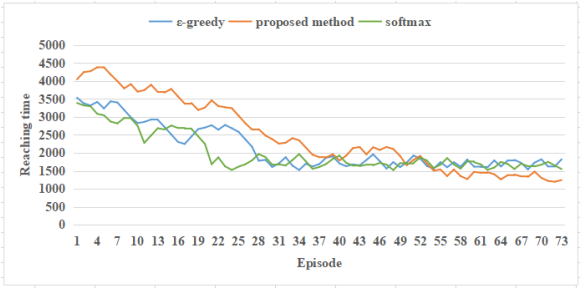

FIGURE 14. The curve of the reaching time.

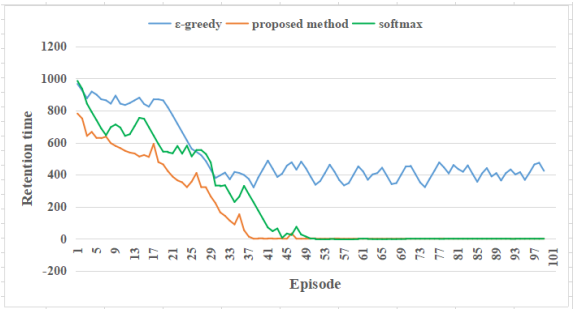

FIGURE 15. The curve of the retention tim.

The average reward is the cumulative reward of the episode divided by the total time of the episode. We take 100 episodes that are to reach the target position in the training process. Fig.13 to Fig.15 shows the experimental results. In Fig.13, the average reward obtained by the proposed method is the largest although the reward of this method is not high at the beginning of the training process. This result is consistent with experiment one. In the process of training, the performance of the ε-greedy method is relatively poor, because it is difficult to balance exploration and exploitation in complex tasks with a fixed threshold. Softmax strategy can reduce the randomness of action selection by temperature function. In the later stage of learning, the robot will choose actions based on previous learning experience, so the curve of the reward will be relatively stable in the later stage. The proposed method enables the robot to collect more experience in the early stage of learning, while in the later stage, the best action is selected through the action policy using a MADM method, and better performance is achieved.

The times for reaching the target for the three methods are shown in Fig.14. Finally, the three methods are stable with

continuous training, but the reaching time is different. The proposed method has the least reaching time, compared with the other two methods.

Fig.15 shows the robot's retention time in front of obstacles. Three different methods all have a large retention time, in the beginning, that is, the robot will stay in front of the obstacles for a long time. However, with the training, the retention time of the three methods is decreasing. The reduction rate of retention time is different for different methods. The experimental results show that the retention time of the proposed method decreases the fastest. The experimental results show that the proposed method can make the robot learn a better strategy to avoid obstacles.

Several simulation experiments show that the proposed method has better performance than competitors. Prior experience can effectively help robots learn the best behavior strategy. Therefore, the prior experience should be collected efficiently. Meanwhile, the action policy based on multi-attribute decision-making can help the robot choose better actions in a task.

VII. CONCLUSION

In this work, to address the dilemma of exploration and exploitation for real-time obstacle avoidance, an experience aggregative method with multi-attribute decision-making is proposed. An experience aggregation method is developed to cluster experiential samples. Meanwhile, the method of expanding the source of prior experience, and the method of integrating these prior experience are investigated. Firstly, the source task is decomposed into several sub-tasks. A double learning scheme was adopted to train these sub-tasks separately. A reward standardization method is proposed to calculate the current reward for the source task. The training method running several learners in parallel can gain more prior experience. The experimental results show that the proposed scheme outperforms the competitors in terms of learning performance.

For real robots, complex communication modules and uncertain environmental factors lead to the sample inefficiency, which is a hot issue at present [40], [41]. In the real world, the samples are often difficult to be collected effectively and the collected samples often have a lot of uncertainty [42]. It is difficult to develop an effective experience model, which also limits the application of reinforcement learning in a real robotic system. In the future, we will try to extend the proposed method to real-world systems. A platform robustness approach is needed to achieve data compatibility for different real-world platforms. In addition, integrating the possibility of applying multi-attribute decision-making in deep reinforcement learning-based systems [43] is also worthy of study.

REFERENCES

[1] J. Liao, Z. Chen, and B. Yao, "Model-based coordinated control of four-wheel independently driven skid steer mobile robot with wheel–ground interaction and wheel dynamics," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1742–1752, Mar. 2019.

[2] W. Li, L. Ding, H. Gao, and M. Tavakoli, "Haptic tele-driving of wheeled mobile robots under nonideal wheel rolling, kinematic control and communication time delay," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 1, pp. 336–347, Jan. 2020.

[3] H. Yang, X. Fan, P. Shi, and C. Hua, "Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 2, pp. 741–746, Mar. 2016.

[4] C.-J. Kim and D. Chwa, "Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 677–687, Jun. 2015.

[5] N. Dilokthanakul, C. Kaplanis, N. Pawlowski, and M. Shanahan, "Feature control as intrinsic motivation for hierarchical reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3409–3418, Nov. 2019, doi: 10.1109/TNNLS.2019.2891792.

[6] W. B. Haskell, R. Jain, H. Sharma, and P. Yu, "A universal empirical dynamic programming algorithm for continuous state MDPs," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 115–129, Jan. 2020.

[7] Z. Xu, F. Zhu, Y. Fu, Q. Liu, and S. You, "A dyna-Q based multi-path load-balancing routing algorithm in wireless sensor networks," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 1–4, doi: 10.1109/Trustcom.2015.522.

[8] W. G. Macready and D. H. Wolpert, "Bandit problems and the exploration/exploitation tradeoff," *IEEE Trans. Evol. Comput.*, vol. 2, no. 1, pp. 2–22, Apr. 1998.

[9] E. Emary, H. M. Zawbaa, and C. Grosan, "Experienced gray wolf optimization through reinforcement learning and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 681–694, Mar. 2018, doi: 10.1109/TNNLS.2016.2634548.

[10] S. Adam, L. Busoniu, and R. Babuska, "Experience replay for real-time reinforcement learning control," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 201–212, Mar. 2012, doi: 10.1109/TSMCC.2011.2106494.

[11] R. Al-Hmouz, T. Gulrez, and A. Al-Jumaily, "Probabilistic road maps with obstacle avoidance in cluttered dynamic environment," in *Proc. Intell. Sensors, Sensor Netw. Inf. Process. Conf.*, Melbourne, VIC, Australia, 2004, pp. 241–245.

[12] M. Alshahrani, Z. Fuxi, A. Sameh, S. Mekouar, and S. Liu, "Influence maximization based global structural properties: A multi-armed bandit approach," *IEEE Access*, vol. 7, pp. 69707–69747, 2019, doi: 10.1109/ACCESS.2019.2917123.

[13] 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)*, Oct. 2016, pp. 4877–4884.

[14] 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.

[15] 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.

[16] M. Khamassi, G. Velentzas, T. Tsitsimis, and C. Tzafestas, "Active exploration and parameterized reinforcement learning applied to a simulated human-robot interaction task," in *Proc. 1st IEEE Int. Conf. Robotic Comput. (IRC)*, Taichung, China, Apr. 2017, pp. 28–35, doi: 10.1109/IRC.2017.33.

[17] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[18] J. Zhang, X. Ruan, J. Huang, J. Chai, and Y. Wu, "A curiosity-based mobile robot path planning method," in *Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Chengdu, China, Dec. 2019, pp. 476–480, doi: 10.1109/IAEAC47372.2019.8997539.

[19] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped DQN," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2016, pp. 4026–4034.

[20] 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

[21] H. Garg and R. Arora, "A nonlinear-programming methodology for multi-attribute decision-making problem with interval-valued intuitionistic fuzzy soft sets information," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2031–2046, Sep. 2017.

[22] 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.
[23] 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.
[24] C. Hu and M. Xu, "Adaptive exploration strategy with multi-attribute decision-making for reinforcement learning," *IEEE Access*, vol. 8, pp. 32353–32364, 2020, doi: 10.1109/ACCESS.2020.2973169.
[25] S. Wen, J. Chen, Z. Li, A. B. Rad, and K. Mohammed Othman, "Fuzzy Q-learning obstacle avoidance algorithm of humanoid robot in unknown environment," in *Proc. 37th Chin. Control Conf. (CCC)*, Wuhan, China, Jul. 2018, pp. 5186–5190.
[26] Y. Zhang, X. Wei, and X. Zhou, "Dynamic obstacle avoidance based on multi-sensor fusion and Q-learning algorithm," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Chengdu, China, Mar. 2019, pp. 1569–1573.
[27] 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.
[28] 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
[29] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep Q-learning model for energy-efficient edge scheduling," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 739–749, Sep./Oct. 2019.
[30] Shina, S. Sharma, and A. Singla, "A study of tree based machine learning techniques for restaurant reviews," in *Proc. 4th Int. Conf. Comput. Commun. Autom. (ICCCA)*, Greater Noida, India, Dec. 2018, pp. 1–4, doi: 10.1109/CCAA.2018.8777649.
[31] M. Khamassi, G. Velentzas, T. Tsitsimis, and C. Tzafestas, "Robot fast adaptation to changes in human engagement during simulated dynamic social interaction with active exploration in parameterized reinforcement learning," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 4, pp. 881–893, Dec. 2018, doi: 10.1109/TCDS.2018.2843122.
[32] M. Zhao, H. Lu, S. Yang, and F. Guo, "The experience-memory Q-learning algorithm for robot path planning in unknown environment," *IEEE Access*, vol. 8, pp. 47824–47844, 2020, doi: 10.1109/ACCESS.2020.2978077.
[33] H. Huang, M. Lin, L. T. Yang, and Q. Zhang, "Autonomous power management with Double-Q reinforcement learning method," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1938–1946, Mar. 2020.
[34] R. Calvo and M. Figueiredo, "Reinforcement learning for hierarchical and modular neural network in autonomous robot navigation," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, USA, vol. 2, 2003, pp. 1340–1345.
[35] Z. Li, Z. An, Z. Yongfang, and S. Zhifu, "Study on group air to ground attack-defends hierarchical dynamic decision-making," *J. Syst. Eng. Electron.*, vol. 18, no. 3, pp. 540–544, Sep. 2007.
[36] B.-Q. Huang, G.-Y. Cao, and M. Guo, "Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Guangzhou, China, 2005, pp. 85–89.
[37] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vols. 2–3, no. 47, pp. 235–256, May 2002.
[38] B. G. Woolley and G. L. Peterson, "Unified behavior framework for reactive robot control," *J. Intell. Robotic Syst.*, vol. 55, nos. 2–3, pp. 155–176, Jul. 2009.
[39] O. Karoui, E. Guerfala, A. Koubaa, M. Khalgui, E. Tovard, N. Wu, A. Al-Ahmari, and Z. Li, "Performance evaluation of vehicular platoons using webots," *IET Intell. Transp. Syst.*, vol. 11, no. 8, pp. 441–449, Oct. 2017, doi: 10.1049/iet-its.2017.0036.
[40] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.
[41] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 3357–3364.
[42] W. Wu and M. Liao, "Reinforcement fuzzy tree: A method extracting rules from reinforcement learning models," in *Proc. IEEE/ACIS 18th Int. Conf. Comput. Inf. Sci. (ICIS)*, Beijing, China, Jun. 2019, pp. 47–51, doi: 10.1109/ICIS46139.2019.8940165.
[43] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020, doi: 10.1109/TITS.2019.2901791.

CHUNYANG HU received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. He was 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 block chain. He is also a member of the China Computer Federation.

BIN NING received the master's degree in software engineering from the Beijing University of Technology, Beijing, China, in 2005. He is currently a Professor with the School of Computer Engineering, Hubei University of Arts and Science, Xiangyang, China. His research interests include software engineering, data mining, and algorithm design. He is also 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 the Northwestern Polytechnical University, Xi'an, China. His research interests include mobile robot, intelligent control, and machine learning. He has won two national scholarships for postgraduate course of China. Besides, he was awarded the Best Presentation Award from IEEE IFuzzy 2018.

QIONG GU received the M.S. degree in computer science and technology and the Ph.D. degree in geosciences information engineering from the China University of Geosciences, Wuhan, China, in 2006 and 2009, respectively. She is currently a Professor with the School of Computer Engineering, Hubei University of Arts and Science. She is particularly interested in data mining and machine learning. She has extended her interests include net-mediated public sentiment and the Internet of Things. She is also a member of the China Computer Federation.