# Multi-robot coverage algorithm in complex terrain based on improved bio-inspired neural network

**Fangfang Zhang[1,2], Mengdie Duan[1], Jianbin Xin[1,2], Jinzhu Peng[1,2]**
[1]School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, China
[2]State Key Laboratory of lntelligent Agricultural Power Equipment, Luoyang, China

## ABSTRACT

Biological neural network (BNN) algorithms have become popular in coverage search in recent years. However, its edge activity values are weak, and it is simple to fall into a local optimum at a late stage of coverage. When applied to complex environments, the 3D BNN network structure has high computational and storage complexity. In order to solve the above problems, we propose an algorithm for multi-robot cooperative coverage of complex terrain based on an improved BNN. The algorithm models the complex terrain using a 2.5-dimensional (2.5D) elevation map. Combining the dual-layer BNN network with the 2.5D elevation map, we propose an elevation value priority mechanism. This mechanism lets the robot make elevation-based decisions and prioritizes higher terrain areas. The dual neural network's first layer plans the robot's path in normal mode. The second network layer helps the robot escape the local optimum. Finally, the algorithm's full coverage effect in complex terrains and the speed of covering high terrain are verified by simulations. The experiments show that our algorithm preferentially covers high points of the region and eventually covers 100% of complex terrain. Compared with other algorithms, our algorithm covers more efficiently and takes fewer steps than others. The speed of covering high terrain areas has increased by 34.51%.

*Corresponding Author:*

Jianbin Xin
School of Electrical and Information Engineering, Zhengzhou University
Zhengzhou, Henan, China
Email: j.xin@zzu.edu.cn

## 1. INTRODUCTION

Robotic systems for full coverage tasks in complex environments are in high demand due to the rapid development of geological exploration, agricultural monitoring, and environmental surveillance [1]. Modern agriculture relies on robots for crop and soil monitoring, while geological exploration uses robots for data collection across large areas. In real-world applications, task requirements often go beyond full coverage. For instance, to optimise water distribution in agricultural irrigation, robots should prioritise higher elevation areas [2], and geological exploration can benefit from focusing on high-elevation points to understand topography. Thus, exploring unknown terrains while prioritising key areas is a major challenge.

Current full coverage path planning algorithms include A* [3], genetic algorithms [4], RRT* [5], the Boustrophedon method [6], and spanning tree coverage (STC) [7]. These algorithms are widely used in various scenarios. Cai *et al.* [3] introduced an algorithm (UAPP) integrating A* and the U-turn algorithm to clean areas obstructed by irregular obstacles. Maxwell [8] proposed a Smooth-STC model that minimises backtracking

and maximises coverage by optimising paths. However, most of these algorithms are designed for single-robot. Boustrophedon methods partition work areas into cells, with robots planning within each cell. Jacobs and Bean [9] examined common cell decomposition techniques, but Boustrophedon coverage lacks multi-robot synergy, with each robot focusing on its partition and minimal coordination [10].

The bio-inspired neural network (BNN) algorithm is able to adjust the robot's path in unknown environments in real time by simulating the behaviour of biological neurons. Additionally, it doesn't require any learning and is computationally straightforward. It has been widely used in unknown and dynamic environments in recent years [11]. Cao and Sun [12] used a BNN network in the domain of multi-robot target search. All robots share environmental information. Each robot perceives other robots in the vicinity as dynamic obstacles. But as the search progresses, the amount of unexplored territory decreases, and the BNN algorithm can quickly put the robot in a deadlock. Zhang *et al.* [13] combined BNN with a model predictive control algorithm. Deadlock states are less common when the activity value goal function is defined to forecast and solve the optimal inputs for the upcoming time sequence. However, the above work is mainly carried out in a 2D grid map, which only divides the environment into obstacles and free grids.

Zhu *et al.* [14] developed a novel BISOM method and tested its performance in an underwater 3D workspace with obstacles by combining self-organising neural networks with BNN networks. The BNN network and potential field were merged in the 3D underwater environment by Cao *et al.* [15]. The BNN network devised the optimal trajectory for the underwater vehicle, while the potential field function refined the path established by the bionic neural network, thereby achieving both static and dynamic target path planning for the AUV. But to deal with the complicated world, the 2D BNN network architecture is changed to 3D, which greatly increases the computing and storage capacity. A coarse and fine scale 3D map was suggested in the literature to address the issue of extensive computation in the 3D BNN algorithm [16]. The D* algorithm calculates the 3D optimal paths for the selected coarse scale map, followed by the application of the improved BNN algorithm to determine the 3D optimal paths for the fine scale map. However, this method depends on the invariance of geomorphological features and the selection of scale. Changes in the environment during task execution, or significant differences in map features across scales, may hinder the method's ability to achieve global optimisation of the path.

In summary, BNN algorithms tend to get trapped in local optima, and 3D architectures are computationally expensive. We propose a multi-robot collaborative coverage algorithm for complex terrains using an improved BNN. Our method employs a 2.5D mesh map to represent complex environments, combining a dual-layer BNN with elevation-first coverage. This allows robots to prioritise high-elevation areas based on terrain differences. The first BNN layer plans normal coverage paths, while the second layer helps avoid local optima. A virtual edge mechanism is also introduced to address weak activity values at the BNN network's edges. The paper is organised as follows: section 2 discusses terrain modeling. Section 3 covers elevation-priority and obstacle avoidance mechanisms. Section 4 details the improved BNN algorithm and robot decision-making. Section 5 presents simulation results and comparisons with existing methods.

## 2.    ENVIRONMENTAL MODELING

In coverage tasks for agricultural monitoring and geological exploration, terrain complexity arises from natural obstacles and varying ground heights. Choosing an appropriate environmental representation method is crucial for robots to interpret terrain. A 3D grid map [17] divides space into voxels, representing complex structures like suspended objects and multi-layered facilities. However, 3D maps are memory-intensive, especially in large-scale or high-resolution environments. In contrast, 2D raster maps use binary values (0 or 1) to denote obstacles and passable areas [18], but fail to capture terrain features. A 2.5D grid map stores height values on a 2D plane, effectively representing terrain undulations [19]. For mountainous or hilly terrains, robots only need surface height data to find optimal paths, avoiding the complexity of full 3D mapping. Thus, we use 2.5D grid maps for environmental representation, as shown in Figure 1. Each grid's number indicates the elevation at its center point, relative to a reference datum [20]. Elevation values are mapped to colors, as shown on the right of Figure 1. The dots represent the robot model, and the arrows show potential robot movement directions. Additionally, we assume that the robot has powerful sensing capabilities and terrain adaptability, enabling it to accurately assess terrain height information and navigate effectively under various conditions.
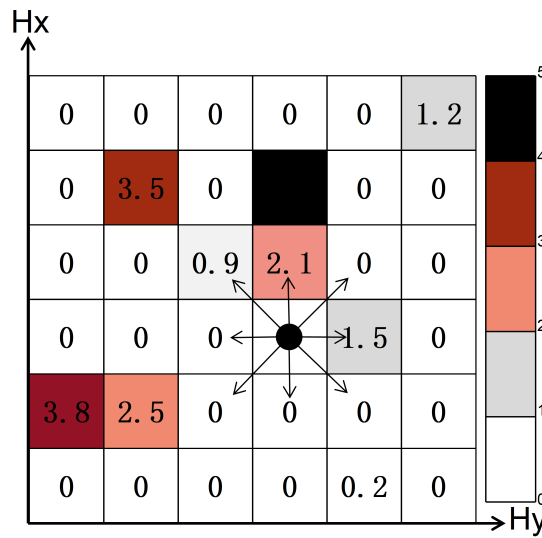
Figure 1. 2.5D grid map and robot motion direction

## 3. ELEVATION-PRIORITY COVERAGE MECHANISM BASED ON BNN

Based on the environmental model and assumptions in section 2, the robot can accurately acquire terrain information. Using this data, the robot makes decisions. To address the need for prioritising higher terrain areas, and given the effectiveness of BNN in dynamic environments, we design a BNN-based elevation-priority coverage mechanism. This mechanism enables the robot to focus on higher terrain areas while ensuring full coverage.

### 3.1. Bio-inspired neural network

The BNN model utilized in this study is a discrete-time Hopfield-type neural network [21], constructed on a grid map that incorporates complex terrain, as illustrated in Figure 2. Each circle represents a neuron corresponding to a grid cell in the map, with each grid cell having an associated neuronal activity value [22]. Neurons are locally connected to one another within a small $(0, r)$ region. The dynamic behavior of neuron activity values is expressed by (1).
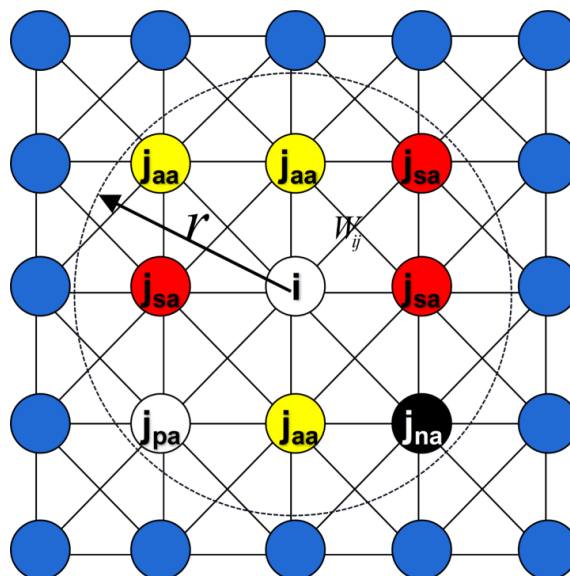


Figure 2. BNN with 2.5D grid map

$$\frac{dm_i}{dt} = -Am_i + (B - m_i)\left([I_i]^+ + \sum_{j \in N_i} w_{ij}[m_j]^+\right) - (D + m_i)[I_i]^- \tag{1}$$

where $A$ represents the decay rate of the neuronal activity value, while $B$ and $-D$ denote the upper and lower limits of the neuronal activity value, respectively. $m_i$ is the activity value of neuron $i$ and $I_i$ corresponds to the external inputs applied to neuron $i$. The terms $[a]^+ = \max\{a, 0\}$, $[a]^- = \max\{-a, 0\}$ are defined accordingly. The expression $[I_i]^+ + \sum_{j \in N_i} w_{ij}[m_j]^+$ are the excitatory input of current neuron $i$, while $m_j$ refers to the activity value of the neighboring neuron $j$. The term $[I_i]^-$ signifies the inhibitory input to neuron $i$. The connection weight coefficients $w_{ij}$ between neuron $i$ and and its neighboring neuron $j$ are defined as (2). $N_i$ is the set of neurons associated with neuron $i$.

$$w_{ij} = g(d_{ij}) = \begin{cases} \frac{u}{d_{ij}}, & 0 < d_{ij} < r \\ 0, & d_{ij} \geq r \end{cases} \tag{2}$$

Here, $d_{ij}$ denotes the Euclidean distance between the locations of neuron $i$ and neuron $j(i \neq j)$. The constant $u$ is a positive value, $r$ represents the radius of the receptive field, as shown in Figure 2, typically ensuring that $w_{ij} \in [0, 1]$. It should be noted that the neural network model presented in this paper differs from existing models. The model is primarily constructed on terrains with height information, incorporating various external inputs($I_i$) at different heights to ensure that the robot can effectively respond in a complex and dynamic environment.

### 3.2. Elevation-priority coverage mechanism

A neuron's external positive input in the neural network model consists of two parts: external excitation ($[I_i]^+$) and surrounding neuron connectivity inputs ($\sum_{j \in N_i} w_{ij}[m_j]^+$), ensuring only positive neuronal activity propagates, while negative values act as local inhibition [23]. Neuronal activity is primarily influenced by inhibitory and excitatory inputs [24]. Model stability theory asserts that the magnitude of $I_i$ does not affect the positivity or negativity of Lyapunov function derivatives [24], meaning model stability is independent of $I_i$'s value. Thus, varying $I_i$ results in different neuronal activity levels for different regions.

Terrain elevation varies by raster, influencing task feasibility and safety [20]. The robot uses this information for real-time decision-making. Elevation differences are more dynamic and informative than absolute values [25], defined as:

$$\Delta h_j = |h_i(x, y) - h_j(x, y)|, j \in N_i \tag{3}$$

where $N_i$ is the set of neurons associated with neuron $i$, and $h_i(x, y)$ and $h_j(x, y)$ are the elevation values at the current and potential decision points, respectively. This allows quick identification of obstacles or unsafe areas. To prioritise higher terrain, we combine height differences with a BNN network to form the elevation-priority coverage mechanism in (4).

$$I_j = \alpha * \Delta h_j, \Delta h_j \leqslant H_{\max}, j \in N_i \tag{4}$$

where $\alpha$ is a constant, and $H_{\max}$ is the maximum passable height difference.

For safety, an obstacle avoidance mechanism applies inhibitory inputs to neurons when the height difference exceeds a threshold or when an obstacle is present, as shown in (5).

$$I_j = \beta, \Delta h_j > H_{\max} \text{ or } p_j(x, y) \in NA, j \in N_i \tag{5}$$

Here, $\beta < 0$, $H_{\max}$ is the maximum height difference the robot can traverse, $p_j(x, y)$ represents the coordinates of neuron $j$, and $NA$ denotes the obstacle grid. $I_j$ is the updated external input of neuron $j$.

## 4. IMPROVED BNN ALGORITHM DECISION

As coverage progresses, fewer meshes remain uncovered, often farther from the robot. Traditional BNN algorithms struggle to guide robots to these areas, leading to repeated coverage and local optimisation [26], [27]. To address this, we propose a locally optimal detrapping mechanism using dual BNN networks, along with a virtual edge mechanism to resolve low activity values at the network's edges.

### 4.1. Virtual edge mechanism

Figure 3 shows the robot operating in a workspace $W$ (red grids), with an inner workspace $W_1$ (light blue grids) and an edge workspace $W_2$ (light green grids). Compared to $W_1$, $W_2$ has fewer surrounding grids, making it less attractive and more likely to remain uncovered. Figure 4 demonstrates a collaborative coverage scenario with four robots covering a 20x20 area. After 65 steps, uncovered areas (grey grids) are mainly at the edge of the workspace ($W_2$). To improve edge grid attraction, we introduce a virtual edge mechanism that adds a virtual grid collection $V_v$ of $v$ layers extending from $W_2$, with all grids set to an elevation of 0. These grids are updated using the same method as (4) and (1). In Figure 3, virtual edges $V_v$ are shown in grey, with dashed arrows indicating areas the robot cannot traverse, and solid arrows showing the next step decisions.
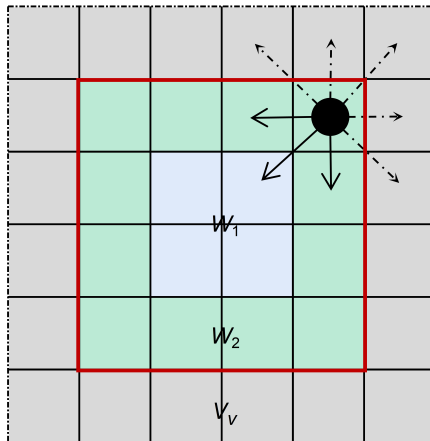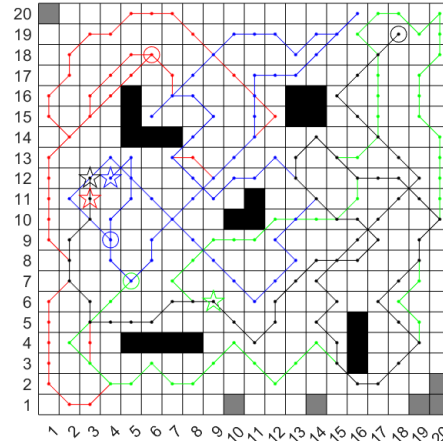


Figure 3. Virtual edge mechanism



Figure 4. Collaborative coverage of late stages

### 4.2. Dual BNN detrapping mechanism

To detect if the robot is trapped in a local optimum, we define the robot's state as in (6):

$$\text{State}(R_i) = \begin{cases} 1, & S_{r_0}(R_i\ right) = 0 \& td > T_d \\ 0, & \text{others} \end{cases} \tag{6}$$

where $S_{r_0}(R_i)$ represents the uncovered area in the robot's detection range, $td$ is the time in the local optimum, and $T_d$ is the threshold. When there is no uncovered area within its range and the time exceeds $T_d$, the robot enters state 1 (local optimum). Otherwise, it operates in state 0.

Based on the robot's state, we define two working modes in (7):

$$\text{Mode}(R_i) = \begin{cases} \text{Breakout}, & \text{State}(R_i) = 1 \\ \text{Coverage}, & \text{State}(R_i) = 0 \end{cases} \tag{7}$$

In the $Breakout$ mode, in order to quickly reach the uncovered area and exit the local optimal state, the robot must find a target point $(T_x, T_y)$ that is as close as possible to its current location and is an uncovered grid.

$$I_j = \begin{cases} G, & j \in T_j \\ \alpha * \Delta h_j, & \Delta h_j \leqslant H_{\max} \end{cases} \tag{8}$$

As the robot executes the $Breakout$ mode, the state may change. In order to identify whether the robot is out of the local optimum, we define the robot termination state discriminant as in (9).

$$\text{State}(R_i) = \begin{cases} 0, & \frac{S_{r_0}(R_i)}{r_0^2} \geqslant \sigma \text{ or } |R_i - T_i| \leq r_{\min} \\ 1, & \text{others} \end{cases} \tag{9}$$

where $\sigma$ is the threshold for uncovered area and $r_{\min}$ is the minimum distance to the target. The robot switches to $Coverage$ mode when the uncovered area exceeds the threshold or it is close to the target; otherwise, it continues the $Breakout$ mode.

### 4.3. Overall decision-making process

The robot's final decision-making process is expressed in (10):

$$P_n \Leftarrow m_{P_n} = \max\{m_j, j = 1, 2, ..., k\} \tag{10}$$

where $P_n$ represents the robot's next position, $m_j$ is the activity value of a potential decision grid, and $m_{P_n}$ is the maximum activity value among these grids. The decision-making flow for each robot is shown in Figure 5. Algorithm 1 details the BNN update process, and algorithm 2 presents the pseudo-code for the local optimum decoupling mechanism. The overall time complexity of the algorithm is $O(N_r * (k * N_1 + N_2 * M_x * M_y))$, where $N_1$ is the number of steps in coverage mode, $N_2$ is the number of steps in breakout mode, $N_r$ is the number of robots, and $M_x * M_y$ is the size of the breakout network, determined by the absolute differences between the robot's current position $(R_x, R_y)$ and breakout point $(T_x, T_y)$:

$$M_x = |R_x - T_x|, \quad M_y = |R_y - T_y| \tag{11}$$

This time complexity arises from Algorithm 1 updating the $k$ surrounding neurons and Algorithm 2 updating the second-layer breakout network. The space complexity is primarily determined by storing the map information, $O(H_x * H_y)$, where $H_x$ and $H_y$ represent the map's dimensions.
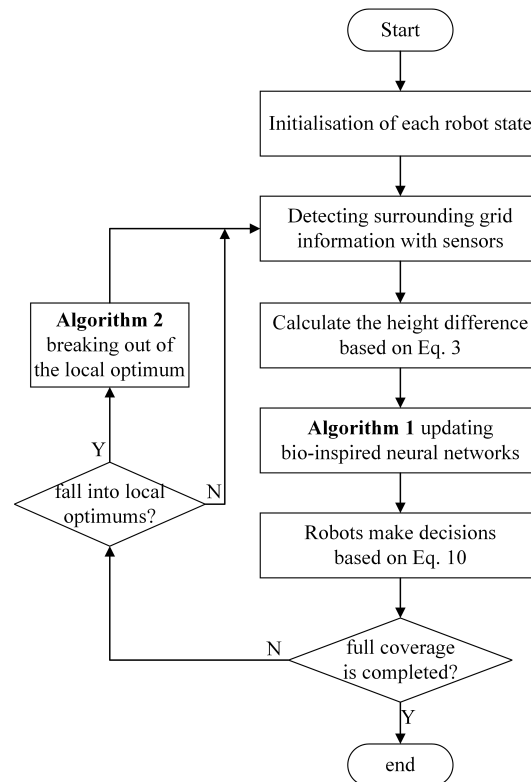


Figure 5. Overall decision-making flowchart

---

### Algorithm 1. Update neuron activity value

Input:  current neuron activity value $m_c(x, y), x \in [1, H_x], y \in [1, H_y]$; elevation value of the robot's current position $h_i(x, y)$; number of robots $N_r$; elevation of the surrounding grid $h_j(x, y), j = 1, 2 \dots, k$
Output:  renewed activity value $m_r(x, y), x \in [1, H_x], y \in [1, H_y]$
1:  for $n = 1 \rightarrow N_r$ do
2:      for $j = 1 \rightarrow k$ do
3:          calculate the $\Delta h_j$ based on (3)
4:          update the $I_j$ based on (4),(5)
5:          calculate the updated neuron activity value based on (1) and $m_c(x, y)$.
6:      end for
7:  end for

---

---

Algorithm 2. VBNN jumps out of the local optimum

---

Input: current neurons activity value $m_c(x, y), x \in [1, H_x], y \in [1, H_y]$; current status State $(R_i)$
Output: Neural network activity values after detrapping
 1: while State $(R_i)==1$ do
 2:     Mode $(R_i)$ = Breakout;
 3:     update the $I_j$ based on (5),8;
 4:     calculate the neuron activity value based on (1);
 5:     make decisions based on (10);
 6:     if $\frac{S_{r_0}(R_i)}{r_0^2} \geqslant \sigma$ or $|R_i - T_i| \leqslant r_{\min}$ then
 7:         State $(R_i)==0$;
 8:         Mode $(R_i)$ = Coverage;
 9:     end if
10: end while

---

## 5. SIMULATION EXPERIMENT

### 5.1. Experimental evaluation indicators

#### 5.1.1. Area coverage

Area coverage (AC) measures the algorithm's path coverage ability. Higher coverage indicates more comprehensive map coverage. It is defined as:

$$AC = \left(\frac{M}{W}\right) \times 100\% \tag{12}$$

where $M$ is the path-covered area and $W$ is the total work area.

#### 5.1.2. Mean coverage

To reduce randomness and better assess the robot's overall coverage, we calculate the mean coverage from multiple experiments:

$$MC = \frac{\sum_{i=1}^{n} AC_i}{n} \tag{13}$$

where $AC_i$ is the area coverage of the $i$-th experiment and $n$ is the total number of experiments.

#### 5.1.3. Standard deviation of coverage

The standard deviation of coverage (SDC) quantifies variability across experiments. A lower value indicates more consistent results:

$$SDC = \sqrt{\frac{\sum_{i=1}^{n} (AC_i - MC)^2}{n}} \tag{14}$$

where $n$ is the total number of experiments.

### 5.2. Process of collaborative coverage

To evaluate the algorithm's performance, we conducted simulation experiments using MATLAB 2021a on a 1.60 GHz processor. The experimental parameters are as follows: the coverage area is a 20×20 grid, with each grid having a unit length of 1. Initially, robots only know the environment boundaries, and each robot has a 3×3 coverage area. The BBN parameters are: $A = 2$, $B = D = 1$, $r = \sqrt{2}$, $u = 0.1$, $\alpha = 0.125$, $\beta = -1$, $H_{\max} = 5$, timestep $dt = 0.2$, $r_0 = 4$, $r_{\min} = 2\sqrt{2}$, $T_d = 1$, $G = 5$.

For collaborative coverage, four robots (R1(14,15), R2(9,5), R3(8,11), R4(6,11)) were tasked with covering an uneven terrain with obstacles. The neural network activity values and robot motion trajectories at different steps are shown in Figure 6. Figure 6(a) shows the trajectory and coverage of the robots when they work collaboratively for 29 steps. Raster elevation values match raster colours in the figure's right legend. Barriers or impassable rasters are black. Light grey rasters represent added virtual edges. Dark grey rasters represent uncovered areas. Figure 6(b) displays the neuronal activity map at 29 cooperative work steps. The light blue filled area is the actual working area. It can be seen that the covered area's neurone activity value drops rapidly, while the uncovered area's remains high, attracting the robot. The obstacle region has a negative activity value, inhibiting the robot from passing through. The trajectory and coverage when completing the full

---

coverage task are shown in Figure 6(c). Full coverage was achieved in 66 steps. Four robots avoided obstacles with little overlap in their trajectories. When this grid is covered, the elevation values at that location are also stored in the 2.5D grid. After completing the coverage task, the neural network activity values in the entire work area have stabilised, leaving only uncovered rasters at virtual edges as shown in Figure 6(d). The above experiments demonstrate that our algorithm, guided by neurone activity values, can avoid obstacles and cover the entire region.



(a)

(b)



(c)

(d)
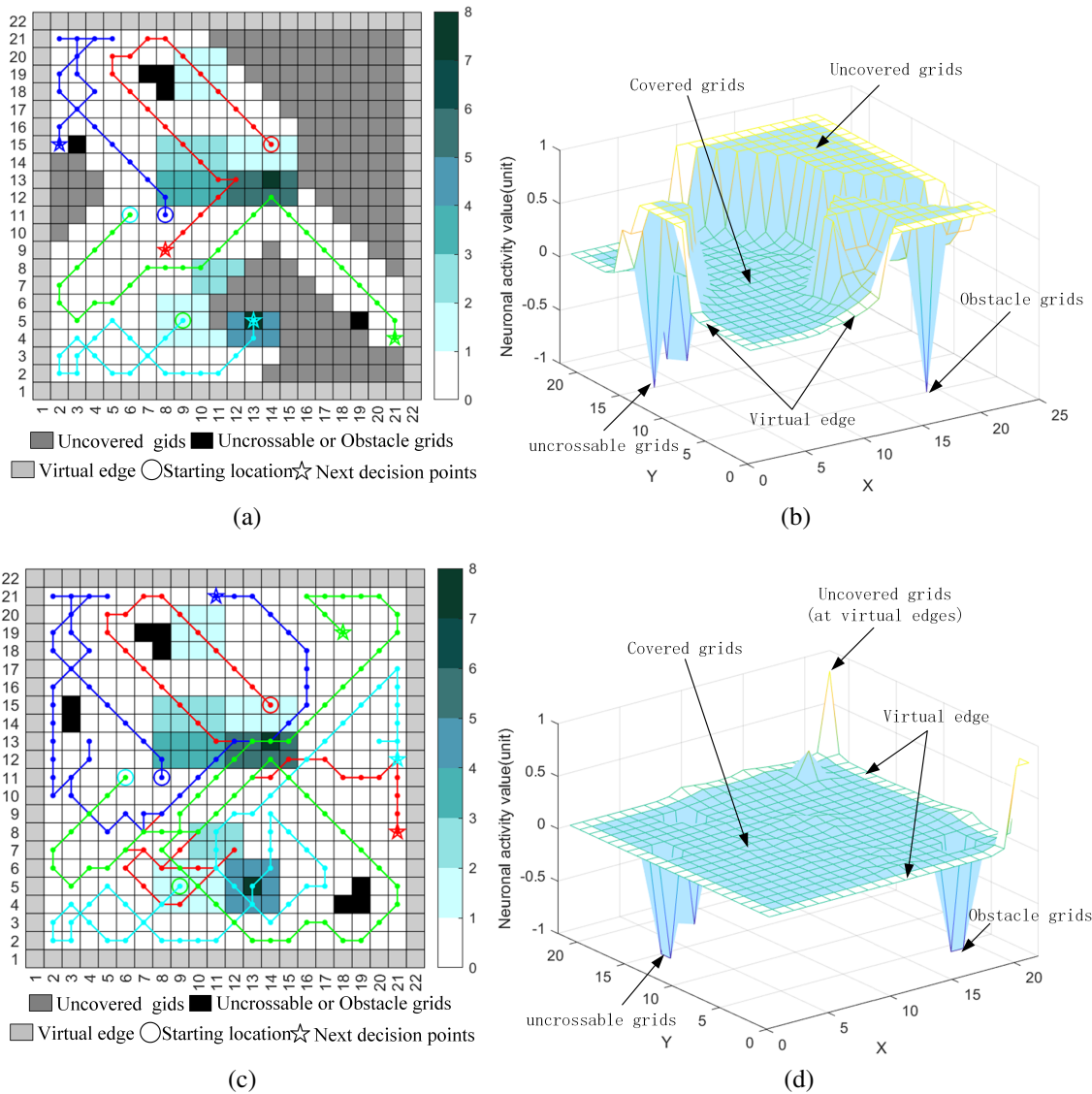
Figure 6. Trajectories of MRS at different movement steps and corresponding activity values (a) motion trajectory (N=29), (b) activity values (N=29), (c) motion trajectory (N=66), and (d) activity values (N=66)

## 5.3. Performance of the improved BNN algorithm

In order to avoid the partiality and randomness of a single experiment, we conducted 50 Monte Carlo experiments in different terrains with different robot starting positions. The experimental platform and parameters are the same as in section 5.2.. The 3D maps of the three different terrains are shown in Figure 7. Specifically, Figure 7(a) illustrates Terrain 1, Figure 7(b) illustrates Terrain 2, Figure 7(c) illustrates Terrain 3. Each of the three terrains has a different number of maxima and different terrain complexity. To fully evaluate the performance of our algorithm, we compared it to traditional neural networks and model predictive control algorithms (DMPC) [13].
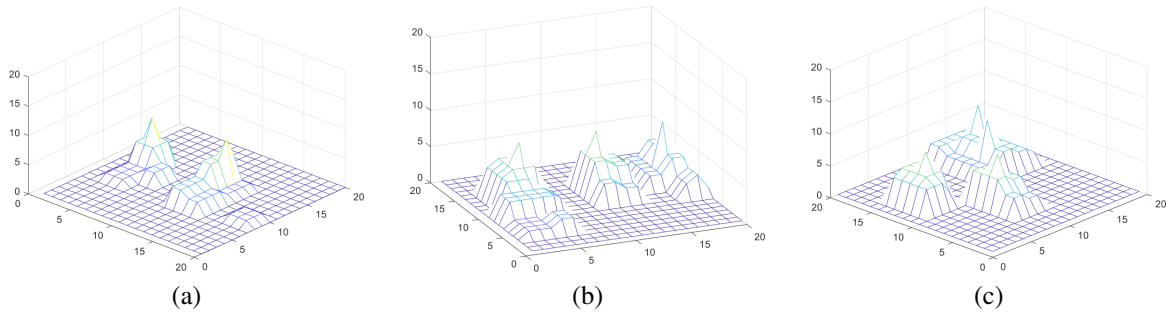
Figure 7. Three different terrains (a) terrain 1, (b) terrain 2, and (c) terrain 3

Table 1 displays the mean coverage (MC) and standard deviation of coverage (SDC) of experiments conducted on various terrains with 105 steps. As can be seen from Table 1, our algorithms achieve 100% coverage under various terrains, completing the full coverage task. The trend of the mean coverage (MC) versus the number of steps for different algorithms is shown in Figure 8. As can be seen from Figure 8, pre-coverage areas are more uncovered, and coverage growth is similar for various algorithms. However, in late coverage stages, our algorithm has faster growth and higher efficiency, while other algorithms have not reached 100% coverage on average. Table 2 displays the average number of steps for the experimental group completing full coverage. As can be seen from Table 2, our algorithm has a superior full coverage completion rate (100% vs. 38% vs. 24% in 50 sets of experiments in terrain 1), with a shorter average number of steps. In summary, our algorithm achieves 100% coverage in various terrains, with fewer steps and higher efficiency.

Table 1. MC and SDC in different terrains

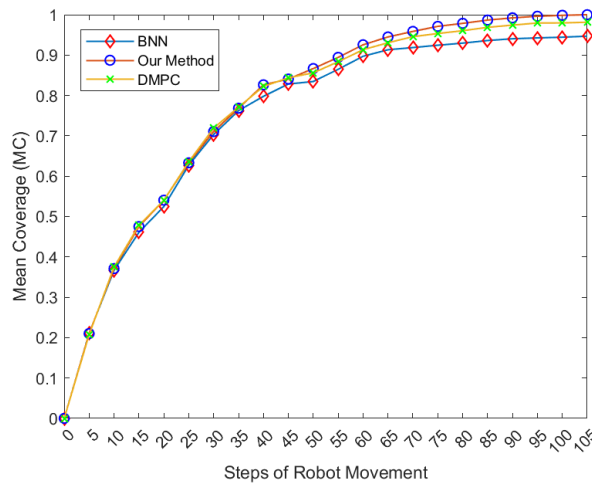| Terrain/indicators | Our method | | GBNN | | DMPC | |
|---|---|---|---|---|---|---|
| | MC | SDC | MC | SDC | MC | SDC |
| terrain 1 | **100%** | **0** | 95.76 | 0.0189 | 98.37 | 0.0131 |
| terrain 2 | **100%** | **0** | 94.17 | 0.0195 | 97.83 | 0.0134 |
| terrain 3 | **100%** | **0** | 95.12 | 0.0206 | 97.21 | 0.0129 |



Figure 8. Mean coverage(MC) with steps for different algorithms

Table 2. Number of groups completed and rate of completion and number of steps covered by completion

| Indicators/method | Our method | DMPC | GBNN |
|---|---|---|---|
| Number of groups completed(50) | **50(50)** | 19(50) | 12(50) |
| Rate of completion | **100%** | 38% | 24% |
| Average steps | **90.90** | 100.82 | 127.25 |

### 5.4. Elevation-priority coverage mechanism validation

Our algorithm uses the height difference information to make decisions, so the robot prefers higher terrain for full coverage. To test the algorithm's preference for higher terrain, we run 50 experiments on terrains with different maximum elevation values. The experimental platforms and parameters are the same as in the section 5.2. Table 3 compares the average number of steps needed to cover all high points in the region for 50 sets of experiments with other algorithms. Figure 9 displays the number of steps needed to cover all high points in Terrain 1 for each experiment group.

Table 3. Mean number of steps when covered all maxima in different terrains

|                  | Our method | BNN     |
|------------------|------------|---------|
| terrain 1 (2max) | **14.0800**| 21.5000 |
| terrain 2 (3max) | **22.7215**| 30.7468 |
| terrain 3 (4max) | **27.6761**| 35.4648 |

According to Figure 9, our algorithm covers the maximum of a region with fewer steps than the traditional BNN algorithm in 38 out of 50 sets of experiments, while only 12 sets have the same steps for both algorithms. Table 3 shows that our algorithm covers all maxima in a region with 14.0800 steps, a 34.51% speedup over the BNN algorithm. In other terrains, our algorithm also covers the region fastest.
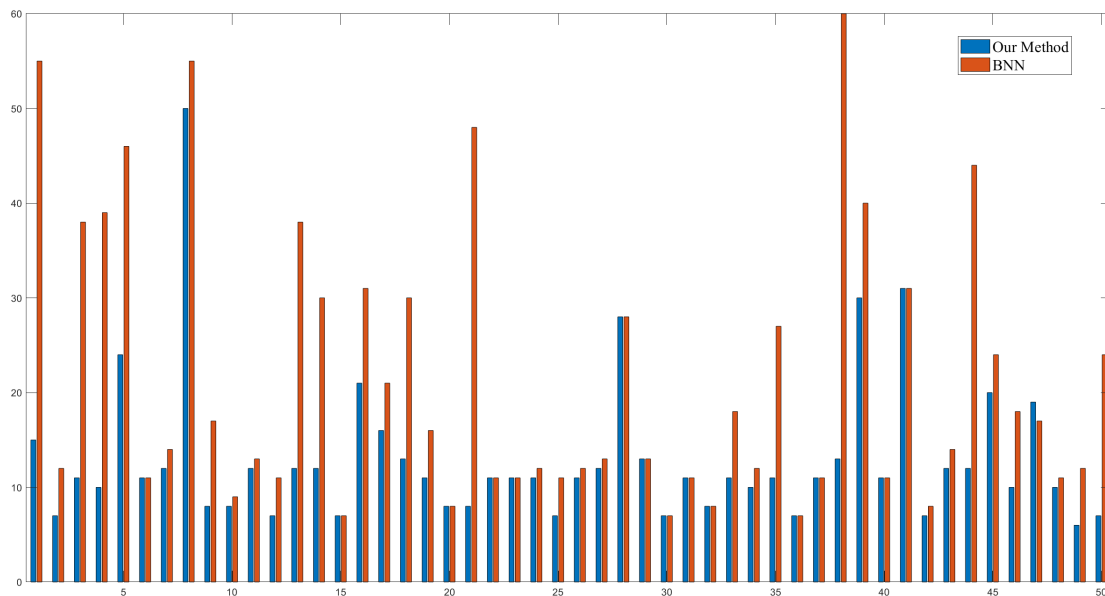


Figure 9. Number of steps when all maxima are covered for 50 sets of experiments

### 6. CONCLUSION

In this paper, we propose an algorithm for multiple robots to collaboratively cover complex unknown terrain based on an improved bio-inspired neural network. The algorithm is improved to address the issues of traditional BNNs falling into local optimums late in covering and the high computational complexity of 3D neural networks in complex environments. The algorithm first builds a 2.5D grid map of the complex terrain environment, preserving height information. Next, an elevation value priority mechanism is proposed by combining a dual BNN model with the 2.5D grid map. This mechanism lets the robot use height difference to make decisions and prefer higher terrain. The first layer of the network plans full-coverage paths in normal mode, and the second layer plans extrication paths when the robot enters a local optimum. In addition, for BNN edges with weak activity values, we added a virtual edge mechanism. The robot can also avoid obstacles and other robots in real time with its obstacle avoidance mechanism. Simulations on different terrains confirm the algorithm's efficacy, achieving 100% coverage, higher efficiency, and faster regional high point detection com-

pared to others. For future work, we aim to transition from simulation to real-world applications by equipping robots with multiple sensors. Focusing on multi-sensor fusion, we will explore how to enhance environmental perception and improve the algorithm's adaptability and robustness in dynamic, real-world scenarios.

## FUNDING INFORMATION

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fangfang Zhang | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | ✓ | | ✓ | ✓ | ✓ |
| Mengdie Duan | | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | | | |
| Jianbin Xin | ✓ | | ✓ | | ✓ | | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Jinzhu Peng | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| C | : **C**onceptualization | I | : **I**nvestigation | Vi | : **Vi**sualization |
| M | : **M**ethodology | R | : **R**esources | Su | : **Su**pervision |
| So | : **So**ftware | D | : **D**ata Curation | P | : **P**roject Administration |
| Va | : **Va**lidation | O | : Writing - **O**riginal Draft | Fu | : **Fu**nding Acquisition |
| Fo | : **Fo**rmal Analysis | E | : Writing - Review & **E**diting | | |

## CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

## INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

## DATA AVAILABILITY

No data was used for the research described in the article.

## REFERENCES

[1] R. I. Mukhamediev, K. Yakunin, and E. A. Aubakirov, "Coverage path planning optimization of heterogeneous unmanned aerial vehicles group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023, doi: 10.1109/ACCESS.2023.3235207.

[2] A. Chen, V. Orlov-Levin, and M. Meron, "Applying high-resolution visible-channel aerial imaging of crop canopy to precision irrigation management," *Agricultural Water Management*, vol. 216, pp. 196–205, 2019, doi: 10.1016/j.agwat.2019.02.017.

[3] Z. Cai, S. Li, Y. Gan, R. Zhang, and Q. Zhang, "Research on complete coverage path planning algorithms based on A* algorithms," *The Open Cybernetics and Systemics Journal*, vol. 8, no. 1, pp. 418–426, 2014, doi: 10.2174/1874110X01408010418.

[4] R. Sun, C. Tang, J. Zheng, Y. Zhou, and S. Yu, "Multi-robot path planning for complete coverage with genetic algorithms," in *Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part V 12*. Springer, 2019, pp. 349–361, doi: 10.1007/978-3-030-27541-9_29.

[5] H. Tu, Y. Deng, Q. Li, M. Song, and X. Zheng, "Improved rapid-exploring random tree global path planning algorithm based on bridge test," *Robotics and Autonomous Systems*, vol. 171, p. 104570, 2024, doi: 10.1016/j.robot.2023.104570.

[6]  R. B"ahnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto, "Revisiting boustrophedon cover-age path planning as a generalized traveling salesman problem," in *Field and Service Robotics: Results of the 12th International Conference*, Springer, 2021, pp. 277–290, doi:10.1007/978-981-15-9460-1_20.

[7]  G. Gao and B. Xin, "A-STC: auction-based spanning tree coverage algorithm for motion planning of cooperative robots," *Frontiers of Information Technology and Electronic Engineering*, vol. 20, no. 1, pp. 18–31, Jan. 2019, doi:10.1631/FITEE.1800551.

[8]  J. C. Maxwell, *A Treatise on Electricity and Magnetism*, vol. 1. Oxford: Clarendon Press, 1873.

[9]  I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, 1963, doi:10.1016/B978-0-12-575303-6.50013-0.

[10]  T. Jingtao and M. Hang, "Large-scale multi-robot coverage path planning via local search," in *Proceed-ings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17567–17574, Mar. 2024, doi:10.1609/aaai.v38i16.29707.

[11]  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, doi:10.1109/ACCESS.2021.3108177.

[12]  X. Cao and C. Y. Sun, "Cooperative target search of multi-robot in grid map," *Control Theory and Applications*, vol. 35, pp. 273–282, Mar. 2018, doi:10.7641/CTA.2017.70242.

[13]  F. Zhang, B. Chen, X. Ban *et al.*, "Multi-robot cooperative search algorithm based on bio-inspired neural net-work and distributed model predictive control," *Control and Decision*, vol. 36, no. 11, pp. 2699–2706, 2021, doi:10.13195/j.kzyjc.2020.0959.

[14]  D. Zhu, X. Cao, B. Sun, and C. Luo, "Biologically inspired self-organizing map applied to task assignment and path planning of an autonomous underwater vehicle system," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 99, pp. 304–313, 2018, doi:10.1109/TCDS.2017.2727678.

[15]  X. Cao, L. Chen, and L. Guo, "Autonomous underwater vehicle global security path planning based on a potential field bio-inspired neural network in underwater environment," *Intelligent Automation and Soft Computing*, vol. 27, no. 2, pp. 391–407, 2021, doi:10.32604/iasc.2021.01002.

[16]  M. Luo, X. Hou, and S. X. Yang, "A multi-scale map method based on bioinspired neural network algorithm for robot path planning," *IEEE Access*, vol. 7, pp. 142682–142691, 2019, doi:10.1109/ACCESS.2019.2943009.

[17]  M. Bagherian and A. Alos, "Three-dimensional unmanned aerial vehicle trajectory planning using evolution-ary algorithms: A comparison study," *The Aeronautical Journal*, vol. 119, no. 1220, pp. 1271–1285, 2015, doi:10.1017/S0001924000011246.

[18]  M. Wang, M. Cong, Y. Du, D. Liu, and X. Tian, "Multi-robot raster map fusion without initial relative position," *Robotic Intelligence and Automation*, vol. 43, no. 5, pp. 498–508, 2023, doi:10.1108/RIA-04-2022-0095.

[19]  A. Wagner, J. Peterson, J. Donnelly, S. Chourey, and K. Kochersberger, "Online aerial 2.5D terrain mapping and active aerial vehicle exploration for ground robot navigation," *Journal of Intelligent and Robotic Systems*, vol. 106, no. 3, p. 58, 2022, doi:10.1007/s10846-022-01751-9.

[20]  S. Choi, J. Park, E. Lim, and W. Yu, "Global path planning on uneven elevation maps," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, IEEE, 2012, pp. 49–54, doi:10.1109/URAI.2012.6462928.

[21]  X. Simon Yang and Max Meng, "Neural network approaches to dynamic collision-free trajectory generation," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2001, doi:10.1109/3477.931512.

[22]  N. Wu, R. Wang, J. Qi, Y. Wang, and G. Wen, "Efficient coverage path planning and underwater topographic map-ping of an unmanned surface vehicle based on A*-improved bio-inspired neural network," *IEEE Transactions on Intelligent Vehicles*, 2024, doi:10.1109/TIV.2024.3422885.

[23]  P. Yao and Z. Zhao, "Improved Glasius bio-inspired neural network for target search by multi-agents," *Information Sciences*, vol. 568, pp. 40–53, 2021, doi:10.1016/j.ins.2021.03.056.

[24]  S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, no. 1, p. 718, 2004, doi:10.1109/TSMCB.2003.811769.

[25]  B. Wang *et al.*, "A noise removal algorithm based on adaptive elevation difference thresholding for ICESAT-2 photon-counting data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 117, p. 103207, 2023, doi:10.1016/j.jag.2023.103207.

[26]  L. Yibing, H. Yujie, Z. Zili *et al.*, "Multi-autonomous underwater vehicle static target search method based on consensus-based bundle algorithm and improved Glasius bio-inspired neural network," *Information Sciences*, vol. 673, p. 120684, 2024, doi:10.1016/j.ins.2024.120684.

[27]  W. Wenhao, Z. Fangfang, X. Jianbin, Y. Hongnian, and L. Yanhong, "An improved multi-robot coverage method in three-dimensional unknown environment based on Glasius bio-inspired neural network," in *Proceedings of Springer*, 2023, pp. 476–483, doi:10.1007/978-981-99-6187-0_47.

# BIOGRAPHIES OF AUTHORS

**Fangfang Zhang** received the B.E. and M.E. degrees in applied mathematics from Shandong University, Jinan, in 2008 and 2011, respectively, and the Ph.D. degree in control science and engineering from Shandong University, in 2015. He is currently an associate professor with Zhengzhou University, Zhengzhou, Henan, China. His research interests include optimal control of multi-agent systems, multi-robot formation, and machine vision. He can be reached by e-mail at zhangfangfang@zzu.edu.cn.
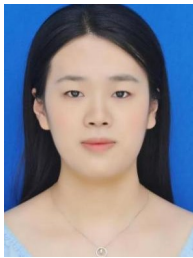
**Mengdie Duan** is a graduate student of Zhengzhou University majoring in electronic information, and her research direction is multi-robot full-coverage search. She received her undergraduate degree from Zhengzhou Aviation Industry Management College. She can be reached by e-mail at 1057127498@qq.com.

**Jianbin Xin** is an associate professor in the Department of Automation at Zhengzhou University. He earned his Ph.D. in Mechanical, Maritime, and Materials Engineering from Delft University of Technology. His research focuses on multi-robot coordination and path planning, as well as intelligent logistics system scheduling and optimization. He can be reached by e-mail at j.xin@zzu.edu.cn.

**Jinzhu Peng** is a professor and doctoral supervisor at Zhengzhou University and holds a Ph.D. from Hunan University. His main research areas are robot motion planning, human-machine collaboration, and intelligent information processing. He has made great contributions in these fields and also guides doctoral students. He can be reached by email at jzpeng@zzu.edu.cn.