An Improved Algorithm for Complete Coverage Path Planning Based on Biologically Inspired Neural Network

Linhui Han, Xiangquan Tan, Qingwen Wu, and Xu Deng

Abstract—**Complete coverage path planning (CCPP) requires the mobile robots to traverse every part of the workspace, which is one of the major challenges in cleaning robots and many other robotic systems. The biologically inspired neural network (BINN) algorithm has been extensively applied in path planning, recently. In this article, a new CCPP strategy with BINN is proposed. The planned path of cleaning robot is not only determined by the dynamic neural activities but also by the distribution of obstacles in the environmental map. By distinguishing the connectivity between different areas of the environmental map, and using the proposed path backtracking algorithm, the improved CCPP algorithm can autonomously plan a collision-free path and reduce the path repetition ratio. Besides, an improved dynamic deadlock escape algorithm is presented to select the optimal escape target point. The simulation results show that the proposed CCPP algorithm without any templates or learning procedures is able to generate an orderly path in both known and unknown environment.**

Index Terms—**Biologically inspired neural network (BINN), cleaning robots, complete coverage, path planning.**

I. Introduction

PATH planning is a fundamentally important issue in mobile robots. Over the last two decades, complete coverage path planning (CCPP) has been considered as an important research issue in the intelligent robotics, which requires the mobile robots to traverse every part of the workspace. The goal of the CCPP is to create an optimal coverage path by reducing the travel time, processing speed, energy consumption, repetition ratio, and the number of turns along the path, which reflect the efficiency of CCPP [1], [2]. This task is requisite to many robotic applications, such as indoor cleaning robots [3], [4], [5], autonomous underwater vehicles [6], [7], wall-climbing robots [8], structural inspection robots [9], [10], and rescue robots [11], [12], just to name a few.

Cleaning robots, as the most common indoor mobile robots, have made a huge commercial success in recent years. However, numerous cleaning robots on the market still adopt the random walking method to the path planning problem. Although the strategy of random walking can cover the whole free space [13], many researchers are devoted to finding a more effective method because a random walk process suffers from time cost and path repetition [14]. The requirements that cleaning robots should follow are shown below [15].

1) Robot should move through every part of the workspace.
2) Robot should avoid the overlap of the path.
3) Robot should avoid collisions with obstacles.
4) Robot should take simple path (e.g., straight lines or circles).
5) Robot should avoid deadlocks.

However, it is difficult to satisfy all these conditions in complex environments. The basic CCPP process consists of the following steps. First, robots use sensors such as lidar to scan environmental boundaries and obstacles to build an environmental map for path planning. Next, robots begin cleaning according to the planned path, such as zigzag path [16] and spiral path [17], to complete the full coverage of the map. During the cleaning process, robots may be surrounded by the cleaned path and get trapped in deadlocks. Robots should be able to escape from deadlocks by using point-to-point path planning algorithms, such as A* [18], [19] and Dijkstra [20], which allow the robots to find a new position that have not been traveled. The robots repeat the above process until the cleaning robots have passed all positions on the map and complete the coverage cleaning task.

The CCPP algorithm requires mobile robots to pass through every position of the workspace, it is necessary to distinguish movable space and obstacles while establishing the environmental map. The grid-based representation method was first proposed by Moravec and Elfes [21] to map an indoor environment using a sonar ring mounted on mobile robots. In this representation, obstacles and free space can be represented by numbers [22], which brings convenience to path planning and has become an important research method for many scholars. An approach of utilizing grid representation is

Manuscript received 4 July 2022; revised 27 October 2022; accepted 12 January 2023. Date of publication 17 January 2023; date of current version 11 September 2023. *(Corresponding author: Xiangquan Tan.)*

Linhui Han and Xu Deng are with the Changchun Institute of Optics Fine Mechanics and Physics, Chinese Academy of Science, Changchun 130033, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: hanlh1996@163.com; dengxujys@163.com).

Xiangquan Tan is with the CAS Key Laboratory of On-Orbit Manufacturing and Integration for Space Optics System, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China (e-mail: tanxq@ciomp.ac.cn).

Qingwen Wu is with the CAS Key Laboratory of On-Orbit Manufacturing and Integration for Space Optics System, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China, and also with the Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wuqw@ciomp.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCDS.2023.3237612.

Digital Object Identifier 10.1109/TCDS.2023.3237612

the spiral-spanning tree coverage (STC) algorithm [23], which generates the systematic spiral path by following a spanning tree of the partial grid map. Le et al. [24] proposed a method of exploiting the Tetris-inspired self-reconfigurable robot hTetro for optimizing area covering. A novel neural network method for CCPP with obstacle avoidance of cleaning robots in nonstationary environments is proposed in [25] and [26].

With the rapid development of artificial intelligence technology, artificial intelligence algorithms, such as neural network algorithm and genetic algorithm, are widely used in CCPP. However, many artificial neural network algorithms suffer from the problem of long learning time and low learning efficiency. Yang and Luo [27] proposed a biologically inspired neural network (BINN) approach for CCPP, and it is a novel algorithm which can autonomously generate robot path from the dynamic activity landscape of the neural network and the previous robot location. Qiu et al. [28] proposed a novel rolling path planning and heuristic searching approach based on the BINNs, which improved applicability and effectiveness in an uncertain environment. Luo and Yang [29] proposed a neural-dynamics-based algorithm for real-time concurrent map building and CCPP in unknown environments. Luo et al. [30] proposed a biologically inspired neural dynamics algorithm for multirobot coordinated navigation of CCPP. The approach reduced completion time by sharing complete coverage tasks. Sun et al. [7] proposed the multi-AUV full coverage discrete and centralized programming method, which achieved full coverage of the mission area in the underwater environment. A CCPP method is proposed for hull inspection robot [31], which has good applicability to the dynamic workspace by considering the energy consumption of the maintenance robot when the direction, distance, and vertical position have changed.

It is convenient to establish the environment maps using the grid method and the CCPP based on the BINN algorithm can deal with changing environments without any learning procedures. However, there are still some shortcomings in the methods introduced above such as the disorderly coverage paths, high repetition ratio in complex environment, and being trapped in deadlocks. In order to make the algorithm can be applied to the complex real indoor environment, it is necessary to improve the original path planning method by optimizing the path selection strategy.

The original algorithm provides a new method for CCPP and gets proved in unstructured environment [27]. However, the original model generates new paths by only computing the data in the neural network, which lacks the consideration of the feature information of the map. In view of the above issues, this article proposed an improved algorithm of CCPP based on BINN, which includes new algorithms of real-time optimization of generated paths and deadlock escape. The improved algorithm is able to reduce path repetitions in both known and unknown complex environment, the main innovations of the proposed algorithm are as follows.

1) In order to reduce the path repetition ratio, the new area connectivity detection and path backtracking algorithm are proposed. When the mobile robot encounters an obstacle or covered location, the area connectivity detection algorithm is executed. If the workspace is detected to be separated by the path of the robot, the path backtracking algorithm is executed. Instead of advancing according to the neural activity, the path backtracking algorithm will select the backtracking point and change the robot's movement direction to another path. The algorithm can effectively reduce path repetitions and improve the applicability of the CCPP algorithm in the complex environment.
2) In order to solve the deadlock problem, a dynamic deadlock escape algorithm is proposed. When the mobile robot is trapped in deadlocks, the robot can dynamically select the optimal escape point by computing 2-D Euclidean distances and neural activities of the possible target points. The algorithm can make the robot escape the deadlock quickly and reduce the path repetitions during the escape process.

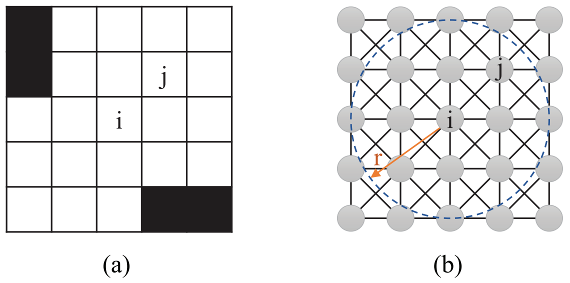


Fig. 1. Correspondence between 2-D environmental map and the neural network. (a) Discretized grid map of workspace. (b) Neural network.

This article is organized as follows. In Section II, the original model of CCPP based on the BINN algorithm is presented. In Section III, the improved approaches are introduced, including area connectivity detection and path backtracking algorithm and dynamic deadlock escape algorithm. Simulation results and discussions of path planning in different environmental maps are presented in Section IV. Finally, the conclusion is given in Section V.

II. Original Model

In this section, the original model of CCPP based on the BINN algorithm is introduced.

A. Grid Map Building and Neural Network Representation

The objective of the algorithm is to propose a dynamic neural network architecture that can reflect the feature information of the environment. The neural activity of the neural network architecture landscape is used to guide the mobile robot to choose the appropriate moving direction. For this purpose, the grid-based representation method is used to build the environmental map in which free space and obstacle can be distinguished by different cells. Each cell has two states: 1) free space and 2) obstacle, as shown in Fig. 1(a), the black cells are obstacles, and the cells in white indicate the accessible areas.

In the grid map, each cell with its own neural activity is considered as a neuron, and each neuron has lateral connections only to neighboring neurons to form a neural network. As shown in Fig. 1(b), a 2-D neural network is built on the grid map, the ith cell and jth cell in Fig. 1(a) correspond to the ith neuron and jth neuron in Fig. 1(b), respectively. In Fig. 1(b), the receptive field of the ith neuron is represented by a circle with radius r, indicating that the neuron responds only to the stimulus within its receptive field.

B. Model Algorithm

A computational membrane model in a biological neural system was proposed by Hodgkin and Huxley [32]. In this model, the dynamics of the membrane voltage V_m is described by a state equation as

$$C_m \frac{dV_m}{dt} = -\left(E_p + V_m\right)g_p + (E_{Na} - V_m)g_{Na} - (E_k + V_m)g_K \tag{1}$$

where C_m is the capacitance of membrane, E_k, E_{Na}, and E_p represent the Nernst potentials for potassium ions, sodium ions, and passive leak current in the membrane, respectively, and g_k, g_{Na}, and g_p are the conductance of potassium, sodium, and passive channels, respectively. A shunting equation can be obtained by changing the formulation of (1)

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e(t) - (D + x_i)S_i^i(t) \tag{2}$$

where x_i is regarded as the neural activity of the ith neuron, A is the attenuation rate, and B and D represent the upper and lower bounds of the neural activity. And S_i^e and S_i^i are the external excitatory and inhibitory inputs of the neuron. The shunting (2) was first proposed by Grossberg [33], which could be applied to visual perception and motor control.

Yang and Luo [27] improved the shunting (2), established a dynamic neural network structure, and applied it to the CCPP of the cleaning robot. By defining the appropriate external inputs, the uncleaned grid, due to its higher neural activity, can globally attract the robot to automatically travel a smooth path to clean all the areas. The topologically organized neural network model is expressed in a 2-D Cartesian space. The neural activity x_i of the ith neuron can be characterized by an equation as

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)\left([I_i]^+ + \sum_{j=1}^{k} \omega_{ij}\left[x_j\right]^+\right) - (D + x_i)[I_i]^- \tag{3}$$

where x_j is the neural activity of the jth neuron, and k represents the number of neurons which share connections with the ith neuron in the receptive field. The terms $[I_i]^+ + \sum_{j=1}^{k} \omega_{ij}[x_j]^+$ and $[I_i]^-$ are the external excitatory and inhibitory inputs of the neuron. Functions $[f]^+$ and $[f]^-$ are defined as $[f]^+ = \max\{f, 0\}$ and $[f]^- = \max\{-f, 0\}$, respectively. ω_{ij} represents the weight of connection between the ith and jth neurons, which is defined as

$$\omega_{ij} = \begin{cases} \frac{\mu}{|i-j|}, & 0 < |i-j| < r \\ 0, & |i-j| \geq r \end{cases} \tag{4}$$

where $|i-j|$ represents the Euclidean distance between the neurons i and j in the state space. μ and r are positive constants, the value of r determines the size of the receptive field of the nerve connection. As the neural network shown in Fig. 1(b), where $r = 2$, the ith neuron has eight external connections with the neighboring neurons. The external input I_i is defined as

$$I_i = \begin{cases} E, & \text{if it is an uncovered area} \\ -E, & \text{if it is an obstacle area} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where E and $-E$ are the value of external excitatory input and inhibitory input. E usually is a very large positive constant which represents the effects of environmental features on neurons.

In (3), only excitatory signals can be transmitted through the connection, the shunting equation guarantees that the positive neural activity can be propagated to the whole state space through the neural network, but the negative neural activity can be restricted to the local location. Therefore, in the actual cleaning process, the uncovered area will globally attract the cleaning robot by the neural activity, and the obstacles will locally reject the robot to approach.

For cleaning efficiency, the robot should travel through an orderly path by having fewer path repetitions and making fewer turns of moving direction. The current position of robot is noted by p_c, the next position p_n is defined as

$$P_n \Leftarrow x_{p_n} = \max\left\{x_j + cy_j\right\}, j = 1, 2, \ldots, k \tag{6}$$

where c is a constant. y_j is a function describing the turning of the robot, which can be determined by the robot's previous position p_p, current position p_c, and next position p_n. y_j can be defined as

$$y_j = 1 - \frac{\Delta\theta_j}{\pi} \tag{7}$$

$$\Delta\theta_j = \begin{cases} \Delta\theta_r, & x \leq \pi \\ |\Delta\theta_r - 2\pi|, & x > \pi \end{cases} \tag{8}$$

$$\Delta\theta_r = |\theta_j - \theta_c| = |\text{atan2}\left(y_{p_j} - y_{p_c}, x_{p_j} - x_{p_c}\right) - \text{atan2}\left(y_{p_c} - y_{p_p}, x_{p_c} - x_{p_p}\right)| \tag{9}$$

where $\Delta\theta_j \in [o, \pi]$ represents the angle change between the moving directions of the current and next moment. Variable $\text{atan2}(y_a, x_a) \in (-\pi, \pi]$ is a four-quadrant arctangent function. When the robot arrives the next position, the next position becomes the new current position. According to the dynamic neural network, the robot can plan a complete coverage path with fewer turns by (3) and (6).

The algorithm for CCPP based on BINN does not need any prior information or manual intervention. In addition, the neural network is a stable system. By using the Lyapunov stability theory, Yang and Meng [34] provided a rigorous proof of the stability and convergence of the neural network model.

III. Improved Model and Complete Coverage Path Planning Algorithm

In this section, the improved algorithm for CCPP based on BINN is proposed. The new area connectivity detection and path backtracking algorithm is introduced. And a dynamic deadlock escape algorithm is presented to generate an optimal escape path from the deadlock.

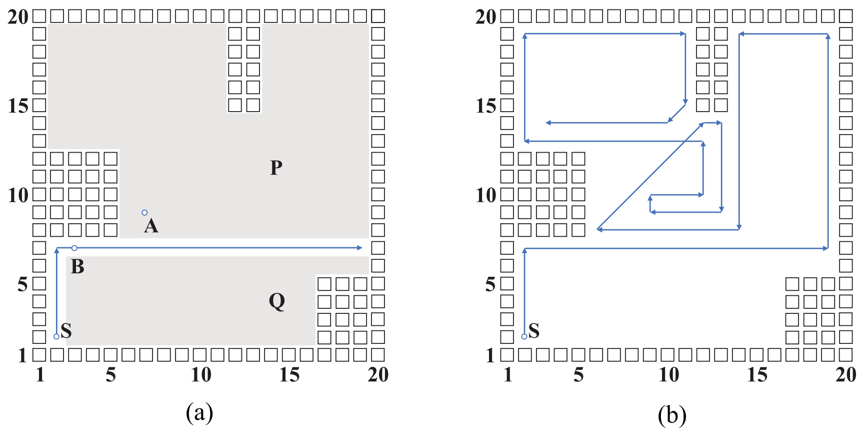


Fig. 2. Disorderly coverage path using the original algorithm. (a) Path without changing direction. (b) Unexpected path.

The following notations are used in this section. And N_x and N_y represent the discretized size of the Cartesian workspace.

(m, n)	The position in the Cartesian workspace, $1 \leq m \leq N_x$, $1 \leq n \leq N_y$.
$N(m, n)$	The neuron at position (m, n), $1 \leq m \leq N_x$, $1 \leq n \leq N_y$.
$f(m, n)$	The flag that indicates the status of the neuron at position (m, n).
$\mathcal{N}(m, n)$	The set of the discretized workspace, $\{(m, n), 1 \leq m \leq N_x, 1 \leq n \leq N_y\}$.
$x(m, n)$	The neural activity at position (m, n)
(m_p, n_p)	The previous position of the robot.
(m_c, n_c)	The current position of the robot.
(m_n, n_n)	The next position of the robot, decided by (3) and (6).
θ_c	The current moving direction of the robot, decided by (m_p, n_p) and (m_c, n_c).
θ_n	The next moving direction of the robot, decided by (m_c, n_c) and (m_n, n_n).
$I(m, n)$	The external input to neuron $N(m, n)$.

In addition, $f(m, n)$ represents the environmental characteristic of the position (m, n), e.g., if (m, n) is obstacle, $f(m, n)$ is defined as -1; if (m, n) is the uncleaned area, $f(m, n) = 0$; if (m, n) is the cleaned area, $f(m, n) = 1$. And $t \in \{t_1, t_2, \ldots, t_c\}$ is the time series in the state space, and $(m_c(t), n_c(t))$ represents the position of the robot at time t. The set of path points at time t is defined as $\mathcal{T}(t) = \{(m_c(t), n_c(t)), t \in \{t_1, t_2, \ldots, t_c\}\}$. $\theta_c(t)$ represents the moving direction of the robot at time t, decided by $(m_p(t), n_p(t))$ and $(m_c(t), n_c(t))$.

A. Area Connectivity Detection and Path Backtracking Algorithm

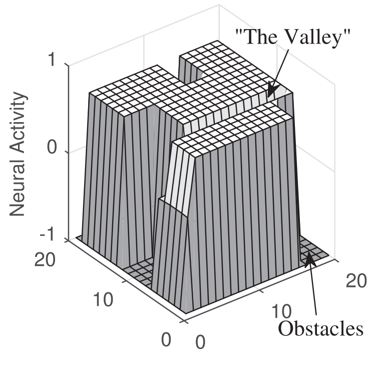
The neural network model proposed above can generate a complete coverage path without prior map information. However, driven by (3) and (6), the robot will stick to moving along in the cleaning process, and the movement direction will change unexpectedly when it encounters the obstacle because the neural activities of neurons close to the obstacles are lower than others. The disorderly coverage path is shown in Fig. 2.

Some scholars [6], [7] normalized the robot's moving method by setting templates when the robot approaches the obstacles, while others [8] optimize the path by changing the path point selection function. The new area connectivity detection and path backtracking algorithm is proposed in this section to choose the appropriate path point.

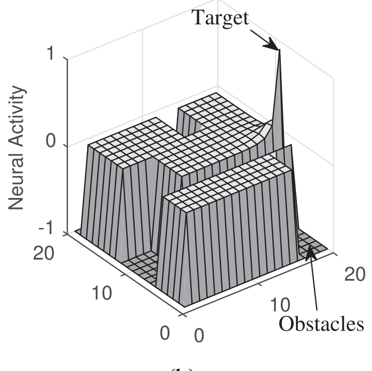
An important requirement of CCPP is to reduce the path repetition ratio. In Fig. 2(a), the path of the robot splits the environmental map into two areas, area P and area Q, the neural activity landscape is shown in Fig. 3(a). If the robot turns to the area P, the overlap of path is inevitable when the robot tries to clean the area Q. This happens frequently in the whole map. An important concept in topology is path connected. For two points A and B in topological space $\mathcal{X}$, if there is a path starting from A and ending at B, then A and B are referred as path connected, noted by $A \sim B$. If any two points in topological space $\mathcal{X}$ are path connected, the space $\mathcal{X}$ is path connected space. If the workspace is always path connected space during the path planning process, the path repetition ratio will be zero.

In order to keep the workspace path connected, the robot will estimate whether area P and area Q are connected by the connectivity detection algorithm when the robot encounters the obstacle or cleaned area, which is given in Algorithm 1. As shown in Fig. 3(a), when areas are disconnected, a "valley" appears on the landscape of neural activity. If the path is considered as obstacles, neural activity in area P cannot propagate to area Q. Therefore, the neural network architecture itself can be used to detect the connectivity of the area. The complete pseudocode of the connectivity detection algorithm is shown in Algorithm 1.

As shown in Fig. 3(b), by setting only one point as the target in area P, which has a large positive external input, the positive neural activity can be propagated to the whole state space through the neural network. But the obstacles between two areas prevent the neural activity from propagating to area Q. The output of Algorithm 1 shows that the areas are not connected, therefore, the path is undesirable. In addition, Algorithm 1 is only executed when the robot encounters the obstacle or cleaned area. To reduce the number of calculations, connectivity detection is performed when the moving



(a)



(b)

Fig. 3. (a) Neural activity landscape when the robot encounters the obstacle. (b) Stable activity landscape of the neural network.

Algorithm 1 Connectivity Detection Algorithm

```
Input: The neural activities of neural network and robot
       path
Output: Logical value of the connectivity $\mathcal{L}$
1. if $\theta_c \neq \theta_n$ then // The robot changes moving direction;
2.   $f(\mathcal{T}(t)) \leftarrow -1$; // Set the path as obstacles;
3.   $x(m,n) \leftarrow 0, \forall 1 \leq m \leq N_x,\ 1 \leq n \leq N_y$;
     // Set all the neural activities as zero;
4.   $I(m,n) \leftarrow 0, \forall 1 \leq m \leq N_x,\ 1 \leq n \leq N_y$;
     // Set all external inputs to neurons as zero;
5.   $\mathcal{N}_w(m_c,n_c) = \{N(m,n) | m \in (m_c-1, m_c, m_c+1)$ and $n \in (n_c-1, n_c, n_c+1)\}$;
     // Find the unknown neighboring neurons;
6.   if $\exists(\alpha_1,\beta_1) \in \mathcal{N}_w(m_c,n_c), s.t.\ f(\alpha_1,\beta_1)=0$ then
7.     $I(\alpha_1,\beta_1) \leftarrow E$;
       // Set external input to one neuron which is not
       path point as $E$;
8.   end if
9.   for $k \leftarrow 1$ to $N_x \cdot N_y$ do
     // Update the neural network;
10.    calculate the neural activities $x(m,n)$,
       $\forall 1 \leq m \leq N_x,\ 1 \leq n \leq N_y$ by (3);
11.  end for
     // $N_x \cdot N_y$ is the total number of neurons in the neural
     network, the neural activities are able to spread to any
     available position after $N_x \cdot N_y$ cycles;
12.  for $i \leftarrow 2$ to 9 do
     // set up a loop to get all the unknown neighboring
     neurons;
13.    get $x(\alpha_i,\beta_i), (\alpha_i,\beta_i) \in \mathcal{N}_w(m_c,n_c)$
       // get the neural activities;
14.    if $\exists(\alpha_i,\beta_i) \in \mathcal{N}_w(m_c,n_c), \alpha_i \neq \alpha_1, \beta_i \neq \beta_1$,
         $s.t.\ x(\alpha_i,\beta_i)=0$ then
15.      $\mathcal{L} \leftarrow 0$; // Areas $P$ and $Q$ are disconnected;
16.    else
17.      $\mathcal{L} \leftarrow 1$; // Areas $P$ and $Q$ are connected;
18.    end if
19.  end for
20. end if
21. return $\mathcal{L}$
```

direction is changed. Without considering the cost of computing time, connectivity detection is required for each step of robot movement.

The backtracking method is widely used in path planning to settle accessibility issues. The efficiency of the backtracking method depends on choosing the appropriate backtracking point, the path points constitute a complete list of backtracking points in CCPP. The set of backtracking points is noted as $B(m_c(t), n_c(t))$, where $t \in \{t_1, t_2, \ldots, t_c\}$.

The path in Fig. 2(a) needs to be optimized to keep the workspace path connected. The robot should change the moving direction in advance rather than follow a fixed direction until it encounters an obstacle. In order to generate the orderly zigzag path, point B in Fig. 2(a) which is the first point after the movement direction of robot changes is selected to be the backtracking point. The complete pseudocode of the path backtracking algorithm is shown in Algorithm 2.

The backtracking point and the new path of the robot are generated by Algorithm 2. The path is backtracked to the first position after the movement direction of the robot changes, and the algorithm guarantees that the robot will make another turn at the backtracking point. The new path is shown in Fig. 4(a), which avoids splitting the map into two parts and leads to an orderly complete coverage path. The activity landscape of the neural network is shown in Fig. 4(b).

The model based on BINN is able to generate a complete coverage robot path without any prior information of the environment or any learning procedures. On the other hand, it is difficult to generate the optimal path without the feature information of the map. The proposed Algorithms 1 and 2 add the feature information of map to the neural network model by the method of *advancing-estimating-backtracking*. Essentially, the proposed algorithms greatly improve the model's ability to solve the global path planning problem by providing the feature information.

In addition, the proposed area connectivity detection and path backtracking algorithm can also be applied in an unknown environment. The environmental map is simultaneously built during the cleaning process in the unknown environment, and the boundary range of the workspace is known before starting the path planning. The robot can sense a limited area by its onboard robot sensors, like Lidar and camera. The range of the map is determined by the sensing range of the robot sensors, and the planned path is restricted in the area of the established part of the map. The shape and size of obstacles can affect the detection of area connectivity. When the size of the obstacle is larger than the detection range of the sensor, the robot cannot distinguish the connectivity of the area. As shown in Fig. 5(a),

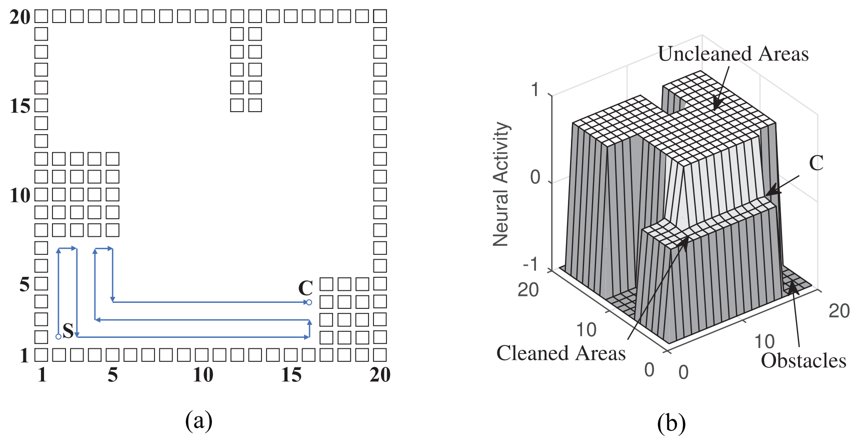


Fig. 4. Generated path when the robot reaches $C(16, 4)$. (a) New path. (b) Activity landscape of the neural network.

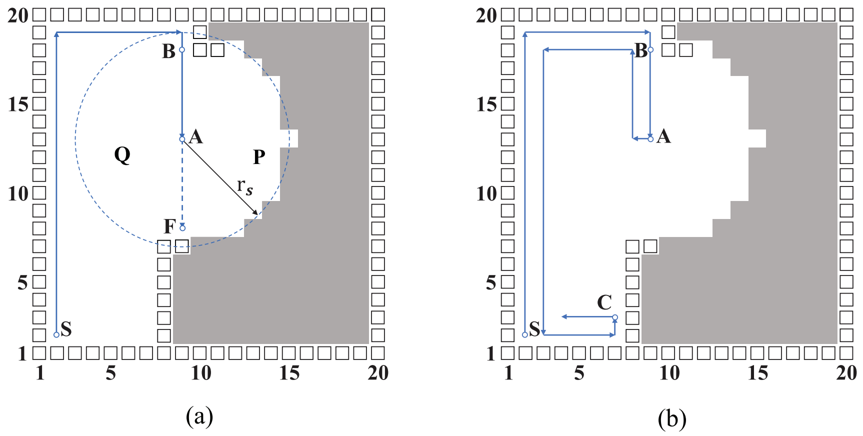


Fig. 5. Generated path in an unknown environment. (a) Path when the robot reaches $A(9, 13)$. (b) Path when the robot reaches $C(7, 3)$.

the gray area is unknown to the robot, and the areas P and Q are first detected as disconnected because the sensing range of the sensor is a circle of radius $r_s = 6$. While the current position of the robot is $A(9, 13)$, the original planned path should be the path from point $A(9, 13)$ to point $F(9, 8)$, but the areas P and Q are disconnected. To avoid the robot moves backward, the backtracking point cannot be selected as $B(9, 18)$, so that the robot will change the moving direction at point A. In Fig. 5(b), when the robot reaches $C(7, 3)$, the local map of the robot's forward direction is completely established, the backtracking point can be selected as point $C(7, 3)$. Therefore, the selection of the backtracking points is related to the sensing range of the sensor and the built map in an unknown environment. Such dynamic selection method of the backtracking points may lead to several overlaps, it is efficient for generating an orderly path in an unknown environment.

B. Dynamic Deadlock Escape Algorithm

During the CCPP, the robot may be trapped into the deadlocks by its own path. In the model based on BINN, the deadlock situation is that the neighboring neurons of the current neural $N(m_c, n_c)$ are either cleaned areas or obstacles. All the neighboring neurons and the central neuron have equally small neural activities, the robot may be trapped into several covered path points because it cannot choose an appropriate path point by (6). One method to escape the deadlock is waiting for the large neural activity spreading through the neural network, the robot will be attracted by the uncleaned areas and escape from the deadlock. But the efficiency of path planning is low because of the long time waiting.

The proposed dynamic deadlock escape algorithm is based on the neural network. It shares some common ideas with the point-to-point path planning algorithm in [34], Yang and Meng proposed an efficient path planning method based on the neural network. The pseudocode of the point-to-point path planning algorithm is shown in Algorithm 3. The robot is attracted by the target point (m_t, n_t) with high external input and moves to the target position automatically. The robot path generated by the point-to-point path planning algorithm is shown in Fig. 6.

By computing 2-D Euclidean distances and neural activities, the proposed Algorithm 4 can provide an appropriate target point at the first time. The current position of the robot is noted by (m_c, n_c), the target position T_n noted by (m_t, n_t) is defined as

$$d_{cj} = \sqrt{\left(m_c - m_j\right)^2 + \left(n_c - n_j\right)^2}, j = 1, 2, \ldots, z \quad (10)$$

$$T_n \Leftarrow x_{t_n} = \min\left\{d_{cj}\right\}, j = 1, 2, \ldots, z \quad (11)$$

Algorithm 2 Path Backtracking Algorithm

```
Input: The neural activities of neural network, robot
       path and logical value of the connectivity $\mathcal{L}$
Output: The new path of robot
1. if $\mathcal{L} = 0$ then // The areas are disconnected;
2.     $\bar{\theta_c} \leftarrow \theta_c$; // Mark the moving direction;
3.     while true do
4.         calculate the moving direction $\theta_c(t_c)$;
           // Calculate the moving direction at time $t_c$;
5.         if $\theta_c(t_c) = \bar{\theta_c}$ then
           // The moving direction does not change;
6.             remove $(m_c(t_c), n_c(t_c)) \in \mathcal{T}(t)$;
7.             $(m_n, n_n) \leftarrow (m_c(t_c), n_c(t_c))$;
8.             $(m_c, n_c) \leftarrow (m_c(t_{c-1}), n_c(t_{c-1}))$;
9.             $(m_p, n_p) \leftarrow (m_c(t_{c-2}), n_c(t_{c-2}))$;
               // Reset the position of the robot;
10.            $(m_c(t_c), n_c(t_c)) \leftarrow (m_c, n_c)$;
               // Remove the last path point;
11.        else // The moving direction changes;
12.            add $(m_n, n_n) \in \mathcal{T}(t)$; // Set the next position
               of robot at time $t_c$ as the last path point;
13.            $(m_c, n_c) \leftarrow (m_c(t_c), n_c(t_c))$;
               // Renew the current position of the robot;
14.            Break;
15.        end if
16.    end while // Get the backtracking point;
17.    while true do
18.        find $(\bar{m_n}, \bar{n_n})$ by (3) and (6); // Find the
           possible next position of robot and mark it;
19.        calculate the possible moving direction $\bar{\theta_n}$;
20.        if $\bar{\theta_c} = \bar{\theta_n}$ then // The robot does not change
           the moving direction;
21.            $I(\bar{m_n}, \bar{n_n}) \leftarrow 0$; // Set external input
               to $(\bar{m_n}, \bar{n_n})$ as zero;
22.        else
23.            $(m_n, n_n) \leftarrow (\bar{m_n}, \bar{n_n})$;
24.            add $(m_n, n_n) \in \mathcal{T}(t)$;
25.            Break; // Make sure the robot change the
               moving direction at the backtracking point;
26.        end if
27.    end while
28. end if
29. return $\mathcal{T}(t)$
```

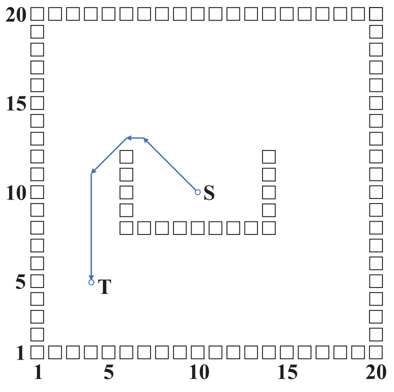


Fig. 6. Path generated by Algorithm 3.

Algorithm 3 Point-to-Point Path Planning Algorithm

```
Input: The 2D Cartesian workspace
Output: The path of robot
1. $x(m, n) \leftarrow 0, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
   // Set all the neural activities as zero;
2. $I(m, n) \leftarrow 0, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
   // Set all external inputs to neurons as zero;
3. $f(m, n) \leftarrow 1, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
   // Set all areas as cleaned;
4. $I(m_t, n_t) \leftarrow E$;
   // Set external input to target point as $E$;
5. repeat
6.     calculate the neural activities $x(m, n)$,
       $\forall 1 \le m \le N_x, 1 \le n \le N_y$ by (3);
7.     $\mathcal{N}_w(m_c, n_c) = \{N(m, n)|m \in (m_c - 1, m_c, m_c + 1)$ and $n \in (n_c - 1, n_c, n_c + 1)\}$;
       // Find the unknown neighboring neurons;
8.     $(m_n, n_n) = \arg\max_{m,n}(x(m, n)) \in \{\mathcal{N}_w|m \in (m_c - 1, m_c, m_c + 1)$ and $n \in (n_c - 1, n_c, n_c + 1)\}$;
       // Find the next position with maximal neural activity;
9.     $(m_c, n_c) \leftarrow (m_n, n_n)$;
       // Renew the current position of robot;
10.    add $(m_c, n_c) \in \mathcal{T}(t)$; // Set the current position of
       robot as the last path point;
11. until $(m_c, n_c) = (m_t, n_t)$
12. return $\mathcal{T}(t)$
```

Algorithm 4 Dynamic Deadlock Escape Algorithm

```
Input: The neural activities of neural network and robot
       path
Output: The path of escape from deadlock
1. repeat
2.     calculate $d_{cj}$ by (10);
3.     find $(m_t, n_t)$ by (11); // Find the target point;
4.     find $(m_n, n_n)$ by Algorithm 3; // Find the next position
       of robot by point-to-point path planning algorithm;
5.     $(m_c, n_c) \leftarrow (m_n, n_n)$;
       // Renew the current position of robot;
6.     add $(m_c, n_c) \in \mathcal{T}(t)$; // Set the current position of
       robot as the last path point;
7. until $(m_c, n_c) = (m_t, n_t)$
8. return $\mathcal{T}(t)$
```

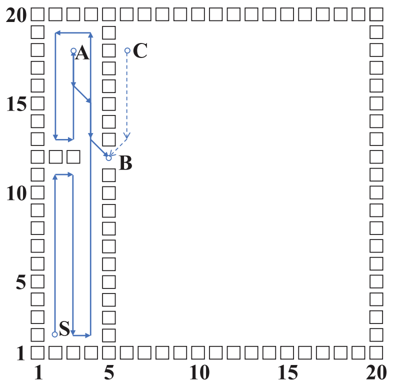
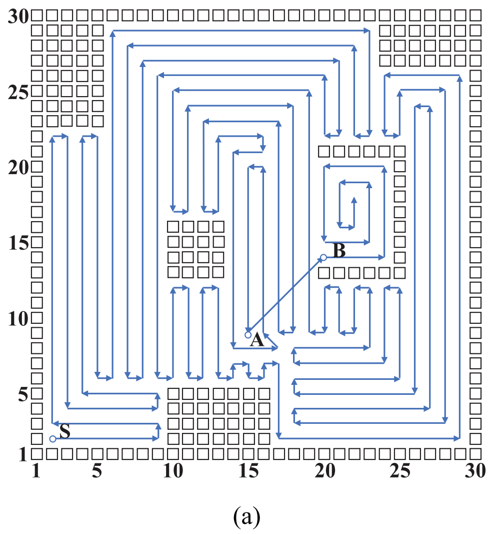


Fig. 7. Path of escape from deadlock.

where d_{cj} is the Euclidean distance between the current position of robot (m_c, n_c) and the possible target position (m_j, n_j), and z is the total number of uncleaned positions in the map. The complete pseudocode of the dynamic deadlock escape algorithm is shown in Algorithm 4.

As shown in Fig. 7, the robot escapes from the deadlock by Algorithm 4. The robot can dynamically change the position of the target point by computing the Euclidean distances between the current position and the uncleaned positions in each step. The position of the target point is changed from



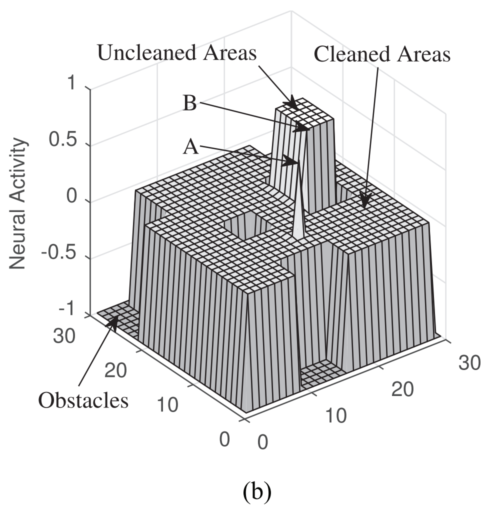


Fig. 8. CCPP. (a) Generated robot path. (b) Activity landscape of the neural network when the robot reaches $A(15, 9)$.

$C(6, 18)$ to $B(5, 12)$. It is obvious that point B is a better escape target than point C.

C. Improved CCPP Algorithm

With Algorithms 1–4, the improved CCPP algorithm based on BINN can plan an orderly path with fewer path repetitions. The pseudocode of the improved CCPP algorithm is shown in Algorithm 5, and the generated robot path is shown in Fig. 8(a).

As shown in Fig. 8(a), the neural network has 30×30 topologically organized neurons, the cleaning robot starts from point $S(2, 2)$. When the robot reaches point $A(15, 9)$ which is a deadlock point, it is able to escape the deadlock by the dynamic deadlock escape algorithm and move to point $B(20, 14)$. The activity landscape of the neural network when the robot reaches $A(15, 9)$ is shown in Fig. 8(b). The improved algorithm generates an orderly path which combines the characteristics of zigzag path and spiral path without any templates or human intervention.

Algorithm 5 Improved CCPP Algorithm

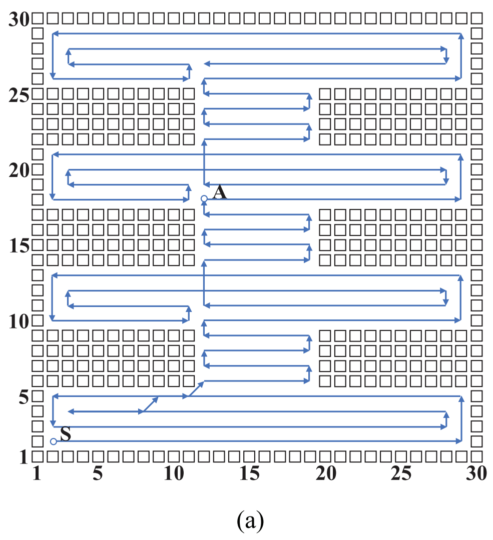
```
Input:  The 2D Cartesian workspace
Output: The path of robot and the neural activities
1. $x(m, n) \leftarrow 0, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
2. $f(m, n) \leftarrow 0, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
   // Initialization;
3. repeat
4.     calculate the neural activities $x(m, n)$,
       $\forall 1 \le m \le N_x, 1 \le n \le N_y$ by (3);
5.     find $(m_n, n_n)$ by (6);
       // Find the next position of robot;
6.     $(m_c, n_c) \leftarrow (m_n, n_n)$;
       // Renew the current position of robot;
7.     add $(m_c, n_c) \in \mathcal{T}(t)$; // Set the current position of
       robot as the last path point;
8.     $I(m_c, n_c) \leftarrow 0$;
       // Set external input to current neuron as zero;;
9.     $f(m_c, n_c) \leftarrow 1$;
       // Set current position as cleaned;
10.    if $\theta_c \neq \theta_n$ // The robot encounters the obstacle
       or cleaned area;
11.        execute Algorithm 1 and Algorithm 2;
12.    end if
13.    $\mathcal{N}_w(m_c, n_c) = \{N(m, n) | m \in (m_c - 1, m_c,$
       $m_c + 1)$ and $n \in (n_c - 1, n_c, n_c + 1)\}$;
14.    if $\nexists(m, n) \in \mathcal{N}_w, \forall m \in \{m_c - 1, m_c, m_c + 1\}$ and
       $\forall n \in \{n_c - 1, n_c, n_c + 1\}, s.t. f(m, m) = 0$ then
       // Robot is trapped in deadlock.
15.        execute Algorithm 4;
16.    end if
17. until $f(m, n) = 1, \forall 1 \le m \le N_x, 1 \le n \le N_y$;
    // Path planning is completed;
18. return $\mathcal{T}(t), x(m, n)$
```

IV. Simulation Studies and Discussion

In this section, the proposed algorithm is applied to a known indoor environment which contains many U-shape obstacles. The result is compared with classical zigzag path planning method and shows the effectiveness of the algorithm. The proposed algorithm is also verified in the completely unknown environment. By changing the number of backtracking steps, the algorithm can also generate an orderly path in the unknown environment. In addition, the proposed algorithm is compared with Oh et al.'s approach.

A. CCPP in Known Indoor Environment

The proposed CCPP algorithm based on BINN is applied to a known environment case, which is complicated with many U-shape obstacles in the workspace. In the simulation, the neural network has 30×30 topologically organized neurons. The parameters of model are set as $A = 100$, $B = 1$, $D = 1$, $\mu = 0.8$, and $r = 2$ for the shunting (3); $E = 100$ for the external inputs; and $c = 0.1$ for the weight of (6). The workspace is shown in Fig. 9(a), where the cleaning robot starts from point $S(2, 2)$. The external inputs of uncleaned areas are initially set as E, after the robot covers a point, the external input of the



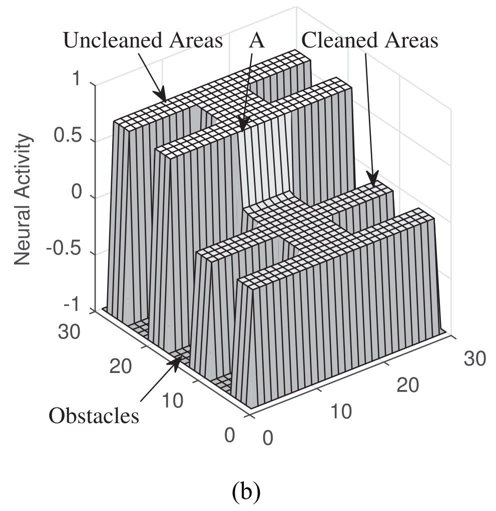


Fig. 9. CCPP in a known environment. (a) Generated robot path. (b) Activity landscape of the neural network when the robot reaches $A(12, 18)$.

TABLE I
COMPARISON IN TERMS OF COVERAGE, TURNS, AND PATH REPETITION RATIO USING THE PROPOSED METHOD AND THE ZIGZAG METHOD

Method	Coverage(%)	Repetition Ratio(%)	Turns
Proposed method	100	2.21	69
Zigzag method	100	11.03	61

corresponding neuron is set to zero. The details are introduced in Algorithm 5.

The simulation result is shown in Fig. 9(a). And the neural activity landscape of the neural network when the robot reaches the point $A(12, 18)$ is shown in Fig. 9(b). The path generated by the proposed algorithm has the characteristics of zigzag path and spiral path, the robot tends to move along the existing path and obstacles. Therefore, in a complex environment with many obstacles, the path described above is more orderly than it generated by (3) and (6), which is cluttered and overlapped. The amount of robot turns and path repetition ratio of the generated path and the classic zigzag path [4] are recorded in Table I. The workspace, start point, and other experimental conditions of the simulations are identical. It shows that the length of path by the proposed algorithm is much shorter than the zigzag path, and the proposed algorithm only has eight more turns.

B. CCPP in Unknown Indoor Environment

The proposed CCPP algorithm is also applied to a completely unknown indoor environment. In this situation, the environment is set to be completely unknown to the robot before the path planning. Unlike the CCPP in a known environment, in this case, the robot does have the map information in the beginning. The proposed CCPP algorithm is capable of generating an orderly path without the prior information, but the dynamic selection of the backtracking points in Algorithm 2 is applied to optimize the generated path.

In the simulation, the robot needs to know the boundary range of the total area before starting path planning, the neural network has 30×30 topologically organized neurons and the parameters of the model are the same as the above simulation. The entire workspace is set as uncleaned, when the sensor detects obstacles, the external inputs of the corresponding neurons are set as $-E$. The cleaning robot starts from point $S(2, 2)$.

The simulation result is shown in Fig. 10. As shown in Fig. 10(a), the gray area is unknown to the robot, when the robot reaches $A(20, 15)$, the robot is only able to detect a limited area in a circle of radius $r_s = 6$. The neural activity landscape when the robot reaches $A(20, 15)$ is shown in Fig. 10(c), in which the undetected areas are defined as uncleaned areas. In Fig. 10(b), due to the different selection of the backtracking points, the generated path in unknown environment is not like a zigzag path, but the path repetition ratio is also as low as 2.40%. The global map of the entire workspace is built when the robot reaches point $B(14, 24)$. The neural activity landscape when the robot reaches the last point $B(14, 24)$ is shown in Fig. 10(d). The result shows that the proposed algorithm can be applied to the unknown environment without any prior information or manual intervention.

The proposed CCPP algorithm is also compared with a well-known template-based CCPP model proposed by Oh et al. [35] using the same workspace. And the other experimental conditions like the start point and the start direction are the same. The neural network has 13×10 topologically and discretely organized neurons and the parameters of the model are the same as the above simulation. All the neural activities are initialized to be zero, and the entire workspace is marked as uncleaned. The cleaning robot starts from point $S(2, 2)$. First, the robot autonomously travels along an orderly spiral path. As shown in Fig. 11(a), when the planning path reaches point $A(7, 7)$, the workspace is divided into two areas, the robot will turn to another direction at point $B(9, 7)$. Finally, there is no overlapped area in the generated path, which is shown in Fig. 11(b). The amount of robot turns and path repetition

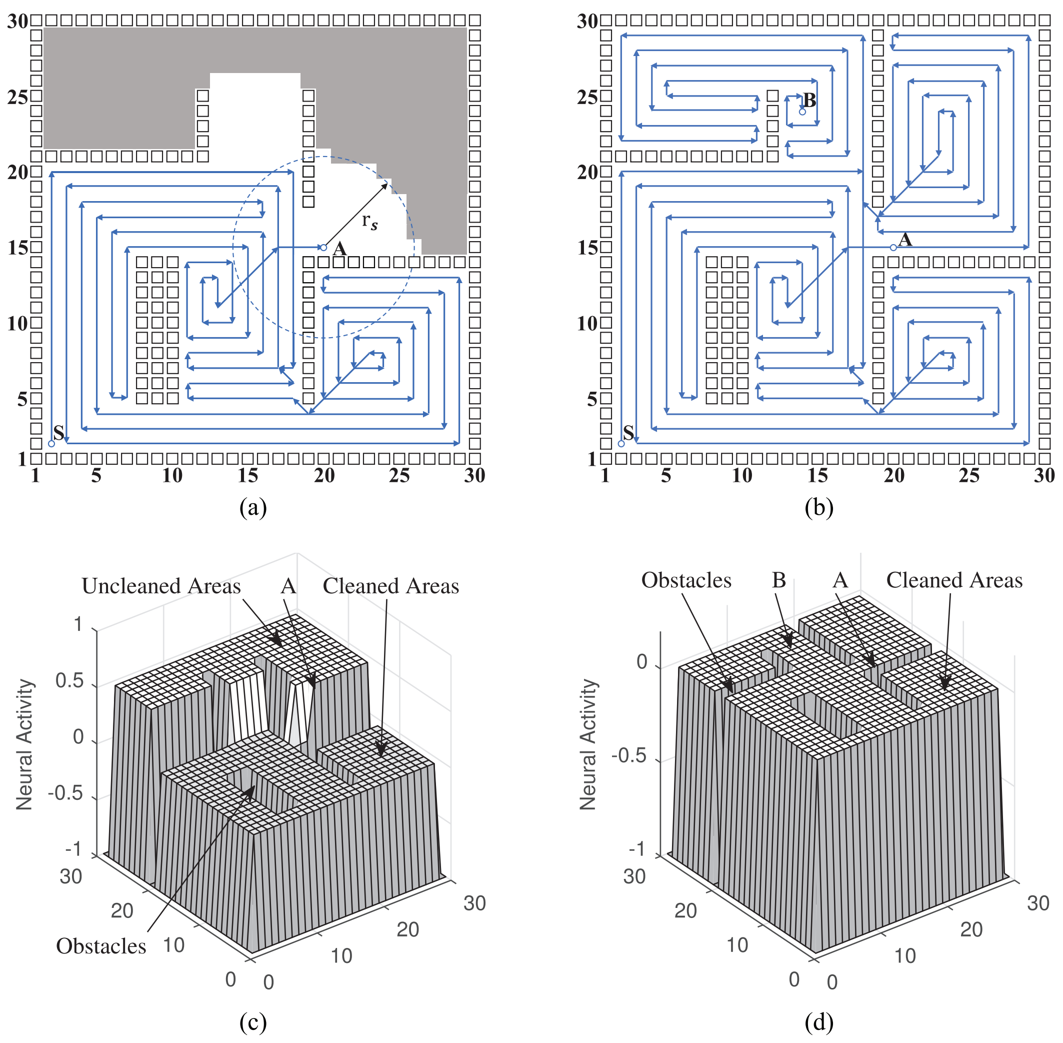


Fig. 10. CCPP in an unknown environment. (a) Built map and generated path when the robot reaches $A(20, 15)$. (b) Generated robot path. (c) Activity landscape of the neural network when the robot reaches $A(20, 15)$. (d) Activity landscape of the neural network when the robot reaches $B(14, 24)$.

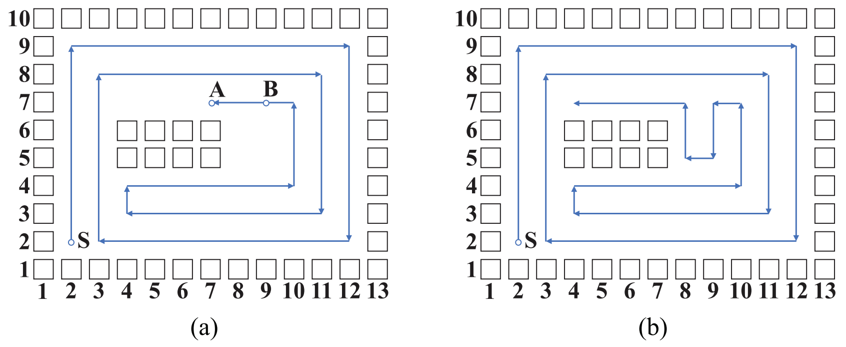


Fig. 11. CCPP in an unknown environment. (a) Planned path when the robot reaches $A(7, 7)$. (b) Generated robot path.

ratio of the generated path and Oh et al.'s path are recorded in Table II. It shows that the path length by our proposed algorithm is shorter than Oh et al.'s model, and the number of turns by the proposed algorithm is also fewer.

C. CCPP in Real Indoor Environments

The proposed CCPP algorithm is also applied to real indoor environments. Maps I and II from the CASAS smart home data set [36], which is collected in the smart apartment testbed located on the WSU campus, are grid maps of the real indoor apartments. In the simulations, the neural networks have 25×40 topologically organized neurons and the parameters of the model are the same as the above simulation. To simulate the actual cleaning process in apartments, there are two path planning simulations with different starting points on each map. The simulation results are shown in Fig. 12.

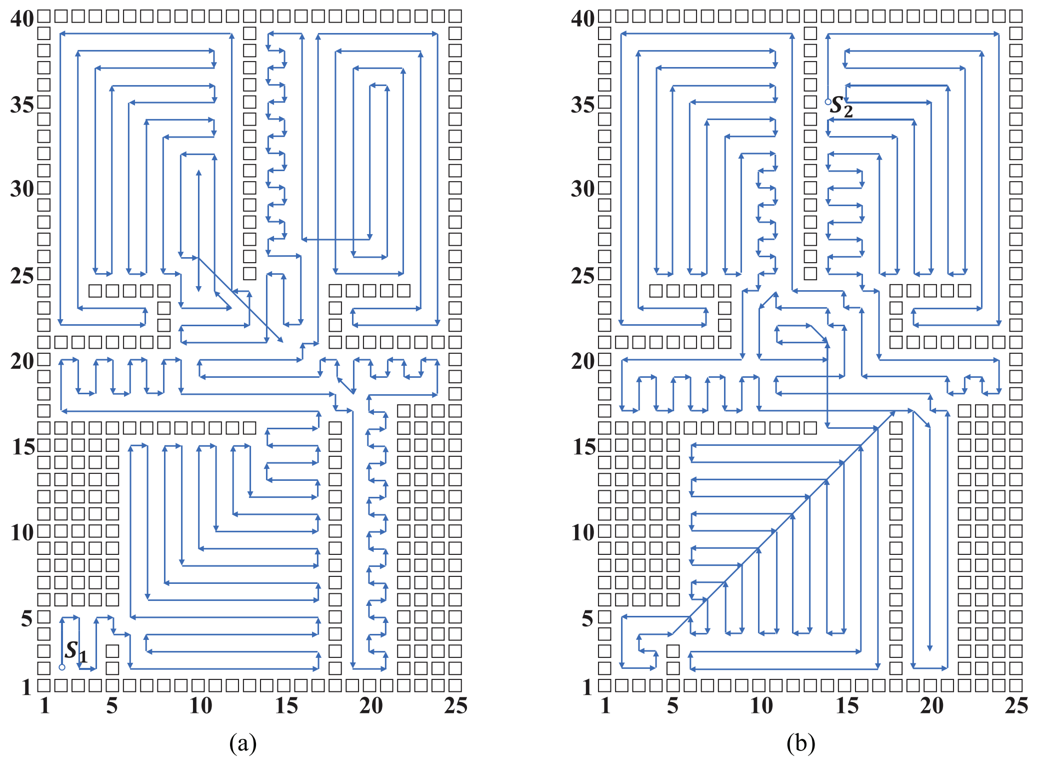
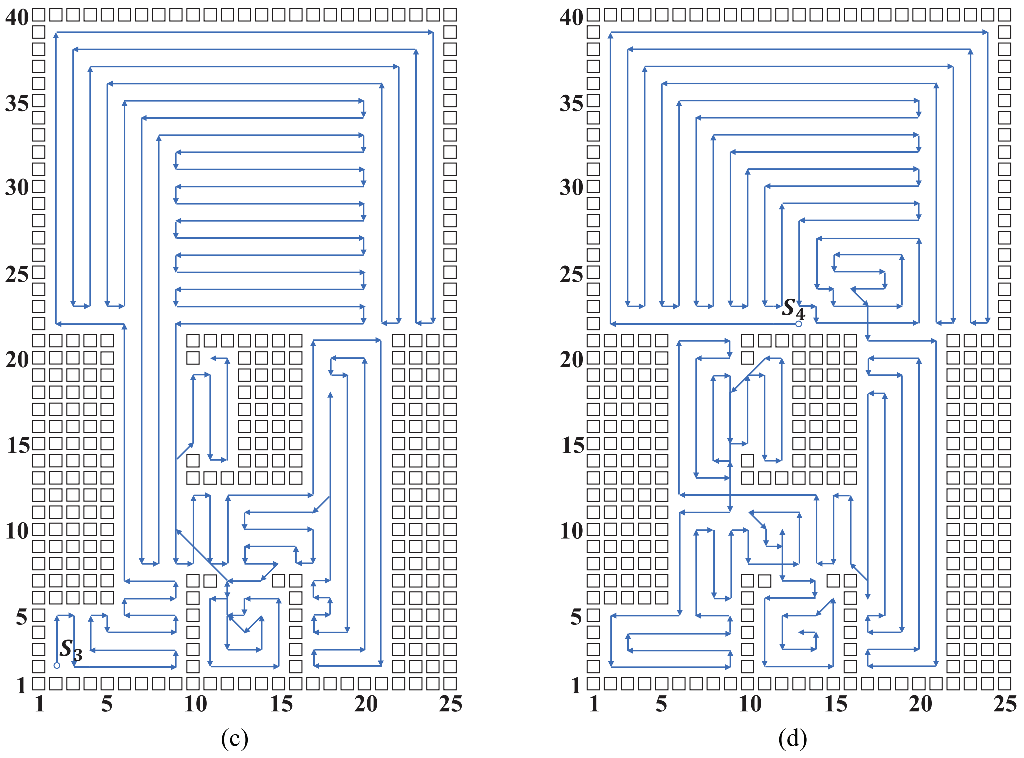



Fig. 12. CCPP in real indoor environments. (a) Generated path on map I started at $S_1(2, 2)$. (b) Generated path on map I started at $S_2(14, 35)$. (c) Generated path on map II started at $S_3(2, 2)$. (d) Generated path on map II started at $S_4(13, 22)$.

The generated path on map I is shown in Fig. 12(a) and (b), the cleaning robot starts from point $S_1(2, 2)$ and point $S_2(14, 35)$. Although the planned paths are quite different, the path repetition ratio of the two simulations is as low as 2.24% and 2.66%, respectively. And the simulations on map II are shown in Fig. 12(c) and (d), and the start points are $S_3(2, 2)$ and $S_4(13, 22)$. The path repetition ratio of the two simulations on map II is 2.46% and 2.17%, respectively. The comparison experiments using the zigzag method [4] are performed under the same experimental conditions. The coverage ratio and path repetition ratio of the generated path and the classic zigzag path are recorded in Table III. It shows that the path repetition

TABLE II
COMPARISON IN TERMS OF COVERAGE, TURNS, AND PATH REPETITION RATIO USING THE PROPOSED METHOD AND OH ET AL.'S METHOD

Method	Coverage(%)	Repetition Ratio(%)	Turns
Proposed method	100	0	15
Oh et al.'s method	100	6.33	26

TABLE III
COMPARISON IN TERMS OF COVERAGE AND PATH REPETITION RATIO USING THE PROPOSED METHOD AND THE ZIGZAG METHOD

Map	Start Point	Method	Coverage (%)	Repetition Ratio(%)
Map I	$S_1(2,2)$	Proposed method	100	2.24
		Zigzag method	100	7.84
	$S_2(14,35)$	Proposed method	100	2.66
		Zigzag method	100	7.14
Map II	$S_3(2,2)$	Proposed method	100	2.46
		Zigzag method	100	5.92
	$S_2(13,22)$	Proposed method	100	2.17
		Zigzag method	100	6.50

ratio by the proposed algorithm is lower than the zigzag path in each simulation, which indicates that our proposed algorithm is more competitive in real indoor environments.

V. CONCLUSION

A CCPP algorithm based on BINN is proposed. In order to generate an orderly path with fewer path repetitions, the new area connectivity detection and path backtracking algorithm and the dynamic deadlock escape algorithm are presented. Without any templates or global cost functions, the improved algorithm is applied to both known and unknown environment, and the results prove its effectiveness. The improved algorithm is applicable in various complex environments. In the future, some potential studies will be extended. First, the selection method of backtracking points should be further studied. Second, the algorithm will be applied to the environment with moving obstacles, such as pedestrians and vehicles. Third, the method of optimizing the path by slightly reducing the ratio of coverage is considered.

REFERENCES

[1] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *SN Appl. Sci.*, vol. 1, no. 8, p. 847, 2019.

[2] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119310–119342, 2021.

[3] X. Miao, J. Lee, and B.-Y. Kang, "Scalable coverage path planning for cleaning robots using rectangular map decomposition on large environments," *IEEE Access*, vol. 6, pp. 38200–38215, 2018.

[4] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. C. Chung, "BA*: An online complete coverage algorithm for cleaning robots," *Appl. Intell.*, vol. 39, no. 2, pp. 217–235, 2013.

[5] H. V. Pham and T. N. Lam, "A new method using knowledge reasoning techniques for improving robot performance in coverage path planning," *Int. J. Comput. Appl. Technol.*, vol. 60, no. 1, pp. 57–64, 2019.

[6] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete coverage path planning of autonomous underwater vehicle based on GBNN algorithm," *J. Intell. Robot. Syst.*, vol. 94, pp. 237–249, Feb. 2018.

[7] B. Sun, D. Zhu, C. Tian, and C. Luo, "Complete coverage autonomous underwater vehicles path planning based on Glasius bio-inspired neural network algorithm for discrete and centralized programming," *IEEE Trans. Cogn. Devel. Syst.*, vol. 11, no. 1, pp. 73–84, Mar. 2019.

[8] K. Chen and Y. Liu, "Optimal complete coverage planning of wall-climbing robot using improved biologically inspired neural network," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, 2017, pp. 587–592.

[9] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *J. Field Robot.*, vol. 32, no. 7, pp. 952–983, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21554

[10] C. Peng and V. Isler, "Visual coverage path planning for urban environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5961–5968, Oct. 2020.

[11] D. Jia, M. Wermelinger, R. Diethelm, P. Krüsi, and M. Hutter, "Coverage path planning for legged robots in unknown environments," in *Proc. IEEE Int. Symp. Saf. Security Rescue Robot. (SSRR)*, 2016, pp. 68–73.

[12] E. Galceran and M. Carreras, "Efficient seabed coverage path planning for ASVs and AUVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 88–93.

[13] T. Palleja, M. Tresanchez, M. Teixido, and J. Palacin, "Modeling floor-cleaning coverage performances of some domestic mobile robots in a reduced scenario," *Robot. Auton. Syst.*, vol. 58, no. 1, pp. 37–45, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889009001171

[14] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, nos. 1–4, pp. 113–126, 2001.

[15] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300167X

[16] T.-K. Lee, S. Baek, and S.-Y. Oh, "Sector-based maximal online coverage of unknown environments for cleaning robots with limited sensing," *Robot. Auton. Syst.*, vol. 59, no. 10, pp. 698–710, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889011000893

[17] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, "BSA: A complete coverage algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2040–2044.

[18] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021.

[19] C. Liu, Q. Mao, X. Chu, and S. Xie, "An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning," *Appl. Sci.*, vol. 9, no. 6, p. 1057, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/6/1057

[20] C. Wang, C. Cheng, D. Yang, G. Pan, and F. Zhang, "Path planning in localization uncertaining environment based on Dijkstra method," *Front. Neurorobot.*, vol. 16, Mar. 2022, Art. no. 821991. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2022.821991

[21] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1985, pp. 116–121.

[22] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Autom.*, vol. JRA-3, no. 3, pp. 249–265, Jun. 1987.

[23] Y. Gabriely and E. Rimon, "Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2002, pp. 954–960.

[24] A. V. Le, V. Prabakaran, V. Sivanantham, and R. E. Mohan, "Modified A-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors*, vol. 18, no. 8, p. 2585, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/8/2585

[25] A. Manimuthu, A. V. Le, R. E. Mohan, P. Veerajagadeshwar, N. H. K. Nhan, and K. P. Cheng, "Energy consumption estimation model for complete coverage of a tetromino inspired reconfigurable surface tiling robot," *Energies*, vol. 12, no. 12, p. 2257, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/12/2257

[26] A. K. Lakshmanan et al., "Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot," *Autom. Constr.*, vol. 112, Apr. 2020, Art. no. 103078. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519305813

[27] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.)*, vol. 34, no. 1, pp. 718–724, Feb. 2004.

[28] X. Qiu, J. Song, X. Zhang, and S. Liu, "A complete coverage path planning method for mobile robot in uncertain environments," in *Proc. 6th World Congr. Intell. Control Autom.*, 2006, pp. 8892–8896.

[29] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1279–1298, Jul. 2008.

[30] C. Luo, S. X. Yang, X. Li, and M. Q.-H. Meng, "Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 750–760, Jan. 2017.

[31] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Toward energy-efficient online complete coverage path planning of a ship hull maintenance robot based on Glasius Bio-Inspired Neural Network," *Expert Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115940. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742101294X

[32] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952. [Online]. Available: https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1952.sp004764

[33] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Netw.*, vol. 1, no. 1, pp. 17–61, 1988.

[34] S. X. Yang and M. Meng, "Neural network approaches to dynamic collision-free trajectory generation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 302–318, Jun. 2001.

[35] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, "Complete coverage navigation of cleaning robots using triangular-cell-based map," *IEEE Trans. Ind. Electron.*, vol. 51, no. 3, pp. 718–726, Jun. 2004.

[36] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009.