

Received April 9, 2020, accepted May 4, 2020, date of publication May 11, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993554

# A Two-Phase Coverage Control Algorithm for Self-Orienting Heterogeneous Directional Sensor Networks

MING LI<sup>1,2</sup>, JIANGPING HU<sup>2</sup>, AND XIAOLI CAO<sup>1</sup>

<sup>1</sup>Engineering Laboratory for Detection, Control and Integrated System, College of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China

<sup>2</sup>School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Ming Li (sshjlm@163.com)

This work was supported in part by the Chongqing Social Science Planning Project under Grant 2017YBGL142, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJ1600627, Grant KJQN201900839, and Grant KJQN201900833, in part by the Chongqing Educational Science Planning Project under Grant 2018-GX-023, in part by the Opening Research Platform of Chongqing Technology and Business University under Grant KFJJ2017048 and Grant KFJJ2019072, in part by the Intelligent ecological Internet of Things innovation and entrepreneurship team under Grant CQYC201903246, and in part by the Chongqing Science and Technology Bureau under Grant cstc2019jcsx-msxmX0007.

**ABSTRACT** Maximizing network coverage is very important for heterogeneous directional sensor networks (HDSNs). HDSNs consist of directional sensor nodes with different parameters in terms of the sensing radius, the communication radius and the angle of view. After the initial deployment of the HDSNs, coverage problems occur, such as overlapped areas and holes in the coverage, which are not covered by any one of the sensor nodes. To solve these coverage problems, motility and mobility are often utilized. Motility means the ability to adjust the working direction of the nodes, while mobility presents the physical movement of the nodes. The coverage problem that we solved in this paper is characterized by the heterogeneity of the directional sensor nodes in terms of the sensing radius, the field of angle, the motility, the mobility, and the communication radius. A two-phase algorithm was proposed to solve the coverage optimization problem of the HDSNs. In the first phase of the proposed algorithm, a stepwise method was used to adjust the working direction of the nodes; in the second phase of the proposed algorithm, a virtual force-directed self-adaptive differential evolution algorithm was proposed to achieve a tradeoff between the coverage and the energy consumption. Extensive simulation experiments were performed to demonstrate the effectiveness of our proposed algorithm over alternative algorithms in terms of the coverage rate and energy consumption.

**INDEX TERMS** Directional sensor networks, coverage control, heterogeneous network, self-adaptive differential evolution algorithm, virtual force.

## I. INTRODUCTION

In recent years, directional sensor networks (DSNs) have drawn a large amount of attention due to their extensive applications in tracking, home automation, environmental monitoring and more [1]. In DSNs coverage is one of the important criteria of Quality of Service and it is determined by dynamic node placement algorithms [2]–[4]. Dynamic node placement means that the directional sensor nodes have the ability to adjust their working directions and change their physical positions. In reality, the strategy of random node placement is mostly adopted in the initial deployment of the dynamic nodes. However, such random placement strategies

often result in overlapped areas and coverage holes in the area. To correct these problems, it is necessary to adjust the working direction of the node or relocate its position after the initial placement to enhance the coverage rate of the network. Although many studies [3]–[21] have focused on the coverage of DSNs, our work is motivated by the following two considerations. First, an assumption that sensors are identical in their parameters is relaxed with the aim of formulating a more realistic DSNs coverage problem. In practice, heterogeneous nodes can extend their network lifetime and enhance their network reliability without significantly increasing the cost. Second, most of the existing solutions to the coverage problem in DSNs largely depend on the right working directions of the sensor nodes. However, this aspect alone is not sufficient for some applications. A hybrid strategy

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

in which motility and mobility are exploited in a cascaded way will be an effective approach to tackling the coverage problem.

In this paper, we proposed an algorithm to address the problem of the dynamic deployment of DSNs. The problem that we aimed to solve is characterized by the heterogeneity of the directional sensor nodes in terms of the sensing radius, the field of the angle and the mobility. In contrast to most of the existing solutions, which rely on only the working direction adjustment, both the motility and mobility of a node are employed in the proposed algorithm to improve the network coverage. To the best of our knowledge, this study is the first research that exploits the motility and mobility of a heterogeneous directional sensor node to maximize the area coverage in heterogeneous DSNs. Heterogeneous DSNs (shortly for HDSNs) consist of directional sensor nodes with different parameters in terms of the sensing radius, the communication radius and the angle of view. Several distinctive contributions of this paper are summarized as follows.

1) The problem of coverage maximization in HDSNs is addressed. The problem that we aimed to solve is characterized by the heterogeneity of the sensor nodes in terms of the sensing radius, the field of angle and the mobility.

2) The motility and mobility of a node is exploited in a cascaded way to maximize the coverage in the HDSNs. The marginal coverage contribution of the sensors is utilized as the objective function in both phases of the proposed algorithm. The stepwise method is used in the phase that addresses the out-of-field coverage avoidance control and the working direction adjustment. In the phase that addresses the physical movement of the nodes, an improved differential evolution algorithm is proposed to achieve the tradeoff between the coverage and energy consumption. In the improved differential evolution algorithm, self-adaptive mutation strategy, random control parameter  $F$  and the good guidance of the virtual forces between the nodes are utilized to enhance the optimization ability of the algorithm. To the best of our knowledge, it is the first-ever consideration of a two-phase coverage enhancement algorithm in HDSNs.

3) Extensive simulations are performed to validate the proposed algorithm. The simulation results show that the proposed algorithm achieves a higher coverage efficiency and lower energy consumption than the other algorithms.

The organization of the remainder of this paper is as follows. In the next section, we review some related work on the dynamic node deployment issue in DSNs. In Section III, several basic definitions, some assumptions and system models that regard the problem are presented. In Section IV, the proposed algorithm is outlined, and its implementation steps are explained in detail. In Section V, the theory correctness and computational complexity of the proposed algorithm are presented. The performance of the proposed algorithm is evaluated in section VI. Finally, Section VII summarizes the concluding remarks.

## II. RELATED WORK

In this section, previous studies related to DSNs coverage optimization are presented. We also make comparisons of those existing studies with our proposed algorithm.

In past decades, a large amount of effort has been made for dynamic sensor deployment in traditional wireless sensor networks (WSNs) [3], [5]. However, the solutions to traditional WSNs cannot be used for nodes deployment in DSNs, because coverage in DSNs is dependent not only on the physical locations of the nodes, but also on their angles of view and working directions. Performing working direction adjustment is an effective way to maximize the coverage rate [6]–[13]. In [6], an improved virtual potential field algorithm in directional sensor networks was proposed to reduce profitless rotations from coverage optimization. However, this algorithm depended largely on the parameter settings, which were difficult to determine. Sungn and Yang [7] proposed a Voronoi-based coverage improvement approach for wireless directional sensor networks. They used the characteristics of Voronoi diagram and direction-adjustable directional sensors to propose a distributed greedy algorithm. All of the studies mentioned above utilize the motility capability of directional sensor nodes, whereas in some cases, a coverage hole could still exist. In [8], the authors proposed the view-coverage model to measure the coverage quality of the camera sensor network with finer granularity, especially when the object was not full-view covered. Based on this model, a distributed view-coverage enhancing algorithm that uses the rotation of the camera sensors was proposed. The weakness of this algorithm is that when the densities of the nodes are low, a coverage hole could exist. The authors in [9] addressed the problem of covering each target at least  $k$  times while using a minimum number of sensors. They devised a greedy approach with two variants of incentive mechanisms to solve this problem. However, the greedy approach cannot guarantee obtaining an optimal solution. The authors in [10] analysed stochastic coverage and connectivity in three-dimensional heterogeneous directional sensor networks and derived the expected probabilistic expressions for  $k$  coverage and  $m$  connectivity. Only the motility of the sensors was explored in the algorithm, whereas in some applications, a coverage hole could still exist. The work in [11] studied the surveillance service problem, which supports the surveillance quality of  $k$ -barrier coverage in DSNs. The authors designed three different algorithms to find a maximal number of different defence barriers, each of which supports  $k$  coverage. The work in [12] investigated the target coverage problem for DSNs, where each target has different coverage requirements. The authors proposed integer linear/quadratic programming formulations and a greedy algorithm to solve this problem. In [13], the authors addressed the problem of maximum full-view target coverage in directional sensor networks, where each node has  $P$  working directions. They proposed an approximation algorithm based on pipage rounding and a heuristic algorithm.

In addition to the method of working direction adjustment, many studies proposed employing a mobility capability for the directional sensors to improve coverage of the network. In [14], the authors proposed two schemes, namely the circumventer-based and incentive-based schemes, to improve coverage. Three moving strategies are adapted to determine the new locations for the mobile sensors to move. However, these three strategies are not energy efficient, which has a negative impact on the lifespans of the networks. The authors of [15] proposed a virtual force-based movement scheme that consists of four different forces introduced by neighboring sensors and coverage holes. By applying the resultant forces on the sensors, each sensor can move to a sparse area as a result of the coverage ratio being increased. The main weakness of that study is the failure to address how to set the parameters of the algorithm properly to maximize the coverage rate. In [16], the authors proposed a hybrid movement strategy in self-orienting directional sensor networks, where the proposed algorithm exploited the motility/mobility in a cascaded manner for the coverage improvement in DSNs. However, the sensors are assumed to be of the same type in this paper, and the authors neglected the impacts of the heterogeneous nodes' on the network. The authors in [17] solved the coverage maximizing mobile sensor deployment problem for optimal coverage with a directional sensing model. They proposed two node placement algorithms to control the node movements and orientations using only local information. However, the authors did not consider the heterogeneity of the nodes in their work. In the above studies [14]–[17], only the mobility of the nodes was used to improve the area coverage. Depending largely on the mobile ability of the nodes can lead to excessive energy consumption of the network. As a result, the lifespan of the network can be shortened. The problem of full-view barrier coverage using mobile camera sensors with a grid-based deployment strategy is addressed in [18]. The objective of this article is to maximize full-view barrier coverage for the target area with a minimum number of camera sensors. However, the sensor nodes have the same parameters, which is not feasible in a real-life scenario. In [19], X. Zhu *et al.* investigated how to deploy a rechargeable DSNs using a mobile charger with the least number of nodes for perpetual target coverage. Their technique is not suitable for our research work because it focuses on only a pre-defined localization selection rather than sensor deployment. Two mobile target tracking algorithms for heterogeneous sensor networks were proposed by Lu *et al.* [20], [21] in their research, but their work focused on a target location evaluation strategy rather than area coverage, which is beyond the scope of our research.

The research on sensor node placements for DSNs has been extensively studied in the literature [3]–[21]; however, no attempt has been made to address coverage problems for DSNs with heterogeneous nodes in terms of the field of angle view, sensing radius and mobility. Furthermore, most of the existing coverage control algorithms for DSNs have not solved the problem of the boundary nodes' working

direction adjustment for the nodes nearby. To overcome the above weaknesses, a coverage control algorithm for HDSNs is proposed. In our proposed algorithm, first, a stepwise method is used to adjust the working directions of the nodes. Then, in the phase of physical movement of the nodes, an improved differential evolution algorithm is proposed to achieve the tradeoff between the coverage and energy consumption. In the improved differential evolution algorithm, virtual force-directed and self-adaptive mutation strategies are adopted to enhance the optimization ability.

### III. PROBLEM STATEMENT

In this section, we describe the HDSNs model assumed in this study. We first give several basic definitions of HDSNs. Then we present the general assumptions of the model. Finally, the formulation of the coverage problem is described.

#### A. DEFINITIONS

*Definition 1 (Heterogeneous Directional Sensor Nodes):* Heterogeneous means that each directional sensor node has different parameters, such as the sensing radius, communication radius, and numbers of working directions. A sensor node  $s_i$  is denoted by a five-tuple  $\langle L_{s_i}, R_{s_i}, \varphi_{s_i}, \vec{\vartheta}_{s_i}, T_{s_i} \rangle$ .  $L_{s_i}$  represents the locations of the nodes, which correspond to  $(x, y)$  in a two-dimensional rectangular coordinate system.  $R_{s_i}$  is the sensing radius of the node, with the field-of-view angle  $0 \leq \varphi_{s_i} \leq \pi$  ( $0 \leq \varphi_{s_i} \leq \pi$ ).  $\vec{\vartheta}_{s_i}$  is the orientation vector and  $T_{s_i}$  is the transmission radius of the sensor  $s_i$ .

*Definition 2 (Sensing Range):* The sensing range of a directional sensor node  $s_i$  is a sector that is defined by

$$C_i \left( R_{s_i}, \phi_{s_i}, \vec{\vartheta}_{s_i} \right) = \{ \xi \in \mathbb{R}^2 : |\xi_i - L_{s_i}| \leq R_{s_i} \text{ and } \theta \left( \vec{\vartheta}_{s_i}, \overrightarrow{\xi_i L_{s_i}} \right) \leq \frac{\varphi_{s_i}}{2} \} \quad (1)$$

where  $|\xi_i - L_{s_i}|$  denotes the Euclidean distance between  $\xi_i$  and  $L_{s_i}$ ,  $\theta \left( \vec{\vartheta}_{s_i}, \overrightarrow{\xi_i L_{s_i}} \right)$  denotes the angle between the orientation vector of sensor ( $\vec{\vartheta}_{s_i}$ ) and the vector from point  $\xi_i$  to  $L_{s_i}$ .

*Definition 3 (Transmission Range):* The transmission range of sensor  $s_i$  is a disk of radius that is defined by

$$D_{s_i} = \{ L_{s_i} \in \mathbb{R}^2 : |\xi_i - L_{s_i}| \leq T_{s_i} \} \quad (2)$$

*Definition 4 (Neighborhood Node):* A sensor  $s_i$  is a neighborhood node of sensor  $s_j$  if and only if the Euclidean distance between the centers of their transmission disks is less than  $R_{s_j}$ .

$$Nbh(s_j) = \{ s_i : |L_{s_i} - L_{s_j}| \leq R_{s_j} \} \quad (3)$$

*Definition 5 (k-Coverage):* A target in the region of interest is covered by at least  $k$  sensor nodes.

*Definition 6 (Marginal Coverage Contribution of a Node  $s_i$ ):* The points or a sub-region that is covered exclusively by the node  $s_i$ . It reflects the coverage effects that a sensor node  $s_i$  has on the monitored area. Therefore, we use the marginal coverage contribution of the node as the objective function in the process of the nodes' working direction adjustments.

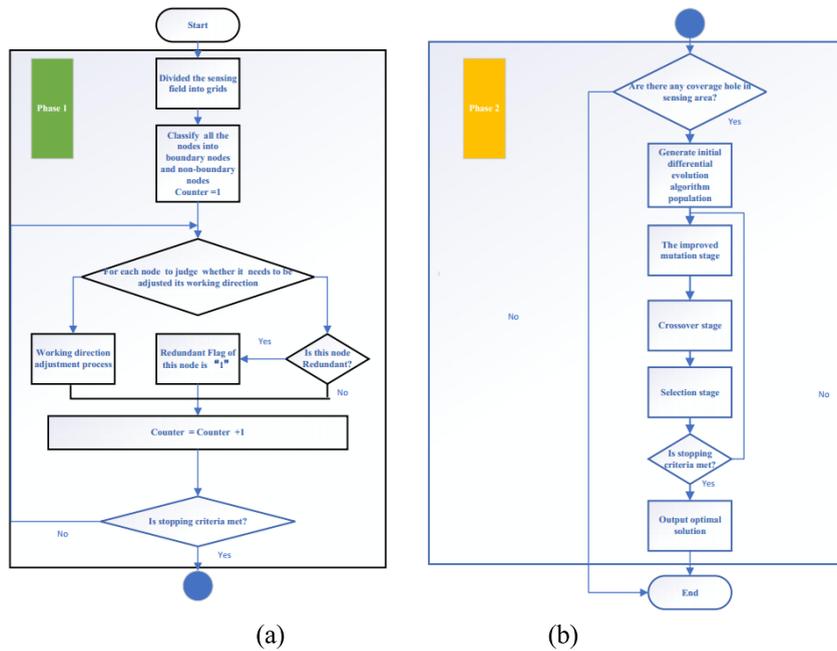


FIGURE 1. (a) Phase 1: Self-Oriented Process. (b) Phase 2: Physical Position Changes Process.

**B. SYSTEM MODEL AND ASSUMPTIONS**

A heterogeneous directional sensor network with  $N$  nodes represented by  $S = \{s_1, s_2, \dots, s_N\}$  is deployed in a two-dimensional field  $Z$ . Each sensor node has different parameters in terms of the sensing radius, the communication radius and the angle of view. Every directional sensor has the ability to adjust its working direction and change its physical position. To calculate the coverage of the monitored region, we use a binary sensing model and a grid-based method. In other words, we first divide the monitored region into small grids, and each grid is represented by its central point. If the central point of a grid is within the sensing range of the nodes, we call such a grid covered. By calculating the ratio between the number of covered grids and the total grids in the observed area, we can obtain the coverage rate of the observed area. The granularity of the grid cells (the distances between the adjacent grid cells) is determined by the requirement precision. When the grid is as small as 0.25%-4% of the monitored area, the error of the calculated value and its precise value are approximately 0.1%-0.5% [22].

**C. FORMULATION OF THE COVERAGE PROBLEM IN HDSNs**

We formulate the coverage optimization issue in heterogeneous directional sensor networks that consist of direction-steerable and position-moveable nodes as follows:

There is a two-dimensional monitored region  $A$  and a set of heterogeneous directional sensor nodes  $S = \{s_1, s_2, \dots, s_N\}$ , which are different in their sensing radius, angle of view and ability to move. The problem that we aim to solve is following: how to optimize their physical positions

$L = \{L_{s_1}, L_{s_2}, \dots, L_{s_N}\}$  and working orientations  $\vec{\vartheta} = \{\vec{\vartheta}_{s_1}, \vec{\vartheta}_{s_2}, \dots, \vec{\vartheta}_{s_N}\}$  to attain the goals of the coverage maximization and the energy efficiency.

**IV. PROPOSED ALGORITHM**

The proposed algorithm is a two-phase algorithm, which consists of self-orientation and physical position changes in consecutive order. Due to high energy consumption of the physical position changes, the self-orientation process precedes the position change in the proposed algorithm. The key idea of the proposed algorithm is to first adjust the sensors' working direction to maximize the coverage rate of the area and minimize the overlapped area. Then, we check whether there are redundant sensors and coverage holes in the area. If there are, then we will make some nodes move to the proper place to heal the coverage hole. The proposed method consists of two phases, as illustrated by the flow chart presented in Fig. 1. In the following subsection, we will explain the proposed algorithm in detail.

**A. SELF-ORIENTED PROCESS**

The aims of the self-oriented process are to reduce the overlapped area and cover the holes in the area. According to sensors' distribution characteristics, there are two types of sensors in the area, namely, boundary nodes and non-boundary nodes. A sensor node is considered to be a boundary node if the distance between it and the sensing field boundary is smaller than  $\mu R_{s_i}$ , where  $\mu$  is the coefficient of  $0 \sim 1$ . By using formula (1), the coverage map, which indicates the coverage status of each grid, can be obtained, where 1 indicates a grid that is covered by one sensor node, 0 means

a grid that is not covered by any nodes, and a value larger than 1 indicates an overlapped grid that is covered by more than one sensor node. Based on the types of sensors, two algorithms are proposed to minimize the overlapped area in the region:

#### 1) BOUNDARY NODES ADJUSTMENT ALGORITHM

Since sensors randomly distribute in the area, it is more likely that some of the nodes are located very close to the boundary of the region or even out of the region. In that case, these sensors contribute little to the coverage rate of the area. An out-of-field coverage enhancement algorithm (OFCE) is given to address this problem. The details of OFCE are given below in the following items.

##### a: DIRECTION\_CHECK SUBROUTINE

The Direction\_check subroutine is performed to determine whether a boundary node must change its working direction or not. The main process of the direction check test is as follows: First, by examining the coverage map, a sensor node obtains the number of covered grids by itself, which is denoted as  $c_i$ . Then, the sensor node makes a comparison with  $c_i$  and the predefined threshold value. If  $c_i$  is larger than the predefined threshold value, the sensor node does not change its working direction and completes the OFCE algorithm; otherwise, it should adjust its working direction. The threshold value is the percent  $p$  of the area of the sector. Eq. (2) presents the formulation of the Direction\_check, where “1” is used to represent the sensor  $s_i$  to adjust its working direction. If the value of  $S_{\text{boundary}}$  for  $s_i$  is 0, then we set the MoveFlag value to be 1, which is indicative of changing  $s_i$ 's physical locations in the second phase.

$$S_{\text{boundary}} = \begin{cases} 1 & c_i \leq p \times 0.5 \times \varphi_{s_i} \times R_{s_i}^2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

##### b: DETERMINATION OF NEW WORKING DIRECTION

After performing the above Direction\_check subroutine, a sensor node that must change its working direction obtains its marginal coverage contribution  $\eta(s_i)$ .  $\eta(s_i)$  can be calculated by counting the number of grids that are exclusively covered by  $s_i$  according to Definition 6. Then, sensor  $s_i$  will conduct anticlockwise rotation until it finds the optimized working direction after the maximum iterations  $B\_MI$ . The rotation angle  $BC$  of the boundary nodes is predefined, and the value of  $BC$  is very small compared to its field-of-view angle  $\varphi_{s_i}$ . In its rotation process, the grids that are exclusively covered by the sensor  $s_i$  are recorded in a specific array. At the end of the iteration, the direction that has the maximum grids exclusively covered by a sensor is selected as its final working direction. Therefore, the working direction of the boundary sensor is determined. After the node changes its working direction, the coverage map of the network is also updated. However, if there are some boundary nodes that cannot find the proper working direction (this situation could occur when the density of the network is very high), we will mark these

nodes and set the value of the array MoveFlag to be 1, which is indicative of changing their physical locations in the second phase.

#### 2) NON-BOUNDARY NODES' ADJUSTMENT ALGORITHM

##### a: DIRECTION\_CHECK SUBROUTINE

The aim of this subroutine is the same as it was for the boundary nodes; in other words, the aim is to determine whether to change the nodes' working directions or not. From the coverage map of the target region, it is easy to know the status of the grids, which are uncovered, 1-covered or  $k$ -covered ( $k > 1$ ). In the coverage map, zeros represent the uncovered points. If a grid is covered by only one sensor node, then its value is one. Those grids that have a value greater than 1 are overlapped grids. For such overlapped grids, several measures must be performed. First, each node obtains the total number of overlapped grids in its sensing range. Then, the sensor compares the total number of overlapped grids with a predefined threshold value  $\theta_t$ . If its value is smaller than the threshold value  $\theta_t$ , then there is no need to adjust its working direction. In that case, the sensor finishes this phase. Otherwise, the working direction of this node should be adjusted to improve the coverage rate of the area. The value of the threshold parameter  $\theta_t$  is  $p'$  percentage of a node's sector area.

##### b: WORKING DIRECTION ADJUSTMENT

After performing the Direction\_check subroutine, we know whether the nodes will change their working directions or not. For those nodes that will change their working directions, they use the same message transmission mechanism as in [16] to obtain the number of neighbors. According to the number of neighbors, every node obtains its priority. A sensor node with a smaller number of neighboring nodes has a higher priority. If there are many nodes with the same number of neighbors, the node that has less marginal coverage contribution has a higher priority. As a result, the node with a higher priority precedes the node with a lower priority to adjust its working direction. Since the sensor nodes do not change their physical locations in this phase, the numbers of their neighboring nodes do not change. The process of changing the working direction is the same as that for a boundary node. The main differences between non-boundary nodes and boundary nodes in the process of the working direction change lie in two additional parameters. Consequently, we will present these two parameters in detail to make the differences more precisely expressed.

① The angle change  $ad_i$  ( $i = 1, 2, \dots, N$ ) for each node  $s_i$  ( $i = 1, 2, \dots, N$ ) in the iteration.

In contrast to fixed-angle changes for the boundary nodes, the angle change for the non-boundary node is self-adaptive according to the number of its neighboring nodes. The details are as follows: First, we obtain the average number of neighbor nodes through dividing the total number of neighbor

nodes by the number of adjusted nodes. Then, for each sensor to adjust in its working direction, we compare the average number of neighbor nodes with the number of the node's neighbor nodes. If the number of the node's neighbors is smaller than the average value, then the changing angle is set to be *Big\_Angle*; otherwise, *Small\_Angle* is used for this sensor node. The key idea behind this step is that in general, if a sensor has more neighbor nodes, it is more likely that the overlap areas between the sensor and its neighbor nodes are larger compared to its counterpart nodes with fewer neighbor nodes. Therefore, we should adjust its working direction in a more casual and precise way.

② The number of maximum iterations  $MI_i$  ( $i = 1, 2, \dots, N$ ) for each node  $s_i$  ( $i = 1, 2, \dots, N$ ).

The number of the node's adjustment iterations  $MI_i$  is determined by its angle change  $ad_i$ . We divide  $2\pi$  by its angle change  $ad_i$  to obtain its maximum iteration number  $MI_i$ . In the process of performing these iterations, we record the marginal contribution gained that is caused by the node's working direction adjustment and choose the working direction that has the greatest marginal contribution as its final working direction.

## B. PROCESS OF PHYSICAL POSITION CHANGES

After the angle change process of the boundary nodes and non-boundary nodes, it is most likely that coverage holes exist within the area due to the energy shortage of the nodes and the damage from the harsh environment. The mobility of the sensor is an efficient way to heal the coverage hole problem. Thus, by exploiting the mobility of the sensors we proposed an algorithm to address the coverage hole problem. The proposed algorithm is based on virtual force and an enhanced differential evolution algorithm. The key points of the proposed algorithm are to solve two problems: the first problem is which sensors are selected to move, and the second problem is how to move these sensors. The details of the two key points for the proposed algorithm are presented in the following subsection.

### 1) WHICH SENSORS ARE TO MOVE?

After the phase of the working direction adjustment, some of the nodes might not find proper working directions due to the high node density in the sensing area. By examining the values of the array *MoveFlag*, we can obtain the information on whether the sensors should change their physical locations or not. If the value of a sensor's array *MoveFlag* is 1, then the sensor is redundant and must change its physical location. Otherwise, the sensor stays in the same location as before.

### 2) HOW TO MOVE?

With the purposes of being comprehensive, in this subsection, we introduce three algorithms i.e., the virtual force (VF) algorithm and the differential evolution (DE) algorithm, and then we introduce the proposed algorithm.

#### a: VIRTUAL FORCE ALGORITHM

The concept of the virtual force field adopted from robotics [23] has been successfully used in mobile omni-directional [5] and directional sensor networks [15] to enhance the coverage rate of the network. In the VF algorithm, each mobile sensor node is treated as a virtual particle and is subject to virtual forces exerted by nearby nodes (virtual particles). More specifically, the virtual force  $F$  exerted on a node is a gradient of a scalar potential field, which can be due to obstacles or other nodes and results in an attractive or repulsive force. The repulsive force that mobile node  $s_i$  receives from a mobile node or static node (a node that does not change its physical position in this phase)  $s_j$  is represented in Eq. (5) [24].

$$F_1(i, j) = \begin{cases} \frac{k_1 m_i m_j}{d(i, j)^{l_1}}, \alpha_{ij} + \pi & 0 < d(i, j) \leq R_{s_i} \\ 0 & d(i, j) > R_{s_i} \end{cases} \quad (5)$$

where  $d(i, j)$  is the Euclidean distance between node  $s_i$  and  $s_j$ ;  $m_i$  is the mass of node  $s_i$ , the value of  $m_i$  is the sensing area of node  $s_i$ , i.e.,  $0.5 \times \pi \times \phi_{s_i} \times R_{s_i}^2$ ;  $k_1$  is a constant,  $l_1$  is a distance coefficient; and  $\alpha_{ij}$  is the orientation of a line segment from  $s_i$  to  $s_j$ .

Since every mobile node receives several forces from its neighbors (i.e., the nodes that are within its communication radius), the node's next position is determined by the composition of the forces exerted by its neighbors. Let  $F_x$  and  $F_y$  denote the x-coordinate and y-coordinate of the sum vector of the virtual forces, respectively, while  $(X_{next}, Y_{next})$  and  $(X_{curr}, Y_{curr})$  are the node's next position and current position, respectively. Then, the next position of the node is updated in the same way as the scheme in [25]. In other words,

$$(X_{next}, Y_{next}) = (X_{curr} + \frac{F_x d_{max}}{\sqrt{F_x^2 + F_y^2}} e^{-\frac{\gamma}{\sqrt{F_x^2 + F_y^2}}}, Y_{curr} + \frac{F_y d_{max}}{\sqrt{F_x^2 + F_y^2}} e^{-\frac{\gamma}{\sqrt{F_x^2 + F_y^2}}}) \quad (6)$$

where  $\gamma$  is a coefficient used to translate the virtual force to the movement; and  $d_{max}$  is the maximum movement distance in one movement. In our experiment, coefficient  $\gamma = 50$ , and the value of  $d_{max}$  is equal to 5% of the node's sensing radius [26].

#### b: STANDARD DIFFERENTIAL EVOLUTION ALGORITHM

The differential evolution (DE) algorithm [27] is a swarm-based stochastic optimizer, which is a simple yet efficient global optimization algorithm. The DE algorithm has been proved to be superior to other evolutionary algorithms such as genetic algorithm and particle swarm optimization, in solving numerical benchmark problems [28]. Suppose that the search space of DE is  $D$ -dimensional. The main process of the classical version of DE is given as follows:

##### ① Initialization

The number of generations is set to be  $G$ . Each population contains  $NP$  individuals. All of the individuals in the first

generation are generated initially by random numbers. The members of the population are limited by the upper and lower limits of the variables.

$$X_{i,j}^0 = X_{i,j}^L + rand \times (X_{i,j}^U - X_{i,j}^L) \quad i = 1, 2, \dots, NP; \\ j = 1, 2, \dots, D \quad (7)$$

where  $X_{i,j}^U$  and  $X_{i,j}^L$  are the lower bound and upper bound of the variable  $X_i$ , respectively; and  $rand$  is a uniformly distributed random number between 0 and 1.

### ② Mutation

The mutation operation plays the most significant role in the performance of a DE algorithm. There are several mutation schemes in DE, and “DE/rand/1” as Eq. (8) and “DE/rand/2” as Eq. (9) are given as follows:

$$U_{i,j}^{t+1} = X_{r1,j}^t + F \times (X_{r2,j}^t - X_{r3,j}^t) \quad (8)$$

$$U_{i,j}^{t+1} = X_{best,j}^t + F \times (X_{r2,j}^t - X_{r3,j}^t) + F \times (X_{r4,j}^t - X_{r5,j}^t) \quad (9)$$

In Eq. (8) and Eq. (9),  $U_{i,j}$  is the element of the mutant individual  $U_i$ ;  $F$  is the real-valued scaling factor which is a positive constant and controls the amplification of the differential vector;  $t$  is the index of the generation;  $X_{r1,j}$ ,  $X_{r2,j}$ ,  $X_{r3,j}$ ,  $X_{r4,j}$  and  $X_{r5,j}$  are five bits of the randomly chosen individuals with index  $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i$ ; and  $X_{best}^t$  defines the individual that has the best fitness function at the  $t$ -th generation. In general, the mutation strategy (8) has strong global exploration ability, and the mutation strategy (9) is suitable for fast convergence.

Note that, each variable of the individuals must satisfy the boundary conditions in the evolutionary process. If these conditions are not satisfied, then the variable must be generated randomly within its permitted range. (This process is the same as the initialization).

### ③ Crossover

The process of crossover is to increase the diversity of the perturbed parameter vectors. The trial individual  $V_i$  is generated by crossing the target individual  $X_i$  with its mutant counterpart  $U_i$ . The widely used binomial crossover is defined by the following equation:

$$V_{i,j}^{t+1} = \begin{cases} U_{i,j}^{t+1}, & \text{if } (rand \leq CR \text{ or } j = rand(i)) \\ X_{i,j}^t, & \text{otherwise} \end{cases} \quad (10)$$

where  $V_{i,j}$  is the element of the trial individual  $V_i$ , and  $CR$  is the crossover probability, whose range is (0,1);  $rand(i)$  are random integers within  $1, 2, \dots, NP$ , where  $NP$  is the length of an individual; and  $j$  is the index of the dimensionality with  $j = 1, 2, \dots, D$ .

### ④ Selection

The generated trial vector  $V_i^{t+1}$  from the crossover operation will be compared with the target vector  $X_i^t$  based on better fitness values. The fittest of these two will survive into the next generation. Therefore, the selection criteria in DE are defined as follows:

$$X_i^{t+1} = \begin{cases} V_i^{t+1}, & \text{if } f(V_i^{t+1}) < f(X_i^t) \\ X_i^t, & \text{otherwise} \end{cases} \quad (11)$$

As shown in Eq. (11), the trial individual  $V_i^{t+1}$  replaces the target individual  $X_i$  if its fitness value is better. Otherwise, the target individual is preserved into the next generation. Therefore, the population is updated according to these three operators.

Selection takes place when a tournament is held between the target vector and trial vector, and the vector with better fitness function will survive into the next generation.

The cyclic implementation of mutation, crossover, and selection is continued until it satisfies the predefined stopping criterion.

Although the DE algorithm can be directly applied to the problem of sensor deployment optimization, it is necessary to solve how to encode individuals in the DE population. In this paper, a position vector  $p_i^c = (x_i^1, y_i^1, x_i^2, y_i^2, \dots, x_i^{n_m}, y_i^{n_m})$  is considered to be an individual in the DE population with the objective to maximize the coverage rate of the network, where  $n_m$  is the number of mobile sensor nodes, and  $x_i^j$  and  $y_i^j$  denote the x-coordinate and y-coordinate of the  $j$ th mobile sensor node, respectively.

### C: THE PROPOSED ALGORITHM

The principle behind the hybridization of the DE algorithm and the VF algorithm proposed in this paper is to furnish an algorithm while blending the innovative structure and good convergence properties of the DE and the good guidance of the VF. The proposed algorithm maintains the swarm intelligence-based nature and iterative searching strategy of their constituent optimizers, but with the procedural changes described in the following steps:

In the mutation step, self-adaptive mutation strategies are adopted, and a virtual force factor is also utilized to direct the updating of the mutant vectors. In other words, in the early stages of the iterations, the mutant vectors in Eq. (12) are used:

$$U_{i,j}^{t+1} = X_{r1,j}^t + F \times (X_{r2,j}^t - X_{r3,j}^t) + AC_1 \times R_i^t \times A_{ij}^t \quad (12)$$

In the final stages of the iterations, the mutant vectors in Eq. (13) are used:

$$U_{i,j}^{t+1} = X_{best,j}^t + F \times (X_{r4,j}^t - X_{r5,j}^t) \\ + F \times (X_{r2,j}^t - X_{r3,j}^t) + AC_1 \times R_i^t \times A_{ij}^t \quad (13) \\ Pr = 1 - t/MaxT \quad (14)$$

where  $F$ ,  $X_{r1,j}^t$ ,  $X_{r2,j}^t$ ,  $X_{r3,j}^t$ ,  $X_{r4,j}^t$  and  $X_{r5,j}^t$  are the same as what was used in Eq. (8) and Eq. (9). We use the relationship between the random value in [0, 1] and the parameter  $Pr$  as in Eq. (14) to decide which mutation strategies are used, where  $t$  is the current iteration number, and  $MaxT$  is the maximum value of the iteration number. In other words, if the random value is smaller than  $Pr$ , then the mutation strategy (12) is used; otherwise, strategy (13) is adopted. From Eq. (14), it can be seen that in the very early stage of the algorithm, the value of  $Pr$  is almost 1. Therefore, it is more likely that the value of  $Pr$  is larger than a random value, and mutation strategy (12) is used. In the final iteration stage, the value of  $Pr$  becomes

smaller than what it was before. As a result, it is probable that the value of Pr is smaller than the random number, and mutation strategy (13) is used.  $AC_1$  and  $R_i^t$  are the acceleration constant and acceleration variable for the virtual force item. Here,  $R_i^t$  is an independent random variable.  $A_i^t$  is a parameter that is used to suggest movement provided by VF:

$$A_i^t = [A_i^t(1), \dots, A_i^t(n_m)]$$

where  $A_i^t(k) = [Ax_i^t(k), Ay_i^t(k)]$  denotes the suggested movement of the  $k$ th mobile sensor node.  $Ax_i^t(k)$  and  $Ay_i^t(k)$  are respectively the x-axis and the y-axis components; and  $F_i^x$  and  $F_i^y$  denote the x-axis and y-axis component of  $F_i$ , respectively.

$$\begin{aligned} Ax_i^t(k) &= \frac{F_i^x}{F_i} \times MaxStep \times e^{\frac{-1}{F_i}} \\ Ay_i^t(k) &= \frac{F_i^y}{F_i} \times MaxStep \times e^{\frac{-1}{F_i}} \end{aligned} \quad (15)$$

In the traditional DE algorithm, the control parameter F is a constant and it cannot be adjusted according to the iteration status, which exerts negative effects on the optimization ability of the DE algorithm. To address this problem and enhance the performance of the DE algorithm, in the proposed algorithm the value of F is set to be a random value in [0.5, 1] as in Eq. (16):

$$F = 0.5 + 0.5 \times rand \quad (16)$$

where rand is a function that can generate a random value in [0, 1]. The advantage of this setting is the reduction of the number of parameters and avoidance of becoming trapped in a local minimum situation due to a wrong value setting of F. As a result, the performance of the proposed algorithm is improved by a self-adaptive mutation strategy, random control parameter F and the good guidance of the VF. In Table 1, the pseudo-code of the proposed algorithm is given.

## V. ANALYSIS OF THE PROPOSED ALGORITHM

This section analyses the computational correction, the computational complexity and the overhead of the proposed algorithm.

### A. COMPUTATIONAL CORRECTION ANALYSIS

*Theorem:* In the HDSNs with a limited number of sensor nodes, the coverage rate of an area obtains its maximum value when the overlap areas between each node of the HDSN and its neighbor nodes obtain the minimum value.

*Proof:* Assuming that the area of the monitored zone is  $S_{area}$ , the total overlap area of the nodes in the HDSN is  $S'$ , while  $\eta(s_i)$  is the marginal coverage contribution of node  $s_i$ ; then, the coverage rate of the zone is

$$C = \frac{S' + \sum_{i=1}^N \eta(s_i)}{S_{area}} \quad (17)$$

Because the value of  $S' + \sum_{i=1}^N \eta(s_i)$  is a constant and does not exceed the value of  $S_{area}$ , when the marginal coverage

TABLE 1. The proposed algorithm.

Algorithm 1. Pseudo code for the proposed algorithm
Input: The parameters(i.e., positions, quantity, sensing radius, angle of view, and so on.) of the static and mobile nodes
Output: improved solution
1. Initialization of Population Vector
2. While not stopCriteria () do {
3. Vector difference-based Mutation using adaptive mutation strategy, which is
If rand < Pr
mutation strategy (12) is adopted
else
mutation strategy (13) is adopted
end if
4. Crossover operation using Equation (10)
5. Section operation using Equation (11)
6. End while }
7. Return solution to the problem

contribution of each node  $s_i$ , namely,  $\eta(s_i)$ , achieves its maximum value, the coverage rate of the area obtains its maximum value. The marginal contribution of node  $s_i$  can be expressed as

$$\eta(s_i) = S_{area-i} - \cup \{s_i \cap Nbh(s_i)\} \quad (18)$$

where  $S_{area-i}$  is the sensing area of node  $s_i$ . From the above Equation,  $\eta(s_i)$  gets its maximum value only when the overlap areas between  $s_i$  and its neighborhood nodes  $Nbh(s_i)$  achieve their minimum value.

### B. COMPUTATIONAL COMPLEXITY

The proposed algorithm consists of two phases. In the first phase, namely, working direction adjustment, the number of sensor nodes is  $N$  and the number of direction adjustments is MI which is a constant. Therefore, the complexity of this phase is  $O(N)$ . In the second phase of the proposed algorithm, the VF directed self-adaptive differential evolution algorithm is used to heal the coverage hole in the sensing area. In this phase, the search of uncovered grids and the mobile directional nodes is necessary. The computation complexity of this process is  $O(N_{uncovered} \times N_m)$ , where  $N_{uncovered}$  is the number of uncovered grids in the area,  $N_m$  is the number of the mobile directional nodes. The total virtual forces between the sensors deployed in the area should be calculated. The computation complexity of the calculation of the virtual forces is  $O(N \times N_m)$ . Therefore, the computation complexity of the proposed algorithm is  $O(N + N_{iter} \times N_m \times Np \times (N_{uncovered} \times N_m + N \times N_m))$ , where  $N_{iter}$  is the number of iterations in the DE algorithm,  $Np$  is the number of population in the DE algorithm. Generally, the value of  $N_{iter}$  is determined by the solved problem.

### C. OVERHEADS

The overhead of the proposed method is working direction adjustment and physical position change. As it was described before, to be energy efficient, the proposed algorithm does not impose trial and error of rotation or relocation on the nodes since working direction adjustment and physical position change are performed after the algorithm.

Fig. 2 demonstrates the average of total rotated angles by the nodes for different node densities. Sensor specifications are listed in Table 2. Fig. 3 shows the average of total travelled distances by the nodes during the coverage optimization approach. Obviously, with the increasing number of the nodes deployed in the area, more rotations and displacements are needed to optimize coverage ratio in the area of interest.

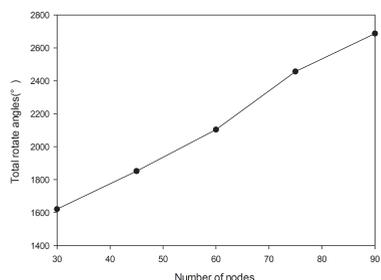


FIGURE 2. Total rotated angles by the proposed algorithm for different node densities.

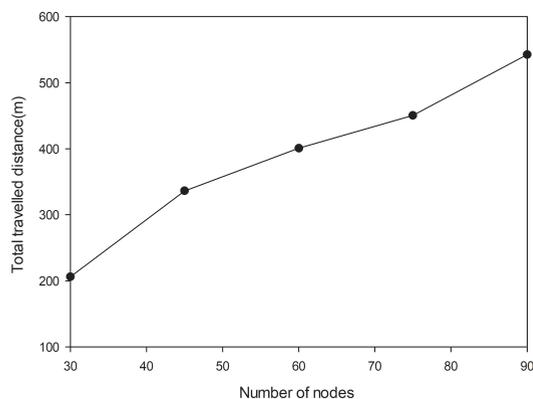


FIGURE 3. Total travelled distances by the proposed algorithm for different node densities.

### VI. PERFORMANCE EVALUATION

In this section, the performances of the proposed algorithm will be investigated. To simulate the algorithm, we used MATLAB2013b on a system with an Intel i7 processor, 2.8 GHz CPU, 16 GB RAM and Microsoft Windows 10 education version as a platform, where all of the experimental values are achieved by over 30 independent runs. The simulation parameters and their values are presented in the following subsections. Then, we simulated the proposed algorithm and compared its performance with existing area coverage algorithms in the literature, namely, HMS [16], random deployment, EEA [29], and two custom VF-based algorithms. In the two custom VF-based algorithms, the virtual force algorithm

proposed in [6] is used in the phase that involves changing the angle of view. The main difference between the two custom algorithms lies in the strategy used in the phase of the nodes' physical position change. In one algorithm of the two custom algorithms, namely VF, the virtual force algorithm proposed in [25] is used to change the physical positions of the nodes. In contrast, the other algorithm, VF-DE, employed a differential evolution algorithm to optimize the physical positions of the nodes to enhance the coverage rate of the network, which uses Eq. (8) as a mutation operator.

### A. SIMULATION SETTINGS

In our simulations, the simulation parameters are listed in Table 2.

TABLE 2. Simulation parameters and values.

Symbol	Declaration	Value
Z	Network size	40×40
N	Number of sensor nodes	45
B_MI	Maximum iterations in self-oriented process	10
BC	Rotation angle	$\pi / 12$
$\mu$	Coefficient of the coverage threshold	0.5
p	Boundary threshold value in percentage	60
p'	Non-boundary threshold value in percentage	90
Big_Angle	Pre-defined changing angle	$\pi / 36$
Small_Angle	Pre-defined changing angle	$\pi / 180$
$k_1$	A constant used in Eq. (3)	1
$l_1$	Distance coefficient used in Eq. (3)	1
tmax	Parameter appeared in [6]	100
G	Number of generations in DE	200
NP	Length of an individual in DE	30
CR	Crossover probability in DE	0.6 [26]
F	Scaling factor in DE	Gaussian distribution $N(0, 1)$ [27]
$AC_1$	Acceleration parameter of the proposed algorithm	1

Three types of the sensors are utilized in the simulations and their parameters are given in Table 3.

TABLE 3. Sensor specifications.

Type	Sensing Radius	Communication Radius	Angle of View	Quantity
1	6	12	$\pi / 6$	15
2	10	20	$\pi / 4$	15
3	15	30	$\pi / 3$	15

### B. SIMULATION RESULTS

#### 1) COVERAGE PERFORMANCE

This experiment aims to evaluate the coverage performance of the proposed algorithm and other algorithms. From Fig. 4, we can observe that HMS, VF, EEA, VF-DE and the proposed algorithm enhance the coverage rate of the network substantially after deployment of the initial nodes (i.e., random deployment). In addition, the proposed algorithm achieves the best results among the compared algorithms with different

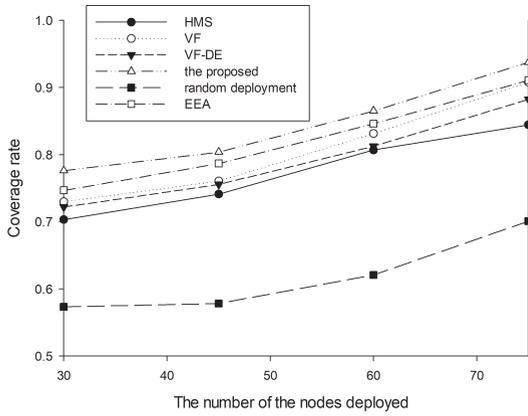


FIGURE 4. Final coverage rate vs. node density.

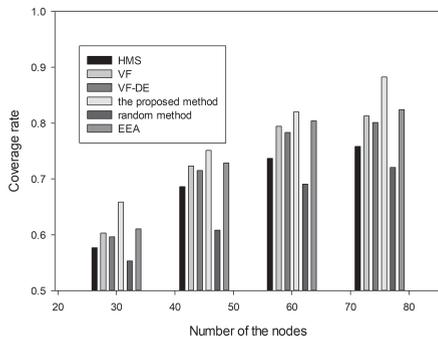


FIGURE 5. Coverage performance comparison after working direction adjustment with different numbers of the nodes.

node densities in the target area. To further evaluate the coverage performance of the algorithms in the working directions adjustment, the results of the coverage rate after the working directions adjustment are demonstrated in Fig. 5. Several observations can be drawn from these results. First, as more sensor nodes are placed in the target area, the performances of all of the algorithms in the coverage rate improve more. Second, it is clear that the proposed algorithm outperforms the other algorithms for different numbers of nodes deployed. There are two reasons for the above results from Fig. 4 and Fig. 5. First, a stepwise method provides a more precise and effective approach than the other compared algorithms to solving the working direction adjustment problem, especially when the density of the sensors is high. In this situation, the virtual force-based algorithm does not work well, because the distances between the sensors are smaller, which has a bad influence on the virtual force. As a result, the performance of the virtual force-based algorithm degraded. For HMS and EEA, the higher the densities of the sensors are, the more probable that the sensor nodes to become boundary nodes. For such boundary nodes, no measures are taken to address this situation in HMS and EEA. Second, in the phase that involves physical position changes, the proposed algorithm can achieve the global optimum with higher probability than the other algorithms. The coverage improvement results generated by the proposed algorithm are illustrated in Fig. 6. As seen from this Figure, with the progress of

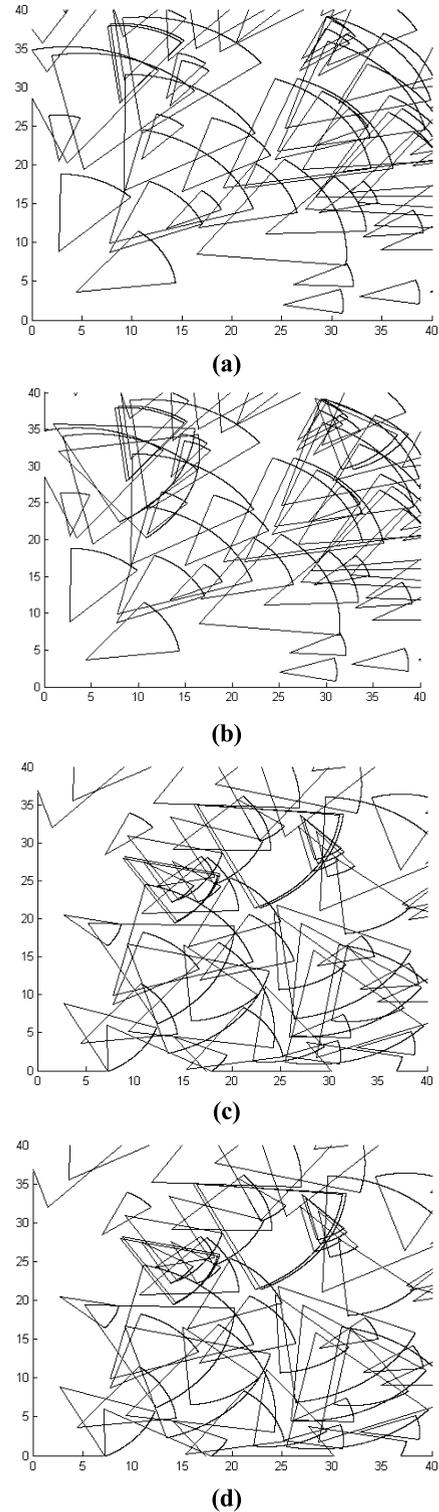


FIGURE 6. Illustration of the solutions obtained by the proposed algorithm. (a) 60 nodes, initial distribution, coverage rate 0.69; (b) 60 nodes, after adjusting boundary nodes, coverage rate 0.71; (c) 60 nodes, after adjusting angle of view, coverage rate 0.80; (d) 60 nodes, after changing physical positions, coverage rate 0.90.

the algorithm, the coverage rate improvement is substantial, which verifies the effectiveness of the proposed algorithm. To further investigate the coverage performance, the results of

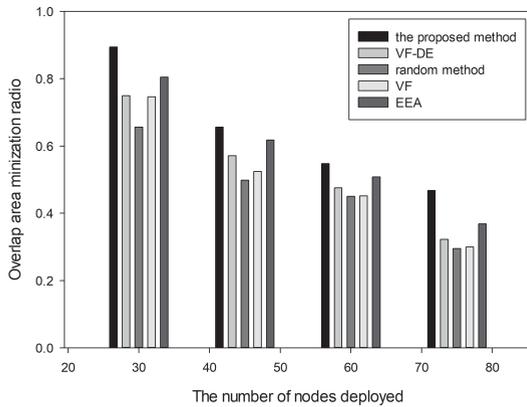


FIGURE 7. Overlap minimization performance comparison.

the overlap minimization ratio (OMR) versus the number of nodes are illustrated in Fig. 7, where each result is the average of 30 independent operations. The OMR can be calculated by Eq. (19):

$$OMR = \frac{A_{pre}^{method} - A_{after}^{method}}{A_{pre}^{method}} \quad (19)$$

where  $A_{pre}^{method}$  and  $A_{after}^{method}$  is the area obtained by the specific method (for example random deployment, VF, VF-DE, HMS, EEA and the proposed method) before and after the process of working direction adjustment, respectively.

As illustrated in Fig. 7, the proposed algorithm can effectively reduce the overlapped regions, and it performs the best among the tested algorithms. The reason for this is that the stepwise method and the function of marginal contribution make the proposed algorithm adjust the working direction in a precise and effective way.

## 2) ENERGY CONSUMPTION

To investigate the energy consumption of the nodes, the results of the increase in the energy consumption versus the number of nodes deployed are illustrated in Fig. 8. The energy consumption values are obtained by the methods proposed in the [30], [31], where a directional node consumes only 1.5 J during a 180°-rotation, while it takes 3.6 J energy in the process of a 1-m physical movement. Fig. 8 shows the simulation results with regard to the energy consumption. It is

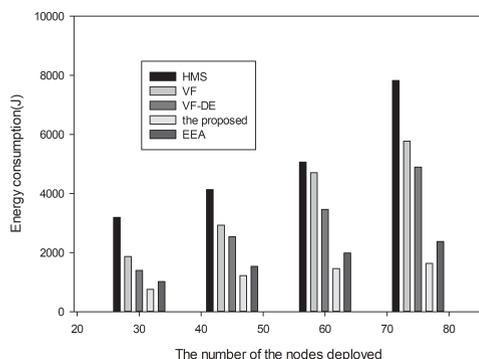


FIGURE 8. Total energy consumption vs. node density.

clear that with more nodes placed in the target area, more energy consumption is required for the process of relocating the nodes. Additionally, the proposed algorithm performs best among the compared algorithms, which is strong proof that the proposed algorithm is an energy-efficient algorithm. The reason is that the proposed algorithm attaches great importance to the rotating process instead of changing their physical positions. Moreover, in the phase of the physical position change, the improved version of the DE algorithm that adopts the self-adaptive mutation strategy, the random control parameter F and the good guidance of the VF in the proposed algorithm can greatly reduce the moving distance. Therefore, its energy consumption is the lowest among the compared algorithms.

## VII. CONCLUSION

In this paper, a coverage control algorithm has been presented to improve the coverage and energy computation of heterogeneous directional sensor networks. In contrast to existing algorithm that address the problems of homogeneous sensor nodes and exploiting motility or mobility only, the proposed method incorporates the motility and mobility of the nodes together to achieve the aims of coverage enhancement and energy reduction. The proposed method consists of two phases: the motility phase and the mobility phase. In the motility phase, the working directions of the heterogeneous sensor nodes are adjusted, and in the mobility phase, the physical positions of the sensor nodes are changed. The stepwise method and virtual force-directed self-adaptive differential evolution algorithm are, respectively, utilized in the two phases. The experimental results prove the effectiveness and efficiency of our proposed algorithm. In future work, extensions to the distributed version of the proposed method will be considered to reduce the energy consumption in the process of exchanging messages.

## REFERENCES

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.
- [2] M. Amac Guvensan and A. Gokhan Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1238–1255, Sep. 2011.
- [3] 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.
- [4] R. Özdağ and A. Karcı, "Probabilistic dynamic distribution of wireless sensor networks with improved distribution method based on electromagnetism-like algorithm," *Measurement*, vol. 79, pp. 66–76, Feb. 2016.
- [5] B. Wang, H. B. Lim, and D. Ma, "A survey of movement strategies for improving network coverage in wireless sensor networks," *Comput. Commun.*, vol. 32, nos. 13–14, pp. 1427–1436, Aug. 2009.
- [6] J. Huang, L. Sun, R. Wang, and H. Huang, "Improved virtual potential field algorithm based on probability model in three-dimensional directional sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 5, May 2012, Art. no. 942080, doi: 10.1155/2012/942080.
- [7] T.-W. Sung and C.-S. Yang, "Voronoi-based coverage improvement approach for wireless directional sensor networks," *J. Netw. Comput. Appl.*, vol. 39, pp. 202–213, Mar. 2014.

- [8] C. Yang, W. Zhu, J. Liu, L. Chen, D. Chen, and J. Cao, "Self-orienting the cameras for maximizing the view-coverage ratio in camera sensor networks," *Pervas. Mobile Comput.*, vol. 17, pp. 102–121, Feb. 2015.
- [9] S. M. B. Malek, M. M. Sadik, and A. Rahman, "On balanced k-coverage in visual sensor networks," *J. Netw. Comput. Appl.*, vol. 72, pp. 72–86, Sep. 2016.
- [10] H. P. Gupta, S. V. Rao, and T. Venkatesh, "Analysis of stochastic coverage and connectivity in three-dimensional heterogeneous directional wireless sensor networks," *Pervas. Mobile Comput.*, vol. 29, pp. 38–56, Jul. 2016.
- [11] C.-Y. Chang, C.-Y. Hsiao, and C.-T. Chang, "QoS guaranteed surveillance algorithms for directional wireless sensor networks," *Ad Hoc Netw.*, vol. 81, pp. 71–85, Dec. 2018.
- [12] A. A. Zishan, I. Karim, S. S. Shubha, and A. Rahman, "Maximizing heterogeneous coverage in over and under provisioned visual sensor networks," *J. Netw. Comput. Appl.*, vol. 124, pp. 44–62, Dec. 2018.
- [13] J. Jia, C. Dong, Y. Hong, L. Guo, and Y. Yu, "Maximizing full-view target coverage in camera sensor networks," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101973, doi: [10.1016/j.adhoc.2019.101973](https://doi.org/10.1016/j.adhoc.2019.101973).
- [14] C.-K. Liang, M.-C. He, and C.-H. Tsai, "Movement assisted sensor deployment in directional sensor networks," in *Proc. 6th Int. Conf. Mobile Ad-Hoc Sensor Netw.*, Dec. 2010, pp. 226–230.
- [15] C. K. Liang, C. Y. Chung, and C. F. Li, "A virtual force based movement scheme for area coverage in directional sensor networks," in *Proc. 10th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Kitakyushu, Japan, Aug. 2014, pp. 718–722.
- [16] M. A. Guvensan and A. G. Yavuz, "Hybrid movement strategy in self-orienting directional sensor networks," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1075–1090, May 2013.
- [17] F. Li, J. Luo, S. Xin, and Y. He, "Autonomous deployment of wireless sensor networks for optimal coverage with directional sensing model," *Comput. Netw.*, vol. 108, pp. 120–132, Oct. 2016.
- [18] X. Liu, B. Yang, and G. Chen, "Full-view barrier coverage in mobile camera sensor networks," *Wireless Netw.*, vol. 25, no. 8, pp. 4773–4784, Nov. 2019.
- [19] X. Zhu, J. Li, and M. Zhou, "Target coverage-oriented deployment of rechargeable directional sensor networks with a mobile charger," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5196–5208, Jun. 2019.
- [20] X. Lu, J. Liu, and H. Zhao, "Collaborative target tracking of IoT heterogeneous nodes," *Measurement*, vol. 147, Dec. 2019, Art. no. 106872.
- [21] X. Lu, Y. Zhang, J. Liu, F. Yuan, and L. Cheng, "Mobile target tracking algorithm for wireless camera sensor networks with adjustable monitoring direction of nodes," *Int. J. Commun. Syst.*, vol. 32, no. 10, Apr. 2019, Art. no. e3944, doi: [10.1002/dac.3944](https://doi.org/10.1002/dac.3944).
- [22] X. Wang, S. Wang, and J. Ma, "Dynamic deployment optimization in wireless sensor networks," in *Intelligent Control and Automation (Lecture Notes in Control and Information Sciences)*, vol. 344, no. 4, 2006, pp. 182–187, doi: [10.1007/978-3-540-37256-1\\_25](https://doi.org/10.1007/978-3-540-37256-1_25).
- [23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, Mar. 1986.
- [24] M. Yang, Y. Cao, L. Tan, and J. Yu, "A new self-deployment algorithm in hybrid sensor network," in *Proc. 2nd Int. Symp. Intell. Inf. Technol. Appl.*, Shanghai, China, vol. 2, Dec. 2008, pp. 268–272.
- [25] S. Li, C. Xu, W. Pan, and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," in *Proc. 7th Int. Conf. Inf. Fusion*, Jul. 2005, pp. 1629–1635.
- [26] R. G. mperle, S. D. M. Iler, and P. Koumoutsakos, "A parameter study for differential evolution," in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, A. Grmela and N. E. Mastorakis, Eds. Interlaken, Switzerland: WSEAS Press, 2002, pp. 293–298.
- [27] H. A. Abbass, "The self-adaptive Pareto differential evolution algorithm," in *Proc. Congr. Evol. Comput. (CEC)*, Honolulu, HI, USA: IEEE Press, 2002, pp. 831–836.
- [28] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," *Eng. Appl. Artif. Intell.*, vol. 90, no. 10379, Apr. 2020, Art. no. 103479. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095219762030004X>
- [29] J. Pournazari, M. Alaei, and F. Yazdanpanah, "An energy efficient autonomous method for coverage optimization in wireless multimedia sensor networks," *Wireless Pers. Commun.*, vol. 99, no. 2, pp. 717–736, Mar. 2018, doi: [10.1007/s11277-017-5142-y](https://doi.org/10.1007/s11277-017-5142-y).
- [30] D. O'Rourke, R. Jurdak, J. Liu, D. Moore, and T. Wark, "On the feasibility of using servo-mechanisms in wireless multimedia sensor network deployments," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, Zürich, Switzerland, Oct. 2009, pp. 826–833.
- [31] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: Enabling mobility in sensor networks," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Seattle, WA, USA, Apr. 2005, pp. 404–409.



MING LI received the B.S. degree in computer science and technology from Yantai University, Yantai, China, in 2004, the M.S. degree in application of computer from the Anhui University of Technology, Anhui, China, in 2008, and the Ph.D. degree in control theory and control engineer from Chongqing University, Chongqing, China, in 2011. Since 2014, he has been an Associate Professor with the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing. His research interests include wireless sensor networks and bio-inspired algorithms.



JIANGPING HU was born in Hubei, China, in 1977. He received the B.S. degree in applied mathematics and the M.S. degree in computational mathematics from Lanzhou University, China, in 2000 and 2004, respectively, and the Ph.D. degree in complex system modeling and control from the Academy of Mathematics and Systems Science (AMSS), Chinese Academy of Sciences (CAS), Beijing, China, in 2007. From September 2007 to August 2008, he held a postdoctoral position at the Royal Institute of Technology, Sweden. He is currently a Professor with the School of Automation Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China. His current research interests are in multi-agent systems, sensor networks, and air traffic control.



XIAOLI CAO was born in Chongqing, China, in 1970. She received the B.S. and M.S. degrees in pattern recognition from Chongqing University in 2001. She is currently a Professor with the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing. Her current research interests are wireless sensor networks and its application.