

Article

# Local Coverage Optimization Strategy Based on Voronoi for Directional Sensor Networks <sup>†</sup>

Guanglin Zhang <sup>1,2,\*</sup>, Shan You <sup>1,2</sup>, Jiajie Ren <sup>1,2</sup>, Demin Li <sup>1,2</sup> and Lin Wang <sup>3</sup>

<sup>1</sup> College of Information Science and Technology, Donghua University, Shanghai 201620, China; 2141077@mail.dhu.edu.cn (S.Y.); renjiajie@mail.dhu.edu.cn (J.R.); deminli@dhu.edu.cn (D.L.)

<sup>2</sup> Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Shanghai 201620, China

<sup>3</sup> Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; wanglin@sjtu.edu.cn

\* Correspondence: glzhang@dhu.edu.cn; Tel.: +86-21-6779-2332

<sup>†</sup> This paper is an extended version of our conference paper published in: You, S.; Zhang, G.L.; Li, D.M. Coverage improvement strategy based on Voronoi for directional sensor networks. In Proceedings of the EAI International Conference on Machine Learning and Intelligent Communications, Shanghai, China, 27–28 August 2016.

Academic Editor: Yu Wang

Received: 31 August 2016; Accepted: 13 December 2016; Published: 18 December 2016

**Abstract:** In this paper, we study the area coverage of directional sensor networks (DSNs) with random node distribution. The coverage of DSNs depends on the sensor's locations, the sensing radiuses, and the working directions, as well as the angle of view (AoV), which is challenging to analyze. We transform the network area coverage problem into cell coverage problems by exploiting the Voronoi diagram, which only needs to optimize local coverage for each cell in a decentralized way. To address the cell coverage problem, we propose three local coverage optimization algorithms to improve the cell coverage, namely Move Inside Cell Algorithm (MIC), Rotate Working Direction Algorithm (RWD) and Rotation based on boundary (RB), respectively. Extensive simulations are performed to prove the effectiveness of our proposed algorithms in terms of the coverage ratio.

**Keywords:** area coverage; Voronoi diagram; wireless sensor networks (WSNs); directional sensor networks (DSNs); angle of view (AoV)

## 1. Introduction

With the rapid development of information technologies, wireless sensor networks (WSNs) are now playing an increasingly important role in environment monitoring, disaster rescuing target tracking, and industrial process control, etc. Generally, the sensor nodes are deployed in a target area to collect environment information for further processing. Such a complex network calls for novel approaches for system design to make it operate efficiently.

The area coverage problem of WSNs is important in both theory and application. As a main measurement of the network quality of service (QoS), the coverage problem attracts a lot of attention and is a hot research topic [1,2]. Previous works focused on the omnidirectional sensor networks, where the sensor's working coverage area is a circle. However, with the increasing demand and more diverse applications, WSNs have some disadvantages in terms of information access, e.g., in the applications of image/video sensing.

Due to its flexibility with rotation and lower energy consumption, directional sensor networks (DSNs) have been widely investigated in the literature. For a DSN, the directional sensor is limited to a certain working direction and the angle of view (AoV), such as camera sensors, infrared sensors, and so on. Different from omnidirectional WSNs, in which the sensing coverage mainly depends on sensing radius ( $R_s$ ) and sensor location, in DSNs, sensing coverage is affected not only by the

location and sensing radius ( $R_s$ ) but also by the working direction and AoV. Furthermore, under the condition of random deployment, directional sensors can improve coverage dynamically by adjusting their working directions. Moreover, compared with the WSNs, the nodes in DSNs have the ability of directional sensing, which results in a lower energy consumption and mutual interference. Recently, the enormous application potential of DSNs attracts lots of attention from researchers [3–6]. In [3,4], the authors investigate how to utilize virtual force between adjacent sensors to improve the coverage. In [5,6], Voronoi tessellation is adopted to tackle the coverage problem. In our previous work [7], we utilize the combination of node's mobility and motility to increase coverage.

In this paper, we adopt a Voronoi diagram to investigate the coverage problem of DSNs with random node deployment. The main characteristics of such an approach are: (i) by the construction of a Voronoi diagram for the random deployed sensors, each Voronoi cell contains one sensor; (ii) every sensor stores all of the vertices' information of the Voronoi cell in which it is located, which is used for decision-making on node moving and rotation; and (iii) every sensor node's actions (i.e., moving or rotation) should be within its related Voronoi cell.

The main contributions of this paper are as follows:

1. We transform the network area coverage problem into cell coverage problems by exploiting the Voronoi diagram, which only needs to optimize local coverage for each cell. The proposed approach solves the problem in a decentralized way.
2. By adopting a node's moving or rotation actions, we propose three local coverage optimization algorithms to improve the cell coverage, i.e., Move Inside Cell Algorithm (MIC), Rotate Working Direction Algorithm (RWD), Rotation based on boundary (RB), respectively.
3. We use extensive simulations to prove the effectiveness of our proposed algorithms in terms of the coverage ratio. Specifically, compared to the benchmark algorithm Distributed Voronoi-Based Self-Redeployment algorithm (DVSA) proposed in [8], our algorithm MIC shows a shorter moving distance and lower energy consumption.

The rest of this paper is organized as follows: we present the literature review in Section 2. In Section 3, we present the system model and problem statement. In Section 4, we show the theoretical analysis and propose the related algorithm. In Section 5, we give the performance evaluation results to demonstrate the effectiveness of our proposed algorithms. Conclusions and future works are shown in the final section.

## 2. Related Work

Many studies have been done on area coverage in both WSNs and DSNs [9,10]. For example, in [11,12], the authors studied the coverage of hybrid WSNs. They employed a Voronoi diagram to address the coverage problem. Specifically, they use static sensors to estimate coverage holes, work out assisted sensor's positions in every cell, and order mobile sensors to heal coverage holes. Liang et al. [13] presented two distributed self-deployment schemes of mobile sensors, namely circumcenter-based and incenter-based. The first scheme constructs a Voronoi diagram by the circumcenter of the sensor's sensing sector. If a sensor does not get full coverage, then a sensor moves until the circumcenter coincides with the centroid of its Voronoi cell. The second scheme uses a similar pattern, and the main difference is that it employs an incenter of the sensor's sensing sector to construct a Voronoi diagram.

In [3,4], the authors presented coverage enhanced algorithms based on virtual force between sensors. Tao et al. [4] presented a potential field based coverage-enhancing algorithm (PFCEA). PFCEA mainly takes a centroid of the sensing sector area as a stressed point, and the sensor adjusts its working direction by a resultant force from neighbor sensors. Liang et al. [3] also employed virtual force, and the difference is that they added a Voronoi diagram based on [4] and proposed a scheme that consists of four different forces caused by neighboring sensors and uncovered regions in the field, namely, the Centroid Push Auxiliary point Force (CPAF), the Centroid Push Centroid Force (CPCF),

the Voronoi point Pull Centroid Force (VPCF), and the Neighbor Repulsive Force (NRF). The main idea of their work is to adjust working direction based on resultant force of sensor. Lin et al. [14] proposed two enhanced deployment algorithms, EDA-I and EDA-II, respectively. EDA-I adopts the concept of virtual force to instruct the sensor to move or rotate, and EDA-II constructs the Voronoi diagram by the centroid of the sensor's fan. If the sensor does not cover all Voronoi vertices surrounding its cell, then the sensor moves until the centroid of the fan coincides with the cell's centroid. Actually, in EDA-II, it is hard to guarantee that all vertices surrounding cells are covered by the sensor itself, which means that all sensors need to move in the scheme of EDA-II. In addition, they proposed scheme Virtual Boundary Torques (VBT) to solve the problem of sensors whose location is outside the boundary. However, they did not solve the problem of whose location is within the boundary but effective working area towards outside the border.

Wen et al. [8,15] adopted Voronoi and Delaunay triangulation theory to design a distributed self-redeployment coverage enhancement algorithm. In [8], they proposed DVSA based on the idea that calculates corresponding field angles and side lengths of each vertex side in each polygon, and then compares AoV and  $R_s$  with field angles and side lengths. If there is no difference between the results, then a sensor moves to that vertex and takes the longer side as one boundary of the sensing sector. In [15], they presented several coverage increment algorithms, vertex-based adjustment with Voronoi diagram (V-VD), edge-based adjustment with Voronoi diagram (E-VD), edge-based adjustment with Delaunay triangulation (E-DT), and angle-based adjustment with Delaunay triangulation (A-DT). In E-VD, the sensor chooses the midpoint of the farthest edge of its own cell as a working direction. In E-DT, the authors utilized sensors to construct Delaunay triangulation, the sensor chooses the midpoint of the farthest edge of its own triangulation as a working direction, and, in A-DT, the sensor chooses the vertex with the maximum angle as the working direction.

In this paper, we study the area coverage of DSNs in the case of random deployment. The sensor makes a decision to adjust working direction and update location based on the vertex information of its Voronoi cell. The goal is to get local maximal coverage in the cell in which the sensor is located.

### 3. System Model and Problem Statement

#### 3.1. DSN Sensing Model

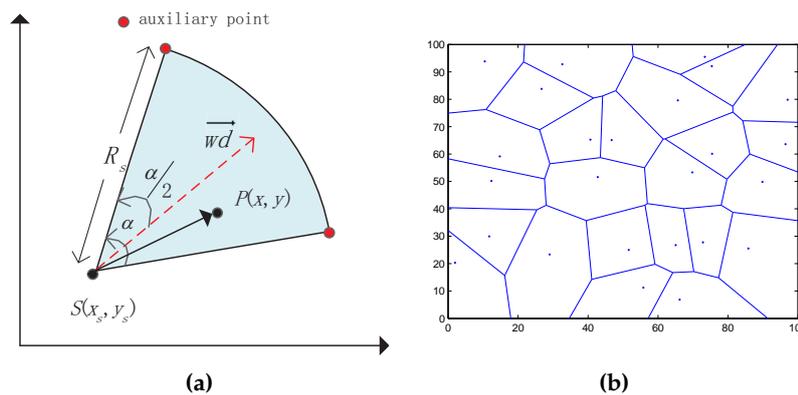
As shown in Figure 1a, a directional sensor is characterized by the directional working direction  $\omega$  ( $-\pi \leq \omega \leq \pi$ ) and the AoV  $\alpha$ . A unit vector  $\vec{wd}$  is defined as the working direction which splits the angle  $\alpha$ . The effective coverage field of sensor is the sector area  $\alpha R_s^2 / 2$ . We introduce two auxiliary points  $a_l$ , and  $a_r$  to help sensors make decisions. We said that a point  $p(x, y)$  is covered by a sensor  $s(x_s, y_s)$ , if the following two conditions are satisfied:

1. Euclidean distance between  $p$  and  $s$  must be smaller than sensing radius  $R_s$ , i.e.,  $d(s, p) \leq R_s$ :

$$d(s, p) = \sqrt{(x - x_s)^2 + (y - y_s)^2} \leq R_s. \quad (1)$$

2. The absolute included angle between  $\vec{wd}$  and  $\vec{sp}$  must be smaller than  $\alpha/2$ :

$$\phi = \arccos \frac{\vec{wd} \cdot \vec{sp}}{\|\vec{sp}\|} \leq \frac{\alpha}{2}. \quad (2)$$



**Figure 1.** Sensing model of directional sensor networks (DSN) and Voronoi diagram (a) Sensing model; (b) Voronoi diagram.

### 3.2. Voronoi Diagram and Some Assumptions

A Voronoi diagram is an important data structure in computational geometry that is widely used in many fields [16,17]. The main properties we need to use in this paper are that: (1) every sensor node is located at the Voronoi cell that is constructed in the initiation and any point  $p$  within the  $s_i$ 's cell has the shortest distance between sensor  $s_i$  and point  $p$  compared to other sensors, i.e.,  $d(s_i, p) \leq d(s_j, p)$ ; (2) a point in a shared Voronoi edge of two sensors has the same distance from two sensors; and (3) each sensor knows the vertex coordinate of its corresponding cell according to the related distributed/localized Voronoi algorithm. Given  $N$  sensors  $s_1, s_2, \dots, s_N$ , we divide it into  $N$  cells based on the Voronoi principle. As shown in Figure 1b, we choose 25 random nodes as an example to generate a Voronoi diagram.

We make the following assumptions in this paper to simplify the problem analysis:

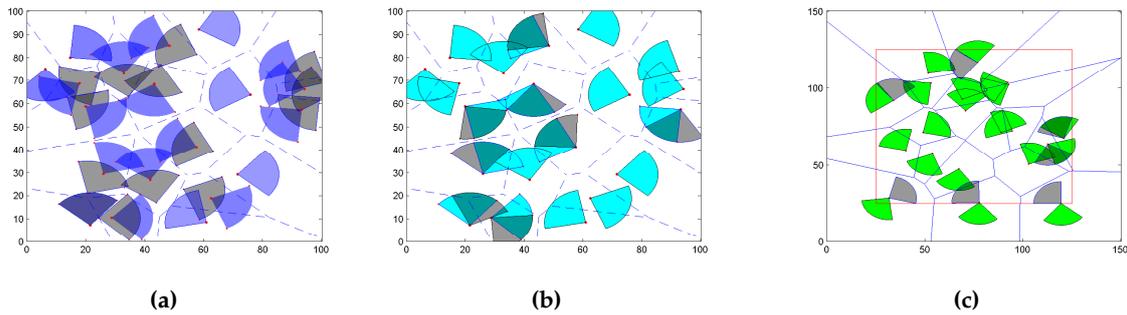
- All of the directional sensors are homogeneous, that is to say, every sensor has the same sensing radius  $R_s$ , viewing angle  $\alpha$ , rotation ability and mobility.
- We can obtain an accurate coordinate of every sensor node by using Global Position System (GPS) or other alternative localization algorithms such as DV-hop, Amorphous, etc.
- All of the sensor nodes have strong transmission ability to ensure the network connectivity and Voronoi diagram constructed successfully.

### 3.3. Problem Statement

We aim to maximize the effective coverage for the target area by utilizing a sensor node's rotation and moving action in the DSNs. The main difficulties that we face is that there exists coverage holes and overlap between a sensor node's sensing area. That is to say, we try to control the sensor node's actions in terms of rotation and movement to reduce the overlap and enlarge the effective coverage. We assume that the sensor's mobility and working direction adjustment should satisfy the following conditions.

Here, we give a simple exhibition of coverage algorithms that we have designed as shown in Figure 2. Figure 2a illustrates the mobility deployment, Figure 2b illustrates the rotation deployment, and Figure 2c illustrates the boundary repair deployment, and the grey area is the status after algorithms. The key part of our designed algorithms lies in taking advantage of the Voronoi information to adjust working directions or update locations. The main goal of a node's movement or rotation is to let the sensor get full coverage (sector working area is completely in the cell) in its own cell and reduce overlap with other sensors:

- Restrict the sensor node's moving trace in its corresponding cell.
- Get maximal coverage in each cell and minimum overlap with another cell's sensor fan.
- Maximize the cell coverage under the constraints of minimized moving distance.



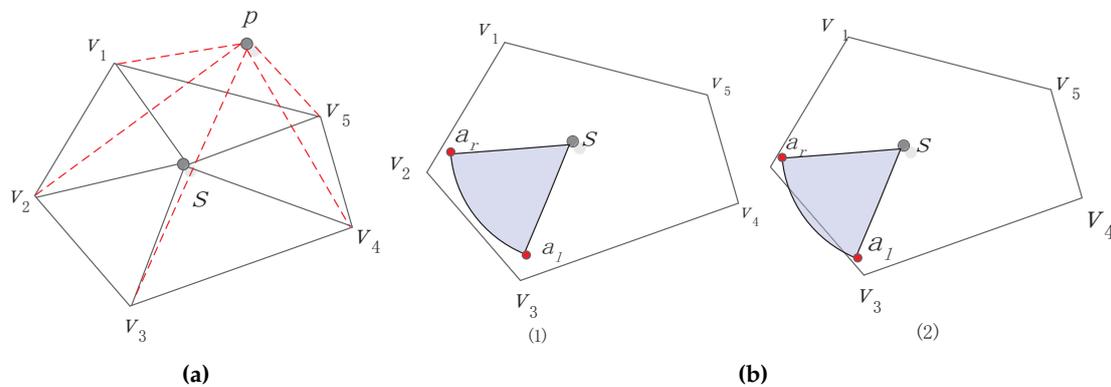
**Figure 2.** Example of mobility and rotation. (a) Move Inside Cell Algorithm (MIC); (b) Rotate Working Direction Algorithm (RWD); (c) Rotation based on boundary (RB).

**4. Theoretical Analysis and Algorithms**

In this section, we propose the theoretical analysis and the corresponding algorithms. We take the advantage of the Voronoi tessellation to divide the network area into Voronoi cells based on the locations of deployed sensors. However, the initial coverage is not optimal in the case of random node distribution, Here, we consider whether the sensor in each cell gets the maximum coverage. If not, the sensor will update location or adjust  $\vec{wd}$ .

*4.1. To Determine Whether a Sensor’s Sector Area is Wrapped in the Cell (Gets Full Coverage)*

If the sensor  $s$  and its auxiliary points  $a_l$  and  $a_r$  are all inside the polygon as shown in Figure 3(b1) and Figure 3(b2), then we say that the sensor  $s$  gets full coverage. In addition, the sensor does not need to move or rotate.



**Figure 3.** Proof point in convex polygon and sector wrapped in convex polygon. (a) Point in polygon; (b) sector wrapped in convex polygon.

To prove that sensor  $s$ ’s working area is wrapped in the cell, we can prove that points  $s$ ,  $a_l$  and  $a_r$  are all in the cell, which is a convex polygon. To prove whether one point (assuming  $p$ ) is in the convex polygon, we employ triangle segmentation to compute the convex polygon area  $S_{v_s}$  and the area  $S_{pv_s}$  constructed by  $p$  and  $V$ , as Figure 3a demonstrates. Here,  $v_s$  denotes a vertex set of sensors:

$$S_{v_s} = \sum_{v_i=1}^{V_s-1} S_{\Delta sv_i v_{i+1}} \quad S_{pv_s} = \sum_{v_i=1}^{V_s-1} S_{\Delta pv_i v_{i+1}}, \tag{3}$$

where  $S_{v_s}$  represents the area of each cell (convex polygon) and  $S_{pv_s}$  represents the area of polygon constructed by the certain point  $p$  and cell vertex. We use Equation (3) to judge whether one point is

in the cell. If  $S_{v_s} = S_{pv_s}$ , then we can know that the point  $p$  is inside the convex polygon. Based on this idea, if two auxiliary points  $a_l$  and  $a_r$  of sensor  $s$  are both in the polygon, we can infer that the sensor gets relatively full coverage, although not completely, as shown in Figure 3(b2). If  $S_{v_s} \neq S_{pv_s}$ , which means that the sensor does not get full coverage in its own cell, then the sensor needs to move or adjust  $\vec{wd}$ .

4.2. Move and Rotate inside the Cell Based on the Vertex

If one auxiliary point ( $a_l$  or  $a_r$ ) of the sensor node is not inside the cell, the location or working direction will adjust. We choose the farthest vertex (assume  $v_1$ ) of the polygon as the  $s$ 's initial working direction, which makes the sensor node get full coverage with a high chance. We find that this decision will lead to minimum moving distance or the rotation angle getting full coverage. Let us look at Figure 4a. First, the sensor selects the farthest vertex  $v_1$  as the working direction  $\vec{sv}_{1,1}$ , and there are two situations here:

4.2.1. Case 1: Rotate Working Direction Algorithm (RWD)

$R_s \leq d(s, v_1)$  and only one auxiliary point ( $a_r$  or  $a_l$ ) is inside its own cell.

For this situation, we can rotate the working direction by  $\theta$  to get the aim of letting the  $a_l$ 's position update to  $a_l'$ . Rotating  $\theta$  may lead  $a_r$  to be out of the polygon, as Figure 4b shows. Therefore, we should primarily figure out  $\theta_{max}$  (see Figure 4c), which is the maximal rotation angle. If  $\theta \leq \theta_{max}$ , the sensor rotates the working direction by  $\theta$ . If not, it rotates by  $\theta_{max}$ .

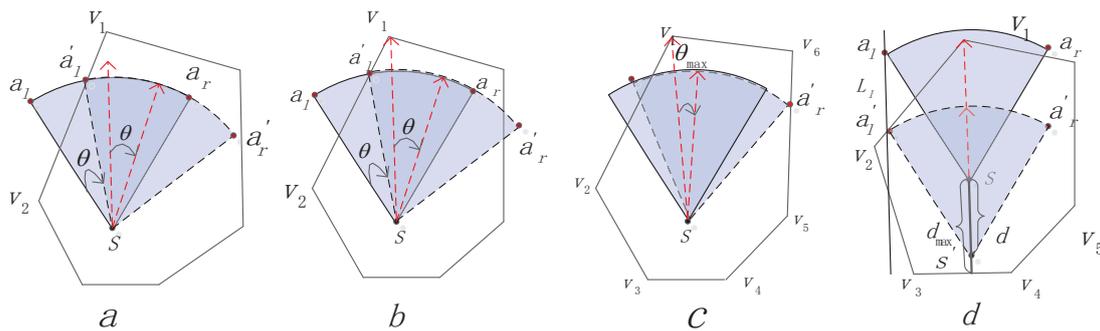


Figure 4. Move and rotate inside the cell. (a) rotate; (b) rotate  $a_r$  leads  $a_r'$ ; (c) compute the  $\theta_{max}$ ; (d) compute  $d_{max}$  and  $d$ .

We state how to compute  $\theta$  and  $\theta_{max}$  in the following: according to  $s(x_s, y_s)$  and the corresponding angle  $\omega$  of  $\vec{wd}$ , we can get the included angle  $\beta$  of  $L_{sa_l'}$  relative to the coordinate axis and coordinate of  $a_l'$ :

$$\psi = \arctan\left(\frac{y_{v_1} - y_s}{x_{v_1} - x_s}\right) \quad (4)$$

if  $\psi \leq 0$ ,

$$\begin{cases} \omega = \psi & x_{v_1} \geq x_s, y_{v_1} \leq y_s \\ \omega = \psi + \pi & x_{v_1} < x_s, y_{v_1} > y_s \end{cases} \quad (5)$$

if  $\psi \geq 0$ ,

$$\begin{cases} \omega = \psi & x_{v_1} \geq x_s, y_{v_1} \geq y_s \\ \omega = \psi + \pi & x_{v_1} < x_s, y_{v_1} < y_s \end{cases} \quad (6)$$

if sensor rotates right,

$$\begin{cases} \beta = \omega + \frac{\alpha}{2} - \theta & \omega \geq 0 \\ \beta = \omega - \frac{\alpha}{2} + \theta & \omega < 0 \end{cases} \quad (7)$$

if sensor rotates left,

$$\begin{cases} \beta = \omega - \frac{\alpha}{2} + \theta & \omega \geq 0 \\ \beta = \omega + \frac{\alpha}{2} - \theta & \omega < 0 \end{cases} \quad (8)$$

We can get the  $\theta$  by solving Equation (9):

$$\begin{cases} x_{a'_1} = x_s + R_s \cos(\beta) \\ y_{a'_1} = y_s + R_s \sin(\beta) \\ L_{v_1v_2} = \frac{y_{v_1} - y_{v_2}}{x_{v_1} - x_{v_2}}(x - x_{v_1}) + y_{v_1} \end{cases} \quad (9)$$

In a similar way, as shown in Figure 4c, we can solve  $\theta_{max}$  according to the above method, and the only difference is that we combine  $a'_r$  with line  $L_{v_5v_6}$ .

#### 4.2.2. Case 2: Move Inside Cell Algorithm (MIC)

Two auxiliary points of the sensor  $s$  are both outside the polygon, Figure 4d.

For this situation, the sensor cannot achieve the goal of getting full coverage if only by rotation, so the sensor should move, and the moving direction and moving distance are primarily under consideration. We choose the reverse direction of  $\vec{sv_1}$  as the moving direction and take  $a_1$ , which is far from polygon boundary, as the reference point. If  $a_1$  is moved into the polygon, then  $a_r$  is also in the polygon. Moving distance  $d = d(a_1, a'_1) = d(s, s')$ , as shown in Figure 4d. If  $d \leq d_{max}$ , the sensor moves by  $d$ . If not, it moves by  $d_{max}$ . If we can solve the coordinate of  $a'_1$ , then the moving distance  $d = d(a_1, a'_1)$ . Each node stores the vertex information of its own cell according to the related Voronoi generation method.  $a'_1$  is the intersection of line  $L_1$  and  $L_{v_1v_2}$ . Then, if we can find out the equations of line  $L_1$  and  $L_{v_1v_2}$ , the problem will be solved. Constructed calculation of  $d$  and  $d_{max}$  is as below:

$$\begin{cases} x_{a_1} = x_s + R_s \cos(\omega + \frac{\alpha}{2}) \\ y_{a_1} = y_s + R_s \sin(\omega + \frac{\alpha}{2}) \\ L_1 = \frac{y_{v_1} - y_{v_s}}{x_{v_1} - x_{v_s}}(x - x_{a_1}) + y_{a_1} \\ L_{v_1v_2} = \frac{y_{v_1} - y_{v_2}}{x_{v_1} - x_{v_2}}(x - x_{v_1}) + y_{v_1} \end{cases} \quad (10)$$

We can get the coordinate of  $a'_1$  by combining  $L_1$  with  $L_{v_1v_2}$ . Similarly,  $d_{max}$  can be worked out by combining  $L_{ss'}$  with  $L_{v_3v_4}$ . The updated sensor location  $(x'_s, y'_s)$  is:

$$x'_s = \begin{cases} x_s - d_{max} \cos(\omega) & d_{max} \leq d \\ x_s - d \cos(\omega) & d_{max} > d \end{cases} \quad (11)$$

$$y'_s = \begin{cases} y_s - d_{max} \sin(\omega) & d_{max} \leq d \\ y_s - d \sin(\omega) & d_{max} > d \end{cases} \quad (12)$$

As shown in Figure 5, there are several major steps of the local coverage optimization algorithm, and the most worthwhile point is that not only can MIC and RWD run independently, but they can also run in combination with RB, and, just as later simulations reveal, the sensor employs the movement followed by rotation and finally calls the RB.

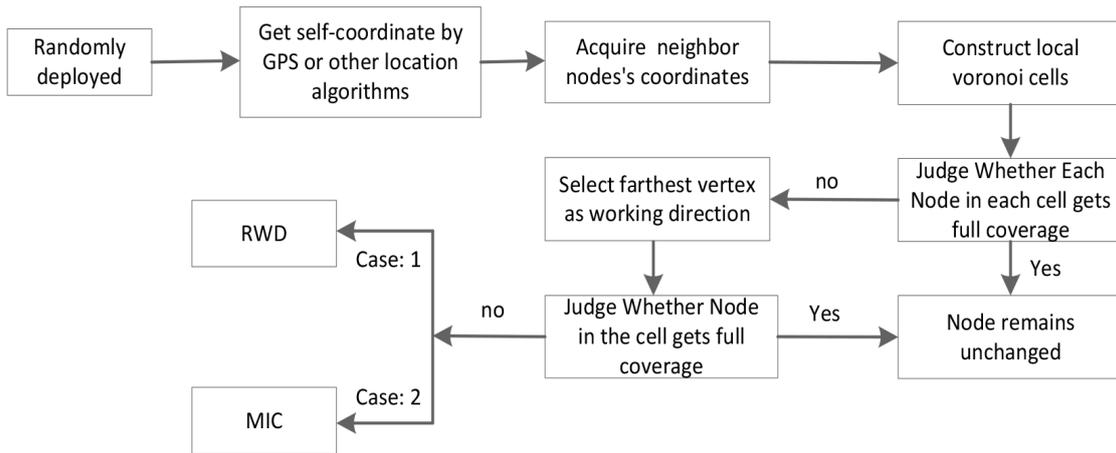


Figure 5. Procedures of the proposed decentralized local coverage optimization algorithm.

---

#### Algorithm 1 Rotate Working Direction Algorithm (RWD)

---

```

1:  $S_{s_i}$  : the area of the sensor.
2:  $S_{v_{s_i}}$  : denotes convex polygon area of the current sensor
3: Input: location  $(X, Y)$  and  $\vec{wd}$ .
4: Output: location and  $\vec{wd}$ .
5: Initialization: Construct local Voronoi cell
6: while  $i \leq N$  do
7:   if  $S_{s_i} \in S_{v_{s_i}}$  then
8:     save  $s_i(x_s, y_s)$  and corresponding  $\vec{wd}$ 
9:   else
10:    select  $s_i v_{farthest}$  as  $\vec{wd}$ 
11:    if  $S_{s_i} \in S_{v_{s_i}}$  then
12:      save  $s_i(x_s, y_s)$  and corresponding  $\vec{wd}$ 
13:    else if  $R_s \leq d(s_i, v_{farthest})$  &&  $(a_i \notin S_{v_{s_i}})$  then
14:      if  $\theta \geq \theta_{max}$  then
15:         $\vec{wd}$  rotate to the right by  $\theta_{max}$ 
16:      else
17:         $\vec{wd}$  rotate to the right by  $\theta$ 
18:      end if
19:    else if  $R_s \leq d(s_i, v)$  &&  $(a_r \notin S_{v_{s_i}})$  then
20:      if  $\theta \geq \theta_{max}$  then
21:         $\vec{wd}$  rotate to the left by  $\theta_{max}$ 
22:      else
23:         $\vec{wd}$  rotate to the left by  $\theta$ 
24:      end if
25:    end if
26:  end if
27: end while
  
```

---

**Algorithm 2** Move Inside Cell Algorithm (MIC)

---

```

1: Input: coordinates  $(X, Y)$  and  $\vec{wd}$ .
2: Output: updated coordinates and  $\vec{wd}$ .
3: Initialization: Construct local Voronoi cell
4: while  $i \leq N$  do
5:   if  $S_{s_i} \in S_{v_{s_i}}$  then
6:     save  $s_i(x_s, y_s)$  and corresponding  $\vec{wd}$ 
7:   else
8:     select  $s_i v_{farthest}$  as  $\vec{wd}$ 
9:     if  $S_{s_i} \in S_{v_{s_i}}$  then
10:      save  $s_i(x_s, y_s)$  and corresponding  $\vec{wd}$ 
11:     else if  $(a_l, a_r \notin S_{v_{s_i}})$  then
12:       if  $d \geq d_{max}$  then
13:          $s$  moves  $d_{max}$  to  $s'$ 
14:       else
15:          $s$  moves  $d$  to  $s'$ 
16:       end if
17:     end if
18:   end if
19: end while

```

---

## 4.3. Rotation Based on Boundary (RB)

From the above analysis, we know that the MIC and RWD can improve the area coverage. However, the performance may not be good enough for sensors that are deployed on the boundary. A sensor makes a decision to move or rotate based on whether auxiliary points are in its own polygon. In Figure 2c, we can see that boundary Voronoi cells are infinite, while, for the infinite cell, it is impossible for the sensor to decide whether auxiliary points are in the cell. Thus, the sensor cannot make a decision to move or rotate.

For this case, we propose a scheme of rotation based on boundary (RB) to enhance boundary coverage. For a given target area,  $A = m \times n$  and the sensor coordinate  $s(x_s, y_s)$ ,  $0 \leq x_s \leq n$ ,  $0 \leq y_s \leq m$ . The boundary sensor makes a decision to rotate according to whether the coordinates of the auxiliary points are inside the target area. In the process of rotation, we need to figure out the minimum rotation angle  $\theta_r$ , shown in Figure 6. To solve  $\theta_r$ , we take Figure 6a as an example. After rotation,  $a_l$  updates to  $a'_l$ , so  $x_{a'_l} = x_s + R_s \cos(\psi - \theta_r)$ . We can get  $\theta_r$  for  $x_{a'_l} = 0$ , and other cases can deduce from this. If the sensor rotates,  $\theta_r$  can achieve the purpose. Then, the sensor rotating contrarily by a certain angle can also achieve the same purpose. From the perspective of energy consumption, the sensor needs to rotate a minimum angle that corresponds to several conditions as follows:

for boundary  $x = 0$ ,

$$\begin{cases} x_{a_l} \leq x_{a_r} \text{ and } x_{a_l} \leq 0 & \text{rotate right} \\ x_{a_l} < 0 \text{ and } x_{a_r} \geq 0 & \text{rotate right} \end{cases}$$

$$\begin{cases} x_{a_r} \leq x_{a_l} \text{ and } x_{a_r} \geq 0 & \text{rotate left} \\ x_{a_l} \geq 0 \text{ and } x_{a_r} < 0 & \text{rotate left} \end{cases}$$

for boundary  $x = n$ ,

$$\begin{cases} x_{a_l} \leq x_{a_r} \text{ and } x_{a_l} \leq n & \text{rotate right} \\ x_{a_l} \geq n \text{ and } x_{a_r} < n & \text{rotate right} \end{cases}$$

$$\left\{ \begin{array}{l} x_{a_r} \geq x_{a_l} \text{ and } x_{a_l} \geq n \text{ rotate left} \\ x_{a_l} < n \text{ and } x_{a_r} \leq n \text{ rotate left} \end{array} \right\}$$

for boundary  $y = 0$ ,

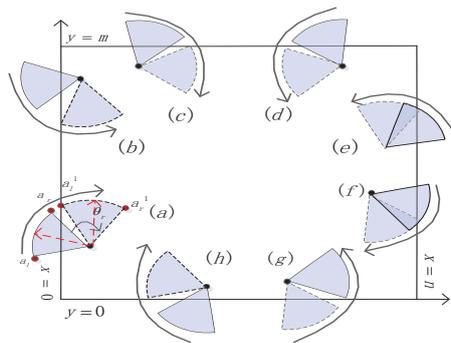
$$\left\{ \begin{array}{l} y_{a_l} \leq y_{a_r} \text{ and } y_{a_r} \leq 0 \text{ rotate right} \\ y_{a_l} \leq 0 \text{ and } y_{a_r} > 0 \text{ rotate right} \end{array} \right\}$$

$$\left\{ \begin{array}{l} y_{a_r} \leq y_{a_l} \text{ and } y_{a_l} \leq 0 \text{ rotate left} \\ y_{a_r} \leq 0 \text{ and } y_{a_l} > 0 \text{ rotate left} \end{array} \right\}$$

for boundary  $y = m$

$$\left\{ \begin{array}{l} y_{a_r} \leq y_{a_l} \text{ and } y_{a_r} \geq m \text{ rotate right} \\ y_{a_r} < m \text{ and } y_{a_l} \geq m \text{ rotate right} \end{array} \right\}$$

$$\left\{ \begin{array}{l} y_{a_l} \leq y_{a_r} \text{ and } y_{a_l} \geq m \text{ rotate left} \\ y_{a_l} < m \text{ and } y_{a_r} \geq m \text{ rotate left} \end{array} \right\}$$



**Figure 6.** Rotation based on boundary. (a) boundary  $x = 0$ , rotate right; (b) boundary  $x = 0$ , rotate left; (c) boundary  $y = m$ , rotate right; (d) boundary  $y = m$ , rotate left; (e) boundary  $x = n$ , rotate left; (f) boundary  $x = n$ , rotate right; (g) boundary  $y = 0$ , rotate left; (h) boundary  $y = 0$ , rotate right.

## 5. Performance Evaluation

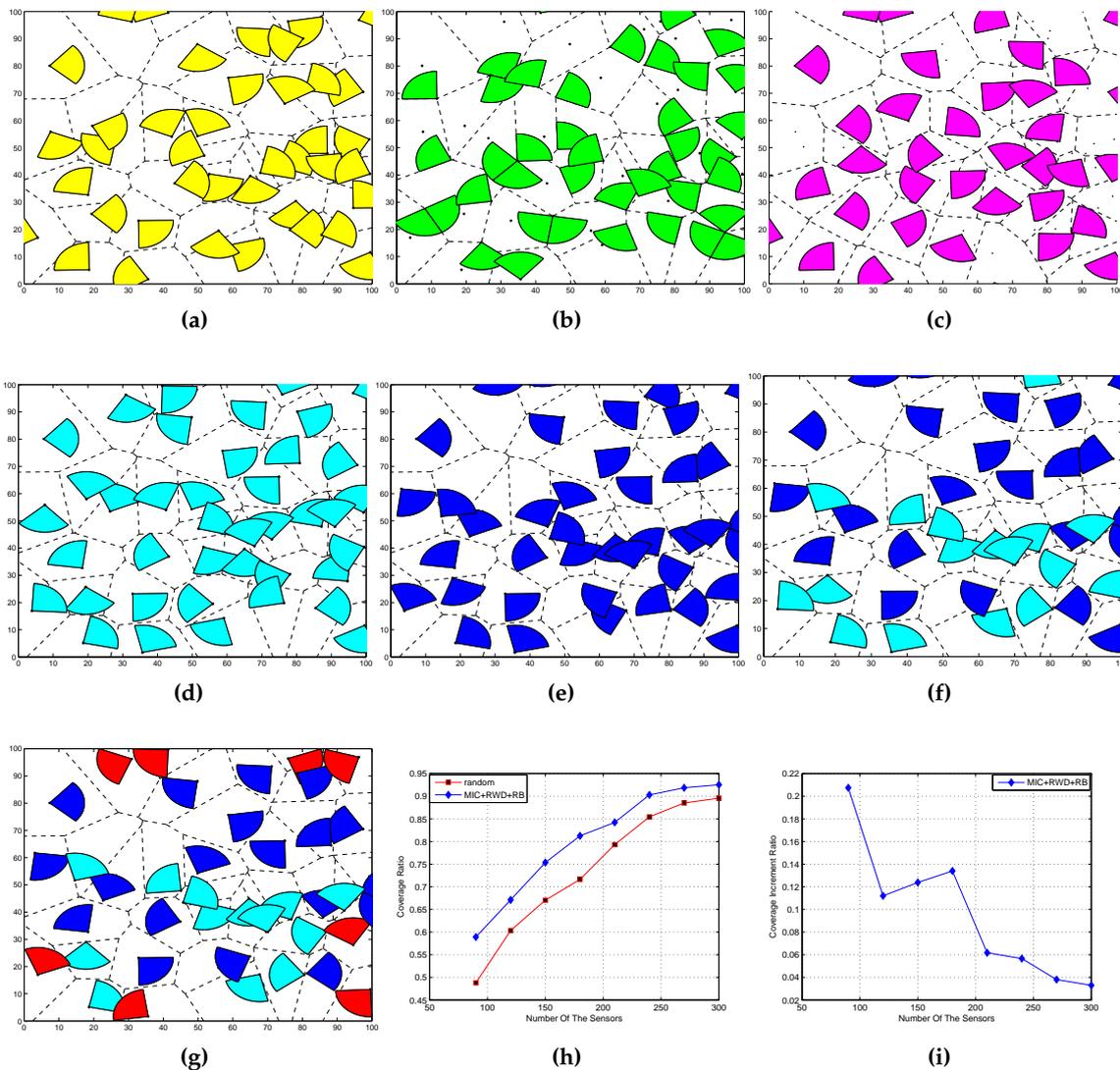
In this section, we conduct extensive simulations to validate the efficiency of the algorithms that we proposed. We compare RWD, MIC and RB with random deployment, DVSA deployment in [8] and EDA-II deployment in [14] because DVSA and EDA-II both employ the movement and rotation characteristics. In addition, we analyze the impact of the sensing radius and AoV on network coverage ratio. For comparison of our proposed algorithm and the random deployment, we also compare the moving track of MIC with DVSA and EDA-II. Some essential simulation parameters are listed in Table 1.

**Table 1.** Main notations.

Parameters	Value
sensing radius	$R_s = 6 \text{ m}, 8 \text{ m}, \dots, 12 \text{ m}$
angle of view (AoV)	$\text{AoV} = 30^\circ, 60^\circ, \dots, 180^\circ$
size of monitoring area	Area = $100 \text{ m} \times 100 \text{ m}$
the number of sensors, $N$	$N = 40, 60, 90, \dots, 300$

5.1. Sensing Coverage

In order to reflect the operation effect of algorithms that we have designed, we take the number of sensor  $N = 40$ ,  $R_s = 10$  m and AoV is  $90^\circ$ . We compare our algorithms with random deployment, EDA-II in [14] and DVSA in [8]. Figure 7a is the random deployment and Figure 7b is the DVSA deployment, Figure 7c is EDA-II deployment, Figure 7d is the RWD deployment, Figure 7e is MIC deployment and Figure 7f is the RWD + MIC. The last one, Figure 7g, is RWD + MIC + RB. By comparison, we can find that the combined algorithms gain an optimal coverage effect. RWD and MIC are based on the Voronoi cell, and, for the sensors not deployed at boundaries, they work well. RB is based on the boundary, and it can help the sensor deployed at the boundary rotate its working direction to the inner border, and this action increases node utilization.



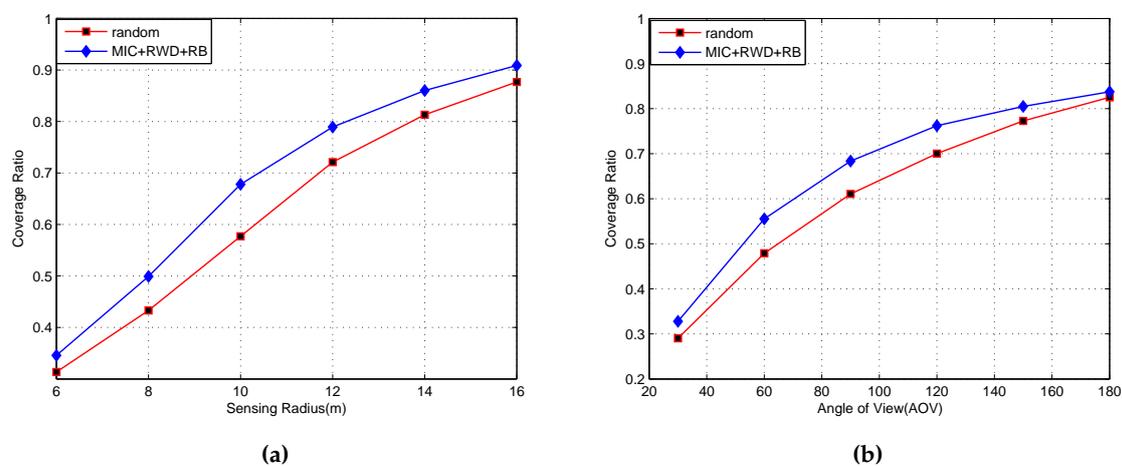
**Figure 7.** Coverage. (a) random; (b) DVSA; (c) EDA-II; (d) RWD; (e) MIC; (f) MIC and RWD; (g) MIC + RWD + RB; (h) Coverage ratio with  $R_s = 10$ , AoV =  $90^\circ$ ; (i) Coverage increment ratio with  $R_s = 10$ , AoV =  $90^\circ$ .

Figure 7h shows the coverage ratio with respect to the number of sensor nodes  $N$  for fixed  $R_s$  and AoV, while Figure 7i shows the coverage ratio increment with respect to  $N$  for fixed  $R_s$  and AoV. Here, the coverage ratio increment is defined as  $\frac{r_{\text{algorithm}} - r_{\text{random}}}{r_{\text{random}}}$ , where  $r_{\text{algorithm}}$  is the coverage ratio

after execution of our proposed algorithm and  $r_{\text{random}}$  is the coverage ratio of the random deployment algorithm. We find that, in Figure 7i, the coverage increment ratio is decreasing with relatively large  $N$ . This is because the method of MIC and RWD are both based on a Voronoi cell, and the generated Voronoi cell is small when  $N$  is large, which results in a bigger working area of the sensor node than the Voronoi cell. Thus, it is hard for the sensor nodes to get full coverage in the cell by actions of movement or rotation. Because of this, the parameters of  $N$ ,  $R_s$ , and AoV jointly affect the coverage ratio. In order to give a more clear illustration, in what follows, we investigate the coverage ratio with the three parameters, respectively.

### 5.2. Comparison from Moving Distance

The moving distance is the main part of the node's energy consumption. Here, we adopt the moving distance as the only measurement of the node's energy consumption, for simplicity of analysis. Our algorithm can also extend to more general scenarios on energy consumption after slight changes. In DVSA, the sensor moves to the vertex with a side length and field angle similar to  $R_s$  and AoV. In EDA-II, if the sensor does not cover all Voronoi vertices surrounding its own cell, then the sensor needs to move, while the MIC sensor makes a decision to move based on whether the auxiliary points are in the cell itself. Figure 8a,b,c represents the moving track of DVSA, EDA-II and MIC. The blue point is the initial position and the red point is the updated position. Clearly, we can see that the moving distance of MIC and EDA-II are shorter than DVSA. By comparing DVSA and EDA-II with MIC, we can find that the first two need all sensors to move despite moving distance being relatively shorter in EDA-II. However, not all sensors need to move in MIC, and network topology evolution is small.



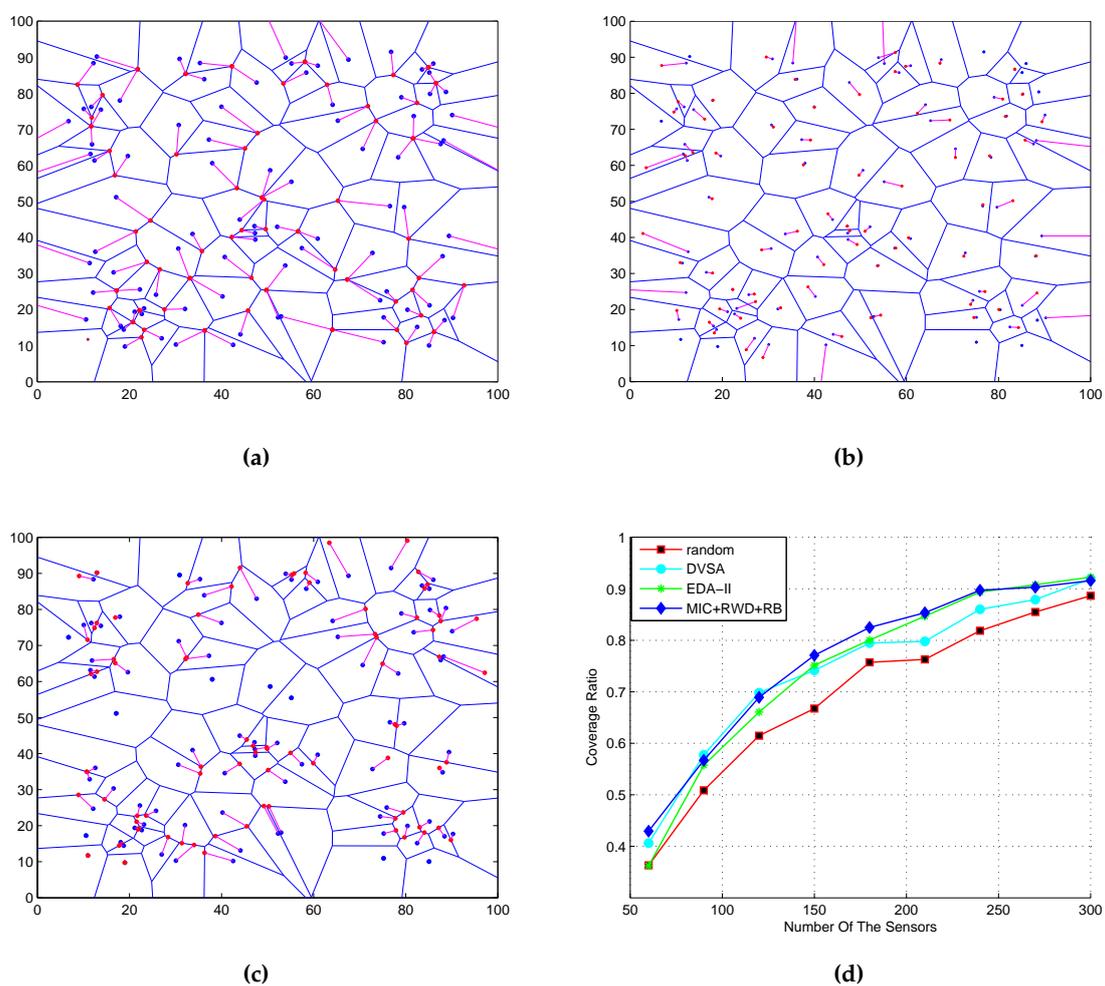
**Figure 8.** The influence of  $R_s$  and AoV on coverage ratio. (a) MIC + RWD + RB compared with random ( $n = 120$ , AoV =  $90^\circ$ ,  $R_s = 6$  m, 8 m, ..., 16 m); (b) MIC + RWD + RB compared with random ( $n = 120$ ,  $R_s = 10$  m, AoV =  $30^\circ$ ,  $60^\circ$ , ...,  $180^\circ$ ).

### 5.3. Coverage Ratios with Different $R_s$ and AoV

In this section, we investigate the influence of different radii and angles of view on coverage. Figure 8 is the simulation result at different  $R_s$  and AoV. In Figure 8a, we can see that when  $R_s$  is smaller, coverage improvement is not very obvious. With  $R_s$  increasing, the degree of improvement increases, and when  $R_s$  is big enough, the improvement gradually disappears. The same process happens in Figure 8b. This is understandable because MIC and RWD make sensors move or rotate based on whether auxiliary points are in the cell. When  $R_s$  and AoV increase to large enough sizes, all of the sensors may move or rotate, but this action does not, in fact, increase coverage. Thus, selecting suitable AoV and  $R_s$  is important.

#### 5.4. Coverage Ratio

Coverage ratio is a basic measure of DSNs. Figure 9d shows four kinds of coverage ratios that correspond to random deployment: DVSA, EDA-II, MIC coupled with RWD, and RB. To calculate the coverage ratio, we first regularly discrete the network area into  $M$  points. We sum all of the points that are covered by the sensors, which is denoted as  $k$ . Then, we can calculate the coverage ratio as  $k/M$ . Specifically, in the simulations, we first define a parameter  $k$ . If a point is covered by the sensor according to the criterion in Section 3.1, Equations (1) and (2), then  $k = k + 1$ . Finally, we get the coverage ratio  $k/M$ . From the simulation results, we can see that MIC + RWD + RB get the better result. When the number of sensors  $N$  is less, the coverage ratio is not so remarkable, and, with  $N$  increasing, the coverage ratio is obvious. If  $N$  is big enough, then the advantage of the coverage increment ratio disappears, as shown in Figure 7i. However, this situation is foreseeable and understandable.



**Figure 9.** Moving track and coverage ratio. (a) DVSA; (b) EDA-II; (c) MIC; (d) coverage ratio.

## 6. Conclusions

In this paper, we studied the area coverage problem of DSNs. We adopted the Voronoi diagram technique to transform the area network coverage problem into a cell coverage problem. We proposed three local coverage optimization algorithms to improve the cell coverage, i.e., MIC, RWD, and RB, respectively. We stated that algorithms have decentralized nature in the sense that sensors only need to construct their local Voronoi cells. We performed extensive simulations to prove the effectiveness of

our proposed algorithms in terms of the coverage ratio. Compared to the benchmark algorithm DVSA that was proposed in the literature, we found that our proposed algorithm shows higher area coverage ratio and shorter moving distance. We also found that there exists a potential to improve the moving distance of our algorithms by comparing it with EDA-II. In the future, we plan to focus on researching energy consumption and networks' lifetime problems.

**Acknowledgments:** This work is supported by the National Natural Science Foundation (NSF) of China under Grant No. 61301118; the Innovation Program of the Shanghai Municipal Education Commission under Grant No. 14YZ130; the International Science and Technology Cooperation Program of the Shanghai Science and Technology Commission under Grant No. 15220710600; and the Fundamental Research Funds for the Central Universities under Grant No. 16D210403.

**Author Contributions:** The work presented in this paper is a collaborative development by all of the authors. Guanglin Zhang and Shan You contributed to the idea of the incentive mechanisms and designed the algorithms. Jiajie Ren, Demin Li and Lin Wang developed and analyzed the experimental results. All authors contributed to the organization of the paper including writing and proofreading.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Cardei, M.; Thai, M.T.; Li, Y.S.; Wu, W.L. Energy-efficient target coverage in wireless sensor networks. In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; pp. 1976–1984.
2. Megerian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M. Worst and best-case coverage in sensor networks. *IEEE Trans. Mob. Comput.* **2005**, *4*, 84–92.
3. Liang, C.K.; Chung, C.Y.; Li, C.F. A virtual force based movement scheme for area coverage in directional sensor networks. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 718–722.
4. Tao, D.; Ma, H.D.; Liu, L. A Virtual Potential Field Based Coverage-Enhancing Algorithm for Directional Sensor Networks. *J. Software* **2007**, *18*, 1152–1163.
5. Li, J.; Wang, R.H.; Huang, H.P.; Sun, L.G. Voronoi-Based Coverage Optimization for Directional Sensor Networks. *Wirel. Sens. Netw.* **2009**, *1*, 417–424.
6. Stergiopoulos, J.; Tzes, A. Voronoi based coverage optimization for mobile networks with limited sensing range: A directional search approach. In Proceedings of the 2009 American Control Conference, St. Louis, Mo, USA, 10–12 June 2009; pp. 1427–1436.
7. You, S.; Zhang, G.L.; Li, D.M. Coverage improvement strategy based on Voronoi for directional sensor networks. In Proceedings of the EAI International Conference on Machine Learning and Intelligent Communications, Shanghai, China, 27–28 August 2016.
8. Sung, T.W.; Yang, C.S. Distributed Voronoi-Based Self-Redeployment for Coverage Enhancement in a Mobile Directional Sensor Network. *Int. J. Distrib. Sens. Netw.* **2013**, *6*, 1456–1459.
9. Wang, B.; Lim, H.B.; Ma, D. A survey of movement strategies for improving network coverage in wireless sensor networks. *Comput. Commun.* **2009**, *32*, 1427–1436.
10. Guvensan, M.A.; Yavuz, A.G. On coverage issues in directional sensor networks: A survey. *Ad Hoc Netw.* **2011**, *9*, 1238–1255.
11. Wang, G.; Cao, G.; Berman, P.; Porta, T.F.L. Bidding protocols for deploying mobile sensors. *IEEE Trans. Mob. Comput.* **2007**, *6*, 515–528.
12. Ghosh, A. Estimating coverage holes and enhancing coverage in mixed sensor networks. In Proceedings of the IEEE International Conference on Local Computer Networks, Alberta, Canada, 2014, pp. 68–76.
13. Liang, C.K.; He, M.C.; Tsai, C.H. Movement assisted sensor deployment in directional sensor networks. In Proceedings of the Sixth International Conference on Mobile Ad-Hoc Sensor Networks, Hangzhou, China, 20–22 December 2010, pp. 226–230.
14. Lin, T.Y.; Santoso, H.A.; Wu, K.R.; Wang, G.L. Enhanced Deployment Algorithms for Heterogeneous Directional Mobile Sensors in a Bounded Monitoring Area. *IEEE Trans. Mob. Comput.* **2016**, *PP*, doi:10.1109/TMC.2016.2563435.

15. Sung, T.W.; Yang, C.S. Localised sensor direction adjustments with geometric structures of Voronoi diagram and Delaunay triangulation for directional sensor networks. *Int. J. Ad Hoc Ubiquitous Comput.* **2015**, *20*, 91–106.
16. Bash, B.A.; Desnoyers, P.J. Exact distributed voronoi cell computation in sensor networks. In Proceedings of the 6th International Conference on Information Processing in Sensor Networks, Cambridge, MA, USA, 25–27 April 2007; pp. 236–243.
17. Kalantari, B. Voronoi diagram properties in polynomials with polynomiography applications and extensions. In Proceedings of the 2012 Ninth International Symposium on Voronoi Diagrams in Science and Engineering, Saint Petersburg, Russia, 27–29 August 2012; pp. 32–40.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).