

# Energy-Efficient Coverage Enhancement Strategy for 3-D Wireless Sensor Networks Based on a Vampire Bat Optimizer

Xiao-Qiang Zhao, Yan-Peng Cui<sup>ID</sup>, Chuan-Yi Gao, Zheng Guo, and Qiang Gao

**Abstract**—As one of the crucial scenarios of the Internet of Things, wireless sensor networks (WSNs) perform tasks invariably on the basis of the effective cover effect of targets. The cover effect is improved commonly by redeployment of the sensor nodes since they tend to deviate from the optimal deployment location frequently. Given the complex and harsh environment of 3-D WSNs, which impede the recharging and restoration of the batteries of the sensor nodes, the intention of our research is to deal with the multiobjective optimization problem of minimizing total energy consumption and balancing the residual energy based on coverage enhancement during redeployment. By using truncated octahedrons to stack the 3-D environment seamlessly, the coverage enhancement and energy optimization problems are transformed into a task-assignment problem of moving nodes to truncated octahedrons, and an energy-efficient coverage enhancement strategy based on the vampire bat optimizer (VBO) is proposed to solve the above problem, which is inspired by the egoistic and altruistic behavior of vampire bats. The simulation results demonstrate that the proposed strategy can enhance the uniformity of residual energy of nodes by 30.53%, 43.44%, and 32.03% compared to the virtual force-directed particle swarm optimization (VFPSO), the 3-D virtual force algorithm (3DVFA), and the Hungarian algorithm (HA), respectively. Additionally, the proposed strategy can reduce the total energy consumption of nodes and also perform well in terms of energy consumption of the maximum energy consumption nodes, final coverage rate, and time consumption.

**Index Terms**—Coverage enhancement, energy consumption optimization, task assignment, 3-D wireless sensor networks (WSNs), vampire bat optimizer (VBO).

Manuscript received October 12, 2019; accepted November 6, 2019. Date of publication November 11, 2019; date of current version January 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61901370, in part by the Science and Technology Innovation Team for Talent Promotion Plan of Shaanxi Province under Grant 2019TD-028, in part by the Special Planned Project for Serving Local Areas of Education Department of Shaanxi Provincial Government under Grant 18JC029, in part by the Science and Technology Program of Xi'an under Grant 201806117YF05NC13-2, and in part by the International Science and technology cooperation project of Shaanxi Provincial Science and Technology Department under Grant 2018KW-025. (Corresponding author: Yan-Peng Cui.)

X.-Q. Zhao is with the School of Communication and Information and the Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710121 (e-mail: zxq7703@126.com).

Y.-P. Cui, C.-Y. Gao, and Z. Guo are with the School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China (e-mail: cuiyanpeng94@126.com; chuanygao@163.com; gz95428@126.com).

Q. Gao is with the College of Mechanical and Electronic Engineering, Northwest Agriculture and Forestry University, Yangling 712100, China (e-mail: hillfinder@163.com).

Digital Object Identifier 10.1109/JIOT.2019.2952718

## I. INTRODUCTION AND RELATED WORK

THE Internet of Things (IoT) paradigm sees computers capable of accessing data about objects and the environment without human interaction, and wireless sensor networks (WSNs) have traditionally been considered its key enablers, since they bring richer capabilities of both sensing and actuation [1]. Three-dimensional WSNs have gradually become a research focus due to their wide application in tactical surveillance, seismic monitoring, auxiliary navigation, pollution monitoring, and other fields [2].

Effective monitoring of targets is the basis for 3-D WSNs to perform tasks [3]. To enhance their monitoring quality and improve network reliability, it is necessary to ensure that the target area can be effectively covered by the sensor nodes [4]. However, considering the limitations of the complex, harsh environment of 3-D WSNs, underwater WSNs (UWSNs), for instance, the initial layout is usually formed randomly, and then the mobile nodes are redeployed to enhance the cover effect. For example, the traditional deployment solution of underwater 3-D WSNs is to use surface buoys, but it is vulnerable to weather and tampering. Hence, the seabed is used to anchor the sensor nodes whose depth is controlled by the wires linked to these anchors [5].

A harsh, remote application area and little human intervention preclude the recharging or replacement of the batteries of the sensor nodes. Consequently, efficient energy consumption of nodes can extend the network lifetime and is a prime design issue for wireless sensor networks [6]. Energy consumption during redeployment mainly results from moving and signal transmission, and the former is of greater importance than the latter for mobile nodes [7]. Therefore, how to reduce the total mobile energy consumption and balance the residual energy of nodes while reaching full coverage effect with a minimum number of nodes is an urgent problem to be solved in regard to a 3-D WSN redeployment strategy.

As the most primitive and effective coverage enhancement algorithm, the virtual force algorithm (VFA) abstracts the sensor nodes into particles in the potential field [8]. Each sensor node is guided by virtual gravitation and repulsive force to perform virtual movement. After the coverage rate reaches the threshold, the node locks the moving destination and moves linearly toward it to enhance the coverage effect. A large number of algorithms [9]–[11] have been derived from VFA. By setting the maximum boundary

coordinates, introducing an effective communication distance, and constraining the maximum step size, improved VFA and exponential VFA were proposed in [9] to reduce the sensitivity of virtual force coefficients and speed up the convergence. By taking the existence of different obstacles and the problem of node's adaptive movement in the network into count, three improved algorithms based on virtual force were proposed in [10]. To solve the deployment problem of 3-D environment, a distributed deployment algorithm based on the 3-D virtual force algorithm (3DVFA) [11] adjusts the parameters of traditional VFA to adapt to the 3-D environment, ensuring the full coverage and network connection effect of the 3-D space.

Evolutionary computation is a subfield of artificial intelligence that is usually used for solving the optimization problems. Swarm intelligence is a branch of algorithms that is based on the self-organization and division of labor among insects [12]. A number of surveys have exploited particle swarm optimization (PSO) to enhance coverage. In [13], a sensing radius adaptive coverage control algorithm based on PSO was proposed, and the sensing radius of each sensor is adjusted according to the coverage rate and energy consumption of each node, which can effectively improve the coverage rate and reduce energy consumption. A virtual force-directed PSO (VFPSO) algorithm combines VFA with PSO [14], where the moving speed of each particle depends not only on the historical optimal position of the individual and the current optimal position of the global but also on the virtual force of the particle. The virtual force-Lévy-embedded grey wolf optimization algorithm is proposed in [15] to obtain a higher coverage rate of the monitoring area, a more uniform sensor distribution, and a shorter average moving distance of the wireless sensors. Dao *et al.* [16] proposed a hybrid Taguchi-cuckoo search to be applied to compatible focus positioning platforms, and to minimize energy consumption. A nature-inspired evolutionary algorithm, the cuckoo search algorithm, was proposed in [17] to optimally locate the sensors. An improved grey wolf optimizer was proposed to efficiently utilize the energy of the sensor nodes and extend the network lifetime maximally [18].

The coverage enhancement problem can be converted to a maximum matching problem of the weighted bipartite graph to reduce the total energy consumption of each node [7], and it has been proved [19] to be NP-complete problem whose optimal solution can be obtained by the extended Hungarian algorithm (HA). The minimum-cost perfect matching of a weighted complete bipartite graph was adopted to select the optimal sinking nodes to ideal pattern locations [20], and a coverage hole and repairing algorithm based on the 3-D Voronoi diagram and a clustering algorithm was proposed to further sinks the remaining nodes. Aiming at the problem of sensor node deployment of underwater WSNs, the double-coverage algorithm was proposed [21] to solve the premature failure caused by excessive energy consumption.

The crucial contributions of our research are as follows.

- 1) A strategy for seamlessly stacking 3-D spaces of a given size by truncated octahedrons of a given radius is proposed.

TABLE I  
ABBREVIATIONS USED IN THIS ARTICLE

| Abbreviations | Explanation  |
|---------------|--|
| WSNs          | Wireless sensor networks                           |
| UWSNs         | Underwater wireless sensor networks                |
| VBO           | Vampire bat optimizer                              |
| VFA           | Virtual force algorithm                            |
| 3DVFA         | Three-dimensional virtual force algorithm          |
| PSO           | Particle swarm optimization                        |
| VFPSO         | Virtual force-directed particle swarm optimization |
| HA            | Hungarian algorithm                                |
| IoT           | Internet of things                                 |
| NPC           | Non-deterministic polynomial complete              |
| NPH           | Non-deterministic polynomial hard                  |

TABLE II  
KEY PARAMETERS USED IN THIS ARTICLE

| Parameters  | Explanation   |
|-------------|---|
| $S_i$       | The $i$ -th sensor node with coordinates $(x_{S_i}, y_{S_i}, z_{S_i})$ .                      |
| $G_j$       | The centroid of $j$ -th truncated octahedron with coordinates $(x_{G_j}, y_{G_j}, z_{G_j})$ . |
| $d_{i,j}$   | The Euclidean distance between $S_i$ and $G_j$ .  |
| $R$         | The perceived radius of node.   |
| $E_{i,j}$   | The residual energy of $S_i$ after moving to $G_j$ .  |
| $b_i, p_j$  | The $i$ -th vampire bat and the $j$ -th prey.   |
| $I_{i,j}^t$ | The interest value of $b_i$ in $p_j$ on the $t$ -th iteration.                                |
| $g_i^t$     | The gene value of $b_i$ on the $t$ -th iteration.   |
| $r_j^t$     | The hunting risk value for $p_j$ on the $t$ -th iteration.                                    |
| $z_i$       | The blood volume taken by $b_i$ .   |
| $B_{i,j}^t$ | The benefit of $b_i$ when capturing $p_j$ on the $t$ -th iteration.                           |
| $D_{i,j}^t$ | The benefits of $S_i$ when moving to $G_j$ on the $t$ -th iteration.                          |

- 2) The coverage-enhancement and energy-optimization problems are transformed into the task-assignment problem of moving nodes to the centroids of truncated octahedrons.
- 3) An energy-optimization algorithm based on the vampire bat optimizer (VBO) is proposed. The coverage enhancement problem and the multiobjective optimization problem of minimizing total energy consumption and balancing residual energy of nodes are solved.
- 4) The performance of the proposed strategy is simulated and compared with VFPSO, HA, and 3DVFA in terms of final coverage, total energy consumption, residual energy balance, etc. The reasons for the performance difference are analyzed.

The organization of this article is as follows. In Section II, the 3-D coverage model, coverage enhancement problem, and energy consumption during coverage enhancement are described. We propose VBO by the egoistic and altruistic behavior of vampire bats in Section III, and the energy consumption of the sensor nodes during redeployment is optimized by VBO. Simulation analysis of VBO, 3DVFA, VFPSO, and HA is given in Section IV. Explanations of the reasons that cause the performance differences in Section IV are discussed in Section V. Finally, we conclude this article in Section VI. In addition, the abbreviations and key parameters used in this article are listed in Tables I and II, respectively.

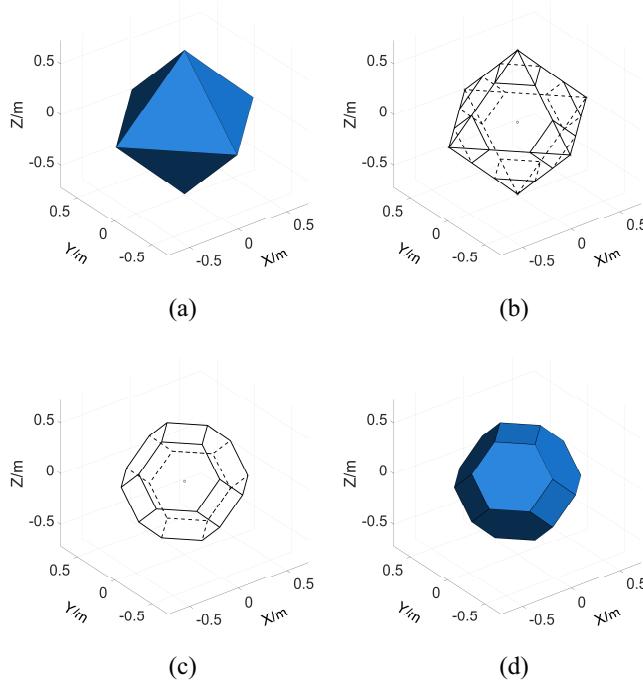


Fig. 1. Process of cutting six vertices of a regular octahedron. (a) Regular octahedron. (b) Perspective drawing of cutting six vertices of the octahedron shown in (a). (c) Perspective drawing after cutting six vertices of (b). (d) Stereogram of (c).

## II. PROBLEM STATEMENT

### A. Three-Dimensional Coverage Model of Sensor Nodes

The 3-D monitoring area can be represented by a 3-D space containing  $K$  grids. Each sensor has a spherical range in the 3-D space; hence, the spherical center represents the sensor node, and the radius represents the sensing range of the node. For a 3-D space  $\Omega$  with  $K$  grids,  $G_j$  can be detected by  $S_i$  when the condition  $d_{i,j} \leq R$  is satisfied, where  $S_i$  is the  $i$ th sensor node with coordinates  $(x_{S_i}, y_{S_i}, z_{S_i})$ ,  $G_j$  is the centroid of the  $j$ th grid with coordinates  $(x_{G_j}, y_{G_j}, z_{G_j})$ ,  $R$  is the perceived radius of the sensor node, and  $d_{i,j}$  is the Euclidean distance between  $S_i$  and  $G_j$ .

### B. Coverage Enhancement Problem

Enhancing the cover effect with the fewest nodes is one of the main research directions in coverage control of 3-D WSNs. Assuming that the sensor nodes can acquire their location information and all sensor nodes have the same perceived radius  $R$ , and ignoring the path loss, end-to-end delay, and data packet dropout rate of acoustic channel caused by the complexity of water environment temporarily, a strategy can be defined as the most efficient strategy of coverage enhancement if it requires a minimum number of nodes when achieving a full cover effect finally, which is equivalent to finding a polyhedron with the largest volumetric quotient [22]. The larger the volumetric quotient is, the less redundant coverage the sensor nodes will have when deployed on the polyhedron.

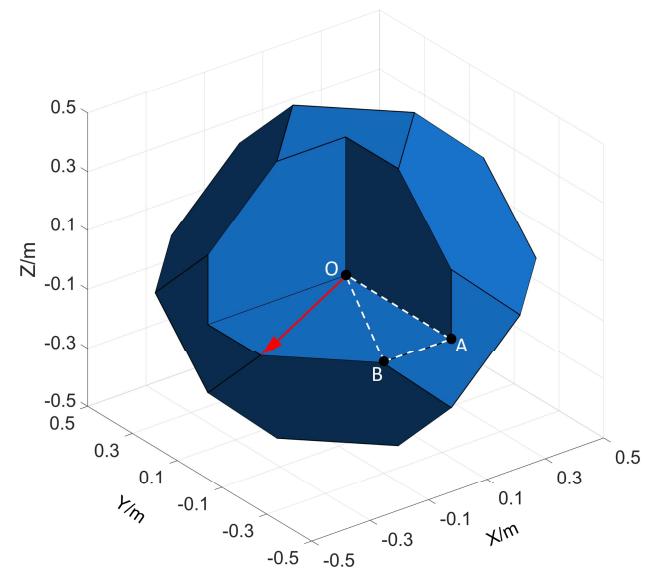


Fig. 2. Stacking interval of truncated octahedrons.

The common regular polyhedrons that can stack  $\Omega$  seamlessly are cubes, hexagonal prisms, and truncated octahedrons. It has been confirmed that the volumetric quotient of a truncated octahedron is the highest in a common regular polyhedron [23]. As the name implies, a truncated octahedron is a polyhedron obtained by cutting six vertices of a regular octahedron, and the cutting process is shown in Fig. 1.

The perceived radius of the sensor nodes, as shown by the red arrow in Fig. 2, is regarded as the radius of the circumscribed sphere of the truncated octahedron. Considering that there is a geometric relationship of  $|\overrightarrow{OA}| = 2\sqrt{5}|\overrightarrow{OB}|/5$  according to the cutting process and geometric relationship shown in Fig. 1, which means that the stacking interval  $\delta$ , as twice the length of  $|\overrightarrow{OA}|$ , satisfies the relationship of  $\delta = 4\sqrt{5}|\overrightarrow{OB}|/5$ .

Consequently, all truncated octahedrons that used to stack  $\Omega$  seamlessly can be classified into two categories according to the positional relationship.

Assuming that the length, width, and height of  $\Omega$  are  $L_1$ ,  $L_2$ , and  $L_3$ , and regarding  $\delta = 4\sqrt{5}R/5$  as the stacking interval, and the centroid  $(0, 0, 0)$  and  $(2\sqrt{5}R/5, 2\sqrt{5}R/5, 2\sqrt{5}R/5)$  as the baseline for the first and second categories of truncated octahedrons, respectively, the centroid coordinates of the first type of truncated octahedrons can be expressed as

$$\begin{cases} (n_1\delta, n_2\delta, n_3\delta) \\ n_i \in \{1, 2, \dots, N_i\}, N_i = \lceil L_i/\delta + 1 \rceil, i = 1, 2, 3 \end{cases} \quad (1)$$

and the centroid coordinates of the second type of truncated octahedrons can be expressed as

$$\begin{cases} (2\sqrt{5}R/5 + m_1\delta, 2\sqrt{5}R/5 + m_2\delta, 2\sqrt{5}R/5 + m_3\delta) \\ m_j \in \{1, 2, \dots, M_j\}, M_j = \lceil L_j/\delta \rceil, j = 1, 2, 3 \end{cases} \quad (2)$$

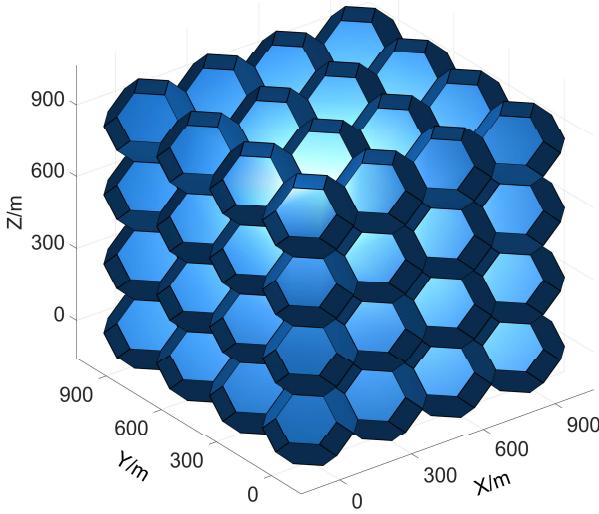


Fig. 3. Stacking effect. Seamlessly stacking the 3-D space with 91 truncated octahedrons. The size of the 3-D space is 900 ft  $\times$  900 ft  $\times$  900 ft, and the radius of the circumscribed sphere of each truncated octahedron is 170 ft.

Therefore, the minimum number of truncated octahedrons required to stack  $\Omega$  seamlessly can be calculated by

$$\begin{cases} N_{\min} = \prod_{i=1,2,3} N_i + \prod_{j=1,2,3} M_j \\ N_i = \lceil \sqrt{5}L_i/4R + 1 \rceil, M_j = \lceil \sqrt{5}L_j/4R \rceil. \end{cases} \quad (3)$$

The stacking effect is shown in Fig. 3. After stacking  $\Omega$  seamlessly using truncated octahedrons with outer sphere radius  $R$ , the strategy of moving the nodes to the centroid of the truncated octahedrons can maximize the final coverage rate while minimizing the number of nodes.

### C. Energy Consumption During Coverage Enhancement

1) *Moving Energy Consumption Model:* After moving to  $G_j$ , the formula for the residual energy  $E_{i,j}$  of node  $S_i$  is

$$E_{i,j} = E_{0i} - e \times d_{i,j} \quad (4)$$

where  $E_{0i}$  is the initial energy of  $S_i$ ,  $e$  is the energy consumption when  $S_i$  moves 1 ft, and  $d_{i,j}$  is the Euclidean distance between  $S_i$  and  $G_j$ .

The uniformity of residual energy of nodes is defined as

$$U = \sqrt{\frac{1}{N} \sum_{i=1}^N \left( E_{i,j} - \frac{1}{N} \sum_{i=1}^N E_{i,j} \right)^2} \quad (5)$$

where  $N$  is the number of nodes and  $E_{i,j}$  is the residual energy of  $S_i$  after moving to  $G_j$ .

Considering that the life cycle of WSNs is usually expressed by that of the node that dies first, the key to energy optimization during redeployment is to reduce the total moving energy consumption and balance the residual energy of nodes, which is equivalent to reduce and balance the moving distance from the nodes.

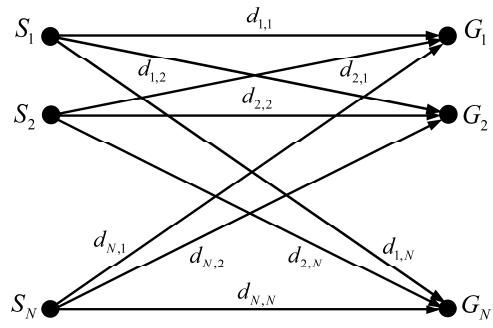


Fig. 4. Bipartite graph model for task-assignment problem.  $S_i$ ,  $G_j$ , and  $d_{i,j}$ , respectively, represent the node  $i$ , the centroids of truncated octahedron  $j$ , and the Euclidean distance between  $S_i$  and  $G_j$ .

2) *Optimization of Energy Consumption:* The redeployment problem can be transformed into a task-assignment problem that moves  $N$  nodes to  $N$  truncated octahedron centroids when it satisfied the condition described in (3). The bipartite graph model of the task-assignment problem is shown in Fig. 4, where  $d_{i,j}$  is the weight of the edge  $\langle S_i, G_j \rangle$  of the bipartite graph, and the distance matrix can be expressed as

$$d_{N \times N} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,N} \\ \vdots & \ddots & \vdots \\ d_{N,1} & \cdots & d_{N,N} \end{bmatrix}. \quad (6)$$

The objective function of the problem is defined as

$$\min(w_1 f_1 + w_2 f_2) \quad (7)$$

where  $f_1$  and  $f_2$  are the fitness functions that consider the total moving distance and the uniformity of moving distance; and  $w_1$  and  $w_2$  are the weights of the fitness functions  $f_1$  and  $f_2$ , respectively.  $f_1$  and  $f_2$  are expressed as

$$\begin{cases} f_1 = \sum_{i=1}^N \sum_{j=1}^N d_{i,j} x_{i,j} \\ f_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \left( d_{i,j} x_{i,j} - \frac{1}{N} f_1 \right)^2} \end{cases} \quad (8)$$

where the constraint conditions are defined as

$$\text{s.t. } \begin{cases} \sum_{i=1}^N x_{i,j} = 1, & j = 1, 2, \dots, N \\ \sum_{j=1}^N x_{i,j} = 1, & i = 1, 2, \dots, N \\ x_{i,j} = \begin{cases} 1, & G_j \text{ is the destination of } S_i \\ 0, & \text{otherwise.} \end{cases} \end{cases} \quad (9)$$

The maximum matching problem with weighted bipartite graphs has been proved to be NP-hard [24]. Although HA can solve it with small time complexity, it still takes a long time if the scale of the problem is large. Furthermore, for the conflicting multiobjective optimization problem described in (7), HA can only optimize  $f_1$  rather than the combination of  $f_1$  and  $f_2$ .

## III. PROPOSED ALGORITHM

### A. Vampire Bat Optimizer

Vampire bats are mammals that feed only on blood, and their feeding habits vary. For example, white-winged vampire bats and hairy-sucking bats feed on bird blood, while the

ordinary vampire bats feed on the blood of other mammals. Vampire bats derive their interest in the target prey based on their own taste and hunger level, the blood volume of the target prey, and the risk of hunting. Then, the vampire bat selects the target prey according to the degree of interest and participates in the predation competition, which is called the egoistic behavior of vampire bats.

However, not every bat can get enough blood due to predation competition, and a vampire bat will die if blood is not drawn for three consecutive nights [25], but few vampire bats will starve to death. Interestingly, almost all vampire bats prefer to share excess blood with a starving bat based on their kinship [26], which is known as altruistic behavior.

Inspired by the egoistic and altruistic behavior of vampire bats, a VBO is proposed to maximize the income of all vampire bats and balance their health status. Ultimately, VBO is used to optimize the energy consumption of nodes during redeployment. VBO has the following steps.

1) *Find the Best Prey for Each Bat:* The interest value of each vampire bat for each prey, the risk value of each prey, and the gene value of each vampire bat are initialized as follows. For convenience, regard the  $i$ th vampire bat and the  $j$ th prey as  $b_i$  and  $p_j$ , and regard the interest value of  $b_i$  in  $p_j$ , the gene value of  $b_i$ , and the hunting risk value for  $p_j$  on the  $t$ th iteration as  $I_{i,j}^t$ ,  $\varphi_i^t$ , and  $r_j^t$ , respectively.

Calculate the benefits of each bat for capturing each prey. For example, the benefits of  $b_i$  when capturing  $p_j$  on the  $t$ -th iteration can be calculated by

$$B_{i,j}^t = I_{i,j}^t - r_j^t. \quad (10)$$

Find the best prey for each vampire bat. For example, the best prey for  $b_i$  on the  $t$ -th iteration is defined by

$$p_{\text{best for } b_i}^t = \arg \max_{o \in \{1, 2, \dots, N\}} B_{i,o}^t \quad (11)$$

where  $B_{i,o}^t$  is the benefit of  $b_i$  when capturing  $p_o$  on the  $t$ -th iteration.

So far, each bat has found its favorite prey. Proceed to step 2) if there are predation conflicts that multiple bats rob the same prey. Otherwise, proceed to step 3) since the benefits of the entire bat population have been maximized, and the process in *finding the best prey for each bat* is called *the part I of VBO*.

2) *Predation Competition:* Predatory competition will not disappear and the benefits of all bats cannot be maximized until all bats have the best prey that no longer conflict.

In the event of a predation conflict, the prey robbed by multiple bats on the  $t$ -th iteration is denoted as  $\Phi_{\text{prey}}^t$ , and the bats involved in robbing preys in  $\Phi_{\text{prey}}^t$  are denoted as a set  $\Phi_{\text{bat}}^t$ . Then traverse all preys in  $\Phi_{\text{prey}}^t$  and the corresponding bats to update interest matrix  $I_{N \times N}^t$  and proceed back to step 1).

Taking  $p_\alpha^t$  in  $\Phi_{\text{prey}}^t$  as an example, the vampire bats involved in robbing  $p_\alpha^t$  are denoted as a set  $\Phi_{\text{bats for } p_\alpha^t}$ . Traverse all bats in  $\Phi_{\text{bats for } p_\alpha^t}$  to update the benefits that capturing  $p_\alpha^t$ . Taking  $b_i$  in  $\Phi_{\text{bats for } p_\alpha^t}$  as an example, update the interest value of it by

$$I_{i,\alpha}^{t+1} = I_{i,\alpha}^t - (\varphi_1^t - \varphi_2^t + \varepsilon) \quad (12)$$

TABLE III  
PSEUDOCODE OF PARTS I AND II OF VBO

| <b>Input:</b> |   |
|---------------|---|
| 1.            | Number of vampire bats and preys: $N$ .   |
| 2.            | Risk value of each prey: $\{r_1, r_2, \dots, r_N\}$ .   |
| 3.            | Interest value of each vampire bat in each prey: $\{I_{1,1}, I_{1,2}, \dots, I_{1,N}; I_{2,1}, I_{2,2}, \dots, I_{2,N}; I_{N,1}, I_{N,2}, \dots, I_{N,N}\}$ . |
| 4.            | Mandatory update factor: $\varepsilon$ .  |
| <b>Steps:</b> |   |
| 1.            | Set $t = 1$   |
| 2.            | <b>While</b> true   |
| 3.            | <b>For</b> $i$ from 1 to $N$  |
| 4.            | Set the best benefit of bat $i$ as $BB_i^t = 0$   |
| 5.            | <b>For</b> $j$ from 1 to $N$  |
| 6.            | Calculate the benefits of $i$ -th bat for capturing $j$ -th prey by $B_{i,j}^t = I_{i,j}^t - r_j^t$   |
| 7.            | <b>If</b> $B_{i,j}^t > BB_i^t$  |
| 8.            | $BB_i^t = B_{i,j}^t$  |
| 9.            | Set the $j$ -th prey as the best prey for $i$ -th bat   |
| 10.           | <b>end if</b>   |
| 11.           | <b>end for</b>  |
| 12.           | <b>end for</b>  |
| 13.           | <b>If</b> there is a predation conflict   |
| 14.           | Denote the prey robbed by multiple bats as $\Phi_{\text{prey}}^t$   |
| 15.           | <b>For</b> $\alpha$ -th prey in $\Phi_{\text{prey}}^t$  |
| 16.           | Denote the optimal and suboptimal benefits of bats in $\Phi_{\text{bats for } p_\alpha}^t$ to capture $p_\alpha$ as $\varphi_1^t$ and $\varphi_2^t$           |
| 17.           | <b>For</b> $i$ -th bat in $\Phi_{\text{bat}}^t$   |
| 18.           | Update the interest value of $i$ -th bat by $I_{i,\alpha}^{t+1} = I_{i,\alpha}^t - (\varphi_1^t - \varphi_2^t + \varepsilon)$                                 |
| 19.           | <b>end for</b>  |
| 20.           | <b>end for</b>  |
| 21.           | $t = t + 1$   |
| 22.           | <b>else</b>   |
| 23.           | <b>Break while</b>  |
| 24.           | <b>end if</b>   |
| 25.           | <b>end while</b>  |
| 26.           | Bats start to absorbing blood, and denote the volume of blood absorbed by each bat as $\{z_1, z_2, \dots, z_N\}$  |

where  $\varphi_1^t$  and  $\varphi_2^t$  are, respectively, the optimal and suboptimal benefits of the vampire bats in  $\Phi_{\text{bats for } p_\alpha}^t$  when robbing  $p_\alpha^t$ . The reason that setting  $\varepsilon$  as a mandatory update factor is that the updating process of  $I_{i,\alpha}^{t+1}$  will fail once  $\varphi_1^t$  and  $\varphi_2^t$  are equal.

Return to step 1) to recalculate the optimal moving destination of each sensor node. The process described in *predation competition* is called *part II of VBO*. The pseudocode of *part I and part II of VBO* is shown in Table III.

3) *Back Feeding:* So far, the benefits of the entire bat population have been maximized, and bats begin to absorb blood once the predation conflicts no longer occur.

Denote the amount of blood absorbed by each bat; for example, the blood volume taken by  $b_i$  is denoted as  $z_i$ . After blood absorption, all bats begin to seek a starving bat for back feeding. For example,  $b_i$  will transmit blood to  $b_j$  when the conditions

$$\begin{cases} z_i - z_j > \tau_1 \\ |g_i - g_j| < \tau_2 \end{cases}, \quad j \in \{1, 2, \dots, N\} \quad (13)$$

are satisfied, where  $\tau_1$  and  $\tau_2$  are, respectively, the threshold to measure the difference in hunger and the degree of

TABLE IV  
PSEUDOCODE OF PART III OF VBO.

|   |
|---|
| <b>Input:</b>   |
| 1. Number of vampire bats and preys: $N$ .  |
| 2. Threshold to measure the difference of hunger and the degree of genetic similarity among bats: $\tau_1$ and $\tau_2$ . |
| 3. Weights for degree of genetic similarity and the difference of hunger among bats: $w_1$ and $w_2$ .                    |
| 4. Blood volume taken by bats: $\{z_1, z_2, \dots, z_N\}$ .   |
| 5. Gene value of each vampire bat:  |
| <b>Steps:</b>   |
| 1. <b>While</b> true  |
| 2. <b>For</b> $i$ from 1 to $N$   |
| 3. <b>For</b> $j$ from 1 to $N$   |
| 4. <b>If</b> $z_i - z_j > \tau_1$ and $ g_i - g_j  < \tau_2$  |
| 5.         Set the $j$ -th bat as a bat that waiting the $i$ -th bat for feeding back                                     |
| 6. <b>end if</b>  |
| 7. <b>end for</b>   |
| 8.     Denote bats that waiting the $i$ -th bat for feeding back as $\Phi_{b_i}$  |
| 9. <b>For</b> $\zeta$ -th bat in $\Phi_{b_i}$   |
| 10.       Calculate the fitness value for $\zeta$ -th bat by  |
| $\text{fit}_{i,j} = \frac{e^{w_1 \times (z_i - z_j)}}{w_2 \times  g_i - g_j }$  |
| <b>end for</b>  |
| 11.       The $i$ -th bat feeds back the $o$ -th if $\text{fit}_{i,o}$ is the largest                                     |
| 12. <b>End for</b>  |
| 13.     If there are no bats need to feed back  |
| 14. <b>Break while</b>  |
| 15. <b>end if</b>   |
| 16. <b>end while</b>  |

genetic similarity between  $b_i$  and  $b_j$ . Formula (13) is called *the conditions for back feeding*.

If there are multiple bats which satisfy *the conditions for back feeding*, then  $b_j$  can be defined as the best transfusion target for  $b_i$  if  $b_j$  has the largest fitness value. The fitness value of  $b_j$  when transfused by  $b_i$  can be calculated by

$$\text{fit}_{i,j} = \frac{e^{w_1 \times (z_i - z_j)}}{w_2 \times |g_i - g_j|} \quad (14)$$

where  $w_1$  and  $w_2$  are, respectively, the weights for the difference of hunger and the degree of genetic similarity between  $b_i$  and  $b_j$ .

The process of back feeding will come to an end once no bats need the transfusion. The amount of blood drawn by each bat can be effectively balanced since the starving bats have been successfully transfused, and the process described in *back feeding* is called *part III of VBO*, whose pseudocode is shown in Table IV.

#### B. Optimize Energy Consumption of Nodes With VBO

For the task-assignment problem of moving nodes to centroids of the truncated octahedron, the nodes and truncated octahedrons are regarded as vampire bats and preys, respectively. VBO is used to solve the multiobjective optimization problem of minimizing total energy consumption and balancing residual energy of nodes. The process is as follows.

1) *Find the Best Destination for Each Sensor Node*: This procedure is corresponding to the calculation of benefits of bats when capturing preys, which is described in (10) of *part I of VBO*.

Calculate the benefits of the sensor nodes moving to different destinations. Given that *part II of VBO* is used to maximize the benefits of the entire bat population, which is opposite to the minimization of the total energy consumption of sensors during redeployment, hence, the distance matrix described in (6) is transformed by  $D'_{N \times N} = -d^t_{N \times N}$ . Accordingly, the opposite value of the distance from the sensor nodes to the truncated octahedrons is regarded as the benefit of bats when capturing preys. For example, the benefits of  $S_i$  when moving to  $G_j$  on the  $t$ -th iteration can be represented by  $D'_{i,j}$ . Determine the best moving destination for the sensor nodes. For example, the best moving destination for  $S_i$  on the  $t$ -th iteration can be determined by

$$G'_{\text{best for } S_i} = \arg \max_{j \in \{1, 2, \dots, N\}} D'_{i,j} \quad (15)$$

where  $D'_{i,j}$  is the benefits of  $S_i$  when moves to  $G_j$  on the  $t$ -th iteration. This procedure is aimed at finding the closest truncated octahedron for each node, which is equivalent to find the best prey for each vampire bat described in (11) of *part I of VBO*.

Calculate the moving task matrix by

$$\text{Task}'_{N \times N} = d^t_{N \times N} \odot X^t_{N \times N} \quad (16)$$

where  $\odot$  is the Hadamard product operator of a matrix, and  $X^t_{N \times N}$  is a mobile indicator matrix, where the meaning of  $x'_{i,j} \in X^t_{N \times N}$  is shown in (9).

So far, each sensor node has found its best moving destination. Proceed to step 2) if the best moving destination of the nodes conflict. Otherwise, proceed to step 3) since the benefits of all sensor nodes have been maximized and the process in *finding the best destination for each sensor node* is similar to *part I of VBO*.

2) *Competition*: The conflict phenomenon is common when nodes choosing the best moving destinations just as the best prey of each bat conflict frequently with each other. The process of competition for destinations of nodes is similar to *part II of VBO*.

In the event of a conflict, regard the conflicting destination as a popular destination and the conflicted sensors as active sensors, which are denoted as sets  $\Phi'_{\text{grids}}$  and  $\Phi'_{\text{sensors}}$ . Then traverse all popular destinations and the corresponding sensors to update the benefit matrix  $D'_{N \times N}$  and proceed back to step 1).

Taking the popular destination  $G'_\alpha$  in  $\Phi'_{\text{grids}}$  as an example, the sensor nodes involved in robbing  $G'_\alpha$  are denoted as a set  $\Phi'_{\text{sensors}}$  for  $G'_\alpha$ . Traverse all active nodes in  $\Phi'_{\text{sensors}}$  for  $G'_\alpha$  to update the benefits of sensors that moving to  $G'_\alpha$ . Taking the active node  $S_i$  in  $\Phi'_{\text{sensors}}$  for  $G'_\alpha$  as an example, update the benefit for  $S_i$  when moving to  $G'_\alpha$  by

$$D'^{+1}_{i,\alpha} = D'^t_{i,\alpha} - (D'^t_1 - D'^t_2 + \varepsilon) \quad (17)$$

where  $D'^t_1$  and  $D'^t_2$  are, respectively, the maximum and second maximum benefit of sensors in  $\Phi'_{\text{sensors}}$  for  $G'_\alpha$  when moving to  $G'_\alpha$ , the reason that setting  $\varepsilon$  as a mandatory update factor is that the process of updating  $D'^t_{i,\alpha}$  will fail once  $D'^t_1$  is equal to  $D'^t_2$ .

Return to step 1) to recalculate the optimal moving destination of each sensor node. Competition will not disappear

TABLE V  
SIMULATION PARAMETERS OF VBO

| Symbol        | Explanation                                     | Value     |
|---------------|---|-----------|
| $L_1$         | length of three-dimensional area                | 900 ft    |
| $L_2$         | width of three-dimensional area                 | 900 ft    |
| $L_3$         | height of three-dimensional area                | 900 ft    |
| $R$           | perceived radius of nodes                       | 170 ft    |
| $K$           | number of grids                                 | 9,261     |
| $N$           | number of nodes                                 | 91        |
| $\mu_1$       | weight of total moving distance                 | 0.001     |
| $\mu_2$       | weight of uniformity of moving distance         | 0.2       |
| $\varepsilon$ | updating factor                                 | 0.1       |
| $Sd$          | distance of single step of movement             | 18 ft     |
| $e$           | energy consumption by moving unit distance [27] | 2.52 J/ft |
| $Eo$          | initial energy of nodes                         | 3,000 J   |

until all sensor nodes have the best moving destination that no longer conflicts. The benefits of all sensor nodes can be maximized once there is no competition.

3) *Exchange the Moving Tasks*: Assuming that all sensor nodes move immediately according to the mobile tasks after step 2), the moving distances of each sensor node will be significantly different, just as the amount of blood absorbed by each bat was significantly different after the end of predatory competition, which leads to the phenomenon that the moving tasks of some sensor nodes will be immensely far.

After competition, all sensor nodes begin to seek other nodes for exchanging moving task. Specifically, exchange the moving tasks of nodes  $S_i$  and  $S_m$  according to *the theorem of task exchange*.

*Condition of Task Exchange*: For node  $S_i$  and its moving destination  $G_j$ , and node  $S_m$  and its moving destination  $G_n$ , the task exchange condition is

$$\begin{cases} d_{i,n} < d_{i,j} \\ d_{m,j} < d_{i,j} \end{cases} \quad (18)$$

which is equivalent to judging whether the difference in the degree of genetic similarity and the hunger degree between bats satisfy *the conditions for back feeding* described in (13).

*Lemma of Task Exchange*: If nodes  $S_i$  and  $S_m$  satisfy the *condition of task exchange*, then their moving distance can be balanced by task exchange, whose formula is

$$\begin{cases} \text{Task}_{i,n}^t = d_{i,n} \\ \text{Task}_{m,j}^t = d_{m,j} \\ \text{Task}_{i,j}^t = 0 \\ \text{Task}_{m,n}^t = 0. \end{cases} \quad (19)$$

*Theorem of Task Exchange*: According to *the lemma of task exchange*,  $K$  schemes can be obtained if there are  $K$  nodes satisfying the *condition of task exchange*. Define  $S_m$  as the optimal exchangeable node for  $S_i$  if the fitness value of  $S_m$  is the largest among  $K$  nodes, and exchange the task of  $S_i$  and  $S_m$  by (19). The fitness value of node  $S_m$  is

$$\text{fit}_m = \frac{e^{-\mu_1 \times (d_{i,n} + d_{m,j})}}{\mu_2 \times |d_{i,n} - d_{m,j}|} \quad (20)$$

where  $\mu_1$  and  $\mu_2$  are the respective weights of the sum and difference of the moving distance of  $S_i$  and  $S_m$  after

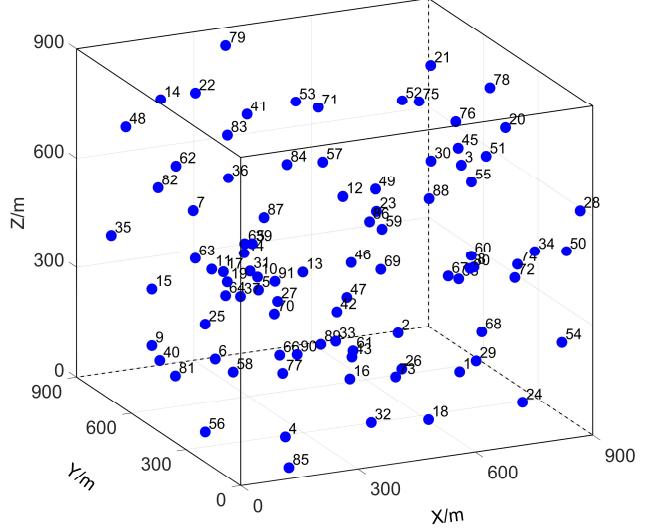


Fig. 5. Initial positions of 91 nodes of four algorithms.

exchanging moving task. The *theorem of task exchange* is equivalent to choose the optimal bat for feeding back, which is corresponding to *part III of VBO*.

The process of exchanging task will come to an end once no sensor nodes satisfied the *theorem of task exchange*. So far, the moving distance from the sensor nodes have been minimized and balanced, just as the amount of blood taken by bats has been maximized and balanced by VBO.

#### IV. SIMULATION RESULTS

##### A. Parameter Setting

The coverage enhancement and energy optimization of 3DVFA, VFPSO, HA, and VBO were compared under the same experimental conditions. The main parameters of VBO are shown in Table V.

##### B. Simulation Comparison

The initial positions of the sensor nodes set for VBO are shown in Fig. 5, which is also used for the other three algorithms for fairness of comparison. The moving trajectories, final positions of the sensor nodes, and final coverage effects of four algorithms are compared in Fig. 6, and it can intuitively show that VBO and HA are better than 3DVFA and VFPSO in terms of the final coverage effect and moving distance of the sensor nodes. Fig. 7 compares the three-view drawing of moving trajectories of four algorithms. Notably, VBO and HA have a shorter moving trajectory than 3DVFA and VFPSO, and the moving task of the node with a long moving distance in HA has been optimized in VBO, which means that VBO is better than HA when considering the maximum moving distance and the uniformity of the moving distance of nodes.

The final coverage rates of HA and VBO both reach 100% while VFPSO and 3DVFA only achieve 96.42% and 98.07% as shown in Fig. 8(a). Given that the final coverage of the

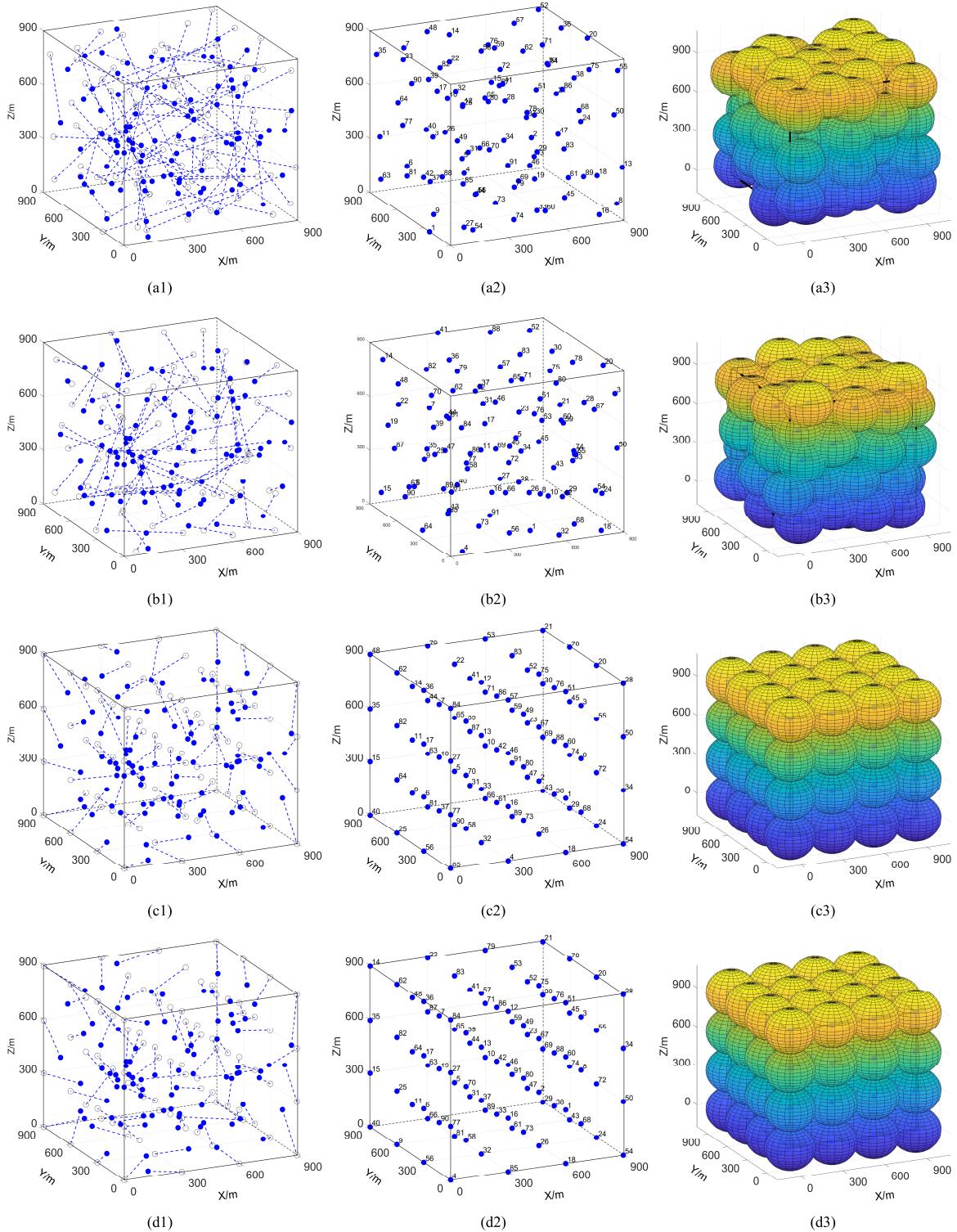


Fig. 6. Comparison of movement and final coverage effects of four algorithms. The 91 blue solid spots and the 91 hollow spots are the respective initial and the final positions of 91 sensor nodes, and the dotted lines connecting the solid and the hollow spots are the moving trajectories of nodes, and the spherical areas with the solid spot at the center are the sensing ranges of nodes. (a1)–(a3), (b1)–(b3), (c1)–(c3), and (d1)–(d3) are the moving trajectories, the final positions, and the final coverage effect of 91 nodes of 3DVFA, VFPSO, HA, and VBO, respectively, after 52 rounds of movement.

four algorithms are different, hence the total energy consumption of the four algorithms when they all reach 95% coverage rate is compared in Fig. 8(b). The total energy consumption of 3DVFA and VFPSO is  $9.1123 \times 10^4$  J and  $5.5642 \times 10^4$  J,

respectively, which are worse than that of VBO by 253.86% and 116.08%, and the former even consume more energy when the latter both reach 100% coverage rate. Unfortunately,  $4.1231 \times 10^4$  J of energy are consumed by VBO when reaches

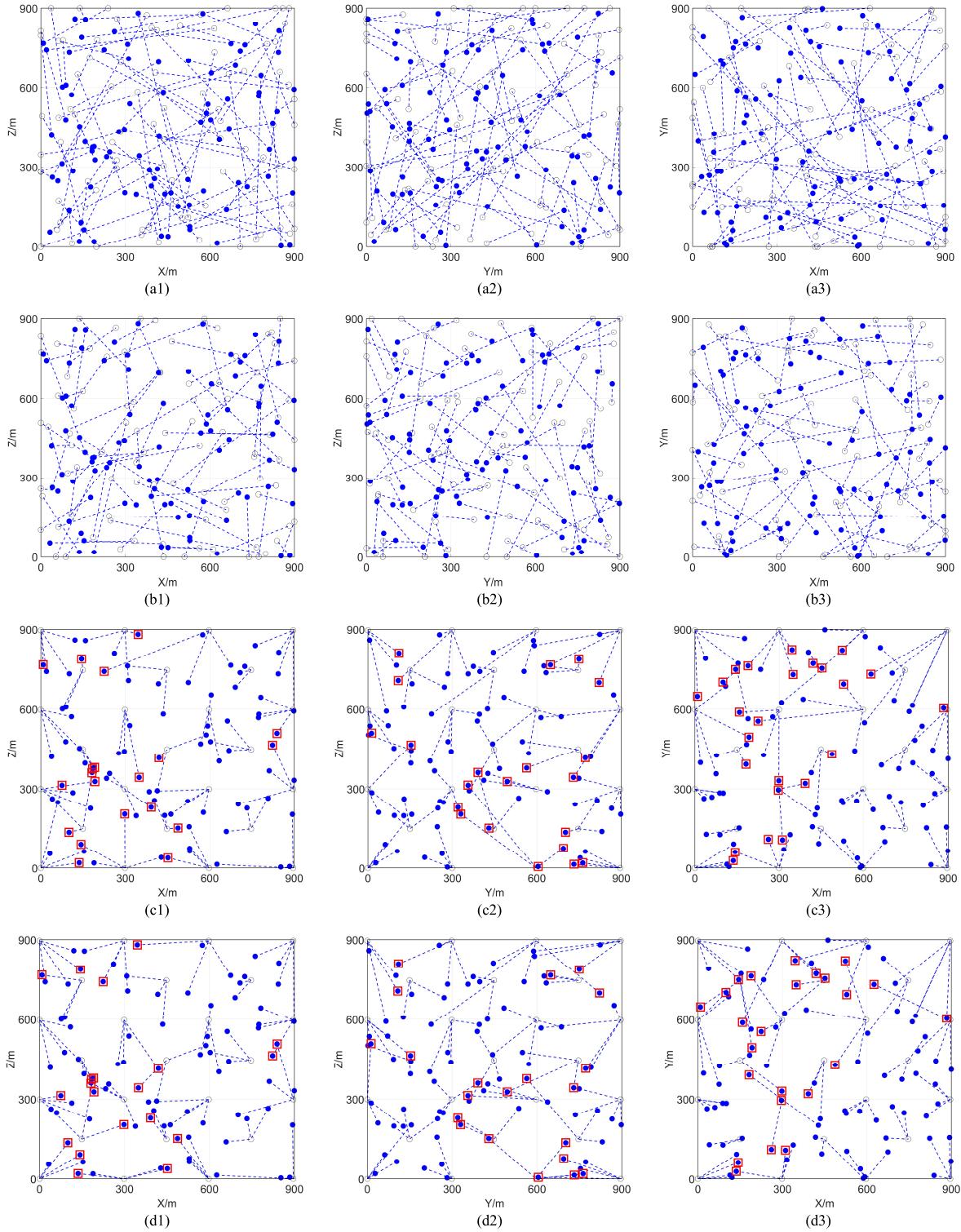
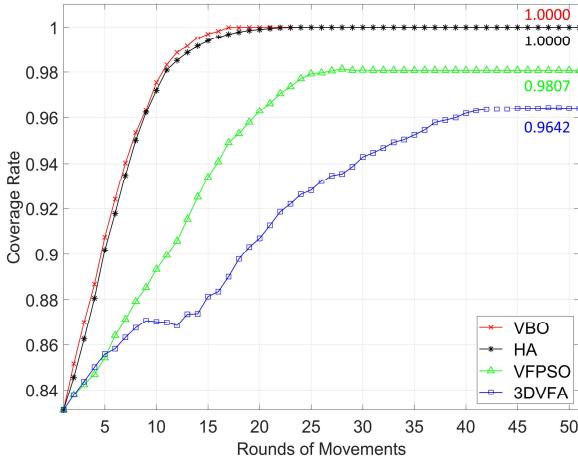


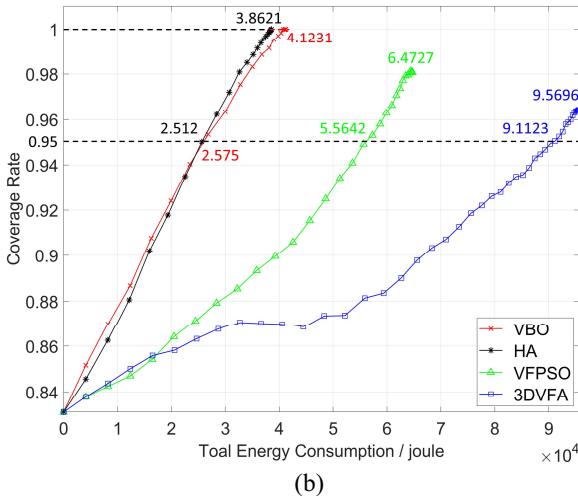
Fig. 7. Comparison of the three-view drawing of moving trajectories of four algorithms. The blue solid and the hollow spots are the respective initial and the final positions of each nodes, and the dotted lines connecting the solid and the hollow spots are the three-view drawing of moving trajectories of nodes. (a1)–(a3), (b1)–(b3), (c1)–(c3), and (d1)–(d3) are the respective front view, end view, and vertical view of the moving trajectories of 3DVFA, VFPSO, HA, and VBO. The moving tasks of nodes with long distance in HA are improved in VBO, which are marked by the red squares in (c1)–(c3) and (d1)–(d3).

full coverage, and it is 6.76% worse than HA who consumes  $3.8621 \times 10^4$  J of energy, as a result of that sacrificing the performance of the minimization of total energy consumption to balance the residual energy of nodes, which are conflicting

multiobjective optimization problems, and the specific reasons are discussed in Section V. As a matter of fact, VBO performs much better than HA in the following comparisons, such as uniformity of residual energy and the energy consumption of



(a)



(b)

Fig. 8. Comparison of coverage rates of four algorithms. (a) Relationship between the coverage rate and rounds of movements. (b) Relationship between coverage rate and energy consumption. Compared with HA, VFPSO, and 3DVFA, VBO can achieve the best coverage rate by 100% with the fewest rounds of movements and the least energy consumption.

maximum energy consumption nodes, which is precisely the focus of this article.

The energy consumption of each node of four algorithms is represented in Fig. 9, which can be seen intuitively that 3DVFA performs worst when considering both total energy consumption and the uniformity of energy consumption of nodes, and VFPSO is slightly better than 3DVFA, and VBO and HA are obviously better than 3DVFA and VFPSO. Furthermore, VBO is better than those of the other algorithms. The energy consumption of the maximum energy consumption node and the uniformity of energy consumption of VBO is effectively reduced when compared with HA since it exchanges the tasks of nodes with long and short moving distances in step 3). As for 3DVFA and VFPSO, they only consider the final coverage as a goal of optimization, and the optimization of energy consumption is even ignored, which leads to the performance differences shown in Fig. 9.

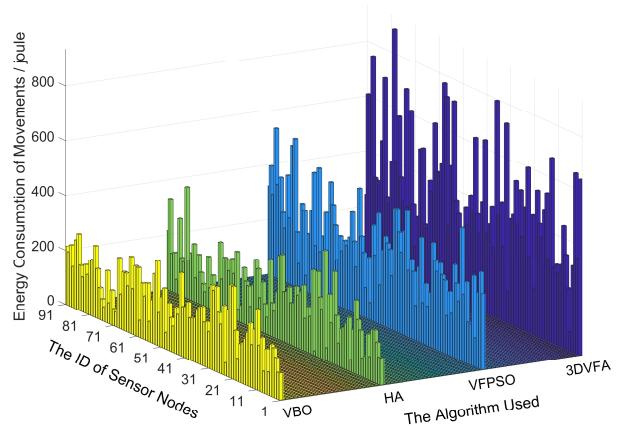


Fig. 9. Comparison of energy consumption about each node of four algorithms. The yellow, green, light blue, and dark blue bar graphs are the energy consumption of the 91 nodes of VBO, HA, VFPSO, and 3DVFA, respectively, after redeployment. The bars from right to left in a given bar graph represent the energy consumption of nodes 1–91 of the corresponding algorithm.

The energy consumption of the maximum energy consumption nodes of four algorithms are compared in Fig. 10(a). As a result, the maximum energy consumption node of VBO, HA, VFPSO, and 3DVFA reaches the best position after 20, 24, 32, and 51 rounds of movement, and the energy consumption of them are 340.67, 405.59, 545.40, and 891.84 J, which means that VBO performs better than 3DVFA, VFPSO, and HA by 61.80%, 37.54%, and 16.01%, respectively, on this performance indicator. The uniformity of residual energy after the final rounds of movements of 3DVFA, HA, VFPSO, and VBO are 173.28, 107.99, 81.42, and 69.43 J, as shown in Fig. 10(b), which indicates that 3DVFA, HA, and VFPSO perform worse than VBO by 59.93%, 35.71%, and 14.72%, respectively.

To evaluate the performance of the VBO, multiple independent simulations with MATLAB 2019a on a computer with 2.5-GHz frequency and 8-GB memory are performed with different initial positions of the sensor nodes to enhance the reliability of the experimental results under the same circumstances. The performance comparison of the final coverage of 3DVFA, VFPSO, HA, and VBO in 200 independent experiments is shown in Fig. 11. It can be seen that the performance of 3DVFA is the worst among them, which basically floats up and down around 95%, and VFPSO, whose average value of final coverage rate is close to 97.5%, is only a little better than 3DVFA. The performance of HA and VBO is the same, which means the final coverage of them reach 100% by every single experiment. Fig. 12(a) shows the total energy consumption after completing the moving task of the four algorithms by 200 experiments, which is similar to the single experiment described above. Specifically, 3DVFA and VFPSO fluctuate around  $9.5 \times 10^4$  and  $8.5 \times 10^4$  J, respectively. The mean value of VBO is close to  $6.8 \times 10^4$  J, which is slightly worse than HA, but much better than 3DVFA and VFPSO. Fig. 12(b) and (c) shows the performance comparison of the energy consumption of

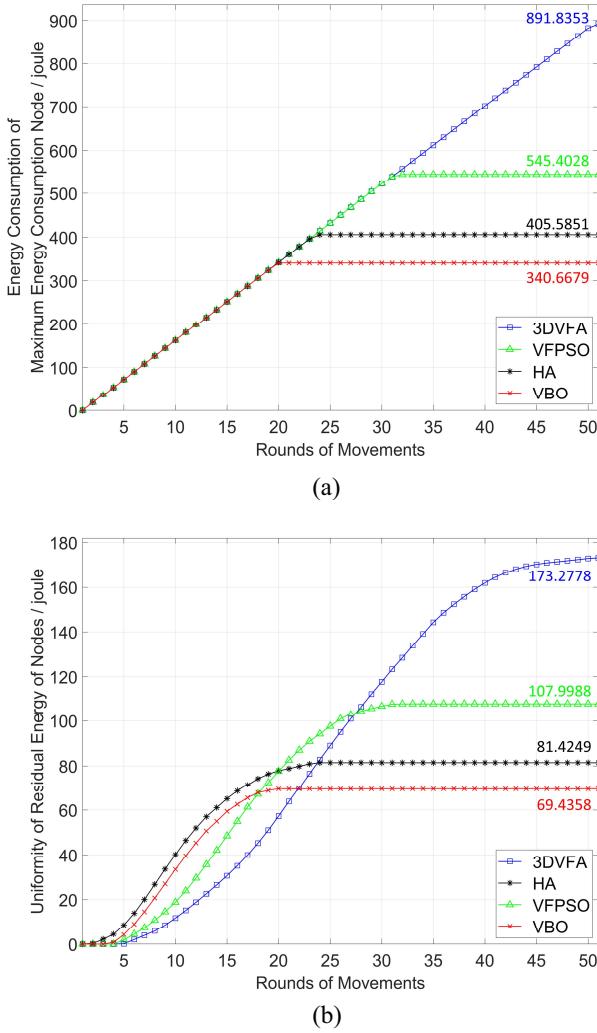


Fig. 10. Comparison of energy consumption of the maximum energy-consumption nodes and uniformity of residual energy of four algorithms. Compared with HA, VFPSO, and 3DVFA, the uniformity of residual energy of 91 sensor nodes is the best after the final round of movement.

the maximum energy consumption nodes and uniformity of residual energy of the four algorithms by 200 independent experiments, and VBO performs better than the other three algorithms.

The difference of performance of four algorithms by 200 experiments are averaged and presented in Table VI and Fig. 13. The proposed strategy can enhance the balance effect of residual energy of nodes by 32.03%, 43.44%, and 30.53%, and also reduce the energy consumption of the maximum energy consumption node by 16.06%, 39.78%, and 29.55%, when compared to HA, 3DVFA and VFPSO. In addition, it can reduce the total energy consumption of nodes by 30.04% and 22.79% when compared to 3DVFA, and VFPSO. It also performs well in terms of final coverage rate and time consumption. The above simulation results can firmly prove the reliability of VBO and verifies the accuracy of the previous single experiment.

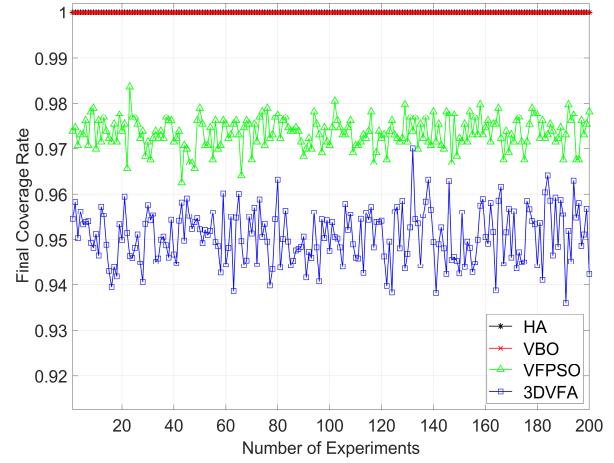


Fig. 11. Comparison of final coverage rate of four algorithms by 200 independent experiments.

TABLE VI  
PERFORMANCE COMPARISON OF THE AVERAGE RESULTS UNDER 200 EXPERIMENTS

| Metrics   | 3DVFA    | VFPSO    | HA       | VBO      |
|---|----------|----------|----------|----------|
| Final coverage rate   | 0.9508   | 0.9734   | 1.0000   | 1.0000   |
| Total energy consumption/J                                  | 96,561.8 | 87,496.0 | 63,557.8 | 67,559.3 |
| Energy consumption of the maximum energy consumption node/J | 2,062.6  | 1,763.3  | 1,479.8  | 1,242.2  |
| Uniformity of residual energy/J                             | 378.9    | 308.5    | 315.3    | 214.3    |
| Time consumption/s  | 0.8493   | 11.5982  | 0.2896   | 0.1010   |

## V. DISCUSSION

### A. Coverage Enhancement

The reasons lead to the performance differences of the final coverage of VBO, HA, 3DVFA, and VFPSO shown in Fig. 11 and Table IV are analyzed in detail as follows.

As task-assignment algorithms, VBO and HA both enhance the final coverage rate by assigning 91 sensor nodes to 91 truncated octahedrons on the basis of meshing the monitoring area, namely, there will be no difference in the final coverage effect between HA and VBO once the stacking strategy is determined. Accordingly, the final coverage rates of HA and VBO can both reach 100% in Fig. 11, thanks to the seamless stacking strategy shown in Fig. 3.

Given that the essence of VFA and its derivative algorithms (such as 3DVFA and VFPSO) is to separate overlapping nodes and fill unmonitored areas, it is noteworthy that moving effect of the sensor nodes so as to the final coverage rate will be inevitably affected by the threshold of the virtual attraction force between the sensor nodes and the uncovered grid spots, the threshold of the virtual repulsive force between the nodes, and the distance of step during every single round of virtual movement. Although these parameters have been adjusted repeatedly to ensure the best performance, the effect is not significantly improved, namely, the performance of the final

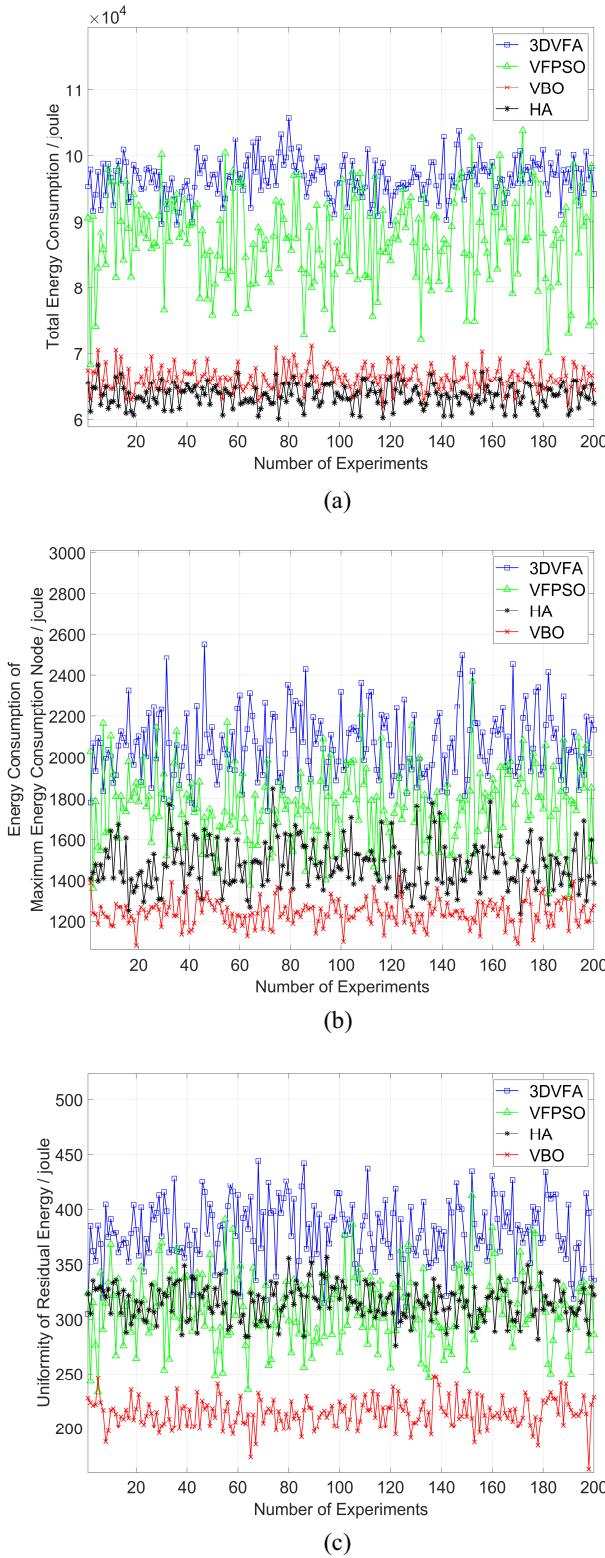


Fig. 12. Comparison of energy consumption of four algorithms by 200 independent experiments. (a)–(c) Comparison of total energy consumption, energy consumption of the maximum energy consumption node, and uniformity of residual energy of four algorithms, respectively.

coverage rate of 3DVFA floats around 95%, which is still worse than that of VBO by 5% approximately. In addition, although VFPSO performs better than 3DVFA occasionally,

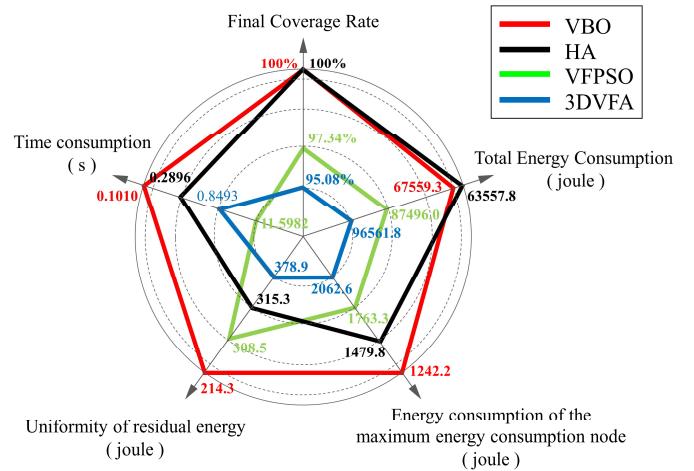


Fig. 13. Performance comparison of the average results under 200 experiments.

the local optimal solution is still unavoidable due to the imperfection of the metaheuristic algorithm, not to mention that it still has the defects similar to 3DVFA naturally as a derivative algorithm of VFA. That is the reason why the final coverage rate of VFPSO is only close to 97% and still worse than VBO and HA by 3% approximately.

### B. Energy Consumption

In addition to the performance differences in coverage enhancements, the performance of the energy consumption of VBO, HA, 3DVFA, and VFPSO is differ during redeployment as shown in Fig. 12, and Table IV, whose reasons can be explained as follows.

Considering that the defects of VFA and its derivative algorithms mentioned above affect the final coverage rate, it can also affect the virtual movement of nodes so as to the energy consumption of movements. Furthermore, 3DVFA and VFPSO only consider the final coverage as a goal of optimization with energy consumption ignored, which enhance the difficulty in accounting the minimization of total energy consumption and the uniformity of residual energy. The task assignment model of HA and VBO determines that their main purpose is to optimize energy consumption based on a full coverage. Therefore, HA and VBO perform better than VFPSO and 3DVFA in terms of energy consumption.

The moving task of the sensor node that needs to move a long distance is exchanged by another node according to *the theorem of task exchange*, just as the reverse blood-transfusion process in VBO ensures the uniformity of blood absorbed by each bat on the basis of maximizing the benefits of all bats, hence VBO is better than HA in terms of energy consumption of the maximum energy consumption node and the uniformity of residual energy.

As shown in Figs. 8(b) and 12(a), VBO is slightly worse than HA in terms of total energy consumption, however, this is not without reason. Given that HA is a task-assignment algorithm for finding the maximum matching of bipartite graphs

by augmented paths, the aim of HA in solving the coverage enhancement problem is to find the assignment strategy with the lowest total energy consumption, which is exactly the same goal of part II of VBO. Consequently, the performance of total energy consumption of HA is the same as that of VBO up to part II. However, VBO achieves a further goal by part III, which is to balance the energy consumption of the sensor nodes by exchanging the moving tasks between nodes with high and low energy consumption on the basis of minimizing total energy consumption, and this process will inevitably sacrifice the performance of the minimization of total mobile energy consumption to balance the residual energy of each node. Essentially, different from the single-objective optimization problems of minimizing total energy consumption which have the only optimum solution, the minimization and equalization of energy consumption are conflicting multiobjective optimization problems [28], [29], whose optimal solutions are a group of tradeoff solutions, namely, Pareto optimum solutions set that no individual can be made better off without another being made worse off [30]–[32]. HA is exactly the only solution in Pareto optimum solutions set that makes the total energy consumption optimal and makes the uniformity of residual energy worst. The performance of minimizing the total energy consumption of any other solution in Pareto optimum solutions set is necessarily worse than that of HA. Accordingly, the total moving energy consumption of VBO is 6.29% worse than HA according to Table IV, which is acceptable. Although the performance of VBO in total energy consumption is sacrificed, the uniformity of the residual energy and the energy consumption of the maximum energy consumption node is optimized as shown in Fig. 12(b) and (c).

## VI. CONCLUSION

In this article, in addition to the demonstrable enhancement of coverage, the conflicting multiobjective optimization problem about the minimization and equalization of energy consumption during redeployment were solved efficaciously by the strategy proposed in this article, namely, VBO, which has superior performances in terms of total energy consumption, uniformity of residual energy, final coverage rate, and time consumption when compared with 3DVFA, VFPSO, and HA in consequence of the competition and feedback strategy in VBO. The application field of VBO is not limited to the energy consumption optimization of 3-D WSNs, and we firmly believe it will perform better in other multiobjective optimization problems. However, there are still some limitations in this article, for example, the constraints on the same perception radius of the sensor nodes. Considering that the challenges of acoustic communication due to variation of the underwater environment, the perceived radius of the sensor nodes, which depends on the link state, is different in UWSNs. Additionally, data packet dropout rate is higher in UWSNs because of acoustic channel refraction, reflection, and ambient noise. Hence, the application of VBO in UWSNs will be our research focus in the future.

## REFERENCES

- [1] M. T. Lazarescu, “Design of a WSN platform for long-term environmental monitoring for IoT applications,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 1, pp. 45–54, Mar. 2013, doi: [10.1109/JETCAS.2013.2243032](https://doi.org/10.1109/JETCAS.2013.2243032).
- [2] M. Ahmed, M. Salleh, and M. I. Channa, “Routing protocols based on node mobility for underwater wireless sensor network (UWSN): A survey,” *J. Netw. Comput. Appl.*, vol. 78, pp. 242–252, Jan. 2017, doi: [10.1016/j.jnca.2016.10.022](https://doi.org/10.1016/j.jnca.2016.10.022).
- [3] X. Hui, W. Bailong, S. Jia, H. Haohan, and Z. Xiaolei, “An algorithm for calculating coverage rate of WSNs based on geometry decomposition approach,” *Peer-to-Peer Netw. Appl.*, vol. 12, no. 3, pp. 568–576, May 2019, doi: [10.1007/s12083-018-0653-1](https://doi.org/10.1007/s12083-018-0653-1).
- [4] X. Wang, H. Zhang, S. Fan, and H. Gu, “Coverage control of sensor networks in IoT based on RPSO,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3521–3532, Oct. 2018, doi: [10.1109/JIOT.2018.2829160](https://doi.org/10.1109/JIOT.2018.2829160).
- [5] K. M. Awan, P. A. Shah, K. Iqbal, S. Gillani, W. Ahmad, and Y. Nam, “Underwater wireless sensor networks: A review of recent issues and challenges,” *Wireless Commun. Mobile Comput.*, vol. 2019, Jan. 2019, Art. no. 6470359. [Online]. Available: <https://www.hindawi.com/journals/wcmc/2019/6470359/cta/>, doi: [10.1155/2019/6470359](https://doi.org/10.1155/2019/6470359).
- [6] V. Pal, G. Singh, and R. P. Yadav, “Balanced cluster size solution to extend lifetime of wireless sensor networks,” *IEEE Internet Things J.*, vol. 2, no. 5, pp. 399–401, Oct. 2015, doi: [10.1109/JIOT.2015.2408115](https://doi.org/10.1109/JIOT.2015.2408115).
- [7] Y.-C. Wang, W.-C. Peng, M.-H. Chang, and Y.-C. Tseng, “Exploring load-balance to dispatch mobile sensors in wireless sensor networks,” in *Proc. 16th Int. Conf. Comput. Commun. Netw.*, Honolulu, HI, USA, 2007, pp. 669–674.
- [8] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc. (IEEE INFOCOM)*, San Francisco, CA, USA, 2003, pp. 1293–1303.
- [9] J. Chen, S. Li, and Y. Sun, “Novel deployment schemes for mobile sensor networks,” *Sensors*, vol. 7, no. 11, pp. 2907–2919, Nov. 2007, doi: [10.3390/S7112907](https://doi.org/10.3390/S7112907).
- [10] F. Zhou, J. Gao, X. Fan, and K. An, “Covering algorithm for different obstacles and moving obstacle in wireless sensor networks,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3305–3315, Oct. 2018, doi: [10.1109/JIOT.2018.2816596](https://doi.org/10.1109/JIOT.2018.2816596).
- [11] N. Boufares, I. Khoufi, P. Minet, L. Saidane, and Y. Ben Saied, “Three dimensional mobile wireless sensor networks redeployment based on virtual forces,” in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Dubrovnik, Croatia, 2015, pp. 563–568.
- [12] S. M. Mohamed, H. S. Hamza, and I. A. Saroit, “Coverage in mobile wireless sensor networks (M-WSN): A survey,” *Comput. Commun.*, vol. 110, pp. 133–150, Sep. 2017, doi: [10.1016/j.comcom.2017.06.010](https://doi.org/10.1016/j.comcom.2017.06.010).
- [13] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G.-J. Kim, “A PSO based energy efficient coverage control algorithm for wireless sensor networks,” *Comput. Mater. Continua*, vol. 56, no. 3, pp. 433–446, 2018, doi: [10.3970/cmc.2018.04132](https://doi.org/10.3970/cmc.2018.04132).
- [14] X. Wang, S. Wang, and D. Bi, “Virtual force-directed particle swarm optimization for dynamic deployment in wireless sensor networks,” in *Proc. Intell. Comput. Int. Conf. Adv. Intell. Comput. Theories Appl.*, vol. 35, Aug. 2007, pp. 292–303, doi: [10.1007/978-3-540-74171-8\\_29](https://doi.org/10.1007/978-3-540-74171-8_29).
- [15] S. Wang, X. Yang, X. Wang, and Z. Qian, “A virtual force algorithm-Lévy-embedded grey wolf optimization algorithm for wireless sensor network coverage optimization,” *Sensors*, vol. 19, no. 12, p. 2735, Jun. 2019, doi: [10.3390/s19122735](https://doi.org/10.3390/s19122735).
- [16] T.-P. Dao, S.-C. Huang, and P. T. Thang, “Hybrid Taguchi-cuckoo search algorithm for optimization of a compliant focus positioning platform,” *Appl. Soft Comput.*, vol. 57, pp. 526–538, Aug. 2017, doi: [10.1016/j.asoc.2017.04.038](https://doi.org/10.1016/j.asoc.2017.04.038).
- [17] D. Arivudainambi, S. Balaji, and T. S. Poorani, “Sensor deployment for target coverage in underwater wireless sensor network,” in *Proc. IEEE Int. Conf. Perform. Eval. Model. Wired Wireless Netw. (PEMWN)*, Paris, France, 2017, pp. 1–6.
- [18] X. Zhao, H. Zhu, S. Aleksic, and Q. Gao, “Energy-efficient routing protocol for wireless sensor networks based on improved Grey wolf optimizer,” *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 6, pp. 2644–2657, Jun. 2018, doi: [10.3837/iiis.2018.06.011](https://doi.org/10.3837/iiis.2018.06.011).
- [19] Z. Liao, S. Zhang, J. Cao, W. Wang, and J. Wang, “Minimizing movement for target coverage in mobile sensor networks,” in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Macau, China, 2012, pp. 194–200.

- [20] Z. Wang, B. Wang, and Z. Xiong, "A novel coverage algorithm based on 3D-Voronoi cell for underwater wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, 2015, pp. 1–5.
- [21] Y. Ding, N. Li, B. Song, and Y. Yang, "The mobile node deployment algorithm for underwater wireless sensor networks," in *Proc. Chin. Autom. Congr. (CAC)*, Jinan, China, 2017, pp. 456–460.
- [22] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, Los Angeles, CA, USA, 2006, pp. 346–357.
- [23] M. Felamian, B. Shihada, and K. Jamshaid, "Optimal node placement in underwater wireless sensor networks," in *Proc. IEEE 27th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Barcelona, Spain, 2013, pp. 492–499.
- [24] W.-C. Ke, B.-H. Liu, and M.-J. Tsai, "The critical-square-grid coverage problem in wireless sensor networks is NP-Complete," *Comput. Netw.*, vol. 55, no. 9, pp. 2209–2220, Jun. 2011, doi: [10.1016/j.comnet.2011.03.004](https://doi.org/10.1016/j.comnet.2011.03.004).
- [25] G. G. Carter and G. S. Wilkinson, "Food sharing in vampire bats: Reciprocal help predicts donations more than relatedness or harassment," *Proc. Roy. Soc. B Biol. Sci.*, vol. 280, no. 1753, Feb. 2013, Art. no. 20122573, doi: [10.1098/rspb.2012.2573](https://doi.org/10.1098/rspb.2012.2573).
- [26] G. Carter and G. Wilkinson, "Does food sharing in vampire bats demonstrate reciprocity?" *Commun. Integr. Biol.*, vol. 6, no. 6, Nov. 2013, Art. no. e25783, doi: [10.4161/cib.25783](https://doi.org/10.4161/cib.25783).
- [27] D. Izadi, J. Abawajy, and S. Ghanavati, "An alternative node deployment scheme for WSNs," *IEEE Sensors J.*, vol. 15, no. 2, pp. 667–675, Feb. 2015, doi: [10.1109/JSEN.2014.2351405](https://doi.org/10.1109/JSEN.2014.2351405).
- [28] D. Alanis *et al.*, "A quantum-search-aided dynamic programming framework for Pareto optimal routing in wireless multihop networks," *IEEE Trans. Commun.*, vol. 66, no. 8, pp. 3485–3500, Aug. 2018, doi: [10.1109/TCOMM.2018.2803068](https://doi.org/10.1109/TCOMM.2018.2803068).
- [29] T. Chen, Y. Du, B. Dong, Y. Chen, and C. Zhang, "Multi-objective learning approach to proactive caching in wireless networks," *IEEE Commun. Lett.*, vol. 23, no. 9, pp. 1538–1541, Sep. 2019, doi: [10.1109/LCOMM.2019.2924422](https://doi.org/10.1109/LCOMM.2019.2924422).
- [30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: [10.1109/4235.996017](https://doi.org/10.1109/4235.996017).
- [31] D. Q. Feng and J. F. Lu, "The research of coverage algorithm for three-dimensional WSN based on MO-DMS-PSO," in *Proc. IEEE Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, 2015, pp. 797–801, doi: [10.1109/IAEAC.2015.7428666](https://doi.org/10.1109/IAEAC.2015.7428666).
- [32] S. Redhu, M. Anupam, and R. M. Hegde, "Optimal relay node selection for robust data forwarding over time-varying IoT networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9178–9190, Sep. 2019, doi: [10.1109/TVT.2019.2929856](https://doi.org/10.1109/TVT.2019.2929856).



**Xiao-Qiang Zhao** received the Ph.D. degree from the Xi'an University of Technology, Xi'an, China, in 2015.

He is a Full Professor with the Xi'an University of Posts and Telecommunications, Xi'an. His current research interests include the technology and application of the Internet of Things.



**Yan-Peng Cui** is currently pursuing the master's degree with the Xi'an University of Posts and Telecommunications, Xi'an, China.

His research interests include the technology and application of the Internet of Things.



**Chuan-Yi Gao** is currently pursuing the master's degree with the Xi'an University of Posts and Telecommunications, Xi'an, China.

His research interests include the technology and application of the Internet of Things.



**Zheng Guo** is currently pursuing the master's degree with the Xi'an University of Posts and Telecommunications, Xi'an, China.

His research interests include the technology and application of the Internet of Things.



**Qiang Gao** received the M.S. degree from the Xi'an University of Posts and Telecommunications, Xi'an, China, in 2017. He is currently pursuing the Ph.D. degree with the College of Mechanical and Electronic Engineering, Northwest Agriculture and Forestry University, Yangling, China.

His research interests include the technology and application of the Internet of Things.