

A Multi-robot Distributed Collaborative Region Coverage Search Algorithm Based on Glasius Bio-inspired Neural Network

Bo Chen, Hui Zhang, Fangfang Zhang*, Yanhong Liu, *Member, IEEE*, Cheng Tan, *Member, IEEE* and Hongnian Yu, *Senior Member, IEEE*, Yaonan Wang

Abstract—There are many constraints for a multi-robot system to perform a region coverage search task in an unknown environment. To address this, we propose a novel multi-robot distributed collaborative region coverage search algorithm based on Glasius bio-inspired neural network (GBNN). Firstly, we develop an environmental information updating model to represent the dynamic search environment. This model converts the environmental information detected by the robot into dynamic neural activity landscape of GBNN. Secondly, we introduce the distributed model predictive control method in search path planning to improve search efficiency. In addition, we propose a distributed collaborative decision-making mechanism among the robots to produce several dynamic search sub-teams. Within each sub-team, collaborative decisions are made among the robot members to optimize the solution and obtain the next movement path of each robot. Finally, we conduct experiments in three aspects to verify the effectiveness of the proposed method. Compared with three algorithms in this field, the experimental results demonstrate that the proposed algorithm exhibits good performance in a multi-robot region coverage search task.

Index Terms—Multi-robot system, region coverage search, Glasius bio-inspired neural network, distributed collaborative decision-making mechanism, search sub-team.

I. INTRODUCTION

Growing evidences show that with the development of robot technology, autonomous robots have been able to assist or replace human to complete many assigned tasks, such as battle reconnaissance [1], space exploration [2], [3], rescue after disaster [4], [5], and floor-cleaning [6]. Those tasks can be summarized as the problem of complete area coverage search (CACS) in an unknown environment. The unknown environment which indicates that the distribution of targets and obstacles in the task area is unknown, but the boundary is known [7]. This requires the robot to continuously search to the unexplored area and avoid repeated exploration. The constraints faced by an autonomous robot in the unknown

This work is supported by the National Natural Science Foundation of China (61603345, 61703372, 61773351).

Bo Chen, Fangfang Zhang, Yanhong Liu and Hongnian Yu are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, Henan, 450001 P.R.China (e-mail: cb233cb@163.com; zhangfangfang@zzu.edu.cn; liuyh@zzu.edu.cn; H.Yu@napier.ac.uk).

Hui Zhang and Yaonan Wang are with School of Robotics, Hunan University, Changsha 410082, China (e-mail:zhanghuihy@126.com; yaonan@hnu.edu.cn)

Cheng Tan is with School of Engineering, Qufu Normal University, Rizhao, Shandong, 276826 P.R.China (e-mail:tancheng1987love@163.com)

Hongnian Yu is also with the School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, EH10 5DT, U.K.

environments are as follows [8]: 1) the detection range of the robot's sensor is limited relative to the task area's size; 2) the autonomous robot has no prior environment information (including the distribution of targets and obstacles in the task area) before the search task starts; 3) the autonomous robot must avoid obstacles at real time in the unknown environment.

Compared with a single autonomous robot, a multi-robot system (MRS) has the advantages of flexibility, robustness and parallelism [9], [10], which can improve the efficiency of a region coverage search task. Therefore, we consider using multiple autonomous robots to conduct the area coverage search task in an unknown environment. The task requires multiple robots obtain environmental information through sensors, and cooperate to complete the coverage search of the unknown environment with the minimum cost [11]. What's more, the communication conditions may not be ideal in some harsh environments such as post-disaster rescue sites. Hence, when the MRS performs the region coverage search task, there are other two constraints need to be satisfied: 1) the collisions among the robots should be avoided in the search process; 2) the communication range of the robots is limited, only local communication can be established among the robots.

Due to the above constraints faced by the MRS, each robot needs to make real-time decisions on the next search path according to the updated environmental information. Several researches on the CACS problem have been conducted. For example, Jia et al. [12] proposed a distributed cooperative search strategy based on the initial target information for multiple underwater robots. However, it is difficult to obtain the precise initial location of the target in an unknown environment.

The researches on improving search efficiency by dividing the task area for each robot have been conducted in the literature. Hu et al. [13] divided the whole surveillance region into cells, the target search problem is transformed into a region coverage problem. Cai et al. [14] divided the task area into Voronoi cells, and the robots were guided by the bacterial chemotaxis (BC) algorithm. Furthermore, the methods for the polygon task area division have also attracted the interests of researchers. Vinh et al. [15] approximated non-convex polygons to convex polygons by removing concave points for the region division. Lasse et al. [16] used the extension line of the edge of the notch to decompose the polygon to realize the area division. However, the practices of dividing the task area in advance may lead to poor robustness of MRS. If a robot is damaged, the corresponding task area may not be searched.

Several researchers have applied the learning-based approach for the region search task. Li et al. [17] proposed a two-stage imitation learning framework for the multi-target search problem in swarm robotics. Liu et al. [18] guided the robot to plan the search path based on reinforcement learning. However, those methods may not be suitable for real-time applications since the learning process is essential and will take substantial time.

Some researchers have also tried to use the evolutionary algorithms to solve the CACS problem, such as particle swarm optimization algorithm (PSO) [19], adaptive robotic PSO (ARPSO) algorithm [20], bat algorithm (BA) [21] and immune genetic algorithm (IGA) [22]. These methods usually design the fitness function of the evolutionary algorithm according to the target signal value to guide the robots to search the task area. However, it is difficult to perceive the target signal globally for the robots in the complex unknown environments.

Some researchers focus on the dynamic task assignment of multiple robots to improve the efficiency of CACS. Hou et al. [23] developed a mission planning system (MPS), consisting of preliminary planning, task assignment, and post-planning layers. Dai et al. [24] adopted an auction-based approach to realize task assignment in the multi-robot region search. Tang et al. [25] proposed an improved grouping strategy based on constrictive factors PSO. Task allocation is an effective way to improve search efficiency [26]. However, it is necessary to realize multi-robot task assignment cooperation under the unknown environment with incomplete environmental information.

Recently, more and more researchers tend to use the Glasius bio-inspired neural network (GBNN) to guide multiple robots to perform CACS tasks under the unknown environments. The GBNN is inspired from the Hodgkin and Huxley's membrane model for a biological neural system [27] and the shunting model [28]. Besides, since the neural connection weights are set in the model design and the selection range is very wide, there is no need to find the optimal connection weights among neurons. Therefore, the GBNN has no learning procedures, which can work in real-time. Compared with other intelligent algorithms, the GBNN based approaches don't rely on any prior information about the dynamic environment. Luo et al. [29] used the GBNN to guide multiple cleaning robots sweeping the task area, and an extension was provided to avoid deadlock situations among the robots [30]. Moreover, the GBNN was further used in the area coverage search of the two-dimensional plane [31], [32] and the three-dimensional space [33], [34] under the unknown environments. In our previous work [11], a distributed model predictive control (DMPC) algorithm based on the GBNN model was proposed for multi-robot area coverage search in unknown environment. However, several simulation experiments show that the GBNN based methods faces difficulty to adapt to the complex obstacle environment and still need to be improved in the performance of multi-robot cooperation.

Considering the above challenges faced in the recent studies, we propose a novel multi-robot distributed collaborative region coverage search algorithm based on the GBNN under the unknown environments. The proposed algorithm not only

meets the the constraints mentioned above, but also has corresponding improvements in search efficiency and collaboration performance among the robots. Compared to the DMPC algorithm [11]), the main contributions of this paper are:

- developing an environmental information updating model to describe the dynamic search environment, in which the varying environmental information is represented by the dynamic activity landscape of the GBNN and the weighted average method is applied to fuse environmental information collected by different robots.
- applying the DMPC method in search path planning to improve search efficiency, in which a new search objective function based on the GBNN (Eq.(12)) under the framework of DMPC is designed for complex obstacle environment.
- proposing a novel distributed collaborative decision-making mechanism (subsection IV-B) among the robots to produce several dynamic search sub-teams, in which each sub-team makes iterative decisions separately to improve cooperation performance.
- demonstrating that the proposed algorithm can maintain effectiveness and superiority in the multi-robot region coverage search task through the experiment study under different obstacle environments.

The remaining sections of this article are arranged as follows. Section II introduces the basic concepts used in this article. Section III shows the environment information update model. Section IV presents the distributed collaborative decision-making process. Section V conducts the corresponding experiments and section VI gives the conclusions.

II. BASIC CONCEPTS

A. Task Area Model

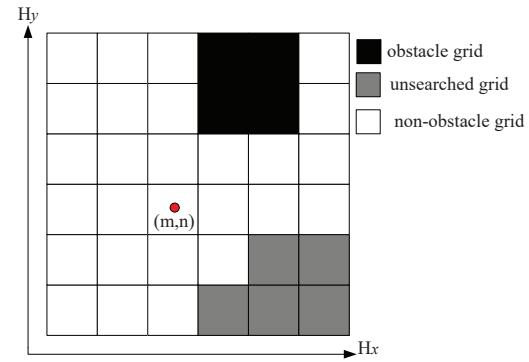


Fig. 1. The grid map

We assume that the shape of the task area to be searched is a rectangle, and the task area is represented by a grid map, as shown in Fig.1. The area is divided into $H_x * H_y$ squares grids with the same size, and the set of all grids is denoted as $E = \{(m, n) | m = 1, 2, \dots, H_x; n = 1, 2, \dots, H_y\}$, $E(m, n)$ represents the grid in position (m, n) . Assuming that the sensor's detection range of each robot is $m_d * n_d$ squares grids, if the target signal or obstacle appears in the detection range of the robot, it is considered that the target has been found.

According to the environmental information detected by the robot in the search process, each grid has one of four states, as shown in Eq.(1). Since all robots have no prior environmental information, the initial state of each grid is K_u .

$$S\{E(m,n)\} = \begin{cases} K_u, & E(m,n) \text{ is the unexplored grid} \\ K_o, & E(m,n) \text{ is the obstacle grid} \\ K_c, & E(m,n) \text{ is the no obstacle and no target grid} \\ K_t, & E(m,n) \text{ is the target grid} \end{cases} \quad (1)$$

where $S\{E(m,n)\}$ represents the state of the grid $E(m,n)$.

B. Robot Motion Model

Supposing N_r robots perform the region coverage search task, and the s -th robot is denoted as R_s , $s = 1, \dots, N_r$, the robot set is $\mathbf{R} = \{R_1, R_2, \dots, R_{N_r}\}$, if the position of R_s at the k -th step is $\{x_s(k), y_s(k)\}$, then the position of R_s at the $(k+1)$ -th step satisfies Eq.(2) below.

$$\begin{bmatrix} x_s(k+1) \\ y_s(k+1) \end{bmatrix} = \begin{bmatrix} x_s(k) \\ y_s(k) \end{bmatrix} + v_s(k)\Delta t \begin{bmatrix} \cos(\omega_s(k)) \\ \sin(\omega_s(k)) \end{bmatrix} \quad (2)$$

where $v_s(k)$ is the motion velocity of R_s , $v_s(k) \in [0, v_{\max}]$; Δt is the interval time between steps; $\omega_s(k)$ is the R_s 's motion direction, $\omega_s(k) \in [0, \omega_{\max}]$, to simplify calculations and fit the grid map, $\omega_s(k) \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ\}$.

Figure 2 shows the Moore neighborhood of the robot at different positions in the grid map. In Fig.2, the blue circles represent the current positions of the robot; the arrows represent the possible motion directions ($\omega_s(k)$) of the robot. When the robot occupies a grid, it will move to one of the adjacent grids in the next step.

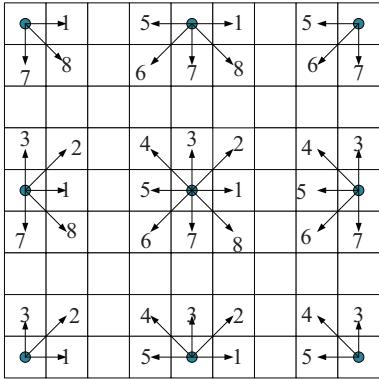


Fig. 2. Possible motion direction of robot.

C. Communication Mechanism Among Robots

Considering the limited communication ability of a single robot, only local communication can be established among robots. The mobile ad hoc network (MANET) is a multi-hop wireless communication network composed of mobile nodes, where the nodes act as both information terminals and routers [35]. Each robot is regarded as a communication node,

then the set of robots that can communicate is defined as follows: $P_z = \{\{R_s, R_h\} : R_s, R_h \in \mathbf{R}; d(R_s, R_h) \leq L_c\}$, $z = 1, \dots, n_c$, n_c is the number of communication sets, L_c is maximum communication radius, $d(R_s, R_h)$ is the Euclidean distance between R_s and R_h , which is defined by Eq.(3). It is noted that the number of communication sets is dynamic when the robots move.

$$d(R_s, R_h) = \sqrt{(x_s - x_h)^2 + (y_s - y_h)^2} \quad (3)$$

where (x_s, y_s) , (x_h, y_h) are the coordinates of R_s , R_h , respectively.

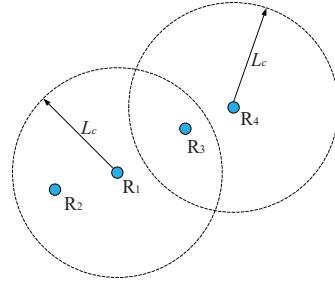


Fig. 3. A collection of communicable robots.

An example diagram of robot communication is shown in Fig.3. Since both R_2 and R_3 are within the maximum communication distance of R_1 , and R_3 also act as the intermediate node between R_1 and R_4 , the corresponding communication set is $\{R_1, R_2, R_3, R_4\}$.

III. ENVIRONMENTAL INFORMATION UPDATE MODEL

Since each robot performs the area search task in an unknown environment, the robot needs to continuously probe and update its surrounding environmental information, which is the basis of its next movement strategy. Therefore, we propose a new environmental information update model in this section. The proposed model mainly includes two parts: (1) the environmental information detected by the robot through the sensor will be transformed into the neuron's activity value of the GBNN; (2) within each communication set P_z ($z = 1, \dots, n_c$), the weighted average method (WAM) is used for information fusion among robots.

A. Glasius Bio-inspired Neural Network

The GBNN model is combined with the grid map to represent the dynamic search environment. The diagram of the GBNN structure is shown in Fig.4. There are only local lateral connections among neurons. Each circle in Fig.4 represents a neuron with a matching neuron activity value Q . The initial activity value of the GBNN is Q_0 and its dynamic characteristics are shown in Eq.(4) [31]. It can be seen from Eq.(4) that the excitatory input of neurons have a global propagation effect in the neural network, while the inhibitory input only has a local influence.

$$\frac{dQ_i}{dt} = -AQ_i + (B - Q_i)([\rho_i]^+ + \sum_{j=1}^M W_{ij}[Q_j]^+) - (D + Q_i)[\rho_i]^- \quad (4)$$

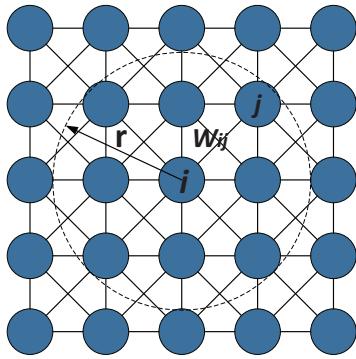


Fig. 4. Diagram of the GBNN structure

where Q_i is the activity value of the neuron i ; ρ_i represents the external stimulus signal of the neuron i , $[\rho_i]^+ + \sum_{j=1}^M W_{ij}[Q_j]^+$ represents the excitatory input, $[\rho_i]^+ = \max(\rho_i, 0)$, $[\rho_i]^-$ represents the inhibitory input, $[\rho_i]^- = \max(-\rho_i, 0)$; $[Q_j]^+ = \max(Q_j, 0)$, Q_j is the activity value of the neuron j adjacent to the neuron i ; M is the number of the adjacent neurons (within its radius r); A , B and D are all positive constants, A represents the decay rate of Q_i , and Q_i is bounded in the finite interval B and D ($Q_i \in [-D, B]$); W_{ij} represents the connection weight coefficient between the neuron i and the adjacent neuron j , which is defined as Eq.(5).

$$W_{ij} = f_1(|e_i - e_j|) = \begin{cases} \frac{\alpha}{|e_i - e_j|}, & d_0 \leq |e_i - e_j| < r \\ 0, & \text{other} \end{cases} \quad (5)$$

where $|e_i - e_j|$ represents the Euclidean distance between the vectors e_i and e_j in the state space; d_0 is the shortest distance between neurons ($d_0=1$ in this paper); α and r are both positive constants, generally make $W_{ij} \in [0, 1]$. Since the GBNN consists of $H_x * H_y$ neurons and each neuron has at most eight local connections, the total number of neural connections is $8H$ ($H=H_x * H_y$). Consequently, the computational complexity of the GBNN is $O(H)$.

The purpose of the proposed model is to represent dynamic environmental information through neuron's activity landscape. It can be seen from Eq.(4) that the change of neuron's activity value depends largely on external stimulus signal ρ_i (excitatory input or inhibitory input). Besides, the term $\sum_{j=1}^M W_{ij}[Q_j]^+$ ensures that the excitatory input can be transmitted globally, while the inhibitory input only acts locally. By defining appropriate external inputs, the targets or unsearched areas will be at the peak of the neuron's activity landscape and the obstacle will be at the lowest point. Due to the propagation effect of positive neuron's activity, the unsearched areas and targets will attract the robots globally. In addition, the obstacles only have local influence in a small area to avoid the collision. The activity value of the unexplored region was the highest before the target signal was detected. Once the target signal is detected, the activity of the corresponding region will be at the peak of the neuron's activity landscape. In the formulation of search strategy, the robots can generally move towards the area with a large

neuron's activity value, which can avoid obstacles and avoid to repeatedly explore the same area.

Based on the above discussion, the association between the GBNN and the grid map is shown in Eq.(6). According to Eq.(6), the activity value of the target grid ($S\{E(m, n)\}=K_t$) is at the peak, the activity value of the unexplored grid ($S\{E(m, n)\}=K_u$) is secondary, the activity value of the obstacle grid ($S\{E(m, n)\}=K_o$) is at the low point, and the activity value of the non-obstacle grid ($S\{E(m, n)\}=K_c$) is only higher than that of the obstacle grid. In the process of searching, each robot updates the activity value of neurons within its detection range according to the sensor information. The activity value matrix of the GBNN stored by the robot R_s is denoted as H_s , $s = 1, \dots, N_r$.

$$\rho_i(m, n) = \begin{cases} I & \text{if } S\{E(m, n)\} = K_t \\ B & \text{if } S\{E(m, n)\} = K_u \\ 0, & \text{if } S\{E(m, n)\} = K_c \\ -I, & \text{if } S\{E(m, n)\} = K_o \end{cases} \quad (6)$$

where $I \gg B$ is a very large positive constant, which ensures that the target signal will be at the peak in the neuron's activity landscape and the obstacle's activity value will be at its lowest; $E(m, n)$ represents the corresponding grid of the neuron i .

B. Environmental Information Fusion

Because the robot's detection range is limited, it can only obtain the environment information within its detection range. Therefore, it is necessary to integrate the environmental information collected by the robots (in the same communication set) to improve search efficiency. The weighted average method is used to fuse the environmental information. For any local communication set P_z ($z = 1, \dots, n_c$), the information fusion method is shown in Eq.(7).

$$H^{P_z}(m, n) = \sum_{g=1}^{n_r^z} W_g * H_g^{P_z}(m, n) \quad (7)$$

where $H_g^{P_z}$ represents the neuron activity value matrix of the g -th robot in communication set P_z ; $H^{P_z}(m, n)$ represents the neuronal activity value of the grid ($(E(m, n))$) after information fusion. $n_r^z = |P_z|$ is the number of robots in the set P_z , and W_g is the corresponding weight coefficient.

The weight coefficient W_g is defined as follows:

$$W_g = \begin{cases} \frac{N_{m,n}^g}{\sum_{g=1}^{n_r^z} N_{m,n}^g}, & \sum_{g=1}^{n_r^z} N_{m,n}^g \neq 0 \\ \frac{1}{n_r^z}, & \sum_{g=1}^{n_r^z} N_{m,n}^g = 0 \end{cases} \quad (8)$$

where $N_{m,n}^g$ denotes the number of explores by robot R_g to the grid ($(E(m, n))$).

Algorithm 1 is the initialization part of the proposed method, including the state of grid map, the GBNN model and the initial position of each robot. **Algorithm 2** describes the updating process of environment information in detail.

Algorithm 1 Initialization

- 1: Set $S\{E(m, n)\} = K_u$, $\forall 1 \leq m \leq H_x, 1 \leq n \leq H_y$
 // Initialize the state of the grid map (set the state of all areas as unexplored).
- 2: Set $Q(m, n) = Q_0$; $\rho(m, n) = 0$, $\forall 1 \leq m \leq H_x, 1 \leq n \leq H_y$.
 // Set the initial state of the GBNN
- 3: **for** $s = 1$ to N_r **do**
- 4: $x_s(0) = P_x^s; y_s(0) = P_y^s$
 // Set the initial positions $(x_s(0), y_s(0))$ of the robot R_s .
- 5: **end for**

Algorithm 2 Updating Environmental Information

Input: current positions of robots $x_s(k), y_s(k)$, activity value matrix H_s , $s \in [1, N_r]$; the state of the grid map $S\{E(m, n)\}$, $m \in [1, H_x], n \in [1, H_y]$.

Output: the renewed activity value matrix H_s , $s \in [1, N_r]$; the state of the grid map $S\{E(m, n)\}$, $m \in [1, H_x], n \in [1, H_y]$.
 // Update the state of the grid map.

- 1: **for** $s = 1$ to N_r **do**
- 2: R_s update $S\{E(m, n)\}$ based on Eq.(1), $\forall 1 \leq m \leq H_x, 1 \leq n \leq H_y$.
 // Each robot updates the grid state based on the information detected by the sensor.
- 3: **end for**
 // Update the activity landscape of the GBNN.
- 4: **for** $m = 1$ to H_x **do**
- 5: **for** $n = 1$ to H_y **do**
- 6: Update $\rho_i(m, n)$ using Eq.(6).
- 7: Calculate H_s using Eq.(4).
- 8: **end for**
- 9: **end for**
 // Fuse the environmental information in each local communication set P_z ($z = 1, \dots, n_c$).
- 10: **for** $z = 1$ to n_c **do**
- 11: Fuse the activity value $H^{P_z}(m, n)$ using Eq.(7), $m \in [1, H_x], n \in [1, H_y]$.
- 12: **end for**

IV. DISTRIBUTED COLLABORATIVE DECISION-MAKING

After the environment information is updated, each robot needs to decide the next movement path in real-time. Considering that the existing methods [11], [31], [33] have the problems of poor cooperation performance among robots and low search efficiency in complex obstacle environment. We propose a new distributed collaborative decision-making method for the MRS. It includes the content of two following aspects concretely: (1) design a new search objective function based on the GBNN under the framework of the DMPC to adapt to complex obstacle environment; (2) establish several dynamic search sub-teams, and the robots in each sub-team make collaborative decisions based on the greedy iteration idea and DMPC to obtain the next movement path.

A. Distributed Model Predictive Control Method

In order to improve the search efficiency of the MRS, the DMPC method is introduced. The core idea of the DMPC is the rolling optimization [36]. Each robot is a subsystem of the MRS. According to the robot motion model established in subsection II-B, the corresponding state space model of each subsystem can be established. For the robot R_s ($s = 1, \dots, N_r$), its state at the k -th step is expressed as $h_s(k) = \{x_s(k), y_s(k)\}$. $\{\omega_s(k), v_s(k)\}$ is selected as the control input $u_s(k)$ of the subsystem, then the state equation of subsystem R_s is shown in Eq.(9).

$$h_s(k+1) = f\{h_s(k), u_s(k)\} \quad (9)$$

where $s = 1, \dots, N_r$; $f\{\cdot\}$ is the state transfer function of the subsystem, determined by Eq.(2).

According to Eq.(9), the state prediction model of the subsystem R_s is shown in Eq.(10).

$$h_s(k+l|k) = f\{h_s(k+l-1|k), u_s(k+l-1|k)\} \quad (10)$$

where $h_s(k+l|k)$ is the predicted state at the $(k+l)$ -th step, and its value depends on the state $h_s(k+l-1|k)$ and the control input $u_s(k+l-1)$; $l = 1, 2, \dots, L$, L is the number of the predicted steps.

For any subsystem R_s , according to Eqs.(9) and (10) establish a predicted L -steps cumulative search objective function $J^{(L)}\{h_s(k), u_s(k)\}$ as

$$J^{(L)}\{h_s(k), u_s(k)\} = \sum_{j=0}^{L-1} J\{h_s(k+j), u_s(k+j)\} \quad (11)$$

where $J\{(h_s(k), u_s(k)\}$ represents the single-step search objective function, defined as follows:

$$J\{(h_s(k), u_s(k)\} = \lambda_1 J_1(k) + \lambda_2 J_2(k) \quad (12)$$

where $J_1(k)$ is the search gain function; $J_2(k)$ is the turning loss function; λ_1 and λ_2 are the corresponding weight coefficients respectively, $\lambda_1, \lambda_2 \in [0, 1]$.

The search gain function should be able to guide the robot to search the target in the unexplored areas, avoid obstacles and collisions among the robots. Based on the discussion in subsection III-A, the activity value of the target is at the peak of the neuron activity landscape, and the activity value of the unexplored grid is secondary. Both of them have a global attraction effect for the robot. In addition, the activity value of the obstacle is at the lowest point in the landscape of neuron activity value and only has local influence. The robot can move towards the area with a high activity value, which is conducive to finding the target and realizing obstacle avoidance. Therefore, $J_1(k)$ is designed as Eq.(13). In $J_1(k)$, the purpose of the first item is to guide the robot to detect the target in the unexplored area, the second item is to avoid obstacles, and the third item is to avoid collisions among robots. Compared to the search gain function designed in the DMPC algorithm [11], only the neuron's activity values of the grids (K_t , K_u , and K_c) are accumulated in the first term of $J_1(k)$. The purpose is to prevent excessive obstacles in the

robot's detection range from adversely affecting the search efficiency.

$$J_1(k) = \begin{cases} \frac{\sum_{z=1}^{\eta} Q_s^z(k)}{\eta * Q_{\max}}, & S\{E(x_s(k), y_s(k))\} = K_c \\ J_{\min}, & S\{E(x_s(k), y_s(k))\} = K_o \\ J_{\min}, & E(x_s(k), y_s(k)) \in U_{\sim s}(k) \end{cases} \quad (13)$$

where $E(x_s(k), y_s(k))$ is the position state of R_s at the k -th step; $U_{\sim s}(k)$ represents the set of motion positions planned by other robots at the k -step; $Q_s^z(k)$ represents the neuron's activity value of the z -th non-obstacle grid (K_t , K_u , and K_c) in the R_s 's coverage range at the k -th step; η represents the number of non-obstacle grids in the R_s 's coverage range; $Q_{\max} = B$; J_{\min} is the minimum value ($J_{\min} = -10$ in this paper).

The energy loss caused by the robot making frequent turns during the search should be considered, so the turning loss function $J_2(k)$ is defined as Eq.(14). $J_2(k)$ is inversely proportional to the turn of the robot.

$$J_2(k) = -\min\{|\omega_s(k) - \omega_s(k-1)|, 360 - |\omega_s(k) - \omega_s(k-1)|\} / \omega_M \quad (14)$$

where $\omega_M = 180^\circ$ in this work.

By maximizing the cumulative search objective function, the optimal solution of the predicted L -steps can be determined, as shown in Eq.(15). The first step ($u_s(k)$) of the L -steps predictive solution ($\hat{U}_s(L)$) is selected as the next motion control input of the subsystem R_s . When the robot moves to the next step, the above process is repeated after updating the environment information. Considering the superiority of the DE algorithm [37] to solve nonlinear problems, we choose the DE algorithm to obtain the optimization solution $\hat{U}_s(L)$. Due to the page limitation, the details of the algorithm will not be described.

$$\begin{aligned} \hat{U}_s(L) &= \arg \max J^{(L)} \{h_s(k), u_s(k)\} \\ \text{s.t. } &\begin{cases} h_s(k+l|k) = f \{h_s(k+l-1|k), u_s(k+l-1|k)\} \\ l = 1, 2, 3, \dots, L \\ h_s(k|k) = h_s(k) \end{cases} \end{aligned} \quad (15)$$

where $\hat{U}_s(L) = \{u_s(k), u_s(k+1), \dots, u_s(k+L-1)\}$.

B. Collaborative Decision-making Within the Search Sub-team

In the DMPC algorithm [11], only environmental information is exchanged between robots, and each robot does not consider the decision information of other robots when making decisions, which may lead to competition among robots and the decline of cooperation performance. To improve the cooperation performance of the MRS, we propose a novel distributed collaborative decision-making method among the robots to produce several dynamic search sub-teams, in which each sub-team makes iterative decisions based DMPC method separately. The advantages of forming multiple sub-teams are: 1) Reducing the communication time of robots through forming multiple sub-teams and making decisions

independently in each sub-team, which can greatly reduce the exchange of redundant information; 2) Improving robots adaptability to unknown environments since all robots can obtain environmental information detected by neighboring robots in each sub-team; 3) Improving efficiency since the robots in the sub-team can make collaborative decisions through decision-making information exchange.

The steps for collaborative decision-making within a search sub-team are as follows: First of all, several robot search sub-teams are established in each local communication set. The conditions for joining a search sub-team are given in Eq.(16). There are two main conditions: (1) the robots belong to a same communication set (see subsection II-C for the definition of communication set.); (2) the prediction range is overlap. The first condition guarantees that the robots in each sub-team could communicate with each other, and the second condition guarantees that the robots with decision-making conflicts will form a search sub-team. It is noted that the number of the search sub-teams and the member of each sub-team are dynamic with the robots move. Figure 5 shows the composition of several search sub-teams (nine robots) at different moments.

$$\begin{aligned} R_s, R_h &\in A^i, i = 1, 2, \dots, n_t \\ \text{s.t. } &\begin{cases} R_s, R_h \in P_z, z = 1, \dots, n_c \\ d(R_s, R_h) < 2L * D_R \end{cases} \end{aligned} \quad (16)$$

where A^i represents the i -th search sub-team; n_t is the number of the search sub-teams; L is the predicted steps of the robot; D_R is the maximum movement distance of the robot at each step ($D_R = v_{\max} * \Delta t$).

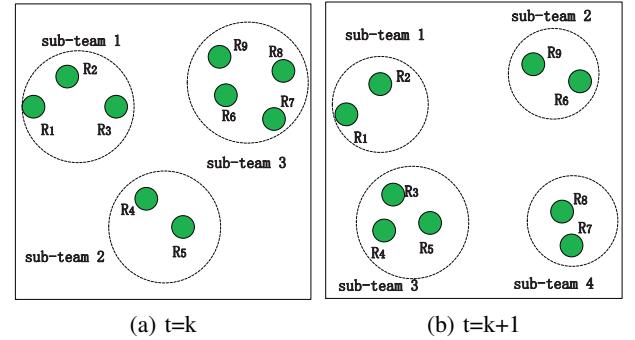


Fig. 5. Search sub-team sample graph.

After the search sub-teams are established, the robots in each sub-team make collaborative decisions based on the greedy iteration idea and the DMPC. Each robot in the sub-team considers the decision information of other robots when making decisions, which can reduce duplicated search and avoid competition among the robots. For any search sub-team A^i ($i = 1, 2, \dots, n_t$), the specific steps are as below.

(1) Determine the sequence of iterative decision. The decision order of robots should meet the principle of self-organization. In search sub-team A^i , the decision sequence of the robots is denoted as $\{R_a^{A^i}, R_{a+1}^{A^i}, \dots, R_{a+n_4^i-1}^{A^i}\}$. $R_a^{A^i}$ is the first robot to make the decision, n_4^i is the number of members in the sub-team A^i . The generation rule of $R_a^{A^i}$ is

shown in Eqs.(17) and (18), $C^{A^i} = (\bar{x}^{A^i}, \bar{y}^{A^i})$ is the centroid of the search sub-team A^i , and $R_a^{A^i}$ is the robot closest to C^{A^i} . $R_{a+1}^{A^i}, R_{a+2}^{A^i}, \dots, R_{a+n_4^i-1}^{A^i}$ are generated from the robots that closest to the previous iterative robot respectively.

$$\begin{cases} \bar{x}^{A^i} = \frac{\sum_{g=1}^{n_4^i} x_g^{A^i}}{n_4^i} \\ \bar{y}^{A^i} = \frac{\sum_{g=1}^{n_4^i} y_g^{A^i}}{n_4^i} \end{cases} \quad (17)$$

where $(x_g^{A^i}, y_g^{A^i})$ represents the coordinate of the g -th robot in the search sub-team A^i .

$$R_a^{A^i} = \arg \min \{d(C^{A^i}, R_g^{A^i}), g = 1, 2, \dots, n_4^i\} \quad (18)$$

where $R_a^{A^i}$ is the first robot to make the decision.

(2) Iterative decision-making. According to its neuron activity value matrix H_a and Eq.(15), $R_a^{A^i}$ obtains the predicted L -steps control input $\hat{U}_a(L)$. Then the predicted L -steps movement positions of $R_a^{A^i}$ are obtained based on Eq.(10), which are sent to the next decision maker $R_{a+1}^{A^i}$. This process is repeated for $R_{a+v}^{A^i}$ ($v = 1, \dots, n_4^i - 1$).

(3) Move to the next step. Each robot in the search sub-team chooses the first component of its L -steps predictive control input to move to the next step.

Algorithm 3 shows the pseudo-code diagram of the distributed collaborative decision-making process among robots.

C. Region Coverage Search Process

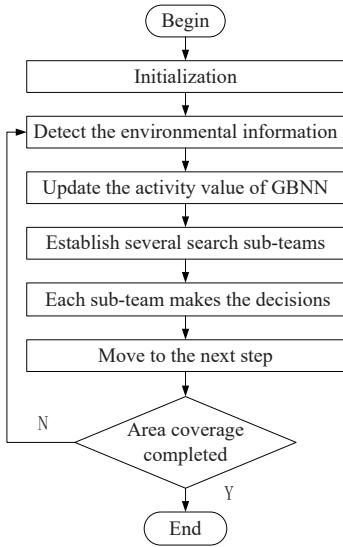


Fig. 6. Region coverage search flow chart.

The region coverage search flow chart of the MRS is shown in Fig.6. Firstly, the corresponding parameters of the proposed method are initialized (see **Algorithm 1**). Secondly, each robot updates the activity value of the GBNN according to the environment information detected by the sensors, and the WAM is used to fuse environment information collected by robots (see **Algorithm 2**). Then several robot search sub-teams are established and the robots in each sub-team make

Algorithm 3 Distributed Collaborative Decision-making

Input: the current positions of the s -th robot $x_s(k), y_s(k)$, activity value matrix H_s , $s \in [1, N_r]$; the state of grid map $S\{E(m, m)\}$, $m \in [1, H_x]$, $n \in [1, H_y]$.

Output: the next movement positions planned by the robot $x_s(k+1), y_s(k+1)$, $s = 1, 2, \dots, N_r$.

- 1: Produce several search sub-teams $A^i, i = 1, 2, \dots, n_t$ using Eq.(16).
// Each search sub-team A^i makes the decision on next movement positions.
 - 2: **for** $i = 1$ to n_t **do**
 - 3: Determine the sequence of iterative decision-making in A^i using Eq.(18).
 - 4: $v = 0$
// Iterative decision-making in each sub-team.
 - 5: **repeat**
 - 6: obtain the predicted L -steps control input $\{u_{a+v}(k), \dots, u_{a+v}(k+L-1)\}$ using Eq.(15).
 - 7: obtain the predicted L -steps movement positions $\{x_{a+v}^p(k+1), y_{a+v}^p(k+1)\}, \dots, \{x_{a+v}^p(k+L), y_{a+v}^p(k+L)\}$ using Eq.(10) and send it to $R_{a+v+1}^{A^i}$.
// The robot $R_{a+v}^{A^i}$ make the decision.
 - 8: $v = v + 1$.
 - 9: **until** $v > n_4^i - 1$
// The sub-team A^i completes the decision.
 - 10: **end for**
// All robots finish the decision process and obtain the next movement positions.
 - 11: Each robot moves to the next position $x_s(k+1), y_s(k+1)$, $s = 1, 2, \dots, N_r$.
-

collaborative decisions to obtain the next movement path (see **Algorithm 3**). Finally, each robot moves to the next step. If the task area is fully searched or all targets are detected, the multi-robot area coverage search task ends. Otherwise, each robot still updates the GBNN and repeats the above process.

D. Algorithm Stability Analysis

The neuron's activity value in Eq.(4) is bounded in the finite interval $[-D, B]$. In Eq.(4), the neuron's activity value increases at a rate of $(B - Q_i)([\rho_i]^+ + \sum_{j=1}^M W_{ij}[Q_j]^+)$, which is proportional to the neuron's external excitation signal $([\rho_i]^+ + \sum_{j=1}^M W_{ij}[Q_j]^+)$ and the auto gain control term $(B - Q_i)$. When the neuron's activity value Q_i is less than B and the neuron receives excitatory input, the closer Q_i gets to B , the slower Q_i grows. Once Q_i is equal to B , the value of $(B - Q_i)([\rho_i]^+ + \sum_{j=1}^M W_{ij}[Q_j]^+)$ becomes zero and Q_i will stop increasing. In case the Q_i is bigger than B , the term $(B - Q_i)$ will become negative, which will pull Q_i back to B . So B is the upper bound of the neuron's activity value. Similarly, the inhibitory term $-(D+Q_i)[\rho_i]^-$ forces the neural activity stay above the lower bound $-D$. Therefore, once the neuron's activity value $Q_i \in [-D, B]$, it stays within this range for any external stimulus.

Besides, the stability and convergence of the proposed GBNN model can be proved by Lyapunov stability theory [38]. Constructing new variables, $\Phi_i = Q_i - B$, three corresponding equations can be obtained

$$Q_i = B + \Phi_i, \frac{dQ_i}{dt} = \frac{d\Phi_i}{dt}, Q_j = B + \Phi_j. \quad (19)$$

Substituting above three equations into Eq.(4) leads to a new shunting equation

$$\begin{aligned} \frac{d\Phi_i}{dt} = & -A(B + \Phi_i) + (-\Phi_i)([\rho_i]^+ + \sum_{j=1}^M W_{ij}[B + \Phi_j]^+) \\ & - (D + B + \Phi_i)[\rho_i]^- . \end{aligned} \quad (20)$$

Eq.(21) is obtained by rearranging Eq.(20).

$$\begin{aligned} \frac{d\Phi_i}{dt} = & -\Phi_i \left\{ \frac{1}{\Phi_i} (AB + \Phi_i(A + [\rho_i]^+ + [\rho_i]^-) + \right. \\ & \left. (B + D)[\rho_i]^-) - \sum_{j=1}^M (-W_{ij})[B + \Phi_j]^+ \right\}. \end{aligned} \quad (21)$$

Then the proposed model in Eq.(4) can be written into the Grossberg's general form [39]

$$\frac{d\Phi_i}{dt} = a_i(\Phi_i) \left(b_i(\Phi_i) - \sum_{j=1}^M c_{ij} d_j(\Phi_j) \right) \quad (22)$$

where

$$a_i(\Phi_i) = -\Phi_i \quad (23)$$

$$b_i(\Phi_i) = \frac{1}{\Phi_i} (AB + \Phi_i(A + [\rho_i]^+ + [\rho_i]^-) + (B + D)[\rho_i]^-) \quad (24)$$

$$c_{ij} = -W_{ij} \quad (25)$$

$$d_j(\Phi_j) = [B + \Phi_j]^+. \quad (26)$$

According to Eq.(5), the weight of neural connection is symmetric, $W_{ij} = W_{ji}$, correspondingly, $c_{ij} = c_{ji}$. Since $Q_i \in [-D, B]$, then $\Phi_i \in [-B - D, 0]$, where B and D are non-negative constants, Φ_i is a non-positive number. Hence, the zoom function $a_i(\Phi_i)$ is non-negative, i.e., $a_i(\Phi_i) \geq 0$. From the definition of function $[B + \Phi_j]^+$, when $\Phi_j < -B$, $d'_j(\Phi_j) = 0$; when $\Phi_j > -B$, $d'_j(\Phi_j) = 1$. Therefore, the signal function $d_j(\Phi_j)$ has a non-negative derivation ($d'_j(\Phi_j) \geq 0$). According to the above derivation, Eq.(4) satisfies all the three stability conditions required by Grossberg's general form [39]. The following Lyapunov function for Eq.(4) is selected:

$$\Omega = - \sum_{i=1}^M \int^{\Phi_i} b_i(\tau_i) d'_i(\tau_i) d\tau_i + \frac{1}{2} \sum_{j,h=1}^M c_{jh} d_j(\Phi_j) d_h(\Phi_h). \quad (27)$$

The time derivative of Ω along all the trajectories is given by

$$\frac{d\Omega}{dt} = - \sum_{i=1}^M a_i d'_i \left(b_i - \sum_{j=1}^M c_{ij} d_j \right)^2. \quad (28)$$

Since both a_i and d'_i are non-negative, then $d\Omega/dt \leq 0$ along all the trajectories. Hence, the proposed GBNN system is stable. The dynamics of the network ensures that the system converges to an equilibrium state. In the proposed distributed collaborative decision-making method, the robot generally advances to the area of high activity value. Furthermore, due to the global propagation effect of positive activity values in the GBNN, the MRS is able to completely cover the task area when the GBNN converges to equilibrium.

V. VERIFICATION AND COMPARISON

To verify the effectiveness of the proposed algorithm, the MATLAB R2020 is selected as the simulation experiment platform. The computer's main frequency is 2.9GHz, and the memory is 16GB. The robot performance parameters: $m_d=3$, $n_d=3$, $L_c=8$, and $D_R=\sqrt{2}$. The GBNN model parameters: $A=0.2$, $B=0.6$, $D=0.6$, $I=3$, $r=\sqrt{2}$, $Q_0 = 0.4$, and $\alpha = 0.1$. The DMPC parameters: $L=3$, $\lambda_1=0.9$, and $\lambda_2 = 0.1$. The experiments are mainly conducted from three aspects: the dynamic search process of multiple robots is analyzed in subsection V-A; the influence of the number of robots on search performance is considered in subsection V-B; the influence of complex obstacle environments on search performance is discussed in subsection V-C. Initially, all robots have no prior environmental information, including the distribution of targets and obstacles in the task area. Assume that the targets are randomly distributed in the task area. The following three indexes are used to evaluate the performance of the proposed algorithm.

(1) The average coverage rate (COV). The coverage rate (cov) is defined in Eq.(29). In order to eliminate the randomness of a single experiment, multiple experiments (select different starting positions of the robots) can be carried out, and the COV is used as the performance index of the algorithm, as shown in Eq.(30).

$$\text{cov} = \frac{S_{\text{searched}}}{S_{\text{Area}}} \quad (29)$$

where S_{searched} is the size of the searched area; S_{Area} is the size of the task area.

$$\text{COV} = \frac{\sum_{i=1}^{N_e} \text{cov}(i)}{N_e} \quad (30)$$

where N_e is the number of experiments; $\text{cov}(i)$ is the coverage rate of the i -th experiment.

(2) The standard deviation of the coverage rate (STD). STD represents the fluctuation range of the coverage rate. The smaller the value of STD, the better the stability of the algorithm, its definition is shown in Eq.(31).

$$\text{STD} = \sqrt{\frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\text{cov}(i) - \text{COV})^2} \quad (31)$$

where N_e is the number of experiments.

(3) The average target search success rate (SUC). If the distance between the robot and the target is less than the threshold set by the user, the target is considered to be found and its influence disappears, as shown in Eq.(32) ($\delta=1$ in

this work). Similar to Eq.(29), the search success rate of the target (suc) in a single experiment is defined in Eq.(33), then the definition of SUC is shown in Eq.(34). Since the targets are assumed to be randomly distributed in the unknown environment, theoretically SUC is proportional to COV.

$$d(R_s, \text{target}) < \delta \quad (32)$$

where $d(R_s, \text{target})$ represents the Euclidean distance between R_s and the target, $s = 1, 2, \dots, N_r$; δ is the threshold set by users.

$$\text{suc} = \frac{O_{\text{searched}}}{O_{\text{All}}} \quad (33)$$

where O_{searched} is the number of the searched targets; O_{All} is the number of targets in the region.

$$\text{SUC} = \frac{\sum_{i=1}^{N_e} \text{suc}(i)}{N_e} \quad (34)$$

where N_e is the number of experiments; $\text{suc}(i)$ is the target's search success rate of the i -th experiment.

Furthermore, we chose three algorithms as baselines to verify the superiority of the proposed algorithm, namely GBNN algorithm [31], DMPC algorithm [11] and A-RPSO algorithm [20] respectively. In the GBNN algorithm, the robot will select the neuron (grid) with the largest activity value of its surrounding neurons as its next movement position. In the DMPC algorithm, each robot determines the next search path through rolling optimization decisions, and there is no decision-making information exchange among robots. As for the A-RPSO algorithm, the potential field function is taken as the fitness function to guide the robot to search the uncovered area.

A. Dynamic Search Process

This subsection mainly analyzes the dynamic search process of multiple robots through an example. Experimental parameters are as follows: the size of the task area is $20 * 20$ squares grid map; the number of the robots is 8 (R_1, R_2, \dots, R_8); the number of the targets is 10 (T_1, T_2, \dots, T_{10}); the initial positions of the robots are $R_1(3,4), R_2(8,2), R_3(11,7), R_5(4,5), R_6(19,14), R_7(10,19)$ and $R_8(3,15), R_8(15,12)$.

The trajectories of the robots at different movement steps are shown in Fig.7, where the robots can avoid obstacles in real time and continuously search the unexplored area, and the trajectories of the robots rarely repeat, which reflects the availability of the proposed algorithm. When the movement steps of the robots reach 160, all targets have been found and the task area is basically explored (cov:97.75%).

TABLE I
THE GROUPING INFORMATION

Group Steps	Communication sets	Search sub-teams
16	$\{R_1, R_2, R_3, R_4\}; \{R_5, R_6, R_7, R_8\}$	$\{R_1, R_2, R_3, R_4\}; \{R_5, R_8\}; \{R_6, R_7\}$
40	$\{R_1, R_3, R_4, R_5, R_6, R_7\}; \{R_2, R_8\}$	$\{R_1, R_3, R_4\}; \{R_5, R_6\}; \{R_7\}; \{R_2, R_8\}$
72	$\{R_1, R_2, R_3, R_6, R_8\}; \{R_4, R_7\}; \{R_5\}$	$\{R_1, R_3, R_6\}; \{R_2, R_8\}; \{R_4, R_7\}; \{R_5\}$
160	$\{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8\}$	$\{R_1, R_2, R_3, R_4, R_5, R_6, R_8\}; \{R_7\}$

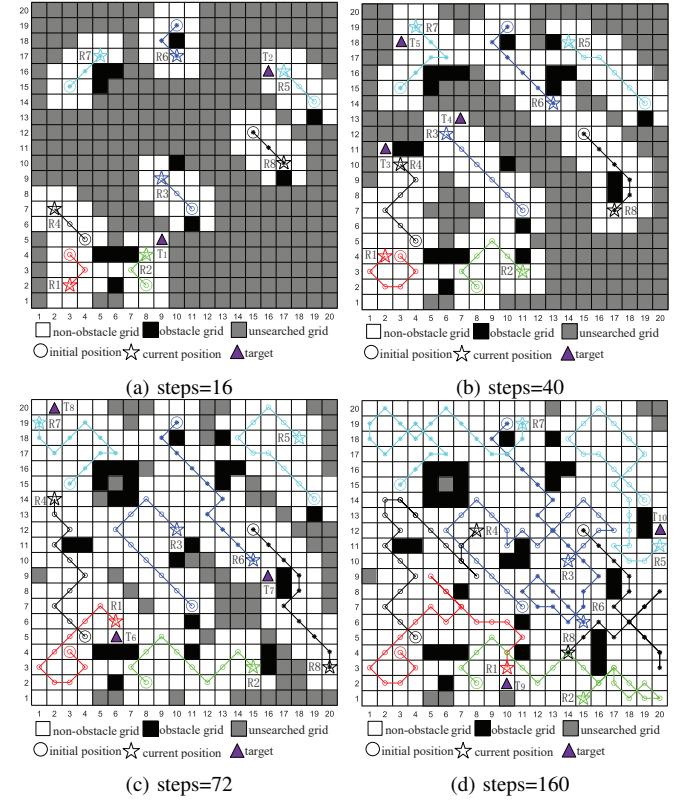


Fig. 7. The movement trajectory of MRS at different movement steps.

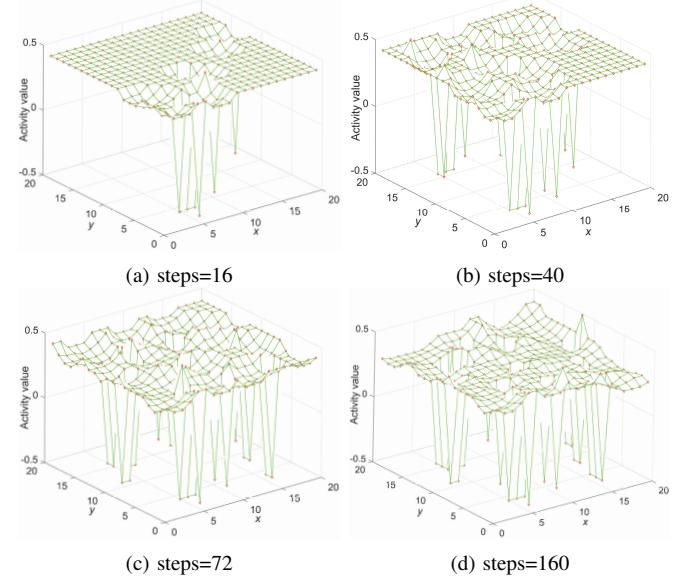
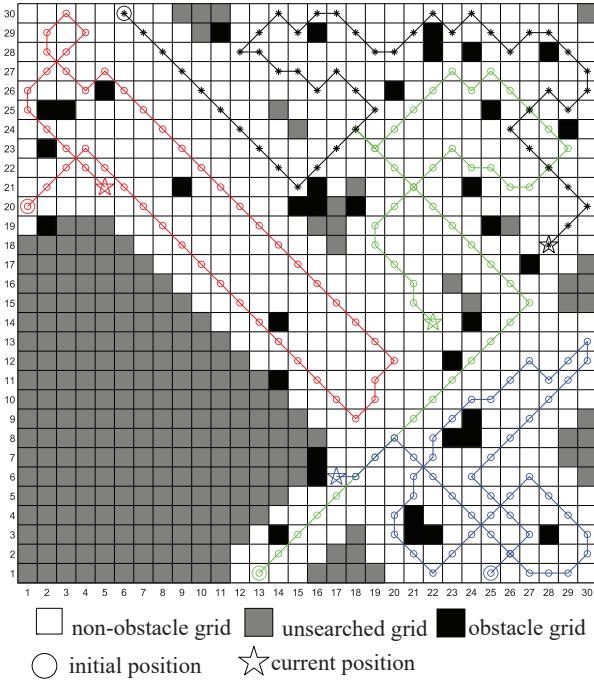


Fig. 8. The neuronal activity landscape of R_1 at different moving steps.

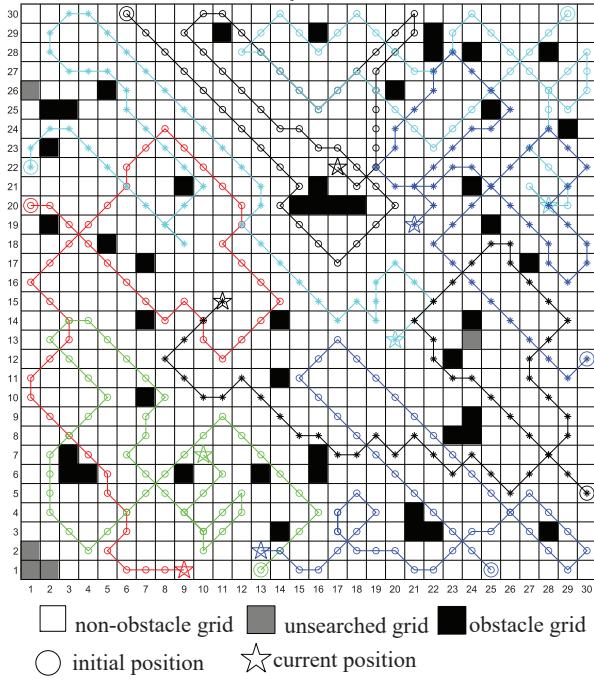
TABLE II
AVERAGE DECISION-MAKING TIME

Robots	4	8	12	16	20	24
$t_d(s)$	0.0149	0.0198	0.238	0.0254	0.0261	0.0287

The neuron activity value matrices of R_1 at different movement steps are shown in Fig.8. By comparing Figs.7 and 8, the



(a) the search trajectories (Robots:4)



(b) the search trajectories (Robots:8)

Fig. 9. The search trajectories of MRS under different numbers of robots.

activity value of the obstacle grid detected by the robot will be at a low point, and the activity value of the searched grid (non-obstacle) will decrease moderately, and the robots will continue to search towards the areas with high activity value. This enables the MRS to quickly search for the targets in an unknown environment. The grouping information of the robots at different movement steps is shown in Table I, including communication sets and search sub-teams information. Table I shows that the communication sets and search sub-teams change dynamically with the movement of the robots. Table

II shows the average decision-making time of robots (different scales) at each step. When the number of robots increases, the decision time of robots remains in a small range.

Hence, although the detection range and communication ability of a single robot are limited, as the robot moves, it can exchange environmental information and decision information with different robots, which improves the search efficiency of the MRS.

B. Influence of Different Number of Robots

This subsection mainly considers the influence of the number of robots on search performance, and the number of the robots are 4, 8, 12, 16, 20, 24, respectively. The search trajectories (proposed algorithm) of the MRS under different numbers of robots are shown in Fig.9. When the number of robots increases from 4 to 8, the area has been basically searched, and the trajectory between robots is repeated less, indicating the effectiveness of the algorithm. To eliminate the randomness of a single experiment, there are 200 times Monte Carlo experiments for each group robots by randomly selecting different initial positions of the robots. The size of the task area is 30 * 30 grid map and the total number of robots' movement steps is 480. Besides, the GBNN algorithm [31], the DMPC algorithm [11] and the A-RPSO algorithm [20] are selected as the comparison group.

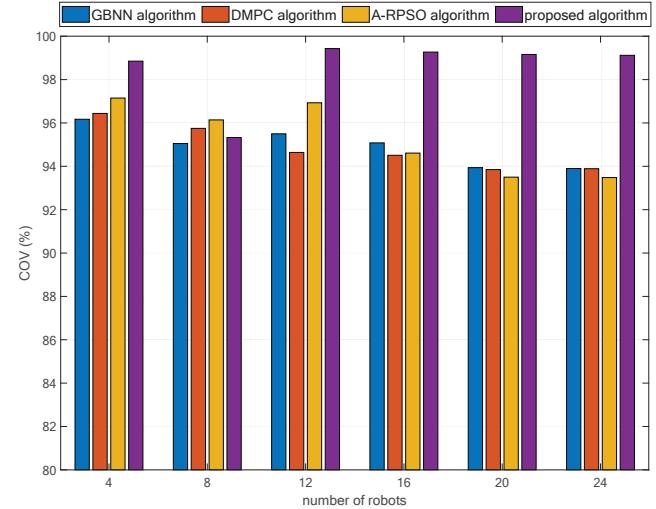
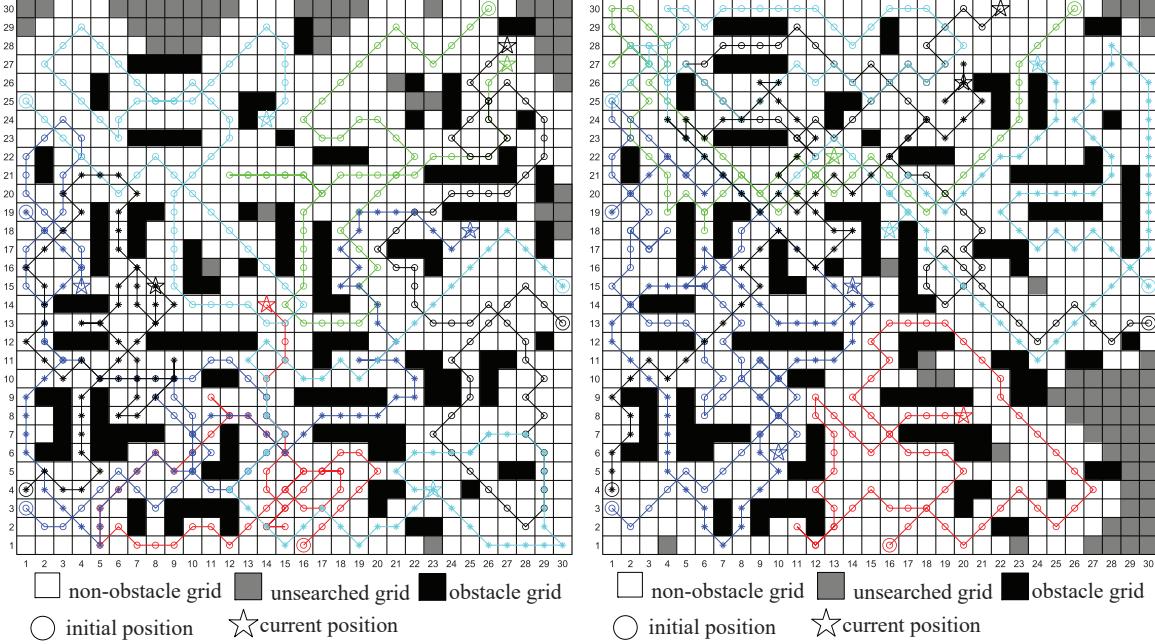


Fig. 10. The comparison of “COV” at different numbers of robots

The performance comparison of four algorithms (different number of robots) are shown in Table III and the comparison of the “COV” of four algorithms is visually shown in Fig.10. Compared to the other three algorithms, with the increase of the robot's number, the advantages of the proposed algorithm become more obvious. When the number of robots is 24, the “COV” of the four algorithms are 93.90% (GBNN algorithm), 93.86% (DMPC algorithm), 93.48% (A-RPSO algorithm), and 99.12% (proposed algorithm) respectively; the “SUC” of the three algorithms are 94.21% (GBNN algorithm), 94.43% (DMPC algorithm), 94.26% (A-RPSO algorithm), and 99.50% (proposed algorithm) respectively; and the “STD” of the proposed algorithm is also the lowest. The above simulation

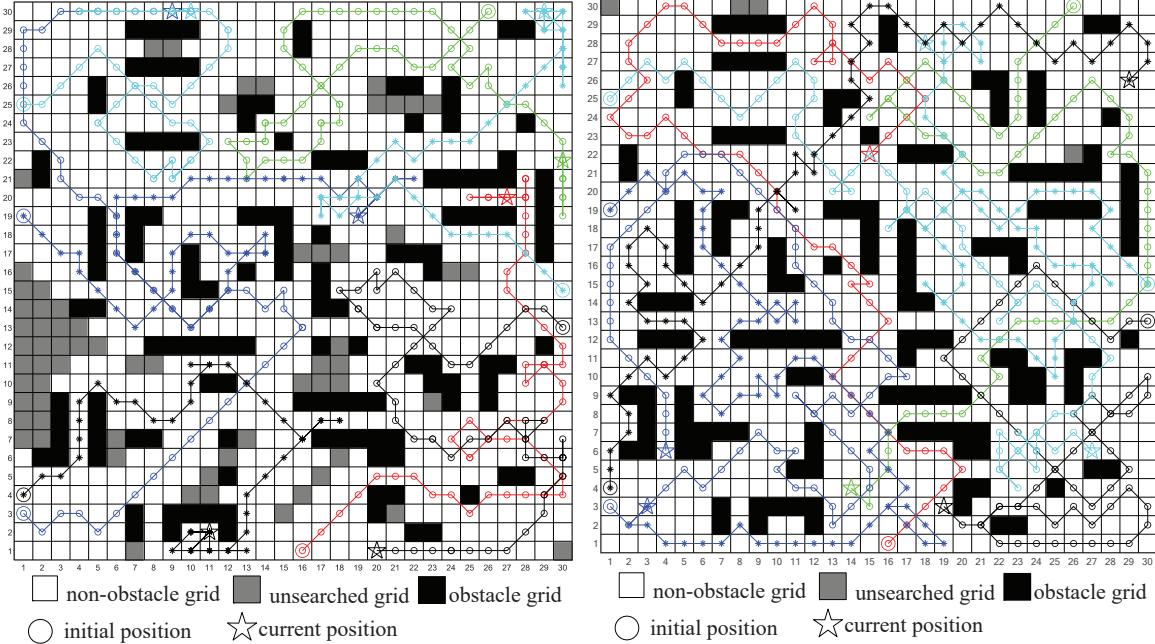
TABLE III
PERFORMANCE COMPARISON OF FOUR ALGORITHMS (DIFFERENT NUMBERS OF ROBOTS)

Robots	Algorithm	GBNN algorithm [31]			DMPC algorithm [11]			A-RPSO algorithm [20]			proposed algorithm		
		COV	STD	SUC	COV	STD	SUC	COV	STD	SUC	COV	STD	SUC
4		96.17%	0.0303	96.53%	96.44%	0.0253	97.46%	97.15%	0.0298	97.38%	98.85%	0.0103	99.16%
8		95.05%	0.0297	95.95%	95.75%	0.0261	96.63%	96.14%	0.0298	96.71%	99.22%	0.0075	99.61%
12		95.33%	0.0308	95.62%	95.50%	0.0230	96.49%	94.64%	0.0298	96.93%	99.43%	0.0045	99.74%
16		95.08%	0.0302	95.23%	94.51%	0.0240	95.34%	94.61%	0.0298	95.46%	99.27%	0.0062	99.68%
20		93.94%	0.0312	94.02%	93.85%	0.0283	94.38%	93.50%	0.0298	94.34%	99.16%	0.0069	99.53%
24		93.90%	0.0305	94.21%	93.86%	0.0214	94.43%	93.48%	0.0298	94.26%	99.12%	0.0076	99.50%



(a) GBNN algorithm [31]

(b) DMPC algorithm [11]



(c) A-RPSO algorithm [20]

(d) proposed algorithm

Fig. 11. The movement trajectories of four algorithms under complex obstacle environments.

TABLE IV
PERFORMANCE COMPARISON OF FOUR ALGORITHMS (DIFFERENT NUMBERS OF OBSTACLES)

Obstacles	Algorithm	GBNN algorithm [31]			DMPC algorithm [11]			A-RPSO algorithm [20]			proposed algorithm		
		COV	STD	SUC	COV	STD	SUC	COV	STD	SUC	COV	STD	SUC
23		95.31%	0.0307	95.64%	97.73%	0.0182	98.46%	96.41%	0.0351	96.15%	99.42%	0.0057	99.69%
45		95.84%	0.0320	95.94%	96.04%	0.0229	96.63%	96.28%	0.0243	95.86%	99.21%	0.0074	99.54%
68		96.09%	0.0295	96.32%	95.82%	0.0197	96.49%	95.66%	0.0271	95.26%	99.27%	0.0060	99.65%
90		95.68%	0.0294	95.84%	93.41%	0.0306	94.25%	95.26%	0.0388	95.74%	99.23%	0.0076	99.52%
154		95.49%	0.0297	95.63%	90.52%	0.0362	91.48%	92.46%	0.0323	92.26%	98.40%	0.0130	99.27%

results show that the distributed collaborative decision-making mechanism of the proposed algorithm can effectively improve the cooperative performance of the MRS in area coverage search.

C. Influence of Complex Obstacle Environments

This subsection mainly considers the influence of complex obstacle environments on search performance. The size of the task area is 30 * 30 grid map, the total number of robots' movement steps is 480, and the number of the obstacle grids is 23, 45, 68, 90, 154, respectively. Similarly, the GBNN algorithm [31], the DMPC algorithm [11] and the A-RPSO algorithm [20] are selected as the comparison group. The movement trajectories of four algorithms under complex obstacle environments (obstacle grids:154) are shown in Fig.11. As can be seen from Fig.11, the three comparison algorithms are difficult to adapt to the complex obstacle environment with narrow passages, and the "cov" is 94.89% (GBNN algorithm), 94.56% (DMPC algorithm), and 93.89% (A-RPSO algorithm), respectively. However, under the same number of motion steps, the proposed algorithm has already guided the robot to basically search the area (cov:99.56%). Compared with the other three algorithms, the proposed algorithm is more suitable for complex environments with narrow obstacle passages.

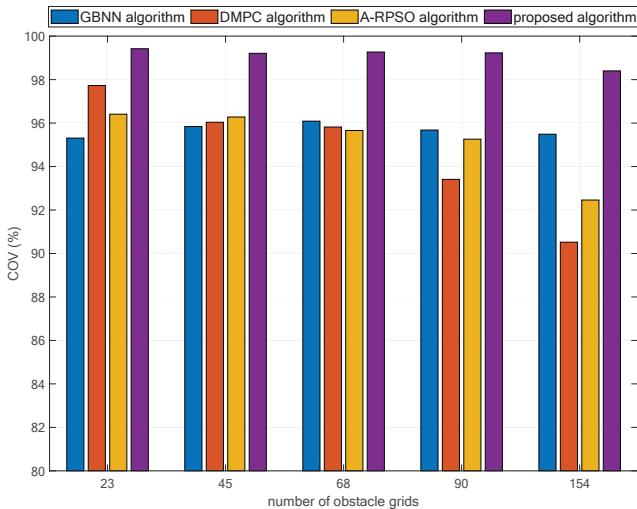


Fig. 12. The comparison of “COV” at different numbers of obstacle grids

To eliminate the randomness of a single experiment, there are also 200 times Monte Carlo experiments for each obstacle

environment by randomly selecting different initial positions of the robots. The performance comparison of four algorithms (different number of obstacle grids) are shown in Table IV and the comparison of the “COV” of four algorithms is visually shown in Fig.12. As the number of obstacle grids in the environment increases, the search performance of the DMPC algorithm and the A-RPSO algorithm degrades substantially. Although the performance of the GBNN algorithm is relatively stable, the proposed algorithm is always optimal and stable among the four algorithms.

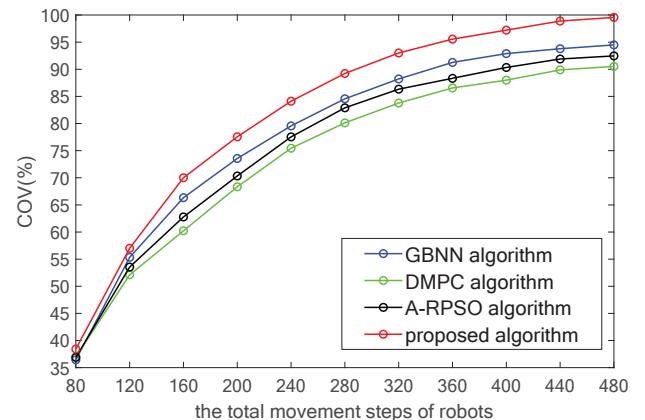


Fig. 13. The COV curves of different algorithms during the search process under the complex obstacle environment

Figure 13 shows the COV curves of different algorithms as the MRS moves in the complex obstacle environment (number of obstacle grids:154). In the early stages of MRS coverage search, the performance of the four algorithms is similar. As the search time increases, the advantages of the proposed algorithm become more obvious compared with the three baseline algorithms. The above results show that the search objective function (Eq.(12)) designed in the proposed algorithm can effectively guide the robot to adapt to the obstacle environment of different complexity.

VI. CONCLUSION

This paper has proposed a novel multi-robot distributed collaborative region coverage search algorithm based on the GBNN. The MRS converts the detected environmental information into the dynamic activity value landscape through the GBNN model, and uses the distributed collaborative decision-making method to plan the next search path in real time to

efficiently complete the area coverage search in the unknown environments. Simulation experiments and comparison studies have shown that the proposed algorithm further improves the cooperation performance of the MRS and enables the robot to adapt to complex obstacle environments. Furthermore, it is still an open issue for minimize the energy consumption of the MRS under the goal of ensuring the shortest area coverage search time in the unknown environments. This will be the topic of our future research.

REFERENCES

- [1] I. Mahmud and Y. Z. Cho, "Detection avoidance and priority-aware target tracking for UAV group reconnaissance operations," *Journal of Intelligent and Robotic Systems*, vol. 92, no. 3, pp. 1–12, 2017.
- [2] S. Chien and K. L. Wagstaff, "Robotic space exploration agents," *Science Robotics*, vol. 2, no. 7, pp. 1–2, 2017.
- [3] 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 Transactions on Cognitive and Developmental Systems*, vol. 11, no. 1, pp. 73–84, 2018.
- [4] M. Atif, R. Ahmad, W. Ahmad, L. Zhao, and J. J. P. C. Rodrigues, "UAV-assisted wireless localization for search and rescue," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2021.
- [5] N. Kumar, S. Misra, and M. S. Obaidat, "Collaborative learning automata-based routing for rescue operations in dense urban regions using vehicular sensor networks," *IEEE Systems Journal*, vol. 9, no. 3, pp. 1081–1090, 2015.
- [6] A. V. Le, P. T. Kyaw, R. E. Mohan, S. H. M. Swe, A. Rajendran, K. Boopathi, and N. H. K. Nhan, "Autonomous floor and staircase cleaning framework by reconfigurable stero robot with perception sensors," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–19, 2021.
- [7] X. Cao, H. Sun, and G. E. Jan, "Multi-AUV cooperative target search and tracking in unknown underwater environment," *Ocean Engineering*, vol. 150, pp. 1–11, 2018.
- [8] M. Tsuru, A. Escande, A. Tanguy, K. Chappellet, and K. Harad, "Online object searching by a humanoid robot in an unknown environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2862–2869, 2021.
- [9] L. Chen, M. Wu, M. Zhou, J. She, F. Dong, and K. Hirota, "Information-driven multirobot behavior adaptation to emotional intention in human–robot interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 647–658, 2017.
- [10] Liang, Ren, Yingying, Yu, Min, Tan, Zhiqiang, Cao, Zhiyong, and Wu, "An optimal task allocation approach for large-scale multiple robotic systems with hierarchical framework and resource constraints," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3877–3880, 2018.
- [11] F. Zhang, B. Chen, and X. Ban, "Multi-robot cooperative search algorithm based on bio-inspired neural network and DMPC," *Control and Decision*, vol. 36, no. 11, pp. 2699–2706, 2021.
- [12] Q. Jia, H. Xu, X. Feng, H. Gu, and L. Gao, "Research on cooperative area search of multiple underwater robots based on the prediction of initial target information," *Ocean Engineering*, vol. 172, pp. 660–670, 2019.
- [13] J. Hu, L. Xie, J. Xu, and Z. Xu, "Multi-Agent cooperative target search," *Sensors*, vol. 14, no. 6, pp. 9408–9428, 2014.
- [14] B. Yang, Y. Ding, Y. Jin, and K. Hao, "Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis," *Robotics and Autonomous Systems*, vol. 72, pp. 83–92, 2015.
- [15] K. Vinh, S. Gebreyohannes, and A. Karimoddini, "An area-decomposition based approach for cooperative tasking and coordination of uavs in a search and coverage mission," in *2019 IEEE Aerospace Conference*. IEEE, 2019, pp. 1–8.
- [16] L. D. Nielsen, I. Sung, and P. Nielsen, "Convex decomposition for a coverage path planning for autonomous vehicles: Interior extension of edges," *Sensors*, vol. 19, no. 19, p. 4165, 2019.
- [17] J. Li and Y. Tan, "A two-stage imitation learning framework for the multi-target search problem in swarm robotics," *Neurocomputing*, vol. 334, pp. 249–264, 2019.
- [18] B. Liu, X. Wang, and W. Zhou, "Multi-UAV collaborative search and strike based on reinforcement learning," in *Journal of Physics: Conference Series*, vol. 1651, 2020, p. 012115.
- [19] Y. Cai and S. X. Yang, "An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments," *International Journal of Control*, vol. 86, no. 10, pp. 1720–1732, 2013.
- [20] M. Dadgar, S. Jafari, and A. Hamzeh, "A PSO-based multi-robot cooperation method for target searching in unknown environments," *Neurocomputing*, vol. 177, pp. 62–74, 2016.
- [21] H. Tang, W. Sun, H. Yu, A. Lin, and M. Xue, "A multirobot target searching method based on bat algorithm in unknown environments," *Expert Systems with Applications*, vol. 141, p. 112945, 2020.
- [22] Z. Zhou, D. Luo, J. Shao, Y. Xu, and Y. You, "Immune genetic algorithm based multi-UAV cooperative target search with event-triggered mechanism," *Physical Communication*, vol. 41, p. 101103, 2020.
- [23] K. Hou, Y. Yang, X. Yang, and J. Lai, "Distributed cooperative search algorithm with task assignment and receding horizon predictive control for multiple unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 6122–6136, 2021.
- [24] W. Dai, H. Lu, J. Xiao, Z. Zeng, and Z. Zheng, "Multi-robot dynamic task allocation for exploration and destruction," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 455–479, 2020.
- [25] Q. Tang, L. Ding, F. Yu, Y. Zhang, Y. Li, and H. Tu, "Swarm robots search for multiple targets based on an improved grouping strategy," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 6, pp. 1943–1950, 2017.
- [26] R. M. de Mendonça, N. Nedjah, and L. de Macedo Mourelle, "Efficient distributed algorithm of dynamic task assignment for swarm robotics," *Neurocomputing*, vol. 172, pp. 345–355, 2016.
- [27] A. L. Hodgkin, "A quantitative description of membrane current and its application to conduction and excitation in a nerve," *Journal of Physiology (London)*, vol. 117, 1952.
- [28] S. Grossberg, "Contour enhancement, short term memory, and constancies in reverberating neural networks," *Studies in Applied Mathematics*, vol. 52, no. 3, 1973.
- [29] C. Luo and S. X. Yang, "A real-time cooperative sweeping strategy for multiple cleaning robots," in *Proceedings of the IEEE International Symposium on Intelligent Control*. IEEE, 2002, pp. 660–665.
- [30] C. Luo, S. X. Yang, and D. A. Stacey, "Real-time path planning with deadlock avoidance of multiple cleaning robots," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 4080–4085.
- [31] C. Luo, S. X. Yang, X. Li, and Q. H. Meng, "Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots," *IEEE Transactions on Industrial Electronics*, pp. 750–760, 2017.
- [32] X. Cao and C. Sun, "Cooperative target search of multi-robot in grid map," *Control Theory and Applications*, vol. 35, no. 3, pp. 273–282, 2018.
- [33] X. Cao, D. Zhu, and S. X. Yang, "Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2364–2374, 2016.
- [34] Z. Huang, D. Zhu, and B. Sun, "A multi-AUV cooperative hunting method in 3-D underwater environment with obstacle," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 192–200, 2016.
- [35] W. Mauve, Martin, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.
- [36] R. Halvgaard, "Distributed model predictive control for smart energy systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1675–1682, 2017.
- [37] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2010.
- [38] S. X. Yang and M.-H. Meng, "Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1541–1552, 2003.
- [39] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks*, vol. 1, no. 1, pp. 17–61, 1988.



Bo Chen received the B.S. degree in automation from the East China University of Technology, Nanchang, China, in 2019. He is currently working toward the M.S. degree with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China.

His research interests include multirobot system, area coverage search, and path planning.



Hongnian Yu received the B.Eng. degree in electrical and electronic engineering from the Harbin Institute of Technology, Harbin, China, the M.Sc. degree in control engineering from Northeast Heavy Machinery Institute, Heilongjiang, China, and the Ph.D. degree in robotics from Kings College London, London, U.K.

He is a Professor and Head of Research with the School of Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, U.K. He has held academic positions with University of Sussex, Brighton, U.K.; Liverpool John Moores University, Liverpool, U.K.; University of Exeter, Exeter, U.K.; University of Bradford, Bradford, U.K.; Staffordshire University, Stoke-on-Trent, U.K.; and Bournemouth University, Poole, U.K. He has extensive research experience in mobile computing, computer networks, control of robots, and neural networks and computing. He has authored or coauthored more than 200 journal and conference research papers.



Hui Zhang received the B.S., M.S., and Ph.D. Degrees in pattern recognition and intelligent system from Hunan University, Changsha, China, in 2004, 2007, and 2012, respectively.

He is currently a professor with the School of Robotics, Hunan University, and He is the Deputy Director of the National Engineering Laboratory for Robotic Vision Perception and Control Technology. His research interests include multi-robot cooperative control, machine vision, image processing, robot vision.



Fangfang Zhang received the B.E. and M.E. degrees in applied mathematics from Shandong University, Jinan, in 2008 and 2011, respectively. He received the Ph.D. degree in control science and engineering from Shandong University, in 2015.

He is currently associate professor at Zhengzhou University, Zhengzhou, Henan, China. His research interests include optimal control of multi-agent systems, multi-robot formation, machine vision.



Yaonan Wang received the B.S. degree in computer engineering from East China Science and Technology University (ECSTU), Fuzhou, China, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively.

From 1994 to 1995, he was a Post-Doctoral Research Fellow with the National University of Defence Technology, Changsha. From 1981 to 1994, he was with ECSTU. From 1998 to 2000, he was a Senior Humboldt Fellow in Germany. From 2001 to 2004, he was a Visiting Professor with the University of Bremen, Bremen, Germany. He has been a Professor with Hunan University since 1995. He is an Academician of the Chinese Academy of Engineering. His current research interests include robotics, intelligent perception and control, and computer vision for industrial applications.



Yanhong Liu received the B.E. degree from the Zhengzhou University of Light Industry, Zhengzhou, China, in 1992, and the M.E. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2002 and 2006, respectively.

From 2012 to 2013, she was a Visiting Scholar with the University of California at San Diego, La Jolla, CA, USA. She is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou. Her research interests include nonlinear system modeling and control, robotic control, and human-robot interactions and collaborations.



Cheng Tan received the PhD degree in control science and engineering from Shandong University, in 2016.

He is currently associate professor at Qufu Normal University. His research interests include networked system control, machine learning and optimal control. He is a Member of IEEE, a Member of Chinese Association of Automation, and a Member of Youth Committee of Chinese Association of Automation.