Received July 15, 2020, accepted July 17, 2020, date of publication July 22, 2020, date of current version August 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011211

Path Planning Method With Improved Artificial Potential Field–A Reinforcement Learning Perspective

QINGFENG YAO[1,2,3], ZEYU ZHENG[1,2,3], LIANG QI[4], (Member, IEEE), HAITAO YUAN[5,6], (Member, IEEE), XIWANG GUO[7], (Member, IEEE), MING ZHAO[1,2,3], ZHI LIU[1,2,3], AND TIANJI YANG[1,2]

[1]Department of Digital Factory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Shenyang 110016, China
[3]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[4]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[5]Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07029, USA
[6]School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China
[7]College of Computer and Communication Engineering, Liaoning Shihua University, Fushun 113001, China

Corresponding authors: Zeyu Zheng (zhengzeyu@sia.cn), Liang Qi (qiliangsdkd@163.com), and Xiwang Guo (x.w.guo@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFF0214704; in part by the Liaoning Revitalization Talents Program under Grant XLYC1907166; in part by the Liaoning Province Department of Education Foundation of China under Grant L2019027; in part by the Liaoning Province Dr. Research Foundation of China under Grant 20170520135; in part by the National Natural Science Foundation of China under Grant 61903229, Grant 61973180, and Grant 61802015; and in part by the Natural Science Foundation of Shandong Province under Grant ZR2019BF004 and Grant ZR2019BF041.

ABSTRACT The artificial potential field approach is an efficient path planning method. However, to deal with the local-stable-point problem in complex environments, it needs to modify the potential field and increases the complexity of the algorithm. This study combines improved black-hole potential field and reinforcement learning to solve the problems which are scenarios of local-stable-points. The black-hole potential field is used as the environment in a reinforcement learning algorithm. Agents automatically adapt to the environment and learn how to utilize basic environmental information to find targets. Moreover, trained agents adopt variable environments with the curriculum learning method. Meanwhile, the visualization of the avoidance process demonstrates how agents avoid obstacles and reach the target. Our method is evaluated under static and dynamic experiments. The results show that agents automatically learn how to jump out of local stability points without prior knowledge.

INDEX TERMS Reinforcement learning, neural network, potential field, path planning.

I. INTRODUCTION

With the development of artificial intelligence, the technology of autonomous mobile agents has been widely adopted in industry, military, and medical fields. At the same time, tasks in uncertain environments become more complex. The agent needs to cooperate with multi-objective tasks. Therefore, intelligent autonomous control technology has attracted extensive attention from academia and industry [1], [2]. As one of the main techniques, path planning is a research hotspot in artificial intelligence.

Path planning requires that a mobile agent finds an optimal or sub-optimal collision-free path from the start point to the destination in the environment. At present, path planning techniques can be divided into two categories, i.e., global planning and regional planning [3]. The former is a path planning in a statically known environment, which is also known as a static path planning method [4]. There are numerous methods such as the greedy algorithm, Dijkstra's algorithm, and A* algorithm. The last one is suitable for the situation where the environmental information is unknown or partial unknown and real-time environmental information is used for path planning. The main approaches include the artificial potential field [5], the genetic algorithm [6], and PSO [7], [8] methods.

To meet the real-time requirement, a fast path planning algorithm, probabilistic signpost algorithm (PRM) [9] is proposed to preprocessing before randomly sampling in the pose space. This algorithm has been extensively applied to path planning within the environment which includes

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

dynamic obstacles. However, it fails to solve the problem of differential constraints in the mobile agent and leads to path planning results unreasonable. In 1998, LaValle and Kuffner [10] proposed a single query Rapidly-exploring Random Trees (RRT) theory. RRT fully considers the quantitative differential constraints of agents and generates the search tree. However, it lacks stability in a dynamic environment [11].

The artificial potential field is a virtual force field method proposed by Khatib [12]. The movement of the agent in the environment is the result of the simulated force field. The target point generates the gravity to the agent and the obstacle generates the repulsive force. The movement of the agent is controlled by both gravity and repulsion. Because of its advantages of simple mathematical analysis, low computational complexity, and a smooth path, the algorithm is widely adopted in the field of real-time obstacle avoidance and path planning [13].

However, traditional artificial potential field method has one inherent defect. An agent will fall into a local-stable-point when the resultant force is zero that happens easily in a complex environment. The reasons for this problem are the various shapes of obstacles and position relations in the environment. A lot of efforts have been made to solve these problems. Jia *et al.* change the repulsive potential of obstacles by discretizing the outline of obstacles [14]. Li *et al.* present an improved artificial potential field based regression search method for autonomous mobile agent path planning in completely known environments [15]. Orozco-Rosas *et al.* propose a membrane evolutionary artificial potential field approach to solve the mobile agent path planning problem. This method finds the parameters for generating a feasible and safe path with the genetic algorithm [16]. Rizqi *et al.* design a potential function to guide the quadrotor to the goal and avoid the obstacle. The algorithm solves the local-stable-point problem by utilizing the wall-following behavior [17]. At present, the main thought to solve the local-stable-point problem is changing the potential field to reduce the occurrence of local-stable-points.

This research explores the ability of reinforcement learning in the artificial potential field. The agent will confront different environments and has restricted access to status information. The agent learns how to jump out of a local-stable-point and achieves the target based on the potential field information. This study makes the following contributions.

1) We propose a method named black-hole potential field (BHPF), which reduces the occurrence of local-stable-points under multi-target circumstances. By combining BHPF and reinforcement learning we propose a black-hole potential field deep Q-learning (BHDQN). The experiments show that an agent can move to the nearest target point and elude obstacles with BHPF information without prior knowledge.

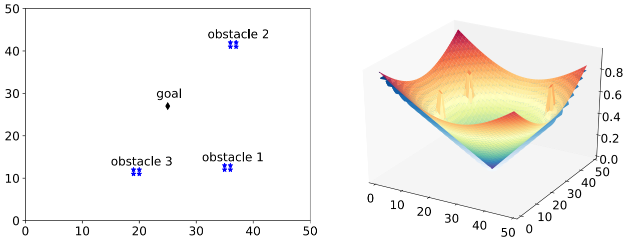
2) We test the adaptability of BHDQN with different shapes of obstacles. The result shows that the trained agent adapts to new surroundings quickly and escapes from new types of the local-stable-point. Besides, the agent can complete path planning in dynamic and static warehouse environments.

The rest of the paper is organized as follows. Section III introduces the artificial potential field and the block-hole potential field. Section IV provides a method that utilizing reinforcement learning in BHPF. Section V presents the experiments and analyzes the experimental results. Section VI concludes this paper and discusses future work.

II. LITERATURE REVIEW

In recent years, deep learning is a widely-used method in computer vision [18], NLP [19], the medical field [20] and shows the power in the path planning problem. At the same time, the network can transfer the knowledge to new scenarios [21]. Yuan *et al.* propose a dynamic path planning method based on a gated recurrent unit-recurrent neural network for path planning in an undiscovered space [22]. Tai *et al.* design a hierarchical structure that adopts a convolutional neural network to avoid indoor obstacles [23]. A. Giusti *et al.* propose an approach that uses a deep neural network as a supervised image classifier and outputs the main direction of the trail by deal with the whole image [24]. M. Dragoicea *et al.* design a system by using the convolutional neural network to learn a control strategy that mimics the behavior of the expert. The quadcopter is applied to autonomously navigate indoors and find the destination with one camera [25].

Reinforcement learning (RL) is an important branch of the artificial intelligence technology that has strong adaptability and self-learning ability in the complex environment. With the development of deep learning, the combination of the deep learning and reinforcement learning has become a research hotspot and has been successfully applied in many fields such as playing games [26], [27] and has potential in many traditional fields such as business process mining [28], transportation system [29], scheduling problems [32] and multiresource-constrained [30], [31]. The agent has the capacity to enhance its strategy to fulfill mission over time with reinforcement learning. Reinforcement learning is inherently suitable for the path planning problem. Wang *et al.* formulate the maximum spatial-temporal coverage optimization issue as a deep reinforcement learning process. A deep reinforcement learning based vehicle scheduling is adopted to produce an optimal solution and maximize the spatial-temporal coverage [33]. Wei *et al.* train a deterministic policy gradient algorithm on an abstracted structure to imitate the deformation of the path under the external force. This method allows unmanned ground vehicle autonomously to find collision-free paths to mobile goals in complicated environments [34]. Tai *et al.* build the environment that regards the coordinate of the agent as input and outputs the continuous steering operation. An end-to-end asynchronous deep reinforcement learning frame enables the partly visible agent to moves to the assigned target without collision [35]. P. Mirowski *et al.* combine the goal-driven reinforcement



(a) The original map.

(b) The potential field energy of the map.

FIGURE 1. The schematic diagram of origin potential field.

learning and auxiliary depth prediction for learning navigation in complicated 3D mazes [36].

III. BLACK-HOLE POTENTIAL FIELD

A. ARTIFICIAL POTENTIAL FIELD

The artificial potential field (PF) method is a path planning method by constructing a virtual potential field in the environment [12]. The potential field is made up of two kinds of potential fields: gravity field and repulsion field. The target exerts a gravity for the agent, forming the gravitational potential field. At the same time, the obstacle generates a repulsive force, forming a repulsive potential field. In the artificial potential field, the potential energy is influenced by the gravitational field and the repulsive field. The potential energy of a location that near an obstacle is high, while the potential energy of a location near a target is low, which is shown in Fig. 1. Therefore, driven by the resultant force of the repulsive and gravitational potential fields, the agent moves from the location with high potential energy to the low, and finds a collision-free path that can reach the target. The gravitational attraction (i.e., gravity) of the target on the map covers the whole map so that the agent moves toward the target point from any location on the map. The obstacles only repel the agent within a certain distance, because an agent avoids obstacles when approach to obstacles.

The gravitational potential field $U_{att}(q)$ of the traditional artificial potential field is defined as:

$$U_{att}(q) = k_{att} * \frac{(q-q_g)^2}{2} \tag{1}$$

where q_g is coordinate of the target point, $U_{att}(q)$ is the gravitational attraction that target point q_g in position q, and k_{att} is the attraction coefficient. The gravitational attraction of the target point augments with increasing gravitational coefficient. The potential field at q_g is zero. The point has higher potential field with increasing distance from the point q_g.

Gravity is obtained from the negative gradient of the gravitational potential field as follows:

$$F_{att}(q) = -\nabla U_{att}(q) = -k_{att}\left|q-q_g\right| \tag{2}$$

The repulsive potential field $U_{rep}(q)$ of the traditional artificial potential field is defined as:

$$U_{rep}(q) = \begin{cases} \frac{k_{rep}}{2}\left(\frac{1}{q-q_0} - \frac{1}{p_0}\right)^2 & q-q_0 \leqslant p_0 \\ 0 & q-q_0 > p_0 \end{cases} \tag{3}$$

$U_{rep}(q)$ is the repulsive force in position q, k_{rep} is the repulsive coefficient, $q-q_0$ is the distance from the obstacle q_0, and p_0 is the range of repulsive field of the obstacles.

The repulsive force is obtained from the negative gradient of the repulsive potential field as follows:

$$F_{rep}(q) = -\nabla U_{rep}(q) \tag{4}$$

Therefore, the total force F_q at position q is calculated by superimposing the potential force both obstacles and targets as follows:

$$F_q = \sum_{i=1}^{n} F_{att}(i) + \sum_{j=1}^{m} F_{rep}(j) \tag{5}$$

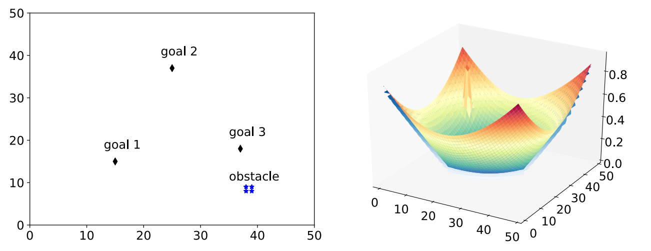
The artificial potential field has the characteristics of simple principle, smooth path, and strong real-time performance. It plays an important role in real-time path planning. However, one drawback that comes up often is the local-stable-point problem. A local-stable-point problem is that agents are trapped in a point that has the lowest potential energy and cannot move to target points. This problem appears on following situations: 1) A special-shaped obstacle appears between the agent and the target, the agent is trapped inside the barrier and cannot reach the target. 2) The environment is relatively complex such as the case of multiple targets. For example, there is $n(n>1)$ targets$(x_1, y_1), (x_2, y_2)\ldots(x_n, y_n)$ on the map, and the total gravitational potential at q_g: (x, y) is:

$$\begin{aligned} U_{att}(q) = {} & \frac{k_{att}}{2}[(x-x_1)^2 + (x-x_2)^2 + \ldots(x-x_n)^2 \\ & +(y-y_1)^2 + (y-y_2)^2 + \ldots(y-y_n)^2] \qquad (6) \\ \frac{\partial U_{att}(q)}{\partial x} = {} & \frac{k_{att}}{2}(2(x-x_1+x-x_2+\ldots+x-x_n)) \\ = {} & k_{att}(nx-x_1-x_2\ldots-x_n) \qquad (7) \\ \frac{\partial U_{att}(q)}{\partial y} = {} & \frac{k_{att}}{2}(2(y-y_1+y-y_2+\ldots+y-y_n)) \\ = {} & k_{att}(ny-y_1-y_2\ldots-y_n) \qquad (8) \end{aligned}$$

The location of the minimum field on the map is $(\frac{x_1+x_2+\ldots+x_n}{n}, \frac{y_1+y_2+\ldots+y_n}{n})$. We provide an example for a demonstration of the formation of local-stable-point. There are three targets and one obstacle on the map. The potential energy of the map as shown in Fig. 2. The accumulation of the potential field of multiple targets generates a huge hole in the center of targets. The agent on the map will drop into the local-stable-point and cannot escape.

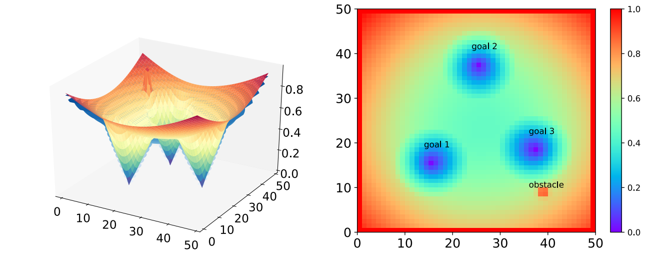
B. BLACK-HOLE FIELD

Multiple targets will lead to the occur of local-stable-point. To overcome this problem, we propose a method called



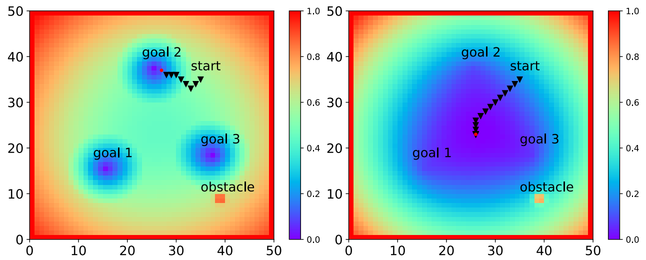
(a) The original map. (b) The potential field energy of the map.

FIGURE 2. The local-stable-point in the multiple targets map.



(a) The black-hole potential energy of the map. (b) The heat map of the black-hole potential energy.

FIGURE 3. The heat map and potential energy based on BHPF.



(a) The result from BHPF. (b) The result from potential field.

FIGURE 4. The contrast of potential field and BHPF.

black-hole potential field method (BHPF). Besides the single gravitational force, we add a black-hole field force. In the original artificial potential field, the gravitational force can be detected globally. The black-hole field force has a small valid range with strong attraction which can prevent multiple gravitational superimposition. The coverage of the black-hole field force is called the domain. Once an agent reaches the domain, it will be pulled to the target point by the black field force. The black-hole field force is obtained as follows:

$$U_{str}(q) = \begin{cases} -\frac{k_{str}}{2}[p_s - (q - q_g)]^2 & q - q_g \leqslant p_s \\ 0 & q - q_g > p_s \end{cases} \tag{9}$$

$U_{str}(q)$ is the black-hole force in position q, k_{str} is black-hole field coefficient, $q - q_g$ is the distance away from the goal q_g, and p_s is the range of black-hole field of the goal. It is worth noting that the value of k_{str} is much larger than k_{att} to overlay origin field force and should less than k_{rep} for avoiding collisions. The black-hole field force $F_{str}(q)$ is calculated as follows:

$$\begin{aligned} F_{str}(q) &= -\nabla U_{str}(q) \\ &= \begin{cases} -k_{str}[p_s - (q - q_g)] & q - q_g \leqslant p_s \\ 0 & q - q_g > p_s \end{cases} \end{aligned} \tag{10}$$

The external force that affects the agent is obtained as follows:

$$F_q = \sum_{i=1}^{n}[F_{att}(i) + F_{str}(i)] + \sum_{j=1}^{m} F_{rep}(j) \tag{11}$$

where n is the number of targets and m is the number of obstacles. For convenience, the potential field is scaled as follows:

$$U'_q = \frac{U_q - U_{min}}{U_{max} - U_{min}} \tag{12}$$

where U_q is the total field in position q, U_{max} is the maximum potential field, and U_{min} is the minimum potential field.

The heat map and potential energy of BHPF on the map are shown in Fig. 3. The potential field transforms slowly at the position further away from the target points, but the potential field collapses rapidly in the position near to the target point. The agent nearby target will reach the target directly under the vigoroso black-hole field and ignore the attraction of other targets. An agent is added to the environment to test the effect of BHPF, as seen in Fig. 4(a), the agent falls into the local-stable-point when the agent is near the middle of multiple target points in the origin PF. The result after adding the block-hole domain is shown in Fig. 4(b). The agent achieves the target directly under black-hole field. The black-hole field could not only reduce the appearance of local-stable-point, but also serve as a pattern to help the agent discover targets with reinforcement learning in the next section.

IV. REINFORCEMENT LEARNING WITH BHPF

A. MARKOV DECISION PROCESS

Reinforcement learning (RL) can learn how to deal with different environmental information. Normally, the environment is a Markov decision process (MDP) [37], which expressed as a tuple $M = (S, A, \rho, R, \gamma)$. In MDP, the change of the state s_{t+1} is only related to the state s_t and the behavior $a_t(a_t \in A)$ of the agent at the previous moment t, and independent of other elements. The agent updates its policy with received reward $r_t(r_t \in R)$. The environment accepts the behavior of the agent and transfers the environment through the environment transfer probability ρ. Finally, the agent receives an overall reward with the step discount factor $r \in (0, 1]$.

However, in most cases, an agent cannot receive the full state of the environment and needs a more general method such as a partially observable MDP (POMDP) [38]. POMDP is described as a tuple $M = (S, A, T, R, \gamma, O)$. Different from MDP, the agent receives an observation $o_t(o_t \in O)$ instead of the state s_t. This observation is obtained by a probability distribution $O(s) = P(o|s)$.

B. REINFORCEMENT LEARNING

Deep reinforcement learning is one of the most popular fields in the artificial intelligence field in recent years. RL trains the model through interactive tests and rewards in the environment. Instead of establishing the control model, it utilizes the reward function to motivate the agent to learn new strategies. An agent interacts with the environment in a real-time situation. By observing the current state, a value function is established to predict rewards of different behaviors. At the same time, the strategy generated by value function map the current state to the corresponding behavior. The environment responds to the behavior of the agent and returns the new state to the agent and corresponding rewards. At this point, the agent receives the reward from the environment and updates its value function. Through the cycle of the above process, the agent is trained to adapt to the environment and make corresponding actions according to different states.

RL relies on the exploration of unfamiliar environments and updates its policy autonomously. In this way, agents acquire knowledge from the environment and improve their strategy to adapt to the environment. In the RL framework, agents interact with the environment through perception and actions. RL can be divided into two types of modeling. The first is the model-based algorithm, which obtains the empirical knowledge from the environment to build the learning model, and then acquires the optimal strategy through the model. The second is the model-free approach, which directly selects the action and interaction with the environment. The common model-free algorithms include AC [39] and Q-learning [40]. Q-learning is an offline strategy and adopts the temporal difference (TD) learning method. The propose of Q-learning is to estimates the cumulative reward that from t to T as follows:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{13}$$

Q function predicts the cumulative reward by current action and current state according to the current policy π as follows:

$$Q^{\pi}(s,a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi] \tag{14}$$

For all strategies, if the expected revenue of one strategy is greater than or equal to the revenue of other strategies, it is the optimal strategy, i.e.,

$$Q^{*}(s,a) = \max_{\pi} \mathbb{E}[R_t | s_t = s, a_t = a, \pi] \tag{15}$$

The optimal strategy conforms to the bellman equation and can be expressed by Q value at the next moment as:

$$Q^{*}(s,a) = \mathbb{E}[r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) | s, a] \tag{16}$$

Traditional methods use an iterative bellman equation to calculate Q value, but it is difficult to achieve convergence in complex environments. The latest methods use neural networks to approximate the Q function. Deep Q-learning (DQN) [27] using a convolution network to predict the Q value and update network parameters with a temporal difference method, which approximates that $Q(s, a; \theta) \approx Q^{*}(s, a)$ and calculates goal as follows:

$$Y_i = r_t + \gamma \max_{a_{t+1}}(s_{t+1}, a_{t+1}) | \theta_{i-1} \tag{17}$$

The update of DQN relies on loss function which is calculated as follows:

$$L(\theta_i) = \mathbb{E}[(Y_i - Q(s_t, a_t) | \theta_i)^2] \tag{18}$$

DQN involves some ways to enhance stability, for example, replay memory and prioritized experience replay set different importance during sampling process [41].

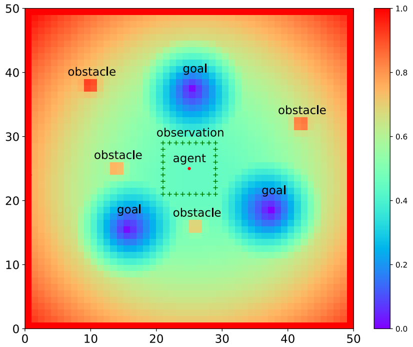


FIGURE 5. The agent observes its state in the environment.

C. RL IN POTENTIAL-FIELD

To verify the method, we evaluate black-hole potential field deep Q-learning (BHDQN) and BHPF based on a grid simulation platform which provides a map with 50 rows and 50 columns. The environment is based on Python 3.6 and Inter Core i5-7200 with 8G RAM. The map has several obstacles and target points, which are generated randomly during training and testing. Therefore, the agent needs enough generalization and robustness to adapt to various conditions. The simulation platform is shown in Fig. 5. The agent can only observe its surrounding environment. The green area is the space that an agent can observe and its selection action relies solely on its surrounding potential field energy. The region of potential field energy is regarded as the input of the network. Then, the field energy is processed by two layers of 2-D convolutions of 16 neurons with a leaky relu activation function. The exporting convolutional feature is operated by max-pooling operation and it is followed by two fully-connected layers of 32 neurons. The last fully-connected layer outputs a vector that represents the estimated reward value of nine actions. The network calculates the loss function based on the real reward value. The loss function and optimizer that we use are mean square error and Adam respectively. The agent does not know its position on the map as well as the position information of the target point and obstacles, which means that it is a typical POMDP. It is worth noting that the potential field function is only affected by the superposition of obstacles and target points and will not be affected by agents.

The action space of an agent includes nine actions, i.e., up, down, left, right, left up, left down, right up, right down, and immobility. The agent selects an optimal based on its policy which finally outputs a one-dimensional vector, i.e., $Q = [q_1, q_2, \ldots, q_9]$, where q_i is the predicted Q value of the i-th behavior. To balance the exploration and exploitation problem in RL, this work adopts a ϵ greedy method. There is a choice for a random behavior with the probability of ϵ. The probability of selection based on DQN is $1 - \epsilon$, ϵ annealed linearly from 0.95 to 0.05 and fixed at 0.05 thereafter.

RL improves its policy with rewards and learns to accomplish goals guided by the reward function, the agent pursues positive rewards and avoid negative rewards. The final reward is obtained as follows:

$$R = W_1 + W_2 + W_3 + W_4 \tag{19}$$

If an agent collides with an obstacle or moves out of the boundary, it gets a penalty of W_1. W_2 is the positive reward received by the agent after it reaches the target point. W_3 is a fixed penalty that urges the agent to reach the target as soon as possible. W_4 is a reward for the change of the potential field. Here, $W_4 = \alpha(p_t - p_{t-1})$, where α is hyper-parameters ($\alpha < 0$), and p_t is the potential field of the position of the agent at time t. It means that the agent is encouraged to get close to the location with a low potential field, and this award will help the agent quickly find the nearby target point and keep away from the obstacles. In BHPF and BHDQN, because the value of k_{str} is larger that k_{att}, agents will receive a high reward in the domain of a target. The general parameters are illustrated in Table 1.

TABLE 1. Parameter values and definitions.

Parameter	Value	Definition
k_{att}	1	Attraction coefficient
k_{str}	8	Repulsive coefficient
k_{rep}	20	Black-hole field coefficient
p_0	3	The range of repulsive field
p_s	8	The range of black-hole domain
W_1	-10	Collision penalty
W_2	10	Reward of completing goals
W_3	-0.2	Penalty for each time
α	-20	Reward coefficient of potential field
γ	0.95	Discount rate

V. EXPERIMENTS

A. BASIC TRAINING

An agent studies in a simple environment first. It is tested in the environment with three target points and ten square obstacles. The locations of target points and obstacles in the environment are randomly generated to help the agent adapt to different situations gradually. The training process converges after 100 epochs. The comparison of reward and success rate of reaching all target points between methods is shown in Table 2. The agent with BHDQN learns to avoid obstacles and reaches the target better in multiple targets circumstances. However, the PF fails to converge because of the local-stable-point produced by multi-targets.

TABLE 2. The experimental result of different methods in 100 rounds of experiments.

Environment	Statistic	PF	BHPF	BHDQN
1 target	Mean	9.38	11.85	11.71
	Std.Dev.	0.45	0.82	0.82
	Done	1	1	1
2 targets	Mean	-4.54	9.29	11.74
	Std.Dev.	14.94	18.23	16.80
	Done	0.28	0.64	0.70
3 targets	Mean	-4.43	11.37	16.06
	Std.Dev.	13.31	21.79	21.48
	Done	0.13	0.47	0.59

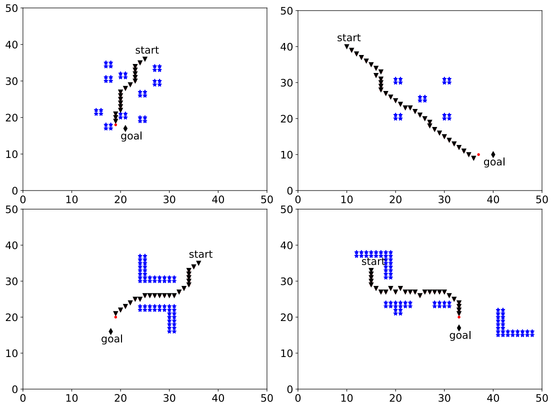


FIGURE 6. Tests of an agent to search the target point. The agent learns to avoid basic obstacles and searches for the target point through field energy.

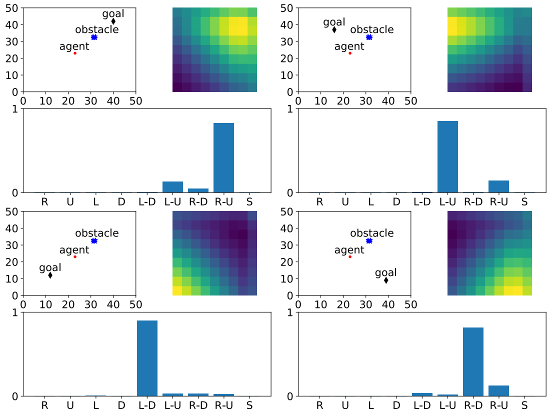


FIGURE 7. A visualization of the interior of the trained convolutional network with different target locations.

To test the generalization of BHDQN with a new environment, a series of dense obstacles [16] is adopted to test the ability of the agent to avoid obstacles directly. The result is shown in Fig. 6. It can be seen that an agent with basic training can cope with different environments and can find target points without additional training.

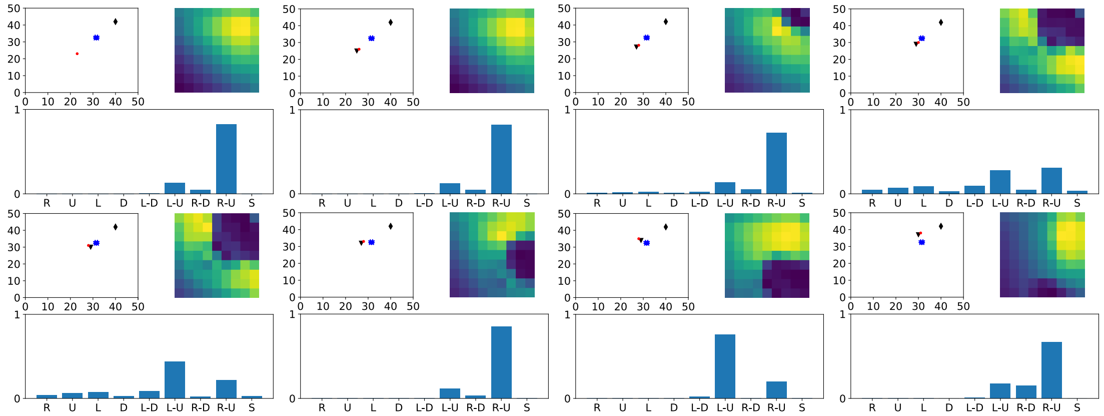
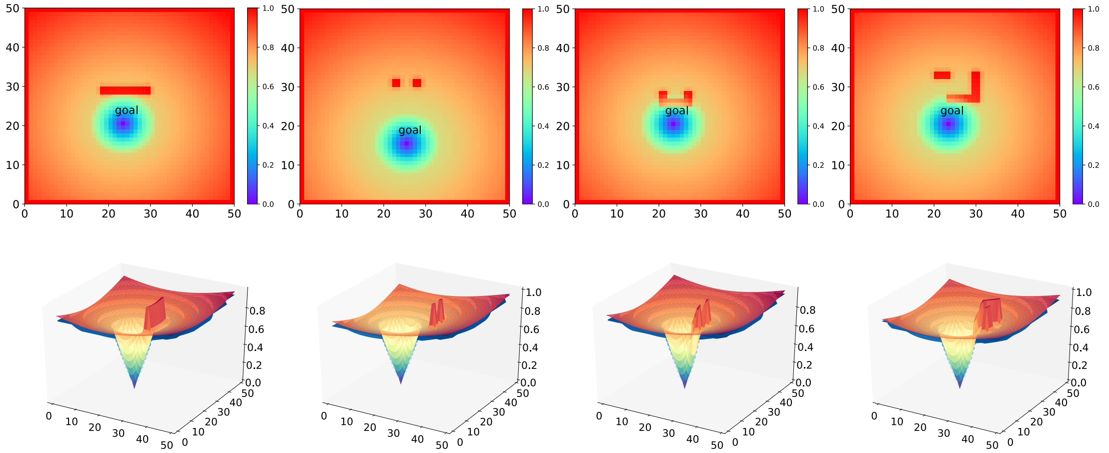


FIGURE 8. A visualization of the process that an agent avoids obstacles and reaches the target.



(a) Env 1. (b) Env 2. (c) Env 3. (d) Env 4.

FIGURE 9. The potential field of different categories of local-stable-points.

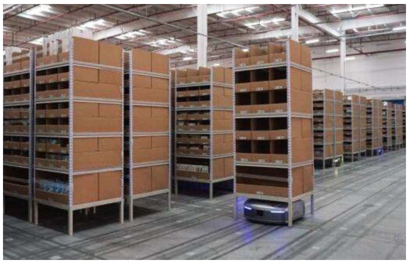


FIGURE 10. The scenario of intelligent warehouse.

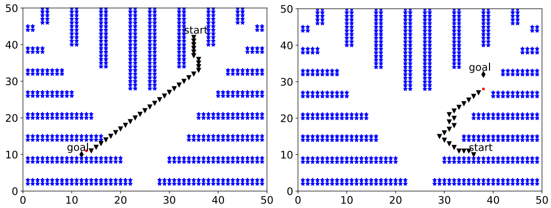


FIGURE 11. The path planning for static fishbone layout warehouse.

In order to analyze how the model makes the decision, we visualize the intermediate layer of the agent so that we could understand whether the trained agent handles the relationship between obstacles and target points. As shown in Fig. 7, agents can identify target points in different directions and act differently. The upper left of each figure is the current situation of the map, the upper right is the intermediate layer visualization of the convolutional network, the activated neuron is lighter, and the below is the evaluation of different actions in the current situation, i.e., the Q value of different actions in this state. R, L, D, U, L-D, L-U, R-U, R-D, S mean right, left, up, lower left, upper left, upper right, lower right, and stagnant, respectively. The result shows that corresponding neurons will be activated to help the agent make decisions when the target occurs in different directions, the agent can learn how to utilize the potential field energy to head for the target without prior knowledge.

The process of an agent searches for a target as shown in Fig. 8, there is an obstacle between the agent and the target point, and the agent needs to bypass the obstacle through

TABLE 3. The experimental results of different methods over independent 20 runs. The significant difference is represented by '+'.

Environment	Statistic	BHDQN		EAPF		PBPF	
		Length	Time(s)	Length	Time(s)	Length	Time(s)
Env1	Mean	24.04	0.89	Fail	Fail	Fail	Fail
	t-test	-	-	-	-	-	-
Env2	Mean	21.82	0.92	24.08	5.30	23.75	14.49
	t-test	-	-	1.68e-4(+)	2.54e-11(+)	3.83e-6(+)	1.01e-13(+)
Env3	Mean	18.38	0.68	29.62	12.49	29.59	17.77
	t-test	-	-	4.25e-28(+)	1.27e-14(+)	2.64e-31(+)	1.20e-10(+)
Env4	Mean	21.21	0.83	20.47	19.04	20.87	65.88
	t-test	-	-	-	4.77e-13(+)	-	1.27e-9(+)

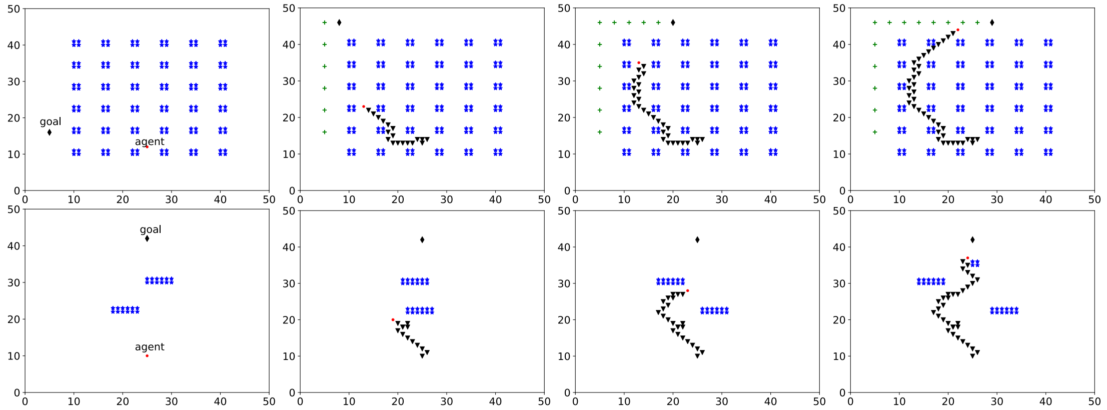


FIGURE 12. Path planning of the agent in dynamic warehouse with moving target or obstacles.

its observation. In the beginning, the agent detects the field energy of the target and move to the corresponding direction because the Q-value of R-U is highest. The Q-value of R-U decreases along with the reduced distance from the obstacle, and the agent turns to L-U to avoid a collision. This experiment proves that trained neurons spontaneously react to field energy of targets and obstacles.

B. CURRICULUM LEARNING

The artificial potential field has the local-stable-point problem which may be caused by different shapes of obstacles such as 'U' shape and orthogonal shape or gravitational superposition of obstacles and target field, as shown in Fig. 9.

For these cases, the agent fails to avoid the obstacles because of new types of local-stable-point. To enhance the adaptability of the agent in BHDQN, we use the curriculum learning (CL) [42]. CL is one of the learning processes of RL. Agents start the training with a simple, basic environment to obtain the initial policies. Then, these policies can be used to adapt to more complex cases. The method which learns a universal policy and applies it into a series of related tasks that have the increasing difficulty is called curriculum learning. It is difficult for the agent to learn how to jump out of the local-stable-point in difficult situations directly. Thus, the CL is used to train the simple square obstacle with 100 epochs, and then the training process can be extended to the complex environment. After the basic training in Section V-A, the agent can identify targets or obstacles and continue training according to different local-stable-point situations in Fig. 9. A trained agent after basic training adopts new surroundings only after 20 epochs. A comparison with the evolutionary artificial potential field (EAPF) [5] and the pseudo-bacterial potential field (PBPF) [6] is shown in Table 3. The result shows that BHDQN has better real-time performance and stable path planning capability.

C. BHDQN IN WAREHOUSE

We test our method in a warehouse which comprises of shelves, a warehouse mobile robot, and free space as shown in Fig. 10. The shelves are regarded as static obstacles and the warehouse mobile robot needs to plan the path within the free space and without collision.

The warehouse environment is represented by a grid map and each shelf or a warehouse mobile robot occupies a grid. The map is based on the parallel layout warehouse and fishbone layout [43]. Results in a static environment are shown in Fig. 11. A local-stable-point exists in the warehouse because of the location of shelves, an agent can jump out from the local-stable-point of warehouse relying on its knowledge.

The agent in the warehouse needs to deal with dynamic situations, such as moving objects, emergent obstacles, and moving obstacles. We test the ability of the trained agents to adapt to dynamic warehouse environments. The target point on the map keeps moving. The trained agent tracks the target point between dense obstacles on the map in the first environment. The target remains stationary and there are moving obstacles and sudden obstacles in the second environment. The agent with BFDQN can adapt to the dynamic environment and complete the target tracking in real-time as shown in Fig.12.

VI. CONCLUSION AND FUTURE WORK

We improve the traditional artificial potential field method with reinforcement learning and propose a new method of path planning. This method enables the agent to find the target point in a multi-target environment. At the same time, a trained agent can adapt to scenarios containing new types of obstacles quickly and dynamic target real-time. Trained neurons react to the partial observation of the potential field and make decisions without human intervention.

The artificial potential field method that combines the black-hole domain can help the agent to jump out of the local-stable-point. The size of the domain needs to be set in advance to adapt to different environments. The range of the too-large domain will cause the superposition of multiple gravitational fields, and the too-small domain value cannot be detected by the agent. We need to further improve the adaptability of BHPF to different environments. We will present the self-adaption black-hole PF which expands its domain according to environment info in our future work.

Additionally, there are multiple agents in real-world scenarios such as intelligent warehouses and unmanned aerial vehicles. New problems arise with multi-agent systems such as collisions and deadlocks that can cause collisions or congestions. Multiple close robots will access to a target meanwhile and cause path redundancy. In order to optimize the application of BHPF in multiple agents situations. We will further improve the BHPF algorithm and establish potential field functions for agents to avoid collisions between agents. We plan to combine the black-hole potential field and multi-agent reinforcement learning algorithms to strengthen agents cooperation in task scheduling and path planning under potential field.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive advice.

REFERENCES

[1] Y. Fu, M. Zhou, X. Guo, and L. Qi, “Artificial-molecule-based chemical reaction optimization for flow shop scheduling problem with deteriorating and learning effects,” *IEEE Access*, vol. 7, pp. 53429–53440, 2019.

[2] X. Guo, M. Zhou, S. Liu, and L. Qi, “Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints,” *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3307–3317, Jul. 2020.

[3] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, “Active autonomous aerial exploration for ground robot path planning,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 664–671, Apr. 2017.

[4] J. Han and Y. Seo, “Mobile robot path planning with surrounding point set and path improvement,” *Appl. Soft Comput.*, vol. 57, pp. 35–47, Aug. 2017.

[5] P. Vadakkepat, T. Heng Lee, and L. Xin, “Application of evolutionary artificial potential field in robot soccer system,” in *Proc. Joint 9th IFSA World Congr. 20th NAFIPS Int. Conf.*, Jul. 2001, pp. 2781–2785.

[6] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, “Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation,” *Int. J. Adv. Robotic Syst.*, vol. 12, no. 7, p. 81, Jul. 2015.

[7] X. Liang, W. Li, Y. Zhang, and M. Zhou, “An adaptive particle swarm optimization method based on clustering,” *Soft Comput.*, vol. 19, no. 2, pp. 431–448, Feb. 2015.

[8] J. Li, J. Zhang, C. Jiang, and M. Zhou, “Composite particle swarm optimizer with historical memory for function optimization,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.

[9] D. Hsu, J.-C. Latombe, and R. Motwani, “Path planning in expansive configuration spaces,” in *Proc. Int. Conf. Robot. Autom.*, Apr. 1997, pp. 2719–2726.

[10] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[11] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” in *Robotics Science and Systems*, vol. 104, no. 2. Cambridge, MA, USA: MIT Press, May 2010.

[12] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.

[13] C.-C. Kao, C.-M. Lin, and J.-G. Juang, “Application of potential field method and optimal path planning to mobile robot control,” in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2015, pp. 1552–1554.

[14] Q. Jia and X. Wang, “An improved potential field method for path planning,” in *Proc. Chin. Control Decis. Conf.*, May 2010, pp. 2265–2270.

[15] G. Li, A. Yamashita, H. Asama, and Y. Tamura, “An efficient improved artificial potential field based regression search method for robot path planning,” in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Aug. 2012, pp. 1227–1232.

[16] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, “Mobile robot path planning using membrane evolutionary artificial potential field,” *Appl. Soft Comput.*, vol. 77, pp. 236–251, Apr. 2019.

[17] A. A. A. Rizqi, A. I. Cahyadi, and T. B. Adji, “Path planning and formation control via potential function for UAV quadrotor,” in *Proc. Int. Conf. Adv. Robot. Intell. Syst. (ARIS)*, Jun. 2014, pp. 165–170.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[19] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[20] Z. Liu, Z. Zheng, X. Guo, L. Qi, J. Gui, D. Fu, Q. Yao, and L. Jin, “AttentiveHerb: A novel method for traditional medicine prescription generation,” *IEEE Access*, vol. 7, pp. 139069–139085, 2019.

[21] D. Xiong and L. Yan, “A classification learning research based on discriminative knowledge-leverage transfer,” *Int. J. Ambient Comput. Intell.*, vol. 9, no. 4, pp. 52–68, Oct. 2018.

[22] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, “A novel GRU-RNN network model for dynamic path planning of mobile robot,” *IEEE Access*, vol. 7, pp. 15140–15151, 2019.

[23] L. Tai, S. Li, and M. Liu, “A deep-network solution towards model-less obstacle avoidance,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2759–2764.

[24] A. Giusti, J. Guzzi, D. C. Ciresan, F.-L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, “A machine learning approach to visual perception of forest trails for mobile robots,” *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 661–667, Jul. 2016.

[25] M. Dragoicea, I. Dumitrache, and N. Constantin, “Adaptive neural control for mobile robots autonomous navigation,” 2015, *arXiv:1512.03351*. [Online]. Available: http://arxiv.org/abs/1512.03351

[26] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, p. 484, 2016.

[27] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[28] L. Qi, W. Luan, X. S. Lu, and X. Guo, "Shared P-Type logic Petri net composition and property analysis: A vector computational method," *IEEE Access*, vol. 8, pp. 34644–34653, 2020.

[29] L. Qi, M. Zhou, and W. Luan, "A dynamic road incident information delivery strategy to reduce urban traffic congestion," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 5, pp. 934–945, Sep. 2018.

[30] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," *IEEE Trans. Autom. Sci. Eng.*, early access, Jun. 19, 2020, doi: 10.1109/TASE.2020.2992220.

[31] X. Guo, S. Liu, M. Zhou, and G. Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2435–2446, Nov. 2016.

[32] Z. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Decomposition method for new single-machine scheduling problems from steel production systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1376–1387, Jul. 2019.

[33] C. Wang, X. Gaimu, C. Li, H. Zou, and W. Wang, "Smart mobile crowdsensing with urban vehicles: A deep reinforcement learning perspective," *IEEE Access*, vol. 7, pp. 37334–37341, 2019.

[34] M. Wei, S. Wang, J. Zheng, and D. Chen, "UGV navigation optimization aided by reinforcement learning-based path tracking," *IEEE Access*, vol. 6, pp. 57814–57825, 2018.

[35] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 31–36.

[36] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," 2016, *arXiv:1611.03673*. [Online]. Available: http://arxiv.org/abs/1611.03673

[37] Y. Aviv and A. Pazgal, "A partially observed Markov decision process for dynamic pricing," *Manage. Sci.*, vol. 51, no. 9, pp. 1400–1416, Sep. 2005.

[38] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Comput. Speech Lang.*, vol. 21, no. 2, pp. 393–422, Apr. 2007.

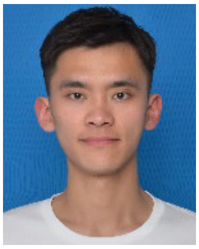
[39] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. and Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1928–1937.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[41] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*. [Online]. Available: http://arxiv.org/abs/1511.05952

[42] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[43] N. V. Kumar and C. S. Kumar, "Development of collision free path planning algorithm for warehouse mobile robot," *Procedia Comput. Sci.*, vol. 133, pp. 456–463, Jul. 2018.



QINGFENG YAO received the B.S. degree in electronic and information engineering from the University of Science and Technology Liaoning, Anshan, China, in 2017. He is currently pursuing the master's degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include neural networks and reinforcement learning.

ZEYU ZHENG received the B.S. degree in mechanical engineering from Zhejiang University, Zhejiang, China, in 1997, and the Ph.D. degree from the Graduate University for Advanced Studies, Japan, in 2005. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Shenyang, China. He has authored nearly 50 technical papers in journals and conference proceedings. His research interests include intelligent optimization algorithms, big data processing technology, data mining, project management, and complex systems.

LIANG QI (Member, IEEE) received the B.S. degree in information and computing science and the M.S. degree in computer software and theory from the Shandong University of Science and Technology, Qingdao, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2017. From 2015 to 2017, he was a Visiting Student with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He is currently with the Shandong University of Science and Technology. He has published over 60 papers in journals and conference proceedings, including the IEEE Transactions on Intelligent Transportation Systems, the IEEE/CAA Journal of Automatica Sinica, the IEEE Transactions on Systems, Man and Cybernetics: Systems, the IEEE Transactions on Computational Social Systems, the IEEE Transactions on Automation Science and Engineering, and the IEEE Transactions on Cybernetics. His current research interests include Petri nets, optimization algorithms, machine learning, and intelligent transportation systems. He received the Best Student Paper Award-Finalist from the 15th IEEE International Conference on Networking, Sensing and Control (ICNSC), in 2018.

HAITAO YUAN (Member, IEEE) received the M.S. and B.S. degrees in Software Engineering from Northeastern University, Shenyang, China, in 2010 and 2012, respectively, the Ph.D. degree in modeling simulation theory and technology from Beihang University, Beijing, China, in 2016, and the Ph.D. degree in computer engineering from the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 2020. He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He was an Associate Professor with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. He was a Ph.D. student with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, from 2013 to 2014. He was also a Visiting Doctoral Student with NJIT, in 2015. He has over 70 publications in international journals and conference proceedings, including ACM Transactions on Internet Technology, IEEE Transactions on Cloud Computing, the IEEE Transactions on Automation Science and Engineering, the IEEE Transactions on Services Computing, the IEEE Transactions on Industrial Informatics and the IEEE Transactions on Cybernetics. His research interests include cloud computing, edge computing, data centers, big data, machine learning, deep learning and optimization algorithms. He was the recipient of the 2011 Google Excellence Scholarship and the recipient of the Best Paper Award-Finalist in the 16th IEEE International Conference on Networking, Sensing and Control.

XIWANG GUO (Member, IEEE) received the B.S. degree in computer science and technology from the Shenyang Institute of Engineering, Shenyang, China, in 2006, the M.S. degree in aeronautics and astronautics manufacturing engineering from Shenyang Aerospace University, Shenyang, in 2009, and the Ph.D. degree in system engineering from Northeastern University, Shenyang, in 2015. From 2016 to 2018, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He is currently an Associate Professor with the College of Computer and Communication Engineering, Liaoning Shihua University. He has authored more than 30 technical papers in journals and conference proceedings, including the IEEE Transactions on cybernetics, the IEEE Transactions on Systems, Man and Cybernetics: Systems, the IEEE Transactions on Intelligent Transportation Systems, and the IEEE/CAA Journal of Automatica Sinica. His current research interests include Petri nets, remanufacturing, recycling and reuse of automotive, and intelligent optimization algorithm.

MING ZHAO received the B.S. degree in automation from Qufu Normal University, Rizhao, China, in 2018. She is currently pursuing the master's degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. Her current research interests include machine learning, deep learning, and computer vision.

ZHI LIU received the B.S. degree in software engineering from Changchun University, Changchun, China, in 2014, and the M.S. degree from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2017. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang. His current research interests include machine learning, deep learning, intelligent medical, natural language processing, and computer vision.

TIANJI YANG received the M.S. degree in industrial engineering and the Ph.D. degree in management science and engineering from the Hefei University of Technology (HFUT), Hefei, China, in 2012 and 2017, respectively. He was a Visiting Research Scholar with the Department of Industrial Engineering, University of Arkansas, Fayetteville, AR, USA, from 2016 to 2017. He is currently an Assistant Researcher with the Shenyang Institution of Automation, Chinese Academy of Sciences, Shenyang, China. He has publications in international journals and conference proceedings, including *Journal of Combinatorial Optimization* and *European Journal of Operational Research*. His research interests include supply chain management, operation research, optimization, reliability, and machine learning.