

---

# DISTRIBUTED DEPLOYMENT ALGORITHM BASED ON BOUNDARY EXPANSION AND VIRTUAL FORCE FOR MOBILE SENSOR NETWORKS

Guofang Nan\*, Zhongnan Chen\*, Minqiang Li\*, Liang Huang<sup>†‡</sup>, Ajith Abraham<sup>‡</sup>

---

**Abstract:** Optimization of sensors' position is a challenging problem in wireless sensor networks since the processing process significantly affects energy consumption, surveillance ability and network lifetime. Vectorbased algorithm (VEC) and Voronoi-based algorithm (VOR) are two existing approaches. However, VEC is sensitive to initial deployment, while VOR always moves to the coverage holes. Moreover, the nodes in a network may oscillate for a long time before they reach the equilibrium state. This paper presents an initially central deployment model that is cost effective and easy to implement. In this model, we present a new distributed deployment algorithm based on boundary expansion and virtual force (BEVF). The proposed scheme enables nodes to move to the boundary rapidly and ultimately reach equilibrium quickly. For a node, only the location of its nearby nodes and boundary information are needed in the algorithm, thereby avoiding communication cost for transmitting global information. The distance threshold is adopted to limit node movement and to avoid node oscillations. Finally, we compare BEVF with existing algorithms. Results show that the proposed algorithm achieves a much larger coverage and consumes lower energy.

Key words: *Wireless sensor networks, sensor deployment, network connectivity, virtual force, boundary expansion*

Received: September 12, 2013

DOI: 10.14311/NNW.2014.24.018

Revised and accepted: June 12, 2014

## 1. Introduction

Wireless sensor network [1] is an emerging technology in recent years. It consists of many sensor nodes with identical or different functions. The nodes have a certain capacity in various applications such as communications, data processing

---

\*Guofang Nan, Zhongnan Chen, Minqiang Li, Institute of Systems Engineering, Tianjin University, Tianjin, 300072, China, E-mail: [gfnan@tju.edu.cn](mailto:gfnan@tju.edu.cn), [z.n.chen2610@gmail.com](mailto:z.n.chen2610@gmail.com), [mqli@tju.edu.cn](mailto:mqli@tju.edu.cn)

<sup>†</sup>Liang Huang – Corresponding Author, School of Information & Electronic Engineering, Zhejiang Gongshang University, 310018, China, E-mail: [huangliang@zjgsu.edu.cn](mailto:huangliang@zjgsu.edu.cn)

<sup>‡</sup>Liang Huang, Ajith Abraham, Machine Intelligence Research Labs – MIR Labs, E-mail: [ajith.abraham@ieee.org](mailto:ajith.abraham@ieee.org)

and data storage, and are reasonably cheap. As an important issue in Wireless sensor networks (WSNs), optimizing sensor deployment [2] is critical for energy consumption, monitoring capability and network lifetime

The method of initially random deployment [3,4] is usually adopted in case of large number of sensor nodes. However, the uniform coverage in the target area is hard to achieve in randomly deployed nodes, which affects the efficiency of the whole network and monitoring capability of the target area. Therefore, the nodes need to be adjusted based on the designed optimization algorithm to achieve maximization and uniform coverage of the target area.

Several deployment strategies have been proposed in recent years. They can be divided into centralized and distributed approaches. Most centralized methods employ the distance threshold to compute the sensor movement, which cannot easily achieve a  $k$ -coverage scenario. In this case, any point in the target field should be covered by at least  $k$  sensors [5]. Moreover, a central server is necessary in a centralized manner. However, gathering information from sensor nodes to the server node and delivering information from the server node to other nodes are difficult, especially for random initial deployment scenarios [6]. Hybrid algorithms combine distributed and centralized algorithms to achieve the required coverage, which also needs a central server for information management and exchange. Whereas in most of the distributed algorithms, the nodes only need limited computing capabilities according to the location information and the dynamic information of the adjacent nodes to execute the algorithm, and independently make the corresponding adjustments. In addition, achieving  $k$ -coverage through a distributed manner is easy. In summary, a better distributed deployment algorithm for randomly initial deployment can be performed. The algorithm can also effectively improve the performance of the network, especially for applications wherein knowledge of the environment is unknown.

In a real environment, a random deployment in the central area is introduced based on the model in [8] because of the high cost and influence of the executing algorithm in the large-scale initial random deployment model [4,5,6,7]. The random deployment method is easy to carry out because the nodes can be deployed by the aircraft to certain coordinates or locations. Considering that all sensors in the central area are randomly and densely deployed, these nodes are assumed to easily receive boundary information of the target area through multi-hop communications from a special node. In fact, using only the initially random deployment scheme to achieve whole area coverage is difficult, especially in case of forests fire monitoring, large scale leakage of chemicals and toxic substances in industry, and battle environments. As a consequence the initially random deployment scheme cannot achieve real-time monitoring of the whole target region. Therefore, under the proposed random initial deployment model, we also present a distributed algorithm based on boundary expansion and virtual force (BEVF). The target region is divided into four sub-regions, and the sensors determine their movement by virtual forces based on boundary information and their neighbor nodes. The sensors also adjust their movement by a distance threshold, which reduces the oscillation of nodes and enables them to reach balance status to maximize sensor coverage. Thus, the nodes move away from a dense area to a sparse area. As such, uniform coverage distribution and maximization are ultimately achieved.

The rest of this paper is organized as follows: Section 2 presents some research works related to ours. In section 3, the system model and assumptions are discussed. Section 4 introduces the structure and process of the BEVF algorithm. In Section 5, we present the simulation and experimental results to demonstrate the efficiency of the system and compare it with other existing deployment algorithms. Finally, the advantages and disadvantages of the proposed scheme are discussed in Section 6, together with open research issues.

## 2. Related Work

Sensor deployment is a fundamental issue in WSN since the sensors' position affects coverage, communication costs and resource management. Sensor deployment problems with more practical considerations have also been studied in-depth over the last decade in a variety of scenarios. Previous studies related to this paper consist of centralized and distributed deployment strategies, together with coverage and connectivity issues.

### 2.1 Centralized Deployment

The virtual force algorithm (VFA) [5] presented by Zou and Chakrabarty is a centralized approach that divides a sensor network into clusters. Each cluster-head is responsible for collecting the position information of nodes, and computing their desired locations. New positions are chosen in such a way that the network coverage is maximized, and sensor movement is minimized. However, this method is based on very demanding assumptions including location-awareness and autonomous sensor mobility, which restrict its applications. Tan et al. extended the VFA to a connectivity-guaranteed and obstacle-adaptive version called CPVF [7].

The authors in [3] proposed a Hungarian algorithm-based centralized solution. In their study, a scan-based movement assisted deployment method (SMART) was used to divide the sensor network into clusters. Each cluster is then scanned to determine the new sensor locations in each stage. This solution can achieve load balancing and minimize the total moving distance of sensors. They also proposed a centralized optimal solution in [9] based on the Hungarian method for 2-D grid-based mesh application.

In [10], the authors proposed the solutions to sensor deployment and dispatch problems. In the sensor dispatch problem, a centralized approach and a distributed method were discussed to determine a set of mobile sensors to be moved to an area of interest with the desired coverage and connectivity properties. In the centralized solution, the sensor dispatch is converted to the maximum-weight maximum-matching problem based on the prior deployment results. The objective of this solution is to minimize the total energy consumption and maximize the average remaining energy of the sensors. Their solution allows arbitrarily shaped obstacles and an arbitrary relationship between sensor communication radius and its sensing radius. However, it cannot be used for k-coverage scenarios.

To extend the network lifetime, the region closer to the sink should be provided with higher density of sensors. Thus, the problem of sensor deployment to increase network lifetime was addressed in [11]. Three solutions including a centralized

integer-programming formulation, a localized matching method, and a distributed corona-radius scanning algorithm are addressed.

In [12], the author considered a two-tiered hierarchal heterogeneous wireless sensor network using the concept of clustering. The author formulated the sensor deployment problem as a multi-constraint optimization problem, and solved it through a binary integer linear programming (BILP) approach, greedy approach (GREEDY) and genetic algorithm (GA) approach. Actually, the use of more high-end sophisticated nodes with significantly additional resources in two-tiered sensor networks is not a cost effective solution. Moreover, fewer studies regarding connectivity, coverage and traffic constraints are discussed in the paper.

Some bio-inspired algorithms were proposed to solve the sensor deployment problems. In [13], the author applied a particle swarm optimization (PSO) approach to optimize the coverage in ad hoc sensor network deployment and to reduce cost through a clustering method. A self-organizing algorithm for enhancing the coverage and detection probability for sensor networks to solve hybrid sensor deployment was presented in [14] Their algorithm combined VFA with PSO, which is called virtual force-directed particle swarm optimization (VFPSO). The VFPSO algorithm regards each mobile node as a particle PSO was used to search the optimal deployment strategy, whereas VFA was used to direct the movement of particles. The authors also presented a distributed particle swarm optimization and simulated annealing for an energy-efficient coverage problem in [15], which considered sensing coverage and energy consumption in their work. They used a grid exclusion algorithm to evaluate the coverage and Dijkstra's algorithm to calculate energy consumption. However, their work did not consider obstacle exits

## 2.2 Decentralized Deployment

The field based deployment algorithm [16] processes the movement of each node by virtual force from other nodes and obstacles. Similar to the particles in the microscopic world, it allows all nodes to explore from a compact region and cover the whole monitoring area. The method can achieve maximum area coverage. However, it may take a long time for the network to reach the static equilibrium. Furthermore, this approach is also constrained due to its boundary restriction.

To reduce oscillation and save energy consumed by node movement, another distributed deployment protocol for MSNs based on Voronoi diagrams (VD) by Wang et al was discussed in [6] In their study, VD was used to discover coverage holes and maximize coverage area. The author also designed three algorithms including the vector-based algorithm (VEC), the Voronoi-based algorithm (VOR) and minimax algorithm to support move assisted sensor deployment with high network coverage and limited moving distance. All these algorithms assume that each node shall exchange its current location information with all other nodes in the network to acquire its corresponding Voronoi vertices and cell Hence, each sensor is required to maintain a large portion of information and consumes more communication cost during the round-by-round deployment phases.

A distributed deployment scheme was presented in [17] to achieve multilevel coverage of the area of interest. The scheme utilizes a competition-based scheme and a pattern-based scheme for the dispatch problem The former allows mobile

sensors to bid for their closest locations, whereas the latter allows sensors to derive the target locations on their own. This scheme can achieve a k-coverage scenario. However, there is an assumption that the initial deployment of the network is connected, which is usually not the case in practice.

Fagiolini et al addressed how mobile sensors with limited sensing capabilities cooperate and adjust their locations to maximize the covered area and minimize coverage holes [18]. They used a distributed motion algorithm based on an original extension of Voronoi tessellation.

Other fundamental issues related to sensor deployment are sensing coverage and network connectivity [19]. The relationship between coverage and connectivity was investigated in [20]. Both the necessary and sufficient conditions are improved to ensure that coverage implies connectivity. Moreover, the problem of maintaining sensor coverage and connectivity by keeping a minimum number of sensor nodes in the active mode is addressed. In [21], the m-coverage and n-connectivity problem under border effects is studied and the metric of sensing coverage and network connectivity is proven. In [22], the sensing coverage and network connectivity are combined into an optimization problem and solved by a single algorithm. The dynamic network configuration was designed and analyzed in [23,24] to achieve guaranteed degrees of coverage and connectivity. Bai et al. proposed an optimal deployment patterns (OPT) to achieve both coverage and connectivity in [25]

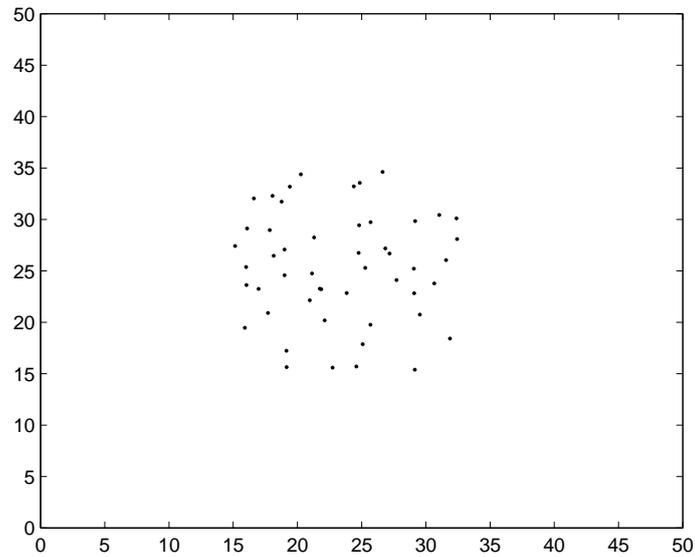
We conclude this section by observing that both the centralized deployment schemes and the distributed deployment strategies have their special applications. To quickly move nodes towards the boundary, as well as to improve the network lifetime and QoS performance, we present a new distributed algorithm based on boundary expansion and virtual force (BEVF).

### 3. System Model and Assumptions

We consider a network of mobile sensors that are randomly deployed over a small central area inside the sensing region of interest [8] to detect and track events, as shown in Fig. 1. In the WSN each sensor has the same role and relies on other sensors to relay its messages to the sink node via multi-hop radio communication [26]. Each point in the area is observed by at least one sensor at any time. The lower bound of the network 1-connectivity is also guaranteed. The objective of the deployment problem is to redeploy the mobile sensors to the whole region to achieve uniform distribution, as well as to minimize the total moving distance and energy consumption of the nodes. Meanwhile, the desired coverage and connectivity constrains should be satisfied.

We consider a monitored region  $A$ . A WSN can be modeled as a graph  $G = (S, E)$ , which consists of  $N$  mobile sensors  $S = \{s_1, s_2, \dots, s_N\}$ , a base station, and a set of wireless links  $E$ . All the sensors have the same communication range  $r_c$  and sensing range  $r_s$  [25]. As with many applications, the sensors can communicate with one another as long as the distance between any two sensors is less than  $r_c$ , and any target in the region can be detected by a sensor according to the binary detection model.

Assuming that sensor  $s_i$  is located at point  $(x_i, y_i)$ ,  $P$  is a point in the region and its coordinate is  $(x, y)$ , the Euclidean distance [14] between  $s_i$  and  $P$  can be denoted as  $d(s_i, P)$ , which is calculated as follows.



**Fig. 1** *Initially central deployment.*

$$d(s_i, P) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

The sensing coverage of a network can be determined by finding the union of the areas defined by the location of each sensor and its  $r_s$ . In the binary detection model [26], the probability of detecting the event of interest is one within the sensing range  $r_s$ . Otherwise, the probability is zero.

For the desired sensor network architecture based on the Voronoi diagram, we make the following assumptions:

- (1) All the nodes can communicate with their neighbors to achieve connectivity for the whole network after random deployment within a small area.
- (2) The sensing field is obstacle-free. Therefore, the nodes need not consider obstacle-avoiding issues.
- (3) Each node knows the boundary information and only needs to maintain the positions of its neighbors.

## 4. BEVF Algorithm

In this section, we present the definitions of boundary nodes and the boundary expansion force, and state the control approach to achieve uniform coverage and node oscillations by a distance threshold. We analyze the total force on the nodes due to their neighbors and boundaries, and then propose the system framework and distributed deployment algorithm (BEVF).

### 4.1 Boundary Nodes

Fig. 2 shows a Voronoi diagram [6] of the nodes that are deployed in the central area of the target field. To construct a Voronoi polygon [6, 27], each sensor node first calculates the bisectors of its neighbors and itself based on the location information, these bisectors and the boundary of the target field then form several polygons. The smallest polygon encircling the sensor is the Voronoi polygon of this sensor.

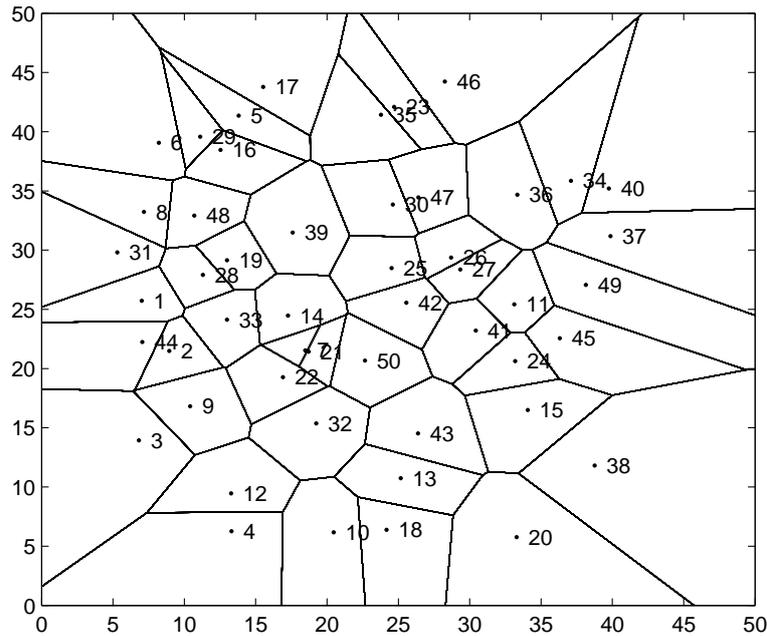


Fig. 2 Voronoi diagram.

As shown Fig. 2, these nodes can be divided into two types, namely, the boundary and internal nodes. Boundary nodes are on the boundary of convex or have at least one Voronoi vertex that falls on the boundary of target area. For example, boundary nodes include the nodes with numbers 3, 4, 10, 18, 20 and 38. Internal nodes are those that have no Voronoi vertex that fall on the boundary of target area. For instance, these include the nodes with numbers 39, 14, 42, 19, 13, and 43.

The nodes move to the boundary based on the moving length  $d_{max}$  (no longer than  $d_{max}$ ) [6] under the effect of boundary expansion force which means that the boundary nodes have the trends of expanding to the boundary by the attraction force exerted by the boundary. In Fig. 3,  $\vec{F}_{ib}$  is the combined force between the horizontal force  $\vec{F}_{ih}$  and vertical force  $\vec{F}_{iv}$ . Therefore, the nodes move toward the angle  $\theta$  under  $\vec{F}_{ib}$ . The angle  $\theta$  is calculated by

$$\theta = \arctan(|\vec{F}_{iv}|/|\vec{F}_{ih}|) \tag{2}$$

where  $|\vec{F}_{ih}|$  and  $|\vec{F}_{iv}|$  are the horizontal and vertical distance between the node to the boundary, respectively

The boundary expansion force is the force exerted by a definable boundary on the boundary nodes to achieve the expansion. The boundary expansion force makes the nodes rapidly move to the boundary and causes the nodes to achieve uniform deployment in the target region. Our BEVF algorithm divides the target area into four parts (Fig. 3). The boundary nodes can determine its location within the target area based on its location information, and calculating angle  $\theta$ , the angle is the combined force of the boundary expansion between the horizontal directions.

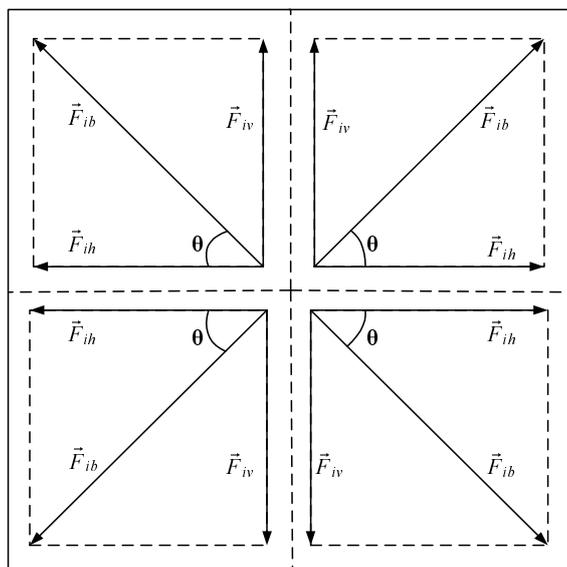


Fig. 3 The expansion force on boundary node.

#### 4.2 The Distance Threshold $d_{th}$

In BEVF, we use the distance threshold  $d_{th}$  that has been introduced in [5] to control the motion of nodes to achieve the uniform coverage, and to reduce the oscillation of nodes to ensure the stability of network. The distance threshold is related to sensing radius  $r_s$ .

The rules of node movement include the exertion of the virtually repulsive force to push each other to move to the sparse area if the distance between two nodes is less than the threshold  $d_{th}$ . However, if the distance is larger than the threshold, there is no virtual force to affect each other. That is, one sensor will exert repulsive force to the other when the distance between them is smaller than the threshold, and they do not affect each other when the distance between them is not smaller than the threshold. For instance, consider four sensors  $s_1, s_2, s_3$  and  $s_4$  as shown in Fig. 4, if we assume that  $d(s_1, s_4) = d_{th}$ ,  $d(s_1, s_2) < d_{th}$  and  $d(s_1, s_3) > d_{th}$ ,

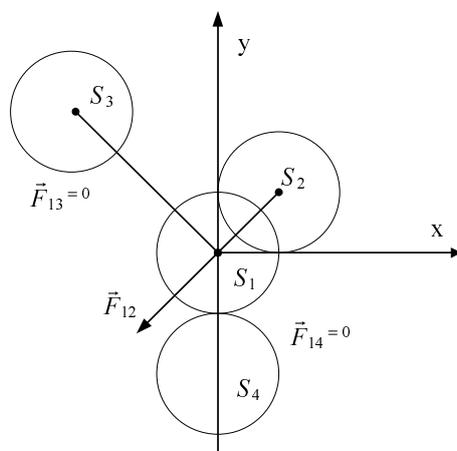


Fig. 4 An example of virtual forces between sensors.

$s_2$  will exert repulsive force on  $s_1$ , and no force from  $s_3$  and  $s_4$  will be exerted on  $s_1$ . Therefore, the nodes will be densely distributed if the distance between each two nodes is much smaller, such as  $d_{th} \leq r_s$ . Furthermore, the optimal network structure is achieved when the distance between two nodes is  $\sqrt{3}r_s$  [20]. Thus, the range of the threshold was set to  $r_s < d_{th} \leq \sqrt{3}r_s$ , as shown in Fig. 5(a).

When  $d_{th} \leq r_s$  as shown in Fig. 5(b), the nodes have less moving distance each time. This increases the unnecessary energy consumption and sensor movement, and more time is needed for the sensor network to achieve static equilibrium. In addition, more coverage overlaps and coverage holes will form in this case. Therefore, the performance of the sensor deployment is decreased and the sensors cannot reach uniform distribution when the proposed algorithm is terminated.

When  $d_{th} > \sqrt{3}r_s$  as shown in Fig. 5(c), the nodes are in an unstable state based on a much larger threshold. This also generates coverage holes and fails to achieve the maximization of the sensing coverage. Reaching equilibrium is difficult

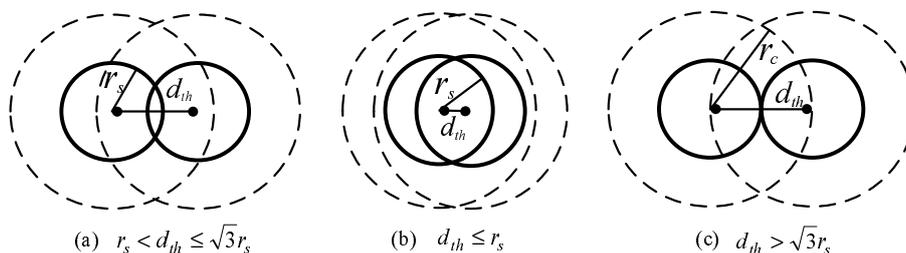


Fig. 5 Three cases of the relationship between  $d_{th}$  and  $r_s$ : the blue dotted lines represent the boundary of communication range with communication radius  $r_c$ , and the red solid lines represent the boundary of sensing range with sensing radius  $r_s$ .

for these nodes because the distance threshold is much larger and the nodes will move if the distance is less than the threshold. As such, the oscillations of the node are increased.

### 4.3 Analyzing the Forces on Nodes

If  $s_i$  is an interval node, it is only affected by virtual repulsive forces exerted by the neighbor nodes. However, if  $s_i$  is a boundary node, it is affected not only by the virtual repulsive forces from its neighbors, but by the expansion force from its nearest boundary. Therefore, the total force  $\vec{F}_i$  on  $s_i$  can now be expressed as

$$\vec{F}_i = \begin{cases} \sum \vec{F}_{ij} & s_i \in S_{inter} \\ \sum \vec{F}_{ij} + \vec{F}_{ib} & s_i \in S_{bound} \end{cases} \quad (3)$$

where  $\vec{F}_i$  is the combined force on  $s_i$ ,  $\vec{F}_{ib}$  is the force exerted by the boundary,  $S_{inter}$  is a set of interval nodes and  $S_{bound}$  is a set of boundary nodes.

Let  $b_i$  be the intersection if a line is drawn through  $s_i$  perpendicular to the nearest boundary line of  $s_i$ , and  $d(s_i, b_i)$  represents the distance between node  $s_i$  and the boundary which is nearest to the node  $s_i$ .  $\vec{b}_i s_i$  is a vector from the nearest boundary to the sensor node  $s_i$ , which is perpendicular to the nearest boundary,  $d_{max}$  is the moving distance of node  $s_i$  affected by the boundary expansion force each time.

If  $d_{th}/2$  is not larger than  $d(s_i, b_i)$ ,  $\vec{F}_{ib} = \vec{F}_{ih} + \vec{F}_{iv} = d_{max} \cdot (\cos \theta, \sin \theta)$  where  $(\cos \theta, \sin \theta)$  is a vector showing  $\theta$  degree with the horizontal direction as shown in Fig. 3  $d_{max} \cdot (\cos \theta, \sin \theta)$  means that node  $s_i$  will move  $d_{max}$  meters each time towards the direction of  $\theta$  degree with the horizontal direction. In other words, it calculates the size and the direction of the force  $\vec{F}_{ib}$ . So the boundary nodes are affected by the boundary expansion force and rapidly move toward the boundary.

If  $d_{th}/2$  is larger than  $d(s_i, b_i)$ ,  $\vec{F}_{ib} = \vec{F}_{ih} + \vec{F}_{iv} = (d_{th}/2 - d(s_i, b_i)) \cdot \vec{b}_i s_i / d(s_i, b_i)$ , where  $\vec{b}_i s_i / d(s_i, b_i)$  is the unit vector of direction  $\vec{b}_i s_i$ , and  $(d_{th}/2 - d(s_i, b_i))$  is the size of force. It means that the boundary will exert a repulsive virtual force to push the node away  $(d_{th}/2 - d(s_i, b_i))$  meters from the boundary, thus avoiding the node move outside the target area, and guaranteeing that it is much closer to the boundary. Therefore, the distance between any boundary node and its nearest boundary of the target field is almost  $d_{th}/2$  meters, which is effective for position adjustment for boundary nodes

$$\vec{F}_{ib} = \begin{cases} d_{max} \cdot (\cos \theta, \sin \theta) & d(s_i, b_i) \geq d_{th}/2 \\ (d_{th}/2 - d(s_i, b_i)) \cdot \vec{b}_i s_i / d(s_i, b_i) & d(s_i, b_i) < d_{th}/2 \end{cases} \quad (4)$$

In a  $50m \times 50m$  target field, we distributed 30 nodes Fig. 6(a) shows the tracks of nodes under the boundary expansion force in the first two rounds. It shows that the nodes move toward the angle  $\theta$  under the boundary expansion force based on the divided part in the target area. The circles represent the initial location of nodes and the boxes are the terminated locations of moving under the force. The

nodes rapidly and effectively move toward boundary based on the combined force to ensure the effectiveness and fast convergence of the proposed algorithm.

Let  $\vec{s_j s_i}$  represent the vector from node  $s_j$  to node  $s_i$ , and  $d(s_i, s_j)$  represent the distance between node  $s_j$  and node  $s_i$ . If the distance between node  $s_i$  and  $s_j$  is not less than the threshold  $d_{th}$ , there is no virtual force between node  $s_i$  and node  $s_j$ , that is,  $\vec{F}_{ij} = 0$  and  $\vec{F}_{ji} = 0$ . When the distance between node  $s_i$  and  $s_j$  is less than the threshold  $d_{th}$ , two different cases are discussed:

(a) If sensor  $s_i$  can completely cover all its Voronoi vertexes,  $\vec{F}_{ij} = 0$ , which shows that sensor  $s_j$  has no virtual force effect on sensor  $s_i$  and sensor  $s_i$  does not move. Meanwhile, sensor  $s_j$  will be pushed  $(d_{th} - d(s_i, s_j))$  meters away from sensor  $s_i$  towards the direction of unit vector  $\vec{s_i s_j} / d(s_i, s_j)$ . Thus, we can guarantee that the distance between any boundary node and its nearest boundary is almost  $d_{th}$  to achieve a uniform distribution that is

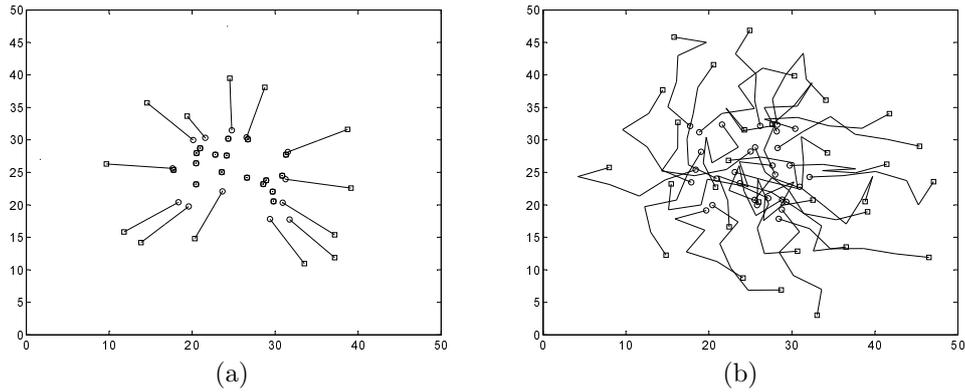
$$\vec{F}_{ji} = (d_{th} - d(s_i, s_j)) \cdot \vec{s_i s_j} / d(s_i, s_j) \quad (5)$$

(b) If sensor  $s_i$  cannot cover all its Voronoi vertexes, the virtual repulsive force is exerted by  $s_i$  to the sensor  $s_j$  and pushes the sensor  $s_j$  to move  $(d_{th} - d(s_i, s_j)) / 2$  meters towards the direction of the virtual force from sensor  $s_j$ . Meanwhile, sensor  $s_j$  exerts a virtual repulsive force to push the sensor  $s_i$  away  $(d_{th} - d(s_i, s_j)) / 2$  meters from sensor  $s_j$ . Thus, we can guarantee the distance between every two nodes is almost  $d_{th}$  to achieve a uniform distribution that is,

$$\vec{F}_{ij} = (d_{th} - d(s_i, s_j)) / 2 \cdot \vec{s_j s_i} / d(s_i, s_j) \quad (6)$$

$$\vec{F}_{ji} = (d_{th} - d(s_i, s_j)) / 2 \cdot \vec{s_i s_j} / d(s_i, s_j) \quad (7)$$

We deployed 30 sensors in a  $50m \times 50m$  target field. Fig. 6(b) shows the tracks of the nodes affected by the boundary expansion force and virtual force in the first



**Fig. 6** (a) The tracks of boundary nodes affected by the boundary expansion forces in the first two rounds, (b) The tracks of nodes affected by the combined forces in the first five rounds.

five rounds. The red circles represent the initial deployment locations whereas the blue boxes represent the terminated locations in round five. We can see that the nodes move toward the direction of the combined forces of the boundary expansion and the virtual forces based on its location information. The nodes constantly adjust their location based on the threshold  $d_{th}$  to push the nodes away from a dense area to a sparse area. Eventually, the network reaches a balance state and achieves a uniform distribution.

#### 4.4 Energy Consumption

In computing the energy consumption of the sensors' movement, we applied the model in [10] which represents the total energy consumption of all nodes:

$$E_i = \sum_j \Delta e \cdot d_{ij} \quad (8)$$

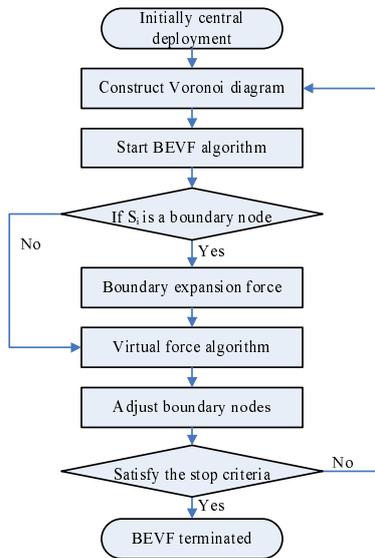
where  $\Delta e$  is the unit energy cost to move a sensor by a distance of one unit, and  $d_{ij}$  is the distance of moving node  $s_i$  in the  $j$  round.

#### 4.5 Application of BEVF in Sensor Deployment

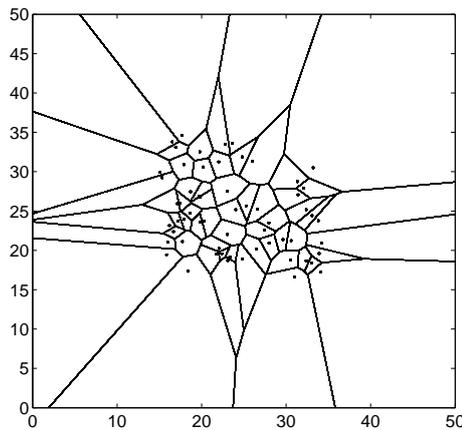
We executed our proposed algorithm using the following steps. To begin with, we deployed nodes in the central part of the target area. The nodes constructed their Voronoi polygons based on the network structure. If node  $s_i$  is the boundary node, then it is affected by the boundary expansion force and virtual force. However, if node  $s_i$  is not a boundary node, it is affected by the virtual forces exerted by the adjacent nodes. Meanwhile, to avoid the movement of the nodes outside the target area or to an area much closer to the boundary, position adjustment for boundary nodes is adopted. The proposed algorithm will be terminated when the system state satisfies the desired criteria. Otherwise, the algorithm will be executed once again until the state of the whole network satisfies the stop criteria. The process of the proposed algorithm is shown in Fig. 7.

Fig. 8 is a Voronoi diagram with 50 sensors deployed into the central area of a  $50m \times 50m$  target field. The initial coverage rate is 28.92% because the nodes are centrally deployed in the target area. Every node maintains its Voronoi polygon and its neighbor nodes are anomalous. Therefore the network is not uniformly covered.

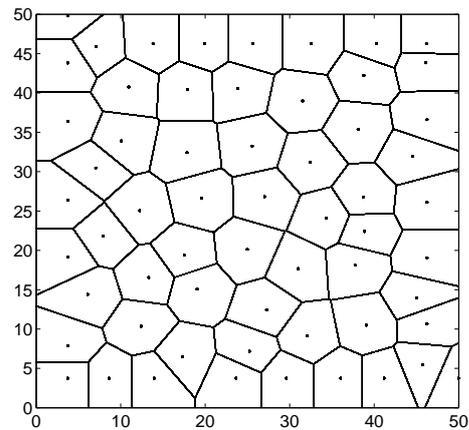
Fig. 9 shows a Voronoi diagram after executing our BEVF for 20 rounds. The coverage is 97.92%. We can see that all the Voronoi polygons are more regular compared with the initial case. The distance between a node and its neighbor is approximately equal to the distance threshold  $d_{th}$ , which shows the entire network has reached a relatively stable state. Each node in the network does not oscillate under the boundary expansion and the virtual force exerted to it. A uniform distribution and maximization of the coverage area are eventually achieved. A formal description of the BEVF algorithm is shown in Fig. 10.



**Fig. 7** The process of BEVF.



**Fig. 8** The initially random deployment.



**Fig. 9** BEVF 20 rounds.

## 5. Performance Evaluation

Our objectives in performance evaluation are twofold. The first is to evaluate the effectiveness of our algorithm in providing high coverage in the target area. The second one is to analyze the efficiency of the proposed algorithm in providing coverage in the target area, average moving distance and average energy consumption

---

BEVF Algorithm

Notation:  $b_i, d_{th}, d_{max}, d(s_i, s_j), d(s_i, b_i), S_{bound}, S_{inter}, \vec{F}_{ib}, \vec{F}_{ij}$  : as defined before  
 $V_i$  : the vertex set of a voronoi polygon of sensor  $s_i$   
 $N_i$  : the neighbor node set of the sensor  $s_i$   
 $A_c$  : the area needs to be central initial deployed in the target field

- (1) Deploy sensor nodes into  $A_c$
- (2) Entering Discovery phase
  - (2.1) sensors discover their voronoi polygon vertexes and neighbors
  - (2.2) broadcast hello to neighbors after a random time
- (3) Sensor Move()
 

for sensor  $s_i$ , if  $s_i \in S_{bound}$  Call Boundary Expansion Force()  
 if  $s_i \in S_{inter}$  Call Virtual Force()  
 calculate  $\vec{F}_i$ , calculate the new position of sensor  $s_i$
- (4) for sensor  $s_i$   
 sensor move (), and update its  $V_i$  and  $N_i$

Boundary Expansion Force()  
 if  $d(s_i, b_i) > d_{th}/2$   
 $\theta = \arctan(|\vec{F}_{iv}| / |\vec{F}_{ih}|)$ ,  $\vec{F}_{ib} = \vec{F}_{ih} + \vec{F}_{iv} = d_{max} \cdot (\cos \theta, \sin \theta)$   
 else  
 $\vec{F}_{ib} = (d_{th}/2 - (d_{th}/2 - d(s_i, b_i)) \cdot \vec{b}_i s_i) / d(s_i, b_i)$   
 /\*position adjustment for boundary nodes \*/  
 end

Call Virtual Force()  
 if  $d(s_i, s_j) \geq d_{th}$   
 $\vec{F}_{ij} = 0$  and  $\vec{F}_{ji} = 0$   
 else  
 if sensor  $s_i$  can completely cover all its Voronoi vertexes  
 $\vec{F}_{ij} = 0$ ,  $\vec{F}_{ji} = (d_{th} - d(s_i, s_j)) \cdot s_i s_j / d(s_i, s_j)$   
 else  
 $\vec{F}_{ij} = (d_{th} - d(s_i, s_j)) / 2 \cdot \vec{s}_i s_i / d(s_i, s_j)$ ,  $\vec{F}_{ji} = (d_{th} - d(s_i, s_j)) / 2 \cdot \vec{s}_j s_j / d(s_i, s_j)$   
 end

end

---

Fig. 10 Pseudocode of the BEVF.

of the nodes. By comparing our BEVF with the existing algorithms based on Voronoi diagram, we are able to show the characteristics and the superiority of our algorithm in some aspects.

We evaluate BEVF from the following aspects: (1) Comparing with VEC and VOR in terms of coverage, average moving distance and average energy consumption in different nodes density. (2) The effects of distance threshold  $d_{th}$  in terms of coverage, average moving distance and average energy consumption in different nodes density.

We randomly distributed four different numbers of sensors in the central area in a  $50m \times 50m$  obstacle-free target field. The number of sensors ranged from 40 to 70, in increments of 10 sensors. We set the sensing range  $r_s = 5$  which is a little smaller than that in [6]. The communication range is set as  $r_c = 10$  which is twice of  $r_s$ . The distance threshold  $d_{th}$  is set as 7. Finally, the energy consumption for one sensor to move by a distance of one unit is set to  $\Delta e = 2$ .

### 5.1 Sensing Coverage

Fig. 11 shows the final coverage of the three algorithms in different numbers of sensors under a randomly central deployment model. We can see that the coverage is greatly increased by all three algorithms compared to the initial random distribution. For example, in the case where 60 sensors are deployed, the three algorithms can increase the coverage rate to more than 95% from 30.4%. When 70 sensors are deployed in the target area, the VOR increase the coverage rate to more than 97%, and BEVF increases the coverage rate to 100%.

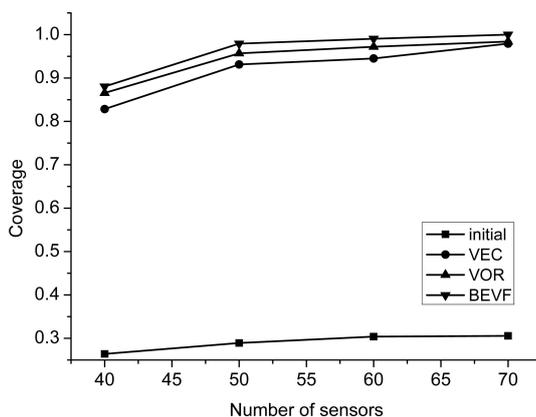


Fig. 11 Coverage comparison.

Fig. 12 shows the coverage in each round and convergence when the number of sensors is 50. We can see that the coverage rate by the three algorithms quickly increased during the first several rounds under the proposed random deployment

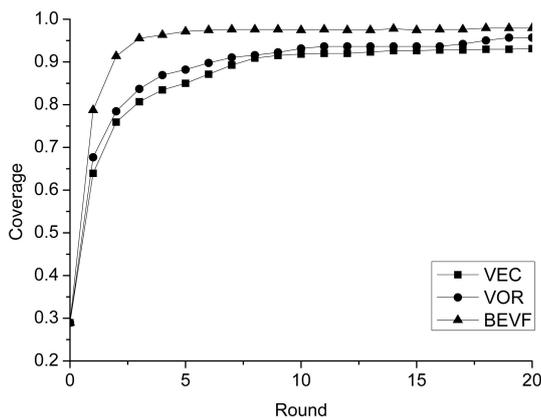


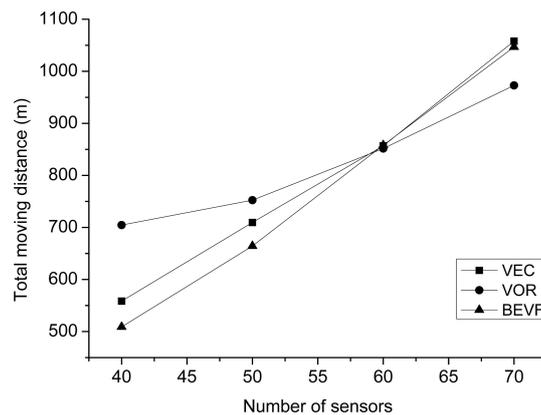
Fig. 12 Coverage and Convergence.

model. In addition, the coverage rate by BEVF and VOR increases much faster than VEC. For example, after 10 rounds, the coverage rate by the three algorithms achieved at least 90%. After five rounds, the coverage rate generated by BEVF is more than 95%.

Among VOR and VEC, VEC performs worse in coverage in all cases of nodes density in [6], and the speed of covering is faster than VEC. It can be seen from Fig. 9 and Fig. 10 that between BEVF and VOR, BEVF performs better in coverage and convergence time. The primary reason is sensitivity of the VEC to the initial deployment. Assuming a situation in which sensors are located in the same line with equal spacing, no sensor will move. Thereby, large holes will be generated. VOR performs better than VEC because it always detects the holes in its Voronoi polygon, and moves toward the holes to heal them without generating new holes. The nodes in the network may oscillate for a long time before they reach the static equilibrium state, which makes the network unstable. Among the three algorithms, BEVF performs best because it can achieve a maximization of coverage in all cases of all nodes density, and it is not sensitive to the initial deployment. From the coverage curve, we can see that the proposed algorithm can rapidly achieve a uniform distribution, and the coverage rate can reach 96% in the first five rounds. Therefore, we can conclude that BEVF can achieve the best coverage performance than VEC and VOR.

## 5.2 Moving Distance and Energy Consumption

The performance of these three approaches described in the previous sections is studied in this section in terms of moving distance and energy consumption. Fig. 13 and Fig. 14 show the total moving distance and average moving distance in different numbers of sensors, respectively. The total moving distance by the three algorithms increase as the number of sensor nodes is also increased, and the total moving distance is in proportion to the nodes density.



**Fig. 13** *The total moving distance.*

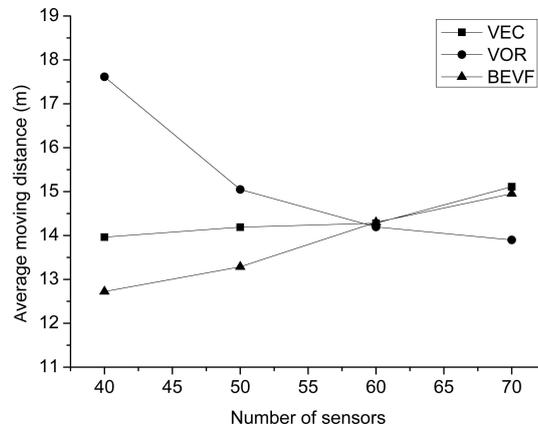


Fig. 14 The average moving distance.

However, the average moving distances has great differences. We can see from Fig. 13 that the average moving distance by VEC only slightly increased with the increase in node density. Under the proposed initial deployment model, the average distance between the two nodes decreases as the number of nodes increase, which also demonstrates that VEC is not sensitive to the node density [6] and the effect of node density on the average moving distance has no obvious difference.

VOR always detects coverage holes in its Voronoi polygon, and moves toward the farthest vertex of polygon with a moving distance of  $d_{\max}$ , unless the hole is healed. As the node density decreases, the size of the coverage holes becomes larger, which leads to more movements of the nodes to heal the holes. For example, when the number of nodes is 40, the average moving distance of VOR is 17.62. As the node density increases in the target area, the coverage holes decreases, and the average moving distance becomes smaller as well. In the case of  $N = 70$ , the average moving distance is 13.89, which shows that the moving distance by VOR is sensitive to the node density.

In our BEVF, the boundary nodes rapidly move toward the boundary, and the internal nodes self-organize to achieve a uniform distribution under the virtual force. The number of boundary nodes decreases because the node density in the target area is low. Therefore, the average moving distance of the nodes is smaller. For example, when the number of nodes is 40, the average moving distance of BEVF is 12.72. As the node density increases, the number of boundary nodes which can move toward the boundary increases as well. Therefore, the average moving distance increases compared to VOR and VEC. The network reaches a balance state because it is affected by the existing virtual force between the nodes. The network avoids node oscillations and is more stable by BEVF than that by VOR.

Fig. 15 and Fig. 16 show the total energy consumption and average energy consumption, respectively, in different numbers of nodes by the three algorithms. We can see that the total energy consumption of BEVF is the lowest and that

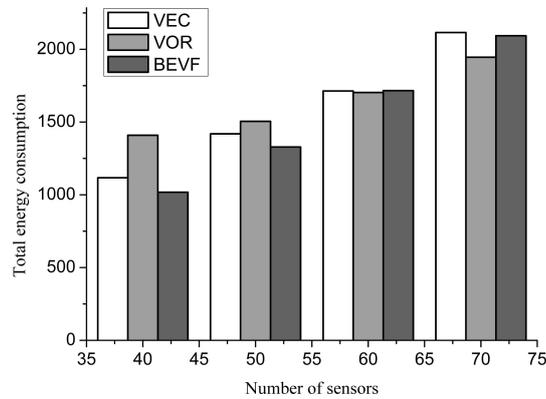


Fig. 15 The total energy consumption.

of VOR is the highest when the nodes density is smaller in Fig. 15. That is because the nodes in the target field will not move if the network comes to stable state, the nodes in VOR always detects coverage holes in its Voronoi polygon, and moves toward the farthest vertex of polygon with a moving distance of  $d_{max}$ , unless the hole is healed. Thus, VOR consumes more energy than BEVF in lower node density. In higher node density mode, though the total energy consumption of VOR is lower than BEVF, BEVF performs better in the coverage and convergence time than VOR. In addition, the energy consumption of BEVF is lower than that of VEC.

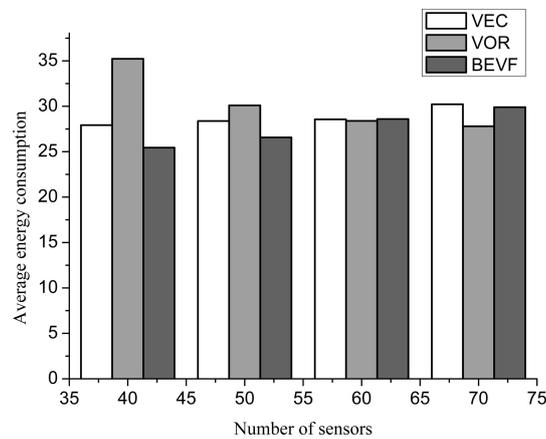


Fig. 16 The average energy consumption.

From Fig. 16 we can see that the average energy consumption of VOR decreases as the nodes density increases, and becomes the smallest among that of three algorithms when the number of nodes is 70. The average energy consumption is

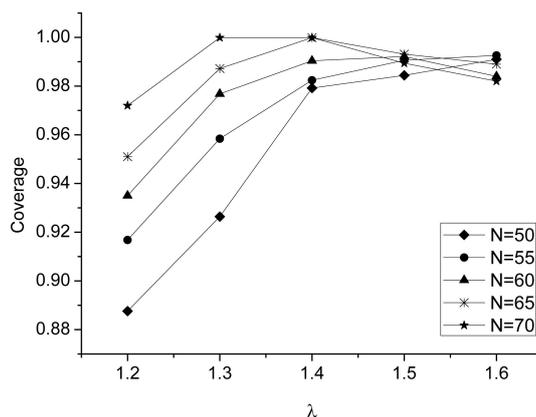
similar in most cases because VEC is not sensitive to the nodes density. BEVF performs the best in lower node density, and the average energy consumption increases smoothly which is much lower than that of VEC in all cases.

Based on the above discussion, we can come to the conclusion that the average moving distance and average energy consumption of BEVF is the smallest in a node with low density. BEVF also obtains a better sensing coverage performance in different nodes densities, as described in Section 5.1.

### 5.3 The Effect of $d_{th}$ on BEVF Performance

From Section 4.2, we know that the threshold  $d_{th}$  must meet the condition of  $r_s < d_{th} \leq \sqrt{3}r_s$ , that is  $1 < d_{th}/r_s \leq \sqrt{3}$ . We set  $\lambda = d_{th}/r_s$ , and the proposed algorithm terminates at round 20.

From Fig. 17 we can see that when the threshold is controlled within a certain range, the sensing coverage increases as the nodes density is increased. After the threshold exceeds the range, the sensing coverage decreases as the nodes density is increased. In the case of  $\lambda = 1.4$ , the sensing coverage is approximately the optimal one in most cases. For example, when the number of nodes is 50, the sensing coverages of  $\lambda = 1.3$ ,  $\lambda = 1.4$  and  $\lambda = 1.6$  are 92.64%, 97.92%, and 99.1%, respectively. When the number of nodes is 60, the sensing coverages of  $\lambda = 1.3$ ,  $\lambda = 1.4$  and  $\lambda = 1.6$  are 97.68%, 99.04%, and 98.4%, respectively. When the number of nodes is 70, the coverages of  $\lambda = 1.3$ ,  $\lambda = 1.4$  and  $\lambda = 1.6$  are 99.99%, 99.99%, and 98.2%, respectively. The primary reason is that a larger threshold can affect the balance of the network. The nodes will move as the distance between the two nodes is smaller than the threshold. Moreover, as the nodes density increases, the distance between two nodes decreases. Therefore, the distance between two nodes may be smaller than the threshold with nodes of high density, where the balance state under the combined forces is difficult to reach. As the value of the threshold increases, the number of unstable nodes also increases. These nodes will move in



**Fig. 17** The effect of  $d_{th}$  ( $\lambda$ ) on coverage.

each round, thus, the network cannot easily reach a stable state. Although the maximization of coverage can be reached, it will lead to an unstable network and cause the deployment process to fail in ultimately meeting the system requirements.

Fig. 18 and Fig. 19 show the effects of the threshold  $d_{th}$  on the average moving distance and the average energy consumption in different nodes density respectively. We can see that when the nodes density is determined, the average moving distance and the average energy consumption increase as the value of  $\lambda$  is also increased. For instance, if the number of nodes is 50, the average moving distances of  $\lambda = 1.2$ ,  $\lambda = 1.3$  and  $\lambda = 1.5$  are 12.55, 12.99 and 15.43 respectively. The average energy consumptions are 25.101, 25.98 and 30.86, respectively. When the number of nodes is 60, the average moving distances of  $\lambda = 1.2$ ,  $\lambda = 1.3$ , and  $\lambda = 1.5$  are 13.48, 14.30

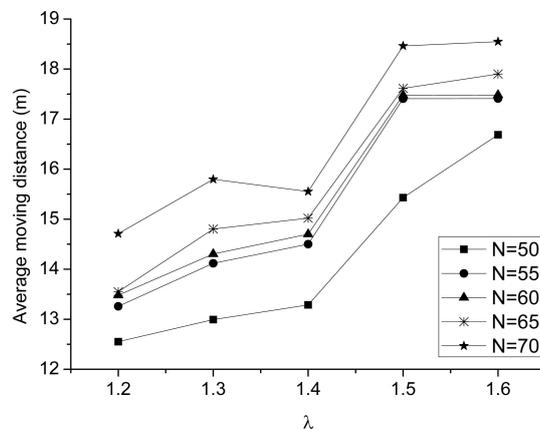


Fig. 18 The effect of  $d_{th}(\lambda)$  on average moving distance.

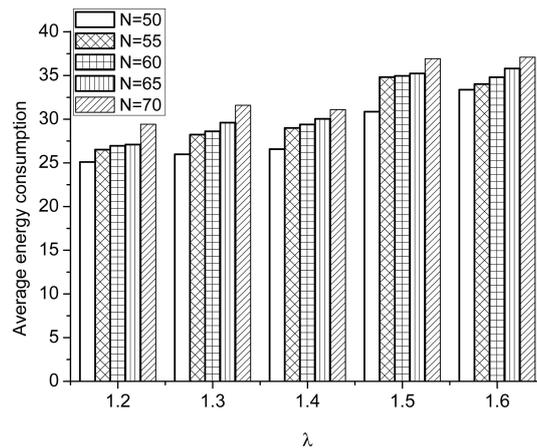


Fig. 19 The effect of  $d_{th}(\lambda)$  on average energy consumption.

and 17.48 respectively. The average energy consumptions are 26.96, 28.61 and 34.96 respectively. When the number of nodes is 70, the average moving distances of  $\lambda = 1.2$ ,  $\lambda = 1.3$ , and  $\lambda = 1.5$  are 14.71, 15.80 and 18.46, respectively. The average energy consumptions are 29.42, 31.59 and 36.93 respectively. The reason is that the number of moving nodes increases as the threshold is also increased for a certain node density, which thereby increases the total moving distance and the total energy consumption, as shown in Fig. 20 and Fig. 21.

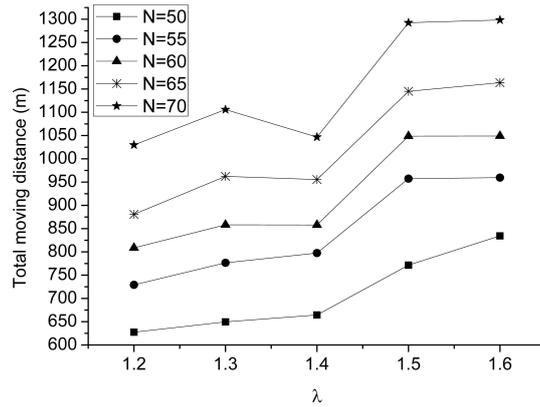


Fig. 20 The effect of  $d_{th}(\lambda)$  on total moving distance.

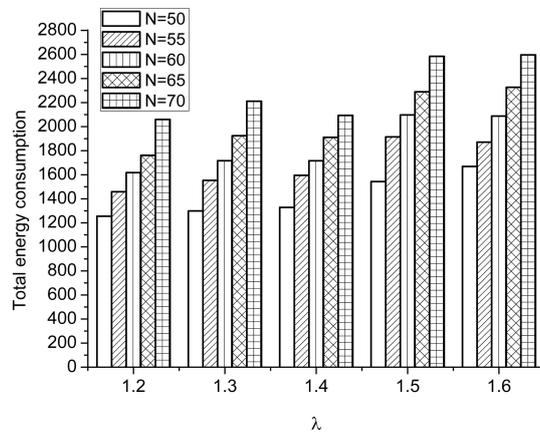


Fig. 21 The effect of  $d_{th}(\lambda)$  on total energy consumption.

We can also see from Fig. 18 and Fig. 19 that the average energy consumption and average moving distance of nodes increase smoothly by a small amount when  $\lambda$  is smaller than 1.4, while they increase sharply when  $\lambda$  is larger than 1.4. Because the increase of  $\lambda$  implies the increase of  $d_{th}$ , the smaller threshold ( $\lambda < 1.4$ ) reduces

the unnecessary move of the nodes and make it easy for a WSN to be in stable state, thereby reducing sensor move and energy consumption while larger threshold ( $\lambda > 1.4$ ) increases the unnecessary move of the nodes and make it hard for a WSN to be in stable state. Thus, combined with the analysis of effect of  $d_{th}$  ( $\lambda$ ) on coverage,  $\lambda = 1.4$  is effective in terms of coverage performance and energy consumption.

## 6. Conclusion

This paper addressed the problem of sensor deployment to achieve coverage performance after a random deployment of sensors into the central area of a large-scale target field. Considering the high cost and inconvenience in the traditional deployment approaches, we introduce an initially random deployment model in the central area of target field, which is not only easy to carry out but cost effective as well. For the proposed deployment model, we also present a distributed algorithm based on boundary expansion and virtual force, which only needs the location information of each node and the boundary. By making full use of the data structure and characteristics of the Voronoi diagram, the motion of the nodes is determined by the angle of the combined force and horizontal direction, the moving length and the distance threshold

Simulation results demonstrate the effectiveness of our BEVF in terms of sensing coverage and energy consumption by comparing it with the existing algorithms based on Voronoi diagram. The proposed algorithm can achieve a maximum sensing coverage, the lowest moving distance and energy consumption in scenarios involving nodes with low density. In the case when the nodes have a higher density, the proposed algorithm can achieve a complete coverage with lower energy consumption. Therefore, our BEVF can achieve the optimal deployment than VEC and VOR. Simulation also discusses the impact of the distance threshold  $d_{th}$  on sensing coverage, average moving distance and average energy consumption.

Our future work will focus on the application of the proposed algorithm to a more complicated environment Our network model will be extended to process any arbitrary obstacles in the target region. We will optimize the destinations of the mobile sensors from the perspective of the overall network to achieve the minimization of the moving distance and energy consumption. We will also derive the relationship of the distance threshold with coverage level and the lower bound of sensors required to achieve an  $m$ -coverage and  $n$ -connectivity scenario.

## Acknowledgements

This research was supported by research grants from the National Science Foundation of China under Grant No.71271148 and 71071105

## References

- [1] Ci S., Guizani M., Sharif H.: Adaptive clustering in wireless sensor networks by mining sensor energy data. Computer Communications, 30, (14-15), 2007, pp. 2968-2975.

- [2] Chen G., Nan G.: Optimization of sensor Deployment for Mobile Wireless Sensor Networks International conference on Computational Intelligence and Vehicular System, 2010, pp. 218-221.
- [3] Yang S., Li M., Wu J.: Scan-Based Movement-Assisted Sensor Deployment Methods in Wireless Sensor Networks IEEE Transactions on Parallel and Distributed Systems 18, (8), 2007, pp. 1108-1121.
- [4] Ma M., Yang Y.: Adaptive Triangular Deployment Algorithm for Unattended Mobile Sensor Networks. IEEE Transactions on Computers, 56, (7), 2007, pp. 946-958.
- [5] Zou Y., Chakrabarty K.: Sensor Deployment and Target Localization Based on Virtual Forces. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, 2, 2003, pp. 1293-1303.
- [6] Wang G., Cao G., Porta L. T. F.: Movement-Assisted Sensor Deployment. IEEE Transactions on Mobile Computing, 5, (6), 2006, pp. 640-652.
- [7] Tan G., Jarvis S. A., Kermarrec A.M.: Connectivity-Guaranteed and Obstacle-Adaptive Deployment Scheme for Mobile Sensor Networks. IEEE Transaction on Mobile Computing, 8, (6), 2009, pp. 836-848.
- [8] Bartolini N., Calamoneri T., Fusco E. G., Massini A., Silvestri S.: Push and Pull: autonomous deployment of mobile sensors for a complete coverage Wireless Networks, 16, (3), 2010, pp. 607-625.
- [9] Wu J., Yang S.: Optimal movement-assisted sensor deployment and its extensions in wireless sensor networks Simulation Modelling Practice and Theory 15, (4), 2007, pp. 383-399.
- [10] Wang Y. C., Hu C. C., Tseng Y. C.: Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network IEEE Transactions on Mobile Computing, 7, (2), 2008, pp. 262-274.
- [11] Cardei M., Yang Y., Wu J.: Non-Uniform Sensor Deployment in Mobile Wireless Sensor Networks. Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2008, pp. 1-8.
- [12] Pandey S., Dong S., Agrawal P., Sivalingam K. M.: On Performance of Node Placement Approaches for Hierarchical Heterogeneous Sensor Networks. Mobile Networks and Applications, 14, (4), 2009, pp. 401-414.
- [13] Wu X., Shu L., Yang J., Xu H., Cho J., Lee S.: Swarm Based Sensor Deployment Optimization in Ad Hoc Sensor Networks. Lecture Notes in Computer Science, December, 3820, 2005, pp. 533-541.
- [14] Wang X., Wang S., Bi D.: Virtual Force-Directed Particle Swarm Optimization for Dynamic Deployment in Wireless Sensor Networks. Lecture Notes in Computer Science, 4681, 2007, pp. 292-303.
- [15] Wang X., Ma J. J., Wang S., Bi D. W.: Distributed Particle Swarm Optimization and Simulated Annealing for Energy-efficient Coverage in Wireless Sensor Networks. Sensors, 7, (5), 2007, pp. 628-648.
- [16] Howard A., Mataric M. J., Sukhatme G. S.: Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems, 2002, pp. 299-308.
- [17] Wang Y. C., Tseng Y. C.: Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multilevel Coverage IEEE Transactions on Parallel and Distributed Systems, 19, (9), 2008, pp. 1280-1293.
- [18] Fagiolini A., Tani L., Bicchi A., Dini G.: Decentralized Deployment of Mobile Sensors for Optimal Connected Sensing Coverage Lecture notes in Computer science, 5067, 2008, pp. 486-491.
- [19] Ghosh A., Das S. K.: Coverage and connectivity issues in wireless sensor networks: A survey Pervasive and Mobile computing, 4, (3), 2008, pp. 303-334.
- [20] Zhang H., Hou J.C.: Maintaining sensing coverage and connectivity in large sensor networks. Wireless Ad Hoc and Sensor Networks, 1, (1-2), 2005, pp. 89-123.

- [21] Jin Y., Wang L., Jo J. Y., Kim Y., Yang M., Jiang Y.: EECCR: An Energy-Efficient m-Coverage and n-Connectivity Routing Algorithm Under Border Effects in Heterogeneous Sensor Networks. *IEEE Transactions on Vehicular Technology*, 58, (3), 2009, pp. 1429-1442.
- [22] Gupta H., Zhou Z., Das S.R., Gu Q.: Connected sensor cover: Self-organization of sensor networks for efficient query execution. *IEEE/ACM Transactions on Networking* 14, (1), 2006, pp. 55-67.
- [23] Wang X., Xing G., Zhang Y., Lu C., Pless R., Gill C.: Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks *ACM Conference on Embedded Networked Sensor Systems* 2003, pp. 28-39.
- [24] Ammari H. M., Das S. K.: Integrated Coverage and Connectivity in Wireless Sensor Networks: A Two-Dimensional Percolation Problem. *IEEE Transactions on Computers* 57(10) 2008, pp. 1423-1434.
- [25] Bai X., Kumar S., Xuan D., Yun Z., Lai T. H.: Deploying wireless sensors to achieve both coverage and connectivity. *ACM international symposium on Mobile ad hoc networking and computing*, 2006, pp. 131-142.
- [26] Liu C., Wu K., Xiao Y., Sun B.: Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 17, (6), 2006, pp. 562-575.
- [27] Wang G., Cao G., Porta T. L.: Movement-Assisted Sensor Deployment. *23th Annual Joint Conference of the IEEE Computer and Communications Societies*, 4, 2004, pp. 2469-2479.