A Virtual Force Interaction Scheme for Multi-robot Environment Monitoring

Kang Ji [a], Qian Zhang [a, 1], Zhi Yuan [b], Hui Cheng [c], and Dingli Yu [a]

[a] *The Department of Electronics and Electrical Engineering, Liverpool John Moores University, Liverpool L3 3AF, U.K.*
[b] *The School of Navigation, Wuhan University of Technology, Wuhan, Hubei 430063, China*
[c] *The Department of Computer Science, University of Hertfordshire, Hatfield, Hertfordshire AL10 9AB, U.K.*

Abstract

The autonomous multi-robot system is an emerging technology that has a wide range of potential applications, such as environmental monitoring, exploration of unknown area, battlefield surveillance, and search and rescue. One major challenge in such applications is how to deploy each robotic agent autonomously in a distributed manner. In this paper, we proposed a distributed coverage control strategy named multi-stage virtual force interaction scheme (VFIS), where the agents' deployment process is split into stages and each agent iteratively seeks its next position according to the interaction among agents and the interaction between agents and the perceived environment. The interactions are realized via virtual repulsive forces and virtual vortex forces, where the latter are newly proposed to enhance the exploration capability of agents. We also designed a group of benchmark testing problems for the mission of monitoring coverage of complex environments with unknown obstacles. Extensive simulation experiments were conducted based on the defined benchmark configurations and the results showed a favourable performance of the invented strategy. In addition, practical experiments were carried out using a group of mobile robots, which validated the effectiveness of the proposed method.

Keywords: Monitoring coverage; multi-robot system; virtual force; benchmark

1. Introduction

Research of autonomous multi-robot systems has gained growing attention [1-7], as such systems have great potential to be used in a wide range of applications, such as exploration of unknown area or space [1, 2], battlefield surveillance [3], environmental monitoring [4], and search and rescue [7]. The path planning problem and coverage problem are of particular interests in the above applications. The robot path planning is a process to reach the destination from a predefined initial position safely through a sequence of rotation and translation. The coverage problem generally can be divide into the static coverage, dynamic coverage, persistent coverage [8]. The static coverage seeks the optimal deployment of a group of agents to cover the environment statically [9]. The dynamic coverage aims that agents visit all the points of the environment at least once or reach the desired coverage level [10]. The persistent coverage requires agents to continually move and visit every place in the environment

1 *Corresponding author.*
E-mail addresses: K.ji@2017.ljmu.ac.uk (K. Ji), Q.Zhang@ljmu.ac.uk (Q. Zhang), D.Yu@ljmu.ac.uk (D. Yu), wisdomyz@whut.edu.cn (Z. Yuan), h.cheng2@herts.ac.uk (H. Cheng)

periodically [11]. This paper tackles a multi-robot monitoring coverage problem, which aims to deploy a swarm of robotic agents in a complex environment with obstacles, to achieve the 'maximum' monitoring coverage of the area of interest (AOI) and is classified as a static coverage problem. The purpose of monitoring may vary from observation of movements or behaviors of interested objects to detection of hazardous substances or fire hazards. Each agent possesses capabilities of perception, monitoring, communication, localization and movement. Besides the area coverage, the travel length and the distribution evenness of agents are also considered as important evaluation metrics of the problem.

The monitoring coverage task can be fulfilled by either centralized approaches or distributed approaches. The centralized approaches require a central unit to gather and integrate environmental information, plan trajectories and distribution of agents using all available information and deliver the computed results to agents. In distributed methods, each agent utilizes its own perception, active communication and control laws to achieve the collective goals. When the number of agents is large, the distributed strategy with the local and scalable features is proven to be more efficient and robust.

The virtual force algorithm [12] provides an effective solution for the deployment of the wireless sensor network in the convex environment and simple non-convex environment. In the real-world scenario, obstacles often exist in the target environment, which are not of interest to agents and. can block agents' movement. The current virtual force related methods cannot tackle the complex non-convex environment well due to agents would be stuck in the dense obstacle area. In this paper, we propose a coverage control strategy named multi-stage virtual force interaction scheme (VFIS), which is a distributed approach that can work properly in complex environments with a large number of obstacles. In VFIS, we introduce the vortex force, a novel genre of virtual force to cooperate with the repulsive force. The new force provides strong exploration capability to agents for overcoming the influence of obstacles and ensuring a balance between exploration and exploitation. In VFIS, the deployment process is split into a number of stages. Each agent iteratively seeks its new position according to the interaction among agents and the interaction between agents and the perceived environment via virtual forces in one stage, and finds the optimal distribution after a number of stages.

In this paper, we also design a group of benchmark testing problems for the monitoring coverage task. They comprehensively consider different characteristics of the environment and agents, including the border shape of the AOI, the number, shape and distribution of obstacles, and the number, starting positions, perception range and monitoring range of agents. Besides the above, the interaction mechanism and the parameter settings are also carefully studied. Moreover, we validate the proposed method through practical experiments using nine mobile robots AlphaBots. This paper is the substantial extension of the work in [13], where the monitoring coverage problem is redefined, the virtual force interaction method is greatly modified and all the works about benchmark problems design, parameters study, comparisons with other literature and robots verification are new.

The remaining part of this paper is organized as follows. In Section 2, we present and discuss the related literature. Section 3 explains various virtual forces proposed and illustrates the execution

process of VFIS. Section IV shows the benchmark problem settings. Section 5 demonstrates the experimental performance of VFIS in solving benchmark problems in the MATLAB simulation and Section 6 discusses some essential parameter settings. Section 7 presents the results of practical experiments using mobile robots. Finally, the conclusion is given in Section 8.

2. Related work

Some research has addressed the tasks similar to the monitoring coverage problem, which relate to wireless sensor networks and the deployment of unmanned aerial vehicles (UAVs) for surveillance. However, these research either considered the problems with none or very few obstacles, or employed centralized methods that cannot efficiently solve the problems when using 'many' agents. The relevant research about the movement-assisted sensors deployment and the coverage problem in UAV networks were surveyed in [14, 15]. The representative approaches to solving multi-agent coverage problems can be divided into the virtual force based approaches, the computational geometry based approaches, the bio-inspired optimization approaches and other approaches.

Zou et al. [12] proposed the idea of virtual force for enhancing the coverage of the wireless sensor nodes, after their initial random deployment. In the designed virtual force algorithm (VFA), the virtual attractive force and repulsive force are exerted on each node by its neighbours, and the virtual movement is computed by a cluster head. Wang et al. [16] combined VFA with particle swarm optimization (PSO) in the deployment of wireless sensor networks. The virtual force of sensor nodes was used for the velocity update of each particle in the iterative optimization process and helped to achieve better regional convergence and global search capability. Delaunay triangulation was introduced to define the adjacent relationship of each agent in [17]. Virtual forces were only exerted from the adjacent agents in the communication range and the convergence time of the deployment was improved to be shorter. Different ratios of communication range to sensing range were also considered and the virtual force function was modified to be suitable in both the high communication range and low communication range conditions. In [18], the factors about the virtual force setting were investigated and a virtual force strategy with energy awareness was proposed for balancing agents' energy consumption. Boufares et al. [19] utilized VFA to solve a three-dimensional area coverage problem, but agents' travel distance requires further improvement due to the large oscillation of agents. In [20], the authors defined the repulsive force exerted upon agents by obstacles for avoiding the collision with obstacles in the environment, however, only one obstacle with different shapes was used to test the algorithm in the simulation. Xie et al. [21] improved VFA by utilizing the area intensity information to select the distance threshold parameters related to the virtual force. A strategy based on the virtual spring force with suitable damping was proposed for decreasing the convergence time in the wireless sensor nodes deployment in [22]. They tested the damping effects and found the proper damping conditions for good distribution performance. In the aforementioned research, researchers developed their methods in the configured environment with no obstacles or merely few simple obstacles, and supposed the agents were able to be randomly placed in the target area at the beginning.

However, such random location initialization and simple environmental configuration could be unavailable in many real-life application scenarios.

The computational geometry-based approaches were mainly developed from the Voronoi partition. Cortes et al. [23] proposed a Voronoi-based coverage control approach for the deployment of wireless sensor nodes. In this method, each agent is directed to the centroids of Voronoi tessellations through the Lloyd descent of the utility function and the agents finally achieve a steady-state coverage distribution in convex environments. Wang et al. [24] proposed three approaches, VEC, VOR, and Minimax, to calculate the target positions for eliminating the coverage holes using the Voronoi diagram. For improving the unsatisfactory performance of VEC considering a large number of agents and the sensitive performance of VOR and Minimax with different initial locations of gents, new edge-based and vertex-based movement methods were introduced in [25]. The multiplicatively weighted Voronoi diagram was then introduced by Sharifi et al. [26] to partition the environment considering the heterogeneity of agents such as their sensing capabilities and dynamics. In [27], a team-based coverage scheme was designed to conduct the optimization process at the team level for allocating each team a sub-region and dividing the assigned region to a team member based on the given environment density function. In [28], the Voronoi-based control strategy was extended to be applied in non-convex environments through a combination of the Voronoi coverage method with a local path planning algorithm TangentBug. The path planning algorithm is used for planning the motions when an agent is close to obstacles and corners. The authors of [29] also compared the influence of the agent's geodesic or Euclidean sensing patterns using the geodesic or Euclidean Voronoi partitioning in non-convex environments. Stergiopoulos et al. [30] addressed the coverage of a moving deformable convex region and introduced a feedforward action to speed up the Centroidal Voronoi Tessellations process. Teruel et al. [31] proposed a distributed control law for the collaborative visual area coverage of the mobile aerial agents with downwards facing camera. For maximizing the defined coverage-quality criterion, a pattern based partitioning scheme and the gradient based control law arei employed to adjust agents' spatial coordinates. All the research above only concerned the obstacle-free environments or environments with a limited number of obstacles. A high computational cost is also required in complex large-scale environments in the implementation.

For a cooperative surveillance mission, finding a proper spatial distribution of agents in a pre-selected AOI can be formulated into a single high-dimensional optimization problem. The problem was solved by a PSO algorithm in [32]. The motion from initial locations to the optimal surveillance locations is controlled by a local controller based on the onboard sensor information. An improved PSO algorithm nPSO was applied in the self-deployment of a large number of agents in a much more complex environment [33]. An approach named stigmergic coverage (StiCo) for multi-robot coverage was proposed in [34], which was designed using an indirect pheromone-based communication strategy. In a basic StiCo, each agent is assumed it can deposit and detect pheromone information and has an initial circling motion. The circling direction and angular velocity would be changed with the detected pheromone density. In [35], an artificial bee colony algorithm was utilized in the wireless sensor nodes deployment. Considering multiple objectives such as coverage, redundancy, and fault-tolerance in

UAVs optimal deployment, a multi-layout multi-subpopulation genetic algorithm (GA) method was developed [36]. The above approaches employing nature-inspired optimization and strategies usually require the environmental information and centralized computation. When the number of agents or obstacles increases, the dimension of the optimization problem will greatly increase and the efficiency of the optimization process may significantly decrease.

A fuzzy-logic-based method was also proposed for determining the movement of mobile robots in the field of random deployment of static sensor nodes in [37]. Raili et al. [38] introduced the game theory for relocating the mobile agents' initial random positions to the optimal coverage configuration. They defined the utility function based on the coverage and energy consumption, and employed a maximum likelihood scheme to estimate the Gaussian mixture model of the sensing area. Renzaglia et al. [39] presented a cognitive-based adaptive optimization (CAO) method to address the multi-robot surveillance problem in a 3D non-convex target area with unknown obstacles. Through the introduction of a learning strategy, the method conducted a coordinated and scalable control for maximizing the monitored area. However, this method has the probability to fail to cover the whole environment due to its local convergence tendency. This method also relies on a centralized computational unit to carry out information fusion and optimization, which limits its applications in some complex and/or extreme situations.

Our first contribution is to consider sufficient conditions in the monitoring coverage problem and design a group of benchmark testing problems and our second contribution is that the distributed/decentralized property of our proposed VFIS would be more flexible and robust compared with the centralized method in the literature. Finally, since the previous solution may cause agents stuck in the dense obstacle area of the complex non-convex environment which lead to the poor coverage performance, the novel vortex force in VFIS could cooperate with the repulsive force and provides strong exploration capability to agents for overcoming the influence of obstacles and ensuring a balance between exploration and exploitation.

3. Methodology

3.1 Problem description

The problem investigated in this paper is monitoring coverage of an unknown area including obstacles using a group of robotic agents. The target area (or called target environment) is initially unknown to the agents, which means that the agents have no prior knowledge about the size and position of obstacles. The area of interest (AOI) of this problem is the obstacle-free region in the environment. The agents travel from their start positions and aim to find appropriate end positions, at which the agents can conduct the best monitoring with respect to the coverage of AOI. The start positions can be located at the environment border or anywhere within the target environment. Each agent is assumed to have a perception range R_p for environment observation, and a communication

rangeR_c, within which the agent can exchange its position information with adjacent agents. Each agent also has a monitoring range R_m, which relates to the assigned monitoring task and depends on the specification of monitoring sensors, such as optical cameras, thermographic sensors, chemical sensors and radars. If a sensor is used for both accomplishing the monitoring task and detecting the environment, R_m and R_p may be the same.

Fig. 1 shows an example of a square target area, where black geometrical objects representing obstacles and red star points indicating agents. The green, blue and red dashed lines represent an agent's monitoring range, perception range and communication range, respectively. The blue and red circles indicate the perception zone border and the communicative zone border individually, and the communicative zone does not have to be greater than the perception zone. It can be seen that the 6th agent can observe the partial border of a rectangular obstacle and communicate with the 5th agent. The light cyan disk zone represents the monitoring area of each agent and the yellow zone represents the unmonitored target area blocked by obstacles.

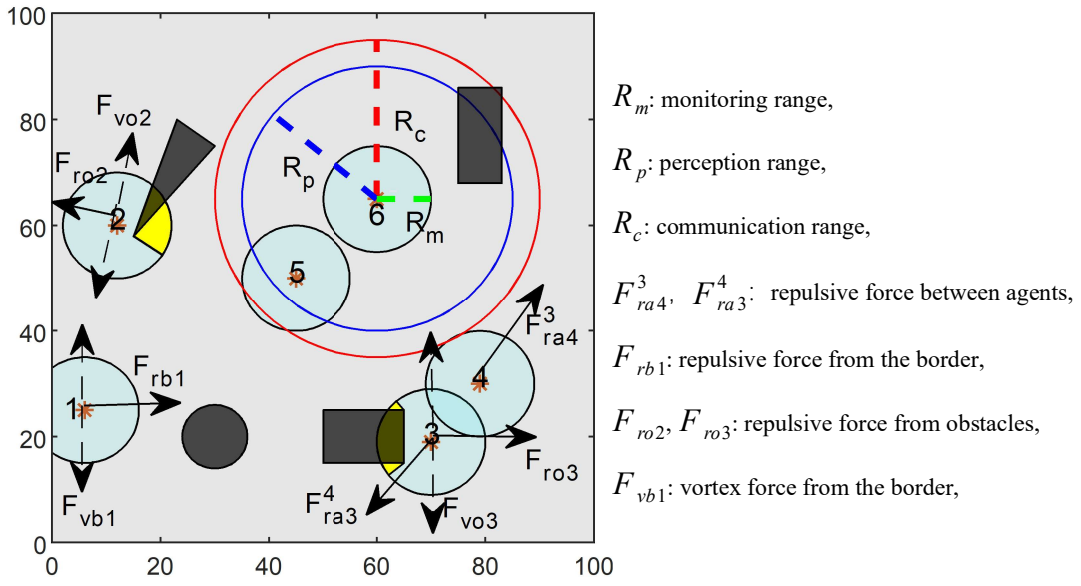


Fig. 1. An example of problem configuration and virtual forces.

Generally, the coverage performance may deteriorate when the agents cannot spread themselves to explore the whole environment. The performance may also deteriorate due to the lack of the position optimization mechanisms for 1) decreasing the overlapped monitoring zone between agents, 2) decreasing the unused monitoring zone outside the target area and 3) decreasing the occlusion area blocked by obstacles. We therefore propose a virtual force iteration scheme to avoid these deterioration situations and achieve the desired coverage performance.

The objectives of the monitoring coverage task include maximizing the monitored region of the AOI and minimizing the total travel length of agents. We introduce three indices for evaluating the

performance of fulfilling the task: 1) the coverage rate based on agents' final positions, 2) the total path length of agents from their initial positions to the final positions and 3) the evenness degree of agents' final distribution. The most important evaluation index is the coverage rate that is calculated as a percentage rate of the effective coverage area over the AOI. The evenness degree is defined as the mean of the sums of the Euclidean distances between each agent and its three nearest agents, and a higher evenness degree value means a more even distribution. The evaluation process is not of the VFIS and it is conducted in the testing experiments only for observing and analyzing the experimental results.

3.2 Virtual force interaction scheme for monitoring coverage

The completion of the monitoring coverage task is divided into a series of stages which combine agents sensing their surroundings and updating temporary positions for covering the currently observed environment. The whole task is fulfilled by agents iteratively achieving the maximum coverage of the currently observed environment with the stage. The final state of each agent in one stage is the initial state of the agent in the next stage and the trajectories for agents' deployment can be directly obtained. The setup of the stages number relates to the size of the target environment and the agents' perception ranges.

We propose the VFIS to give agents proper global exploration capability and local exploitation capability in the form of virtual forces and update agents' positions until achieving a distribution with favorable coverage performance. In our method, each agent can respond to its current environmental observation via the virtual forces, which include the repulsive force from other nearby agents, the repulsive and/or vortex forces from the border of the given environment, and the repulsive and/or vortex forces from the observed obstacles when the agent is close to obstacles. These virtual forces are illustrated as the arrow lines in Fig. 1. The repulsive forces and the vortex forces are distinguished as solid lines and dashed lines. The direction of one vortex force is randomly selected from two converse directions, shown as dashed arrow lines. It can be seen that the 3rd agent and the 4th agent have the mutual repulsive force; the 2nd agent and the 3rd agent receive their repulsive and vortex forces individually from the nearby obstacles; and the 1st agent receives the repulsive and vortex forces from the environment border.

A repulsive force will push an agent directly away from an object, such as an adjacent agent, a nearby obstacle or the environment boundary, to improve its coverage performance locally; while a vortex force can motivate the agent to spread away using the environmental traits. In each stage, the distribution of agents can be determined by calculating all virtual forces and updating the agents' positions with the effect of these virtual forces, iteratively.

The execution of VFIS is divided into a series of stages. In one stage, each agent records the perceived environment information firstly, exchanges its position information with other agents in its communication zone and calculates its internal position information which is saved as a virtual position and is used as the current position in the calculation of the next iteration using the defined equations

iteratively. This means that the repulsive force between two agents is calculated using their real positions in the first iteration and using the mutual virtual positions exchanged via their communication afterward. However, the physical position of one agent is only updated once at the end of the single stage. Before the stage end, the agent will refuse to move to the final calculated position if the distance from the current physical position to the calculated position is less than a threshold. The Pseudo code of the proposed distributed approach is presented in Table 1, where Part A presents the agents' initial configuration and Part B shows how a single agent calculates its position only based on its local observation and the communication with the adjacent agents in one stage. The agent would update its physical position with its stage index until the index reaches to the predefined value N_S. The explanation of all types of virtual forces is presented after the algorithm pseudocode.

Table 1

Pseudo code of VFIS algorithm

Algorithm: Multi-stage Virtual Force Interaction Scheme

Part A: Initial configuration of the whole group of agents

Configure agents' perception range R_p, communication range R_c, monitoring range R_m, the number of agents N_a, their initial physical positions (x_{pi}, y_{pi}), movement step size L, the number of stages N_S, and the iteration times in one stage N_I. Set values for the virtual force coefficients w_{ra}, w_{rb} and w_{ro}, the distance thresholds d_{ath} and d_{bth}, and other parameters including R_l and R_h.

Part B: Distributed algorithm for the single agent a_j in one stage

1: Initialize the stage index n and the virtual position $x_{vi}=x_{pi}, y_{vi}=y_{pi}$
2: Observe the environment and save the perceived information.
3: for iteration index $t=1$ to N_I do
4: for $j=1$ to N_a do
5: if the agent a_j is connected
6: Send the virtual position x_{vi} and y_{vi}
7: Receive the virtual position x_{vj} and y_{vj}
8: Compute F_{rai}^{j} using Eq. (1)
9: end if
10: end for
11: Compute F_{rai} using Eq. (2) and compute F_{rbi} using Eq. (3)
12: Update r_i using (6)
13: for $k=1$ to N_o do
14: Compute F_{roi}^{k} using Eq. (4) and compute F_{voi}^{k} using Eq. (8)
15: end for
16: Compute F_{roi} F_{vbi} and F_{voi} using Eq. (5), Eq. (7) and Eq. (9)
17: Generate random values in [0, 1] $r1$ and $r2$
18: if $r1<P1$
19: $F_{roi}=0$
20: else
21: $F_{voi}=0$
22: end if
23: Compute $M_{x_{vo}}$ and $M_{y_{vo}}$ using Eq. (11) and Eq. (12)

24: if $r2 < P2$
25: $F_{rbi} = 0$
26: else
27: $F_{vbi} = 0$
28: end if
29: Compute $M_{x_{vb}}$ and $M_{y_{vb}}$ using Eq. (11) and Eq. (12)
30: Compute F_{ri} using Eq. (10)
31: Compute M_{x_r} and M_{y_r} using Eq. (11) and Eq. (12)
32: Update x_{vi}, y_{vi} using Eq. (13) and Eq. (14)
33: end for
34: Update $x_{pi} = x_{vi}, y_{pi} = y_{vi}$ and move its position
35: Update the stage index $n = n + 1$

The repulsive force between two agents is designed for adjusting their relative positions. The repulsive force exerted upon an agent a_i from a_j, F_{rai}^j, is defined to be inversely proportional to the distance between agents as Eq. (1). The magnitude of F_{rai}^j will be exceptionally large when the distance is small. The combined repulsive force from all other agents F_{rai} is defined as Eq. (2).

$$F_{rai}^j = \begin{cases} \left(w_{ra} \times \frac{1}{d_i^j},\ \alpha_{rai}^j \right), & \text{if } d_i^j < d_{ath} \\ 0, & \text{if } d_i^j \geq d_{ath} \end{cases} \tag{1}$$

$$F_{rai} = \sum_{j=1, j \neq i}^{N_{ai}} F_{rai}^j \tag{2}$$

where d_i^j denotes the Euclidean distance between a_i and a_j, d_{ath} represents a distance threshold, α_{rai}^j is the direction angle of the vector from a_j to a_i, w_{ra} represents the coefficient for the agent-to-agent repulsive force, and N_{ai} is the number of agents in the perception zone of the agent a_i.

We define the virtual repulsive force exerted on a_i from the environment boundary as F_{rbi}, which is used for avoiding the agent being too close to the boundary. The magnitude of F_{rbi} varies with a distance variable to ensure a small distance leads to an extremely big force. F_{rbi} is defined as Eq. (3).

$$F_{rbi} = \begin{cases} \left(w_{rb} \times \frac{1}{d_{bi}},\ \alpha_{rbi} \right), & \text{if } d_{bi} < d_{bth} \\ 0, & \text{if } d_{bi} \geq d_{bth} \end{cases} \tag{3}$$

where d_{bi} denotes the shortest distance from a_i to its surrounding map border, d_{bth} represents a distance threshold, α_{rbi} denotes the direction from the nearest border point to a_i, and w_{rb} represents the coefficient for the border-to-agent repulsive force.

For each agent, its occlusive area refers to the area blocked by the obstacles in the monitoring range owing to the light-of-sight matter. The repulsive force contributed by obstacles relates to both the agent's relative position and the shape of the obstacles, as shown in Fig. 1. Therefore, the repulsive force exerted on a_i by an obstacle o_i is defined to vary with an area variable rather than a distance variable, as shown in Eq. (4). When the quantity of the observed obstacles is N_{obi}, the combined repulsive force from all observed obstacles $F_{roi}$is defined as Eq. (5).

$$F_{roi}^{k}=\left(w_{ro}\times A_i^k,\quad \alpha_{roi}^{k}\right) \tag{4}$$

$$F_{roi}=\sum_{k=1}^{N_{obi}} F_{roi}^{k} \tag{5}$$

where A_i^k is the occlusive area blocked by one obstacle O_k within the agent's provisional monitoring range, α_{roi}^{k} is the direction of the vector from the closest location of the occlusive area to a_i, $\mathrm{w_{ro}}$ denotes the coefficient of the obstacle-to-agent repulsive force, and N_{obi} is the number of obstacles in the perception zone of the agent a_i.

In this work, the occlusive area A_i^k is calculated based on a changeable monitoring range r_i, which is designed for adaptation in the environment with dense obstacles. We define this provisional monitoring range r_i varying between a higher limit and a lower limit in one stage as Eq. (6).

$$r_i=\begin{cases} R_h-\dfrac{R_h-R_l}{\frac{N_I}{2}}\times t_i,\ if\ 0<t_i\le\dfrac{N_I}{2} \\ R_l+\dfrac{R_h-R_l}{\frac{N_I}{2}}\times\left(t_i-\dfrac{N_I}{2}\right),\ if\ \dfrac{N_I}{2}<t_i\le N_I \end{cases} \tag{6}$$

where N_I is the total iteration times in one stage, t_i is the current iteration index, R_l and R_h are the lower limit and the higher limit of the provisional monitoring range respectively.

Despite the intention of introducing the vortex force as different from the repulsive force, they are closely linked with each other. The orientation of a border-to-agent vortex force is perpendicular to the orientation of the relevant border-to-agent repulsive force, and the magnitude is proportional to the corresponding repulsive force's magnitude, as Eq. (7).

$$F_{vbi}=\left(w_{vb}\times\left|F_{rbi}\right|,\quad \alpha_{vbi}\right) \tag{7}$$

where α_{vbi} is the direction angle of F_{vbi} with a value either $\alpha_{rbi}-pi/2$ or $\alpha_{rbi}+pi/2$ whenF_{rbi} is not zero, the coefficient w_{vb} is a random value uniformly distributed in (0.4, 0.8).

Similarly, the direction of an obstacle-to-agent vortex force is perpendicular to the orientation of the homologous obstacle-to-agent repulsive force. The magnitude is proportional to that of the corresponding repulsive force, as shown in Eq. (8). The combined vortex force from all observed obstacles F_{voi} is defined as Eq. (9).

$$F_{voi}^{k}=\left(w_{vo}\times\left|F_{voi}^{k}\right|,\quad \alpha_{voi}^{k}\right) \tag{8}$$

$$F_{voi}=\sum_{k=1}^{N_{obi}}F_{voi}^{k} \tag{9}$$

where $\alpha_{voi}^{k}$is the direction angle of F_{voi}^{k} with a value either $\alpha_{roi}^{k}-pi/2$ or $\alpha_{roi}^{k}+pi/2$ when F_{roi}^{k} is not zero, w_{vo} is a random value uniformly distributed in (0.4, 0.8).

A vortex force is assumed to motivate an agent to adjust its position at a certain probability. The border-to-agent vortex force F_{vbi} and the obstacle-to-agent vortex force F_{voi} take effect at probability $P1$ and $P2$ respectively. When the agent adopts the movement caused by a vortex force F_{vbi} or F_{voi} , the corresponding repulsive force F_{rbi} or F_{roi} is modified to be 0. The combined repulsive force F_{ri} upon the agent a_i is calculated as (10) and we always allow the repulsive force to push the agent to move.

$$F_{ri}=F_{rai}+F_{roi}+F_{rbi} \tag{10}$$

The movement generated by either F_{ri}, F_{voi} or $F_{\mathrm{vb}i}$ follows the rules defined in Eq. (11) and Eq. (12). We introduce an exponential format in the equations for limiting the agents' movement to a small range.

$$M_{x_s}=W_m\times L\times\frac{\left|F_{x_s}\right|}{\left|F_s\right|}\times e^{\frac{-1}{|F_s|}} \tag{11}$$

$$M_{y_s}=W_m\times \mathrm{L}\times\frac{\left|\mathrm{F}_{y_s}\right|}{\left|F_s\right|}\times e^{\frac{-1}{|F_s|}} \tag{12}$$

where M_{x_s} and M_{y_s} are motion displacements along the horizontal and vertical coordinates in the Cartesian coordinate system respectively; L is the maximum movement related to the number of agents and the area of the map; $\mathrm{W_m}$is a random coefficient uniformly distributed in (0.5, 1); F_s, F_{x_s} and F_{y_s}

represent one virtual force and its component forces along the horizontal and vertical coordinates, respectively.

The positions of agents are updated using the following equations:

$$x(t+1)=x(t)+M_{x_r}(t)+M_{x_{vo}}(t)+M_{x_{vb}}(t) \quad (13)$$

$$y(t+1)=y(t)+M_{y_r}(t)+M_{y_{vo}}(t)+M_{y_{vb}}(t) \quad (14)$$

where $(x(t),y(t))$ and $(x(t+1),y(t+1))$ are the current location and the newly computed location of the agent; M_{x_r}, $M_{x_{vo}}$ and $M_{x_{vb}}$ are the motion displacements along the x-coordinate under the received virtual forces F_{ri}, F_{voi} and F_{vbi}, respectively; M_{y_r}, $M_{y_{vo}}$ and $M_{y_{vb}}$ are the motion displacements along the y-coordinate under F_{ri}, F_{voi} and F_{vbi}, respectively.

4. Benchmark problem design

We defined 22 benchmark configurations for the monitoring coverage problem, considering various characteristics of the target environment and various features of the robotic agents, which are summarized in Table 2. Eight maps are utilized in the benchmark problems, as shown in Fig. 2(a-h). The area of each map is less than 10,000 square units and the area occupied by obstacles remains at about 12% of the total area in all eight maps. The static obstacles in the map are represented by the black zone. Agents are denoted as red star points, and their perception scope and the ideal monitoring area are represented by blue circles and light blue disks respectively. The properties of the target environments include the shape of the environment border, such as square, circle and triangle, and the obstacles' quantity, shapes and distribution. In the first three maps Fig. 2(a-c), the environment is a 100-unit by 100-unit square containing 6, 12, and 20 obstacles, which are with regular geometric shapes and in even distributions. Their complexity for a monitoring coverage problem increases with the increase of the obstacles number. The fourth map (Fig. 2(d)) contains 12 obstacles in an uneven distribution. To investigate the influence of the map border, we set 16 evenly distributed obstacles in a circular border with a 50-unit radius as the fifth map (Fig. 2(e)) and 10 evenly distributed obstacles in a triangular border with the area of 5000 square units as the sixth map (Fig. 2(f)). We also set an extra square map and an extra circular map that contain the obstacles with irregular geometric shapes as Map 7 and Map 8 (Fig. 2(g-h)), respectively.

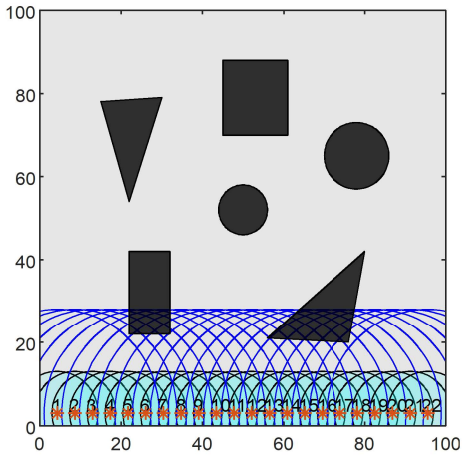
The number of agents is an important element considered in the design of benchmark problems. The ratio of the target environment area to the monitoring area of one agent provides a clue for estimating the number of agents needed to cover the entire AOI. This ratio is assumed to be the sufficient number for agents, which is denoted as N_{su}. We set the first eight benchmark configurations as initializing an adequate number ($0.67\,N_{su}$) of agents at the border line of Maps 1-8. For observing the influence of the number of agents, we used $0.5\,N_{su}$ and N_{su} agents in Maps 3, 6 and 8 in the following six

configurations (Problems 9-14). The next four benchmark problems concern agents' initial positions. We chose $0.67N_{su}$ and N_{su} agents and set agents' start positions at the central area of Map 3 in Problems 15 and 16. $0.67N_{su}$ and N_{su} agents are initialized at a border corner of Map 4 in Problems 17 and 18. We then considered the case with larger monitoring ranges in Problems 19 and 20, and the case that the perception range is equal to the monitoring range in Problems 21 and 22.

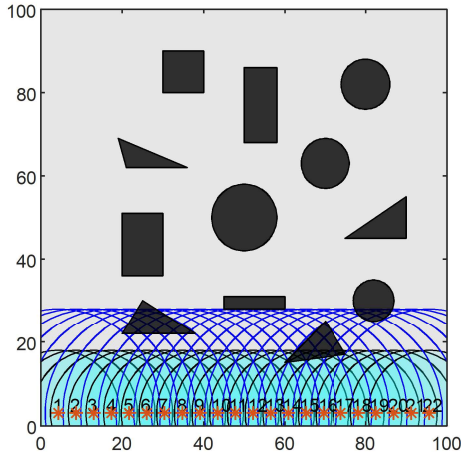
Table 2

Benchmark problem configurations.

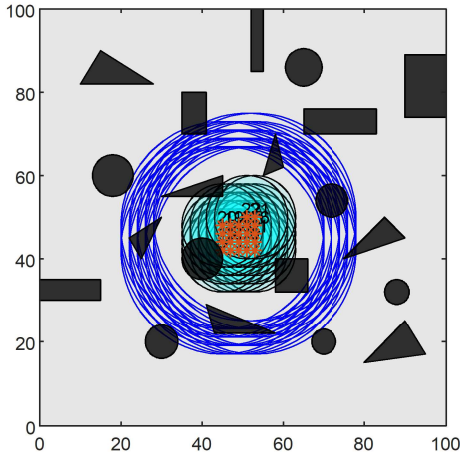
Benchmark problem	**Map**	**Map Border**	N_{ob}	**Distribution of obstacles**	N_a	R_m	R_p	**Initial positions**
No. 1	Map 1	Square	6	Even	22	10	25	Border line
No. 2	Map 2	Square	12	Even	22	10	25	Border line
No. 3	Map 3	Square	20	Even	22	10	25	Border line
No. 4	Map 4	Square	12	Uneven	22	10	25	Border line
No. 5	Map 5	Circle	16	Even	17	10	25	Border line
No. 6	Map 6	Triangle	10	Even	11	10	25	Border line
No. 7	Map 7	Square	6	Overlapped	22	10	25	Border line
No. 8	Map 8	Circle	6	Overlapped	17	10	25	Border line
No. 9	Map 3	Square	20	Even	**16**	10	25	Border line
No. 10	Map 3	Square	20	Even	**32**	10	25	Border line
No. 11	Map 6	Triangle	10	Even	**8**	10	25	Border line
No. 12	Map 6	Triangle	10	Even	**16**	10	25	Border line
No. 13	Map 8	Circle	6	Overlapped	**13**	10	25	Border line
No. 14	Map 8	Circle	6	Overlapped	**26**	10	25	Border line
No. 15	Map 3	Square	20	Even	22	10	25	**Centre**
No. 16	Map 3	Square	20	Even	32	10	25	**Centre**
No. 17	Map 4	Square	12	Uneven	22	10	25	**Border point**
No. 18	Map 4	Square	12	Uneven	32	10	25	**Border point**
No. 19	Map 2	Square	12	Even	**10**	**15**	25	Border line
No. 20	Map 2	Square	12	Even	**15**	**15**	25	Border line
No. 21	Map 5	Circle	16	Even	17	10	**10**	Border line
No. 22	Map 6	Triangle	10	Even	11	10	**10**	Border line



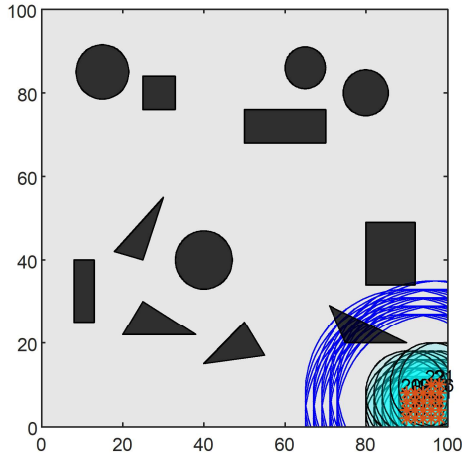
(a) Problem 1 in Map 1



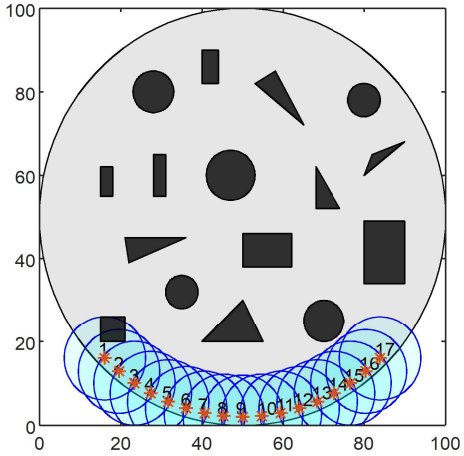
(b) Problem 19 in Map 2



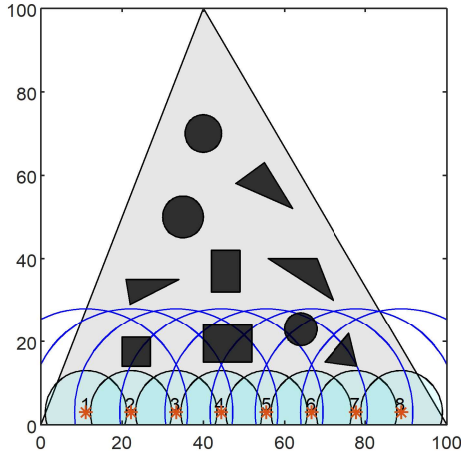
(c) Problem 15 in Map 3



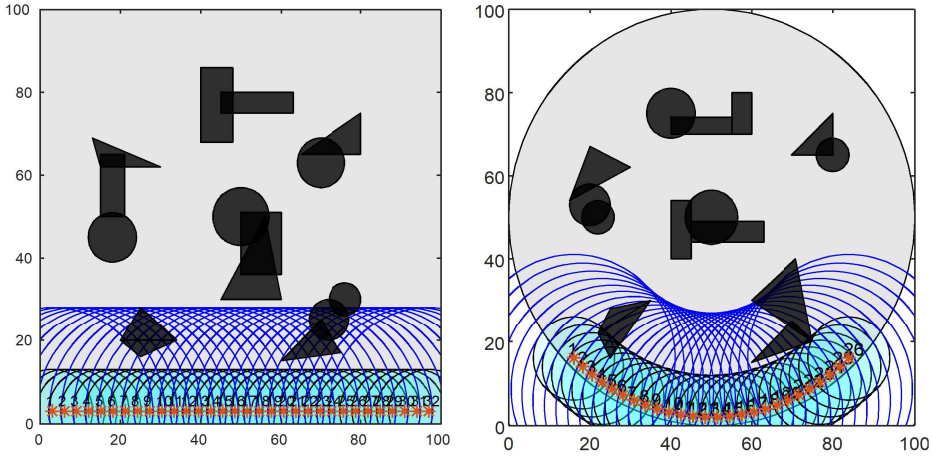
(d) Problem 17 in Map 4



(e) Problem 21 in Map 5



(f) Problem 11 in Map 6



(g) Problem 7 in Map 7 (h) Problem 14 in Map 8

Fig. 2. Examples of benchmark problem settings.

5. Experiments and discussion

5.1 Parameter Setting

The parameters of VFIS used in testing all benchmark problems were fixed as shown in Table 3. We chose 100 iteration times in one stage to ensure adequate iterative computation and set 15 stages for exploring the whole target area in one experiment. Three distance thresholds of the virtual force were related to the monitoring rangeR_m; $d_{\text{bth}}$was fixed toR_m; the provisional coverage range r_i varied between $0.2\,R_m$ and R_m following (6); and d_{ath} varied from either $2\,R_m$ or $\sqrt{N_{su}/N_a}\times 2\,R_m$ (R_c *if* $\sqrt{N_{su}/N_a}\times 2\,R_m > R_c$) at the same probability. The movement step size L was set to $d_{\text{ath}}/12$ and the vortex force probability was fixed as 0.6. Twenty-two groups of experiments have been executed against the benchmark problems and experiment has been repeated 20 times for each benchmark problem.

Table 3

Parameters setting of VFIS.

Parameter	N_S	N_I	w_{ra}	w_{rb}	w_{ro}	d_{bth}	R_h	R_l	P
Value	15	100	20	30	1	R_m	R_m	$0.2R_m$	0.6

5.2 Experimental results

Table 4 shows the performance of VFIS in terms of several performance indices, the coverages rate, the distribution uniform degree and the average path length. The uniform degree U_d is mainly used for evaluating the distribution performance, and is a two-variable vector in this paper. We take the average of the sum of each agent's distances to three closest agents as the first term, and choose the standard

deviation of each distance sum as the second term, and normalize these two variables in R_m unit as (15).

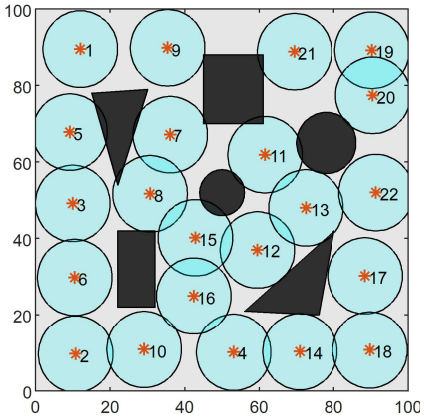
$$U_d=[\frac{\sum_{i=1}^{N_a} D_i}{N_a \times R_m}, \sqrt{\frac{\sum_{i=1}^{N_a}(D_i-\frac{\sum_{i=1}^{N_a} D_i}{N_a})^2}{(N_a-1)\times R_m^2}}] \quad (15)$$

where D_i is the sum of distances between a_i and its three closest agents.

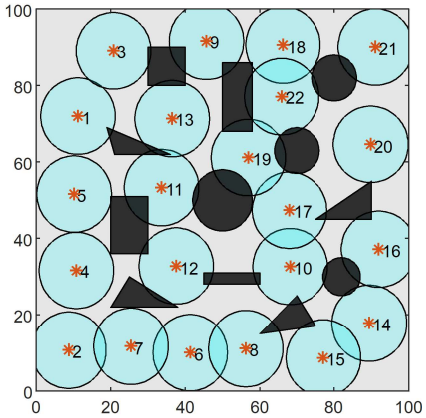
Table 4

Experimental results.

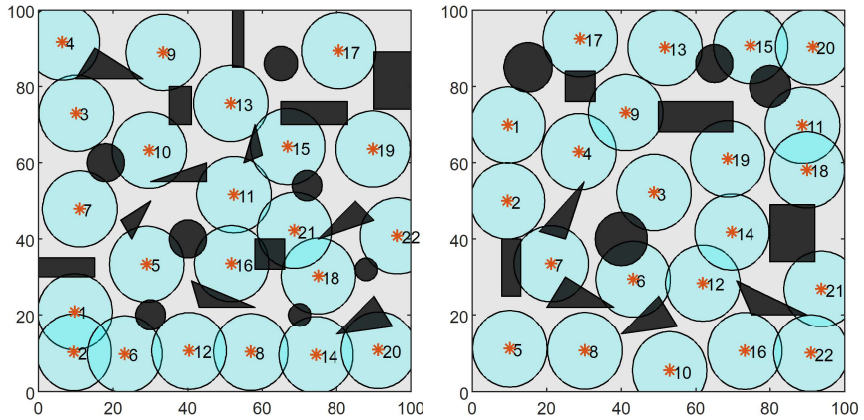
Exp. No.	Average Coverage	Coverage Rate (%)	Uniform Degree	Average Path Length
1	6499.3±31.4	73.9±0.4	[6.47, 0.80]	66.0±2.8
2	6155.2±92.0	69.9±1.0	[6.54, 0.87]	72.0±2.9
3	5912.8±103.7	67.2±1.2	[6.30, 0.91]	76.8±4.2
4	6111.2±110.8	69.4±1.3	[6.45, 0.79]	76.1±2.9
5	4640.1±73.8	67.1±1.1	[6.69, 0.60]	77.8±4.8
6	2682.7±85.1	61.0±1.9	[6.69, 1.19]	78.9 ±6.2
7	6274.2±118.1	71.3±1.3	[6.47, 0.86]	73.3±2.9
8	4871.2±66.2	70.5±1.0	[6.52, 0.74]	78.0±4.7
9	4598.1±67.2	52.3±0.8	[7.64, 1.21]	80.5±3.8
10	7260.7±136.1	82.5±1.5	[5.06, 0.76]	77.4±1.5
11	2098.7±60.0	47.7±1.4	[7.60, 1.44]	87.6±10.0
12	3483.1±117.0	79.2±2.7	[5.68, 1.12]	69.8±4.4
13	3892.0±45.0	56.3±0.7	[7.80, 0.76]	74.5±3.1
14	6010.8±69.7	87.0±1.0	[4.92, 0.64]	75.0±2.6
15	5935.8±57.5	67.5±0.7	[6.45, 0.94]	69.7±3.5
16	7461.5±186.0	83.8±2.1	[5.10, 0.74]	68.8±2.0
17	6149.1±66.8	69.9±0.8	[6.42, 0.87]	92.2±2.2
18	7461.5±82.3	84.8±0.9	[5.04, 0.84]	95.1±2.7
19	5787.8±161.5	65.8±1.8	[6.77, 1.20]	86.5±5.4
20	7114±227.9	80.8±2.6	[5.21, 1.11]	90.8±6.0
21	4584.6±72.4	66.3±1.0	[6.52, 0.66]	83.8±3.0
22	2617.2±120.9	59.5±2.7	[6.30, 1.29]	88.1±4.4



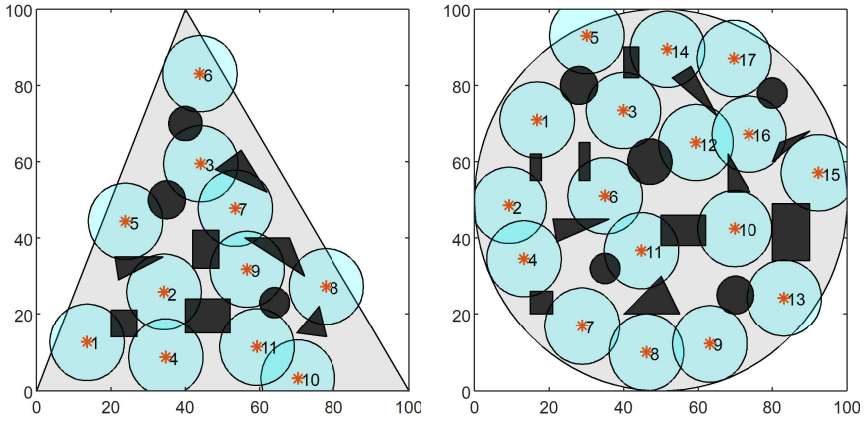
(a) Exp. 1



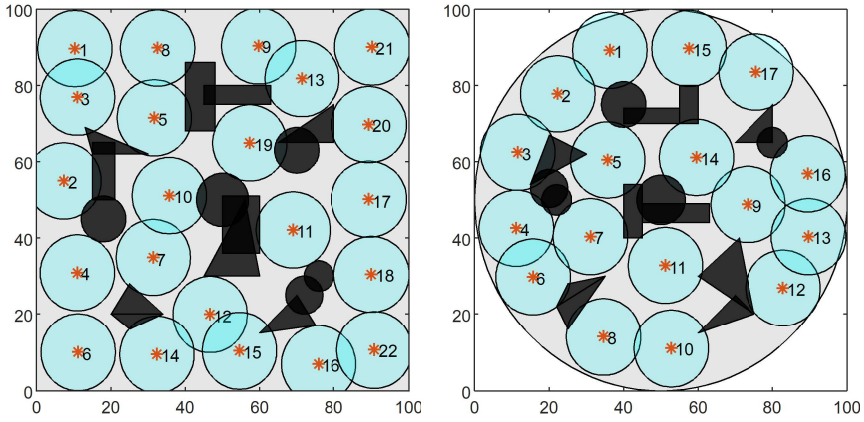
(b) Exp. 2



(c) Exp. 3 (d) Exp. 4



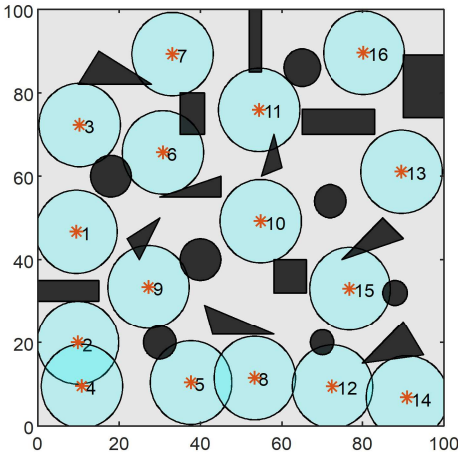
(e) Exp. 5 (f) Exp. 6



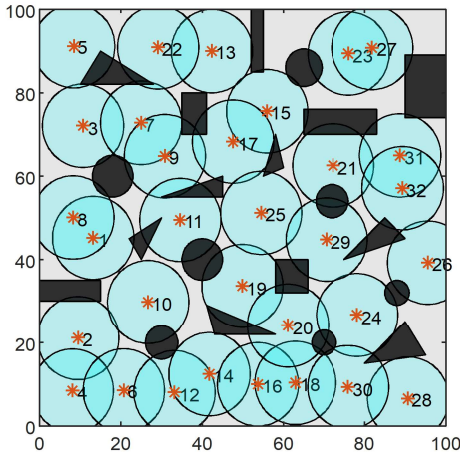
(g) Exp. 7 (h) Exp. 8

Fig. 3. Coverage performance of benchmark problems No. 1-8.

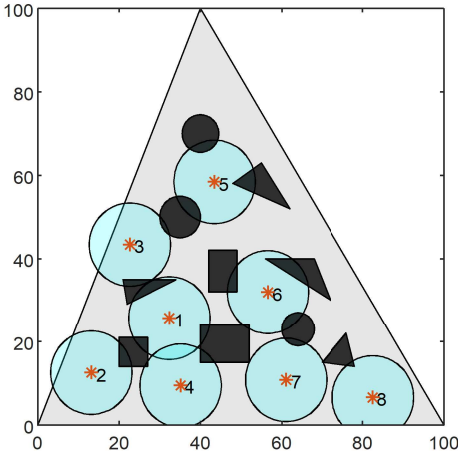
Fig. 3 illustrates the final deployment of the first eight benchmark problems and shows that the overall monitoring coverage states are favourable in all eight maps. From Exps. 1-3 in Table 4 and Fig. 3, we can observe that the coverage rate decreases slightly with the number of obstacles increasing. If Exp. 4 is compared with others, such as Exp. 2, we can see that the distribution of obstacles has no obvious effect on the coverage performance of the designed method. The average coverage rate in Exps. 1-8 is close to 70%, except for Exp. 6. The smaller coverage rate in Exp. 6 is mainly due to the triangular map border and a smaller AOI, which leads to more overlap between the agents and the environment border. The coverage uniform degree in Exps. 1-8 keeps similar.



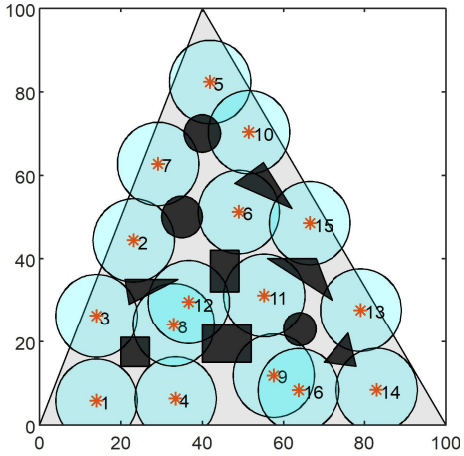
(a) Exp. 9



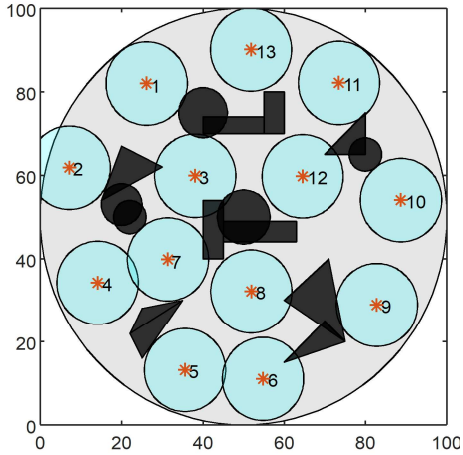
(b) Exp. 10



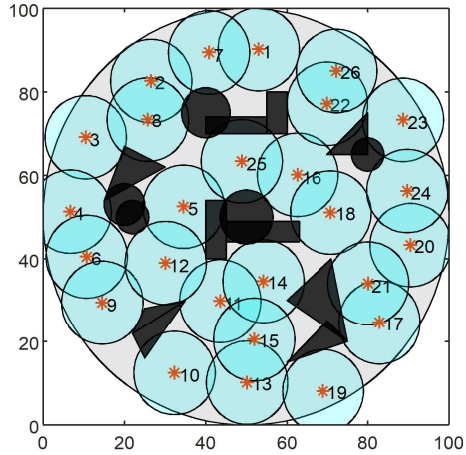
(c) Exp. 11



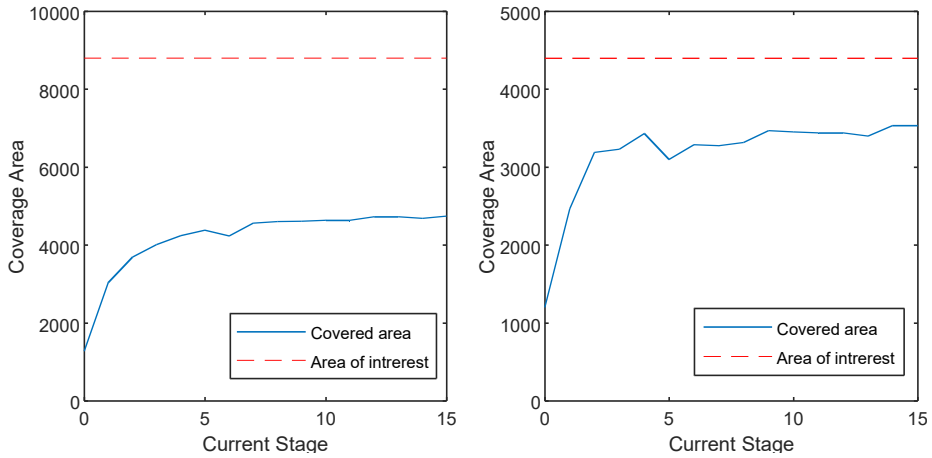
(d) Exp. 12



(e) Exp. 13



(f) Exp. 14



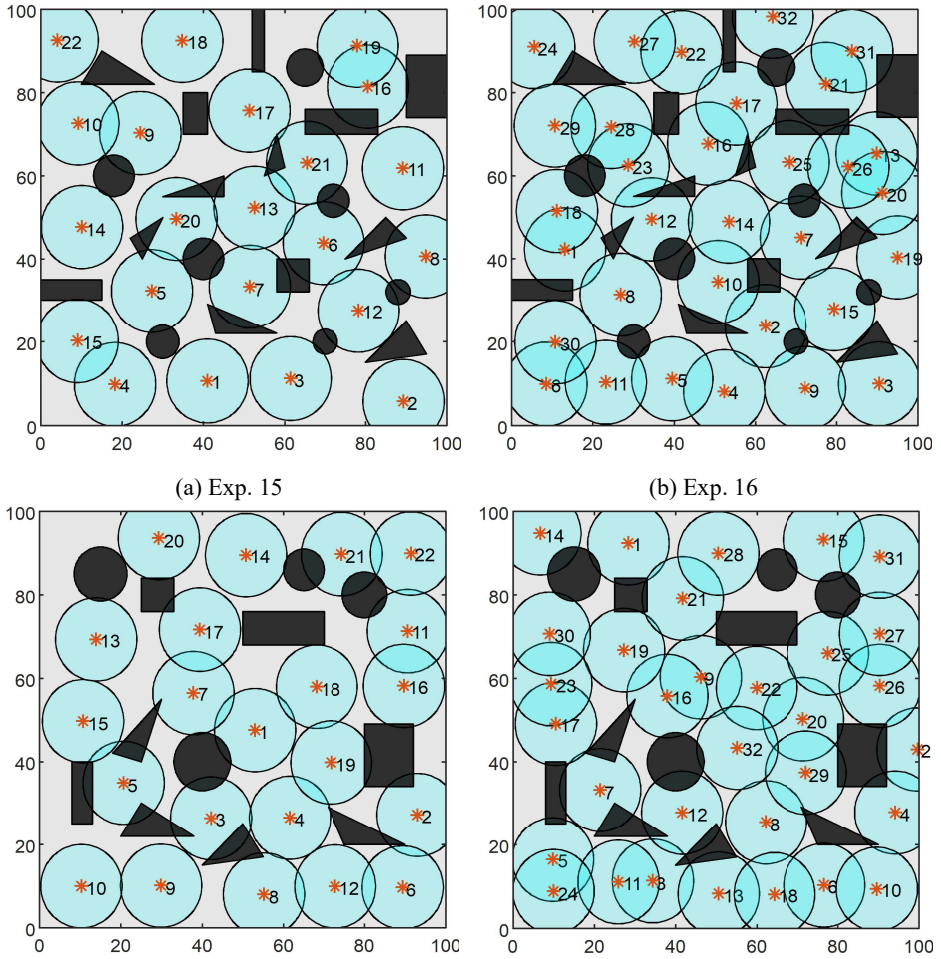
(g) Area Coverage trend in Exp. 9 (h) Area Coverage trend in Exp. 12.

Fig. 4. Coverage performance of benchmark problems No. 9-14.

Fig. 4 presents the coverage performance of the benchmark problems No. 9-13. The coverage rate increases with the number of agents, and the coverage rate is great than 79% when the number of agents is sufficient. The coverage rate in the triangular environment is still lowest under different settings of the number of agents. Fig. 4(g-h) presents the changing trends of the area coverage with the stage. We can see the area coverage initially increases with stage and may then oscillate around the 'best' value due to the stimulative effect of vortex forces.

Fig. 5 presents the final coverage performance in the benchmark problems No. 15-18. Comparing the results of Exps. 15-17 with the results of Exps. 3, 10 and 4 in Table 4, we can find that the final coverage rate and distribution state are similar although the initial positions of the agents in the map are different. Fig. 6 demonstrates the effectiveness of VFSI under the cases with a larger monitoring range or a smaller perception range. A larger monitoring range means less agents are needed and the monitoring area of a single agent may be larger than a single obstacle. A smaller perception range also makes the monitoring coverage task more difficult to complete. From the results in Table 4 and Fig. 6, it is shown that VFIS works well with different combination of R_m, R_p and R_c.

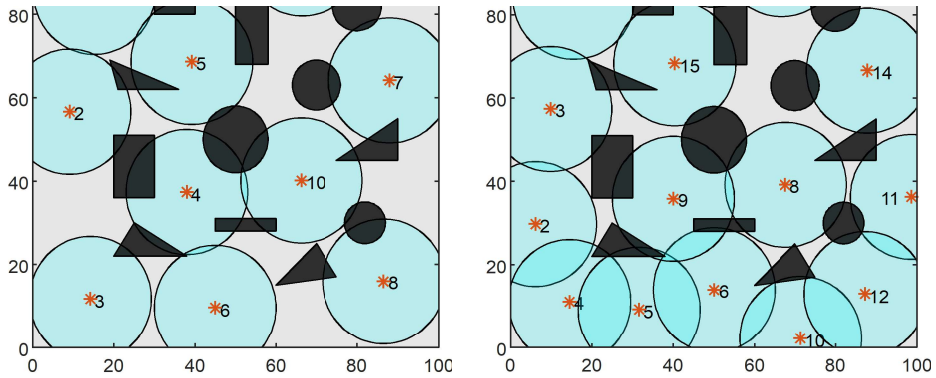
For the uniform degree, we expect to see a large first term and a small second term for an even distribution. For a given problem, the first term is small at the initial stage because of agents gathering together. From Table 4, we can find that the first term is less than 6 when the number of agents is N_{su}, and the second term is relatively large when the number of agents is small. According to the results of Exps. 1-4 and 7 in Table 4, one can find that the average path length of agents increases with the complexity of obstacle distribution increasing. Different from the coverage rate, the average path length relates to the initial positions of agents. The average path length is shortest when the agents start from the center of the target area and it is longest when the agents start from the corner of the target area. The average path in the cases with a larger monitoring range or a smaller perception range is greater.



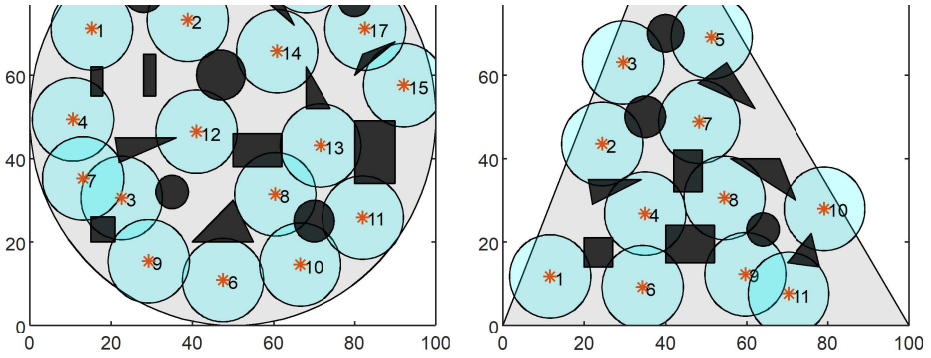
(a) Exp. 15 (b) Exp. 16

(c) Exp. 17 (d) Exp. 18

Fig. 5. Coverage performance of benchmark problems No. 15-18.



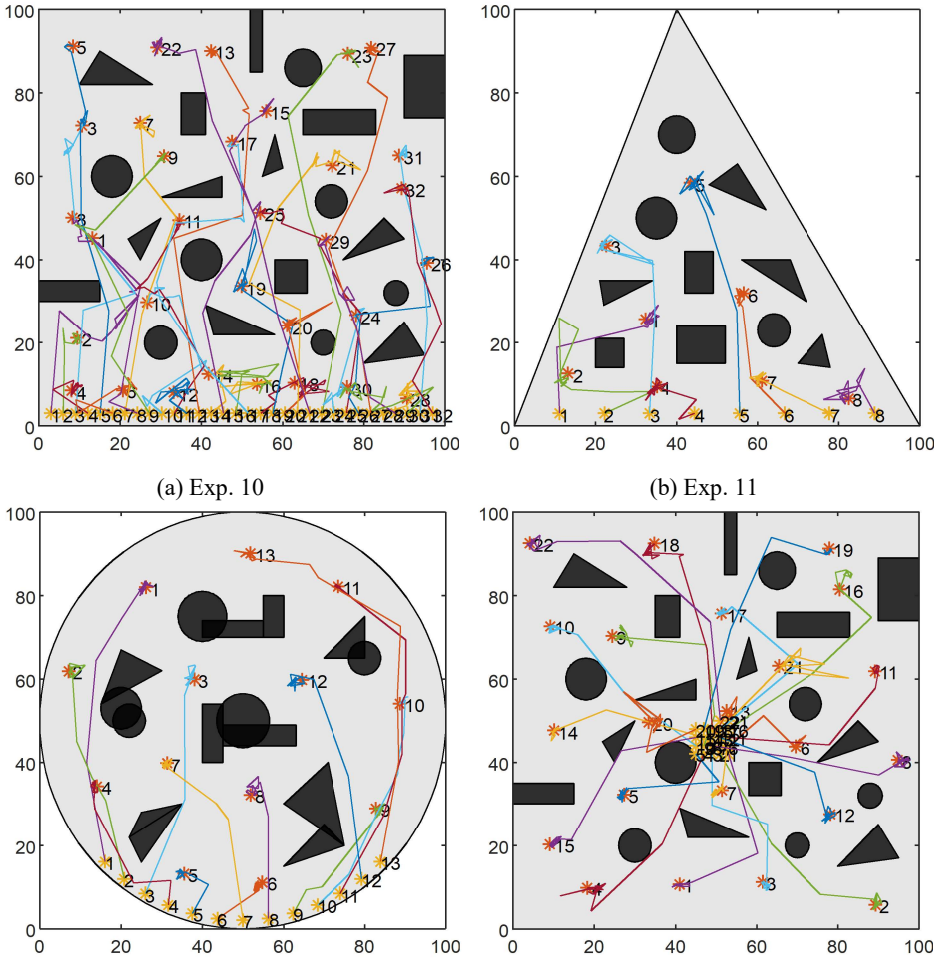
(a) Exp. 19 (b) Exp. 20



(c) Exp. 21 (d) Exp. 22

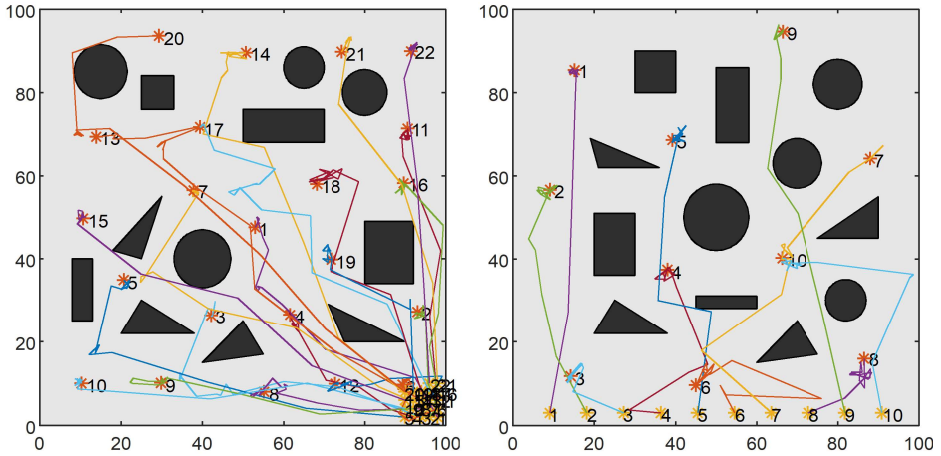
Fig. 6. Coverage performance of benchmark problems No. 19-22.

Fig. 7 presents six examples of agents' paths in different benchmark problems. The paths of agents are overall reasonable except for some oscillatory movements near their final positions. The average path length in Exp. 17 is longest and the one in Exp. 15 is relatively short according to Table 4. Some obvious near-end oscillation can be observed in some experiments, for example Exp. 11, as shown in Fig. 7(b). Such oscillation may cause longer paths, since the agents continue to update their positions in the final stages due to the effect of vortex forces.



(a) Exp. 10 (b) Exp. 11

(c) Exp. 13 (d) Exp. 15



(e) Exp. 17 (f) Exp. 19

Fig. 7. Examples of generated paths using VFIS.

Table 5

Results of Comparative Experiments.

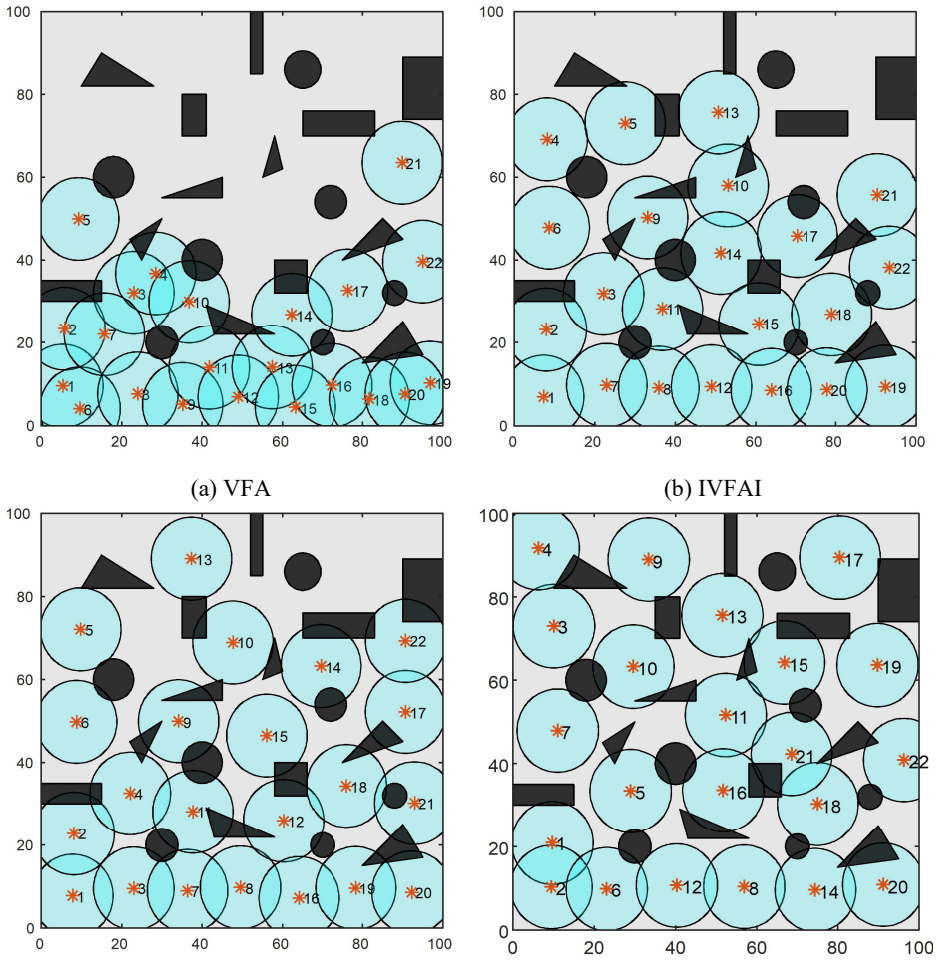
Exp. No.	Algorithm	Benchmark problem	**Coverage Rate(%)**	Uniform Degree	Average Path Length
1	VFA	No. 1	68.4	[6.28,1.30]	55.7
2	IVFAI	No. 1	64.2	[5.62,0.79]	38.1
3	OAVFA	No. 1	73.7	[6.63,0.71]	57.2
4	VFIS	No. 1	73.9	[6.47,0.80]	66.0
5	VFA	No. 3	45.2	[4.56,1.74]	31.1
6	IVFAI	No. 3	64.2	[5.78,0.87]	41.6
7	OAVFA	No. 3	64.5	[6.18,1.24]	43.0
8	VFIS	No. 3	67.2	[6.30,0.91]	76.8
9	VFA	No.9	34.1	[4.51,1.59]	17.3
10	IVFAI	No.9	45.9	[6.39,1.10]	33.8
11	OAVFA	No.9	44.8	[6.41,1.34]	33.5
12	VFIS	No.9	52.3	[7.64,1.21]	80.5
13	VFA	No.10	61.2	[4.25,1.48]	38.7
14	IVFAI	No.10	58.4	[3.80,0.78]	29.7
15	OAVFA	No.10	58.1	[3.84,0.75]	29.1
16	VFIS	No.10	82.5	[5.06,0.76]	77.4

5.3 Algorithms Comparison

To verify the effectiveness of our algorithm in improving coverage performance, the basic virtual force algorithm [12] (VFA), obstacle avoidance virtual force algorithm [20] (OAVFA) and improved virtual force algorithm based on area intensity [21] (IVFAI) are selected for comparative analysis and discussion based on the benchmark problems No. 1, No. 3, No. 9 and No.10. The influence of the

number of obstacles and the number of agents on the algorithms' quality can be observed. For IVFAI, we modified the air intensity calculation in distributed form and added the basic repulsive force from the environment border and obstacle due to this algorithm was tested only in obstacle-free environment in literature. The iteration times in one stage is set as only 20 and the stage number is set as 30 in the above three algorithms considering no periodic randomization scheme for virtual force proposed in VFIS and this means these algorithms would require less iterative calculation to achieve their final deployment state compared with VFIS. The simulation results are presented in Table 5.

From Exps. 1-4 in Table 5, we can observe that the coverage rate and uniform distribution degree of VFIS in the simple environment with a moderate number of agents are similar to the results of OAVFA and are better than the results of IVFAI and VFA. As Exps. 5-9 in Table 5, the coverage quality of VFIS in the complex environment with a moderate number of agents exceeds IVFAI and OAVFA cases a bit and is much better than the VFA case. Fig. 8 illustrates the results intuitively.



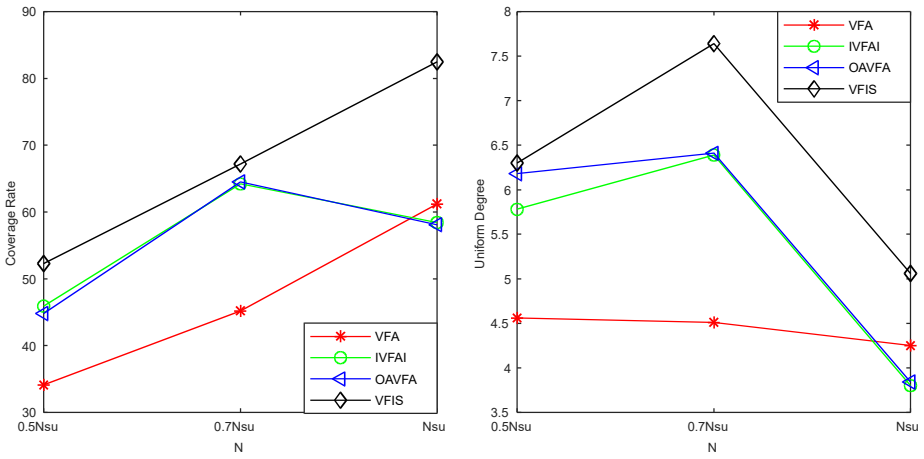
(a) VFA (b) IVFAI

(c) OAVFA (d) VFIS

Fig. 8. Coverage performance of benchmark problems No. 3 based on various algorithms.

In Benchmark problem No.3, No. 9, and No. 10, the maps keep the same and the number of agents are $0.67\,N_{su}$, $0.5\,N_{su}$ and N_{su} respectively. Fig. 9 presents the performance of coverage rate and

uniform degree with the number of agents changes according to the results of Exps. 5-16. In this specific environment, the coverage quality of VFA is the worst among these algorithms in most conditions, the performance of IVFAI and OAVFA improved with increase of the number of agents firstly and then deteriorate and VFIS always keeps the best coverage rate and uniform degree among these algorithms. As the results in the final column of Table 5, one can find that the average path length of agents with VFIS is greatest compared with other algorithms in all 4 selected benchmark problems, but this is necessary for maximizing coverage rate in the complex environment. Overall, VFIS is the most suitable for environment monitoring task in the complex environment considering three indices' results compared with other algorithms although it cost more running time.



(a) coverage rate (b) the first term of uniform degree

Fig. 9. Comparison of coverage performance in various algorithms.

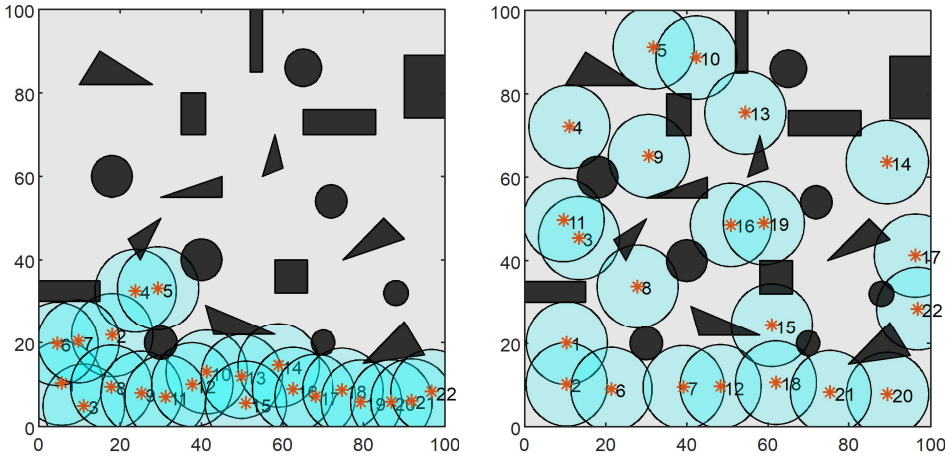
6. Parameter Analysis

6.1 Vortex force mechanism

Fig. 10 illustrates the coverage performance of two non-vortex-force solutions to the benchmark problem No. 3. Agents are blocked by obstacles and cannot spread when the obstacle-to-agent repulsive force is calculated based on a fixed provisional monitoring range, as shown in Fig. 10(a). The blocking situation is eased by calculating the repulsive force based on a varying provisional monitoring range, as shown in Fig. 10(b). When the vortex force is exerted at the probability $P=0.6$, the coverage performance becomes much better as shown in Fig. 3(c). The vortex force in VFIS can provide active exploration ability to agents so as to improve the coverage performance in monitoring coverage problems.

The action times of the vortex force is mainly controlled by the probability P. In this section, we investigate the influence of different probability values. Table 6 present the coverage performance against benchmark problem No. 12 when P is 0.2, 0.4, 0.6, 0.8, 0.9 and 1. Fig. 11 illustrates the change of three performance indices with the increase of P from 0 to 1. We can see that the coverage

rate increases with P increasing from 0.2 to 0.8 and then decreases. We can also find that the travel distance increases with the increase of P but keeps relatively stable in the range of 0.5 to 0.8. The uniform degree becomes better when P increases from 0 to 0.95 and it becomes worse when P is over 0.95. Considering the overall performance for all benchmark problem configurations, the value in the range between 0.6 and 0.9 is an appropriate choice for P.



(a) fixed monitoring range. (b) varying provisional monitoring range

Fig. 10. Two non-vortex-force solutions for Problem 3

Table 6

Experiments on Vortex Force Probability.

P	Average Coverage	Coverage Rate (%)	Uniform Degree	Average Path Length
0.2	3106.0±90.9	70.6±2.1	[4.81, 1.16]	56.2±3.4
0.4	3262.5±85.0	74.1±1.9	[5.19, 1.17]	66.0±2.6
0.6	3483.1±117.0	79.2±2.7	[5.68, 1.12]	69.8±4.4
0.8	3633.8±93.5	82.6±2.1	[6.16, 1.10]	71.8±4.2
0.9	3631.6±113.1	82.5±2.6	[6.32, 1.08]	75.8±5.7
1	3465.2±126.6	78.8±2.8	[6.18, 1.07]	87.8±7.8

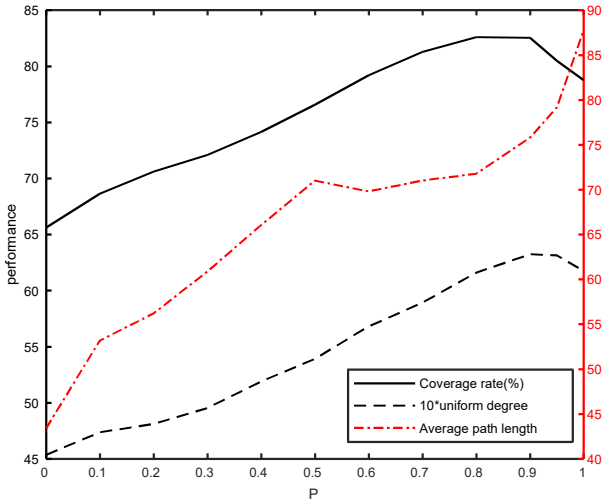


Fig. 11. Effects of the vortex force probability P based on Problem 12. The coverage rate and the first term of uniform degree are based on the left column axis, and the average path length refers to the right column axis.

6.2 Step size in one iteration

We set the step size L in one iteration to different values and investigated the influence of the step size in the benchmark problem No. 3. Each experiment was repeated 10 times and the selected results are recorded in Table 7. Fig. 12 illustrates the change of three performance indices with the increase of L from $d_{\mathrm{ath}}/50$ to $d_{\mathrm{ath}}/5$. We find that the coverage rate and uniform degree increase with the step size increasing in a certain range and it deteriorates when the step size is too large. When the step size is larger, the average path length becomes longer. The step size value $d_{\mathrm{ath}}/12$ has the best performance considering all three indices.

Table 7

Experiments on Step Size.

L	Average Coverage	Coverage Rate (%)	Uniform Degree	Average Path Length
$d_{\mathrm{ath}}/30$	5478.2±223.6	62.3±2.5	[5.82,1.37]	54.4±3.6
$d_{\mathrm{ath}}/20$	5787.0±144.8	65.8+1.6	[6.14,1.10]	64.6±2.3
$d_{\mathrm{ath}}/12$	5912.8±103.7	67.2±1.2	[6.30,0.91]	76.8±4.2
$d_{\mathrm{ath}}/5$	5671.5±152.0	64.4±1.7	[6.23,0.96]	123.2±3.1

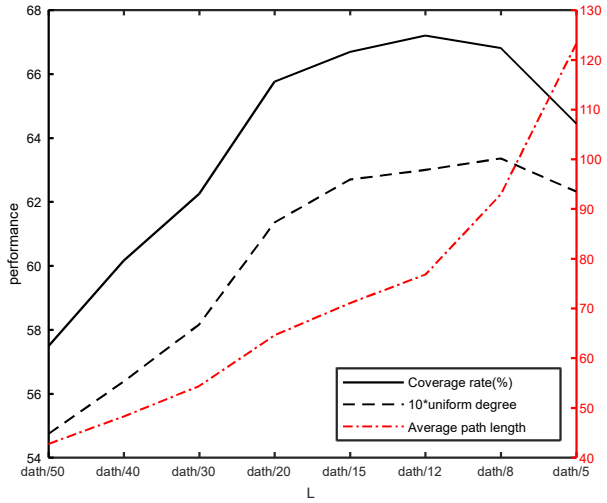
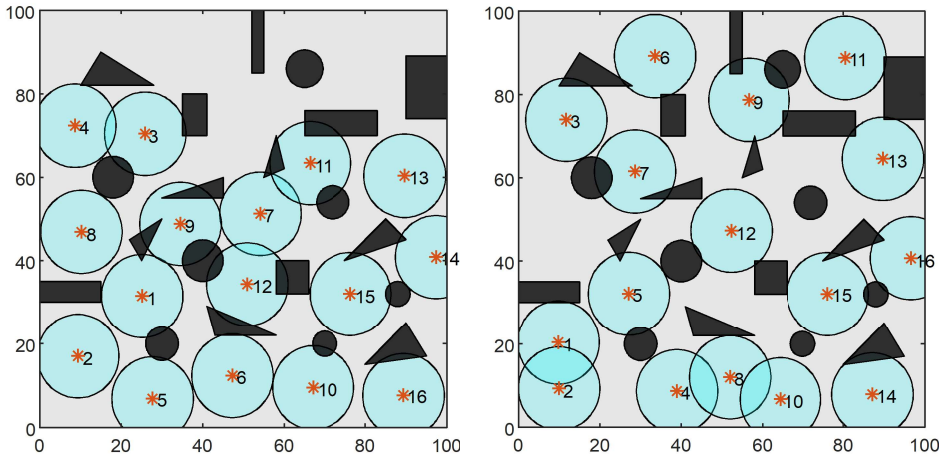


Fig. 12. Effects of the step size ⌊ based on Problem 3. The coverage rate and the first term of uniform degree are based on the left column axis, and the average path length refers to the right column axis.

6.3 Repulsive force range between agents

The effective range for generating repulsive force between two agents d_{ath} should be at least $2R_m$ to decrease the overlap between the monitoring zones of the agents. When it is fixed at $2R_m$, the agents' distribution is not very even across the whole target area, although the coverage rate is high, where an example relating to Problem 9 is shown in Fig. 13(a). We defined a coefficient C_a as $\sqrt{N_{su}/N_a}$ and used it in setting the d_{ath} value. C_a is greater than 1 when N_a is less than N_{su}. The uniform degree of agents' final distribution is better when d_{ath} increases from $2R_m$ to $C_a \times 2R_m$, as Fig. 13 and Table 8 show, though the coverage rate slightly decreases. To take a balance between coverage and distribution performance, we set d_{ath} randomly vary as either $2R_m$ or $C_a \times 2R_m$ and this achieved the most desirable overall performance compared to the fixed d_{ath} settings.



(a) $d_{\text{ath}} = 2R_m$ (b) $d_{\text{ath}} = C_a * 2R_m$

Fig. 13. Effects of repulsive force range d_{ath} between agents.

Table 8

Experiments on Effective Range of Repulsive Force.

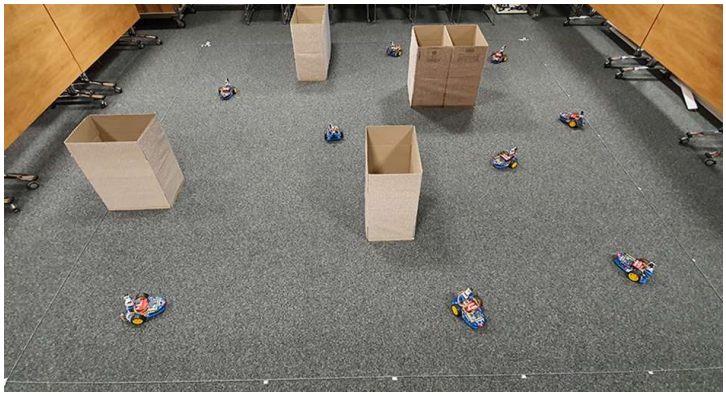
d_{ath}	Average Coverage	Coverage Rate (%)	Uniform Degree	Average Path Length
$2R_m$	4526.6±107.5	51.4±1.2	[6.85, 1.21]	62.8±3.1
$C_a \times 2R_n$	4504.8±155.2	51.2±1.8	[7.66, 1.14]	80.6±4.8
Varied	4598.1±67.2	52.3±0.8	[7.64, 1.21]	80.5±3.8

7. Experiments using mobile robots

We also conducted practical experiments to validate the proposed VFIS approach using nine AlphaBot robots. Each robot was equipped with two micro-controllers: an Arduino board was used to accomplish 1) localization based on odometers and inertial measurement units, 2) environmental perception through an ultrasonic range finder with the 360-degree rotation, and 3) movement control; a Raspberry Pi was used for communication and the calculation of position update based on the proposed algorithms through MATLAB Support Package for Raspberry Pi Hardware. The experimental environment was a 4m by 4m square and several cardboard boxes were used as obstacles, whose area is about 5.5% of the target area. The surface of obstacles was roughened for better reflections of ultrasonic signals, and ultrasonic sensors were set a 30-degree upward tilt angle for better detection of obstacles without the influence of other robots.

We set the stage number as 10, the coverage distance as 70 cm, the perception distance as 120 cm and the communication distance as 210 cm. The value of the coefficient W_m was doubled in (11) and (12) considering a relatively large map area and relatively small obstacles. At the beginning of the experiment, each robot was located along the edge of the target environment, as shown in Fig. 14(a). The experiment was repeated five times and its performance was compared with the results in the MATLAB simulation with the same configuration and parameter settings.

(a) Initial position of the robots



(b) Three-stage movement

(c) Final position of the robots

Fig. 14. Experiments based on nine AlphaBot robots.

Fig. 14(b-c) illustrates examples of multi-robot deployment after three-stage movement and ten-stage movement. Fig. 15 presents the agents' final deployment in the comparative simulation experiment. Table 9 illustrates the comparison of all three performance indices between the AlphaBot experiments and the simulation experiments. We can find that the coverage rate and uniform degree are similar between these two sets of experiments. In the AlphaBot experiments, the coverage rate is slightly smaller and the uniform degree is even better. It is noticeable that the average path length in the AlphaBot experiments is much greater. This is caused by inaccurate obstacle detection from ultrasonic sensors. Due to the ultrasonic sensors' beam pattern, each robot might detect the obstacles' border differently at different positions. This would cause a frequent update of obstacles' locations and more movement of agents according to the updated perception of surroundings.

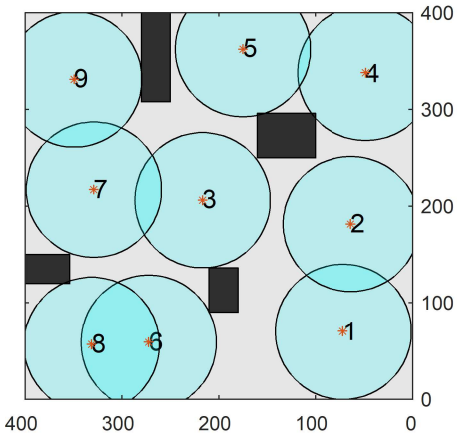


Fig. 15. Final deployment of the control experiment in MATLAB simulation

Table 9

Coverage Performance of Comparative Experiments.

Comparative Experiments	Average Coverage	Coverage Rate(%)	Uniform Degree	Average Path Length
AphaBot	113628±4603	75.2±3.0	[6.57,0.82]	410.0±24.7
Simulation	116871±4396	77.3±2.9	[6.32,0.62]	286.7±20.4

8. Conclusion

In this work, we have addressed the problem of applying a large number of autonomous agents for monitoring complex unknown environments and designed a series of benchmark testing problems for the monitoring coverage task. In this paper, we developed a multi-stage virtual force interaction scheme, where the virtual vortex forces were introduced to collaborate with repulsive forces to enhance the global exploration ability of agents. Our method has been investigated in simulation according to the benchmark configurations and the results revealed that each agent could achieve the good deployment with a desired coverage rate and uniform distribution. We have also investigated the key parameters and provided guidelines on their settings. In addition, some experiments using Alphabots were conducted and the effectiveness of VFIS was successfully verified. Future works will focus on the convergence analysis, the extension to the dynamic 2D environment and 3D network scenarios with UAVs.

Reference

[1] K. Senthilkumar, K.K. Bharadwaj, Multi-robot exploration and terrain coverage in an unknown environment, Robotics and Autonomous Systems, 60 (2012) 123-132.
[2] T.-M. Liu, D.M. Lyons, Leveraging area bounds information for autonomous decentralized multi-robot exploration, Robotics and Autonomous Systems, 74 (2015) 66-78.
[3] R. Calvo, J.R. de Oliveira, M. Figueiredo, R. Romero, Inverse aco applied for exploration and surveillance in unknown environments, in: COGNITIVE 2011, the Third International Conference on Advanced Cognitive Technologies and Applications, Citeseer, 2011.

[4] D.A. Lima, C.R. Tinoco, J.M. Viedman, G.M. Oliveira, Coordination, Synchronization and Localization Investigations in a Parallel Intelligent Robot Cellular Automata Model that Performs Foraging Task, in: ICAART (2), 2017, pp. 355-363.
[5] G. Sun, R. Zhou, B. Di, Z. Dong, Y. Wang, A novel cooperative path planning for multi-robot persistent coverage with obstacles and coverage period constraints, Sensors, 19 (2019) 1994.
[6] C.R. Tinoco, D.A. Lima, G.M. Oliveira, An improved model for swarm robotics in surveillance based on cellular automata and repulsive pheromone with discrete diffusion, International Journal of Parallel, Emergent and Distributed Systems, 34 (2019) 53-77.
[7] Q. Tang, Z. Xu, F. Yu, Z. Zhang, J. Zhang, Dynamic target searching and tracking with swarm robots based on stigmergy mechanism, Robotics and Autonomous Systems, 120 (2019) 103251.
[8] J.M. Palacios-Gasós, E. Montijano, C. Sagues, S. Llorente, Multi-robot persistent coverage using branch and bound, in: 2016 American Control Conference (ACC), IEEE, 2016, pp. 5697-5702.
[9] L. Siligardi, J. Panerati, M. Kaufmann, M. Minelli, C. Ghedini, G. Beltrame, L. Sabattini, Robust area coverage with connectivity maintenance, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 2202-2208.
[10] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider, Coordinated multi-robot exploration, IEEE Transactions on robotics, 21 (2005) 376-386.
[11] S.L. Smith, M. Schwager, D. Rus, Persistent robotic tasks: Monitoring and sweeping in changing environments, IEEE Transactions on Robotics, 28 (2011) 410-426.
[12] Y. Zou, K. Chakrabarty, Sensor deployment and target localization based on virtual forces, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, IEEE, 2003, pp. 1293-1303.
[13] K. Ji, Q. Zhang, H. Cheng, D. Yu, A Virtual Force Interaction Scheme for Monitoring Complex Unknown Environments by Autonomous Mobile Robots, in: 2019 18th European Control Conference (ECC), IEEE, 2019, pp. 95-100.
[14] M.R. Senouci, A. Mellouk, K. Asnoune, F.Y. Bouhidel, Movement-assisted sensor deployment algorithms: A survey and taxonomy, IEEE Communications Surveys & Tutorials, 17 (2015) 2493-2510.
[15] Y. Chen, H. Zhang, M. Xu, The coverage problem in UAV network: A survey, in: Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on, IEEE, 2014, pp. 1-5.
[16] X. Wang, S. Wang, D. Bi, Virtual force-directed particle swarm optimization for dynamic deployment in wireless sensor networks, in: International Conference on Intelligent Computing, Springer, 2007, pp. 292-303.
[17] X. Yu, W. Huang, J. Lan, X. Qian, A novel virtual force approach for node deployment in wireless sensor network, in: Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on, IEEE, 2012, pp. 359-363.
[18] G. Sallam, U. Baroudi, M. Al-Shaboti, Multi-Robot Deployment Using a Virtual Force Approach: Challenges and Guidelines, Electronics, 5 (2016) 34.
[19] N. Boufares, I. Khoufi, P. Minet, L. Saidane, Y.B. Saied, Three dimensional mobile wireless sensor networks redeployment based on virtual forces, in: 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2015, pp. 563-568.
[20] M. Rout, R. Roy, Dynamic deployment of randomly deployed mobile sensor nodes in the presence of obstacles, Ad Hoc Networks, 46 (2016) 12-22.
[21] J. Xie, D. Wei, S. Huang, X. Bu, A Sensor Deployment Approach Using Improved Virtual Force Algorithm Based on Area Intensity for Multisensor Networks, Mathematical Problems in Engineering, 2019 (2019).
[22] X. Yu, N. Liu, X. Qian, T. Zhang, A deployment method based on spring force in wireless robot sensor networks, International Journal of Advanced Robotic Systems, 11 (2014) 79.
[23] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, IEEE Transactions on robotics and Automation, 20 (2004) 243-255.
[24] G. Wang, G. Cao, T.F. La Porta, Movement-assisted sensor deployment, IEEE Transactions on Mobile Computing, 5 (2006) 640-652.
[25] H. Mahboubi, K. Moezzi, A.G. Aghdam, K. Sayrafian-Pour, V. Marbukh, Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors, IEEE Transactions on Industrial Informatics, 10 (2013) 163-174.
[26] F. Sharifi, A. Chamseddine, H. Mahboubi, Y. Zhang, A.G. Aghdam, A distributed deployment strategy for a network of cooperative autonomous vehicles, IEEE transactions on control systems technology, 23 (2015) 737-745.
[27] F. Abbasi, A. Mesbahi, J. Mohammadpour, Team-based coverage control of moving sensor networks, in: 2016 American Control Conference (ACC), IEEE, 2016, pp. 5691-5696.
[28] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, D. Rus, Voronoi coverage of non-convex environments with a group of networked robots, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 4982-4989.
[29] Y. Stergiopoulos, M. Thanou, A. Tzes, Distributed collaborative coverage-control schemes for non-convex domains, IEEE Transactions on Automatic Control, 60 (2015) 2422-2427.
[30] E. Teruel, R. Aragues, G. López-Nicolás, A distributed robot swarm control for dynamic region coverage, Robotics and Autonomous Systems, 119 (2019) 51-63.
[31] S. Papatheodorou, A. Tzes, Y. Stergiopoulos, Collaborative visual area coverage, Robotics and Autonomous Systems, 92 (2017) 126-138.

[32] M. Saska, V. Vonásek, J. Chudoba, J. Thomas, G. Loianno, V. Kumar, Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles, Journal of Intelligent & Robotic Systems, 84 (2016) 469-492.
[33] S. Spanogianopoulos, Q. Zhang, S. Spurgeon, Fast Formation of Swarm of UAVs in Congested Urban Environment, IFAC-PapersOnLine, 50 (2017) 8031-8036.
[34] B. Ranjbar-Sahraei, G. Weiss, A. Nakisaee, Stigmergic coverage algorithm for multi-robot systems, in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1497-1498.
[35] C. Ozturk, D. Karaboga, B. Gorkemli, Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm, Sensors, 11 (2011) 6056-6065.
[36] D. Reina, H. Tawfik, S. Toral, Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks, Ad Hoc Networks, 68 (2018) 16-32.
[37] O. Banimelhem, M. Naserllah, A. Abu-Hantash, An efficient coverage in wireless sensor networks using fuzzy logic-based control for the mobile node movement, in: 2017 Advances in Wireless and Optical Communications (RTUWO), IEEE, 2017, pp. 239-244.
[38] S. Rahili, J. Lu, W. Ren, U.M. Al-Saggaf, Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory: Algorithms and Experiments, IEEE Transactions on Mobile Computing, 17 (2017) 1303-1313.
[39] A. Renzaglia, L. Doitsidis, A. Martinelli, E.B. Kosmatopoulos, Multi-robot three-dimensional coverage of unknown areas, The International Journal of Robotics Research, 31 (2012) 738-752.