

Article

# A Virtual Force Algorithm-Lévy-Embedded Grey Wolf Optimization Algorithm for Wireless Sensor Network Coverage Optimization

Shipeng Wang, Xiaoping Yang, Xingqiao Wang and Zhihong Qian \*

College of Communication Engineering, Jilin University, Changchun 130012, China; spwang17@mails.jlu.edu.cn (S.W.); yxp@jlu.edu.cn (X.Y.); xqwang18@mails.jlu.edu.cn (X.W.)

\* Correspondence: dr.qzh@163.com; Tel.: +86-135-0441-5955

Received: 10 May 2019; Accepted: 11 June 2019; Published: 18 June 2019



**Abstract:** The random placement of a large-scale sensor network in an outdoor environment often causes low coverage. In order to effectively improve the coverage of a wireless sensor network in the monitoring area, a coverage optimization algorithm for wireless sensor networks with a Virtual Force-Lévy-embedded Grey Wolf Optimization (VFLGWO) algorithm is proposed. The simulation results show that the VFLGWO algorithm has a better optimization effect on the coverage rate, uniformity, and average moving distance of sensor nodes than a wireless sensor network coverage optimization algorithm using Lévy-embedded Grey Wolf Optimizer, Cuckoo Search algorithm, and Chaotic Particle Swarm Optimization. The VFLGWO algorithm has good adaptability with respect to changes of the number of sensor nodes and the size of the monitoring area.

**Keywords:** wireless sensor network; Lévy-embedded Grey Wolf Optimization algorithm; Virtual Force algorithm; coverage optimization

---

## 1. Introduction

Wireless sensor networks (WSNs) were originally designed for military applications, however currently WSNs are also widely used in civilian applications, including vehicle tracking, forest monitoring, seismic observation, building monitoring, and water resource monitoring [1,2]. The coverage rate is an important measure of WSN performance. How to use the minimum number of sensor nodes to monitor the target area has been a question of particular interest regarding WSN technology. Wireless sensor nodes are typically placed randomly in the monitored area, which results in a non-uniform distribution of sensor nodes and a low coverage rate of the monitored area. Therefore, it is of great significance to improve the coverage rate of WSNs in the monitored area by adjusting the position of sensor nodes.

In 1999, the concept of swarm intelligence was formally proposed. Algorithms and solutions for a distributed problem inspired by the collective behavior of social insects and animal groups are part of swarm intelligence research. This type of algorithm is called a swarm intelligence optimization algorithm. Among intelligent algorithms, the Taguchi-teaching learning-based optimization algorithm and Jaya algorithm are also widely used [3,4]. In 2009, Yang and Deb of Cambridge University proposed a new biological heuristic algorithm, namely the Cuckoo Search algorithm [5]. This algorithm is widely used because of its simple structure, few control parameters, and strong search capability. Thanh-Phong Dao [6] proposed a hybrid Taguchi-Cuckoo Search (HTCS) to be applied to compatible focus positioning platforms (CFPP). The proposed HTCS approach can effectively optimize multiple objectives for the CFPP, and would be a useful technique for related optimization problems.

In recent years, scholars have carried out a lot of research on the GWO (Grey Wolf Optimizer). Heidari combined the GWO algorithm with Levi's flight to propose the Lévy-embedded GWO (LGWO) [7]. This algorithm has a stronger search ability than the GWO. At the same time, Heidari has set up multiple sets of experiments to prove that the LGWO algorithm is better than PSO (Particle Swarm Optimization), CS (Cuckoo Search), DE (Differential Evolution), and other algorithms in most cases [7]. Therefore, the LGWO algorithm is applied to wireless sensor node deployment in this paper. The application of the LGWO algorithm to wireless sensor node deployment improves the coverage rate of the wireless sensor network. However, the sensor nodes are still non-uniform. In addition, many studies lack a design for how to complete wireless sensor node deployment with the shortest moving distance.

This paper proposes a wireless sensor network coverage optimization algorithm, the Virtual Force-Lévy-embedded Grey Wolf Optimization (VFLGWO) algorithm, which is intended to obtain a higher coverage rate of the monitoring area, a more uniform sensor nodes distribution, and a shorter average moving distance of the wireless sensor nodes. Typical performance evaluation indicators, such as the coverage rate, uniformity, average moving distance of the sensor nodes, and running time are simulated. Compared with wireless sensor network coverage optimization algorithms using Chaos Particle Swarm Optimization (CPSO), Cuckoo Search (CS) and Lévy-embedded Grey Wolf Optimization (LGWO), the simulation experiment proves the performance superiority of the VFLGWO.

The contributions of this paper are:

1. This paper applies the LGWO algorithm to wireless sensor node deployment.
2. For wireless sensor node deployment, this paper proposes the VFLGWO algorithm, which achieves a higher coverage rate.
3. This paper designs sensor node matching algorithms to make the average moving distance of sensor nodes shorter.

The remainder of this paper is structured as follows. Section 2 describes the related work. Section 3 describes the proposed VFLGWO based sensor deployment scheme. The simulation results and comparison of the CPSO algorithm, CS algorithm, and LGWO algorithm are presented in Section 4, and conclusions are drawn in Section 5.

## 2. Related Work

### 2.1. Research Situation

In Guanglin Zhang's work [8], Guanglin Zhang and Shan You studied the coverage of WSNs. They employed a Voronoi diagram to address the coverage problem. The authors transformed the network area coverage problem into cell coverage problems with the Voronoi diagram, which only needs to optimize local coverage for each cell in a decentralized way.

The research on using swarm intelligence optimization algorithms to solve WSN coverage optimization problems is gradually increasing. Ning-ning Qin and Leopoldo Eduardo Cárdenas-Barrón [9,10] applied differential evolution and its improved algorithm to complete wireless sensor node deployment. Differential evolution has the characteristics of a simple structure, easy implementation, fast convergence, and strong robustness. Jia, Jie [11] adopted a genetic algorithm to solve the WSN coverage optimization problem. Genetic algorithms have a better ability to conduct a global search, but their implementation is relatively complex and their convergence speed is slow. The PSO algorithm [12–14] has been applied to the deployment of wireless sensor networks by scholars. The PSO is easy to implement and can effectively balance the directionality, diversity, and balance in the search. In other research [15], the PSO algorithm was employed to solve the optimization problem for a compliant rotary joint for an upper limb assistive device, and obtained a CRJ with a smaller error. However, the PSO algorithm has some shortcomings, including relying too much on search parameters, easily falling into local extremum, and premature convergence. In Celal Ozturk's work [16], an artificial

bee colony algorithm was used to maximize area coverage in WSNs. The artificial bee colony algorithm has the advantage of easy realization, and the shortcoming of easily falling into the local solution. Some scholars have adopted the Chaotic Artificial Fish Swarm algorithm to solve this problem [17]. The advantage of the Artificial Fish Swarm algorithm is that it has better robustness. The algorithm adopts an adaptive step size, which effectively increases the coverage of the algorithm.

## 2.2. An Overview of the GWO Algorithm

Inspired by wolf hunting behavior, Mirjalili et al. proposed the Grey Wolf Optimization (GWO) [18], which is easy to implement because of its few control parameters. The GWO algorithm is inspired by the social hierarchy and hunting strategies of grey wolves in the wild. Each grey wolf represents a potential solution to the population. The position of  $\alpha$  wolf, which is the leading wolf, is the best position. The position of the  $\beta$  wolf in the second class of the wolf rank, and the  $\delta$  wolf responsible for reconnaissance, vigilance, warding and guarding, are the second-best solution and the third best solution, respectively. The other candidate solutions are the lower-level  $\omega$  wolf positions. In the conventional GWO, the motion of wolves is described as follows.

### 2.2.1. Population Initialization

Since the performance of the GWO is less affected by the initial value of the population, it is initialized in the algorithm by randomly generating:

$$X_{i,j} \sim U(lb_j, ub_j) \quad (1)$$

where  $X$  is the grey wolf population,  $i \in [1, 2, 3 \dots, N]$  and  $j \in [1, 2, 3 \dots, sizepop]$ ,  $N$  is the number of grey wolf populations,  $sizepop$  is the population dimension;  $lb$  and  $ub$  are the lower bound and upper bound of the search interval respectively; and  $U$  is a random uniform distribution function.

### 2.2.2. Encircling Prey

As mentioned above, grey wolves encircle prey during the hunt. In order to mathematically model encircling behavior, the following equations are proposed:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (2)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (3)$$

where  $\vec{D}$  is the distance between the prey and grey wolf;  $t$  indicates the current iteration,  $\vec{A}$  and  $\vec{C}$  are coefficient vectors,  $\vec{X}_p$  is the position vector of the prey; and  $\vec{X}$  indicates the position vector of a grey wolf.

The vectors  $\vec{A}$  and  $\vec{C}$  are calculated as follows:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (4)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (5)$$

where components of  $\vec{a}$  are linearly decreased from 2 to 0 over the course of iterations, and  $\vec{r}_1, \vec{r}_2$  are random vectors in  $[0,1]$ . Candidate solutions tend to diverge from the prey when  $|\vec{A}| \geq 1$  and converge towards the prey when  $|\vec{A}| < 1$ .

### 2.2.3. Population Location Update

By calculating the objective function value of the GWO algorithm, the best solution, the second best solution, and the third best solution are set to  $\alpha$  wolf,  $\beta$  wolf, and  $\delta$  wolf. The position of other

grey wolves is determined by the position of  $\alpha$  wolf,  $\beta$  wolf and  $\delta$  wolf, as shown in Equations (6)–(8). After generating a new group, boundary control is performed on the elements in the population to complete an iteration. The above process is repeated until the algorithm termination condition is satisfied, and finally the optimal solution is outputted.

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \end{cases} \quad (6)$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{cases} \quad (7)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (8)$$

### 2.3. Lévy-Embedded Grey Wolf Optimization Algorithm

The GWO algorithm has the advantages of not relying on parameter settings and convenient implementation, however the GWO algorithm is still prone to prematurely falling into local extremum when solving complex optimization problems; that is, the phenomenon of premature convergence occurs. In order to alleviate the above problems, Heidari proposed a Lévy-embedded Grey Wolf Optimization (LGWO) algorithm based on the GWO algorithm combined with Lévy Flight [7]. The Lévy Flight can assist the GWO in searching based on deeper searching patterns. Using this concept, it can be ensured that the GWO can handle global searching more efficiently. The LGWO algorithm mainly improves the GWO algorithm in three aspects.

1. The role of  $\delta$  wolves in the social hierarchy will be played by other wolves.
2. The Lévy Flight concept is embedded into the modified GWO.
3. The greedy selection (GS) strategy is employed for the LGWO.

The specific process of the LGWO algorithm is summarized as follows, and the pseudo code of the LGWO algorithm is described in detail, as shown in Table 1.

1. Population initialization. The process is the same as Equation (1) of Section 2.2.1 for the GWO algorithm.
2. Population search. The process is the same as Equations (2) and (3) of the GWO algorithm in Section 2.2.2.
3. By calculating the objective function value of the LGWO algorithm, the best solution  $\alpha$  wolf and second-best solution  $\beta$  wolf are obtained. The positions of other grey wolves are determined by the position of  $\alpha$  wolf and  $\beta$  wolf, as shown in Equation (9):

$$\vec{X}(t+1) = \begin{cases} 0.5 \times (\vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha + \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta) + \alpha \oplus \text{Levi}(\beta) & |A| \geq 0.5 \\ 0.5 \times (\vec{X}_\alpha - \vec{A}_1 \vec{D}_\alpha + \vec{X}_\beta - \vec{A}_2 \vec{D}_\beta) & |A| < 0.5 \end{cases} \quad (9)$$

where

$$\alpha \oplus \text{Levi}(\beta) \sim 0.01 \frac{u}{v^{-\beta}} (\vec{X}(t) - \vec{X}_\alpha(t)) \quad (10)$$

where  $u$  and  $v$  follow a normal distribution:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2)$$

$$\sigma_u = \left[ \frac{\Gamma(1 + \beta) \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\beta \times 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}}, \quad \sigma_v = 1$$

In the LGWO Algorithm 1 proposed by Heidari, the parameter  $\beta$  is a random number of [0,2].

---

**Algorithm 1** Pseudo code for the LGWO algorithm

---

1. Determine the initial swarm size  $N$  and number of iterations  $T$
  2. Randomly generate the initial population of wolves
  3. Initialize  $a, p, \vec{A}, \vec{C}$
  4. Compute the fitness of each wolf
  5. Set to  $\vec{X}_\alpha$  be the best wolf
  6. Set to  $\vec{X}_\beta$  be the second best wolf
  7. **While** ( $t < T$ ) or (stopping condition) **do**
  8.     **For** each wolf
  9.         Update the position of current wolves by Equation (13)
  10.         Perform GS by Equation (15)
  11.     **end for**
  12. Update  $a, p, \vec{A}, \vec{C}$
  13. Calculate the fitness of wolves
  14. Update  $\vec{X}_\alpha$  and  $\vec{X}_\beta$
  15.  $t = t + 1$
  16. **end while**
  17. Return  $\vec{X}_\alpha$
- 

After generating a new group, Heidari [7] selected whether to retain the updated grey wolf through the greedy selection strategy depicted by Equation (11), and completes an iteration. The above process is repeated until the algorithm termination condition is satisfied, and finally the optimal solution is outputted.

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t) & \text{if } f(\vec{X}_{new}(t)) > f(\vec{X}(t)) \text{ and } r_{new} < p \\ \vec{X}_{new}(t) & \text{otherwise} \end{cases} \quad (11)$$

where  $r_{new}$  and  $p$  are random values inside (0, 1).

Additionally, referring to the greedy selection (GS) strategy from the DE algorithm, the concept of “survival of the fittest” is utilized here with the probability  $p$ . According to this strategy, new superior positions in each generation can continue to be more enriched for the next generations and the worse ones are disregarded, so the LGWO algorithm has a stronger search capability.

#### 2.4. Virtual Force Algorithm

The Virtual Force Algorithm (VFA) was originally an algorithm that allowed mobile robots to evade obstacles in an unknown environment [19–21]. The VFA has been applied to WSN coverage optimization, and can quickly disperse sensor nodes. The VFA treats the sensor nodes as charged particles. The forces between the sensor nodes are presented by attractive and repulsive patterns. When two nodes are close enough, smaller than the threshold  $D_{th}$ , the force is in a repulsive pattern, which intends to separate them. When the distance between two nodes is larger than the threshold  $D_{th}$ , the force becomes an attractive pattern, which draws them closer. Assuming that the set of sensor

nodes in the region is  $S = \{s_1, s_2, s_3, \dots, s_l\}$ , then the distance  $d_{ij}$  between the  $i, j$  sensor nodes is  $s_i(x_i, y_i)$  and  $s_j(x_j, y_j)$ :

$$d_{ij} = d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (12)$$

Then, the force of the sensor node  $s_j$  on the sensor node  $s_i$  is:

$$\vec{F}_{ij} = \begin{cases} \omega_\alpha(d_{ij} - D_{th}), \alpha_{ij} & d_{ij} > D_{th} \\ 0 & d_{ij} = D_{th} \\ \omega_\gamma/d_{ij}, \alpha_{ij} + \pi & d_{ij} < D_{th} \end{cases} \quad (13)$$

where  $\omega_\alpha$  and  $\omega_\gamma$  are the virtual force coefficients,  $\omega_\gamma \gg \omega_\alpha$ ;  $D_{th}$  is the threshold for generating gravitation and repulsive force, and the value of  $D_{th}$  is  $\sqrt{3}R_s$  [20]; and  $\alpha_{ij}$  is the angle of the line between sensor node  $i$  and sensor node  $j$  with respect to the y-axis. If  $d_{ij} < D_{th}$ , it is considered that sensor node  $i$  is too close to sensor node  $j$ , and the force is in a repulsive pattern; if  $d_{ij} > D_{th}$ , the force becomes an attractive pattern. The resultant force of the sensor node  $i$  is  $\vec{F}_i$ , the resultant force in the horizontal direction is  $\vec{F}_{ix}$ , the resultant force in the vertical direction is  $\vec{F}_{iy}$ , and the position of the node is updated by Equations (14) and (15).

$$x(k+1) = x(k) + \frac{\vec{F}_{ix}}{F_i} \times MaxStep \times e^{-1/F_i} \quad (14)$$

$$y(k+1) = y(k) + \frac{\vec{F}_{iy}}{F_i} \times MaxStep \times e^{-1/F_i} \quad (15)$$

where  $x(k)$  and  $y(k)$  are the horizontal and vertical coordinates of sensor node  $s_i$  at time  $k$ , respectively; and  $MaxStep$  is the step size for the wireless sensor node to move.

### 2.5. Improved Virtual Force Algorithm

Generally, the communication radius  $R$  of the sensor node is defined as twice the perceived radius  $R_s$ . Actually, when  $d_{ij}$  is larger than  $R$ , they cannot exchange information with each other, they cannot calculate or obtain the distance, and the VFA cannot be used in this case. The purpose of applying the improved virtual force algorithm in this paper is to apply it to wireless sensor network node deployment, and specifically to apply it to wireless sensor network nodes, which not only have sensor sensing capabilities, but also have communication capabilities. Therefore, the distance between sensor nodes must be less than the communication radius, and the sensor node distribution should not be too dense.

In the improved virtual force algorithm applied in this paper, when the  $d_{ij}$  is greater than  $R$ , there is no virtual force between the sensor nodes. Equation (13) of the sensor node  $s_j$  to the sensor node  $s_i$  is modified:

$$\vec{F}_{ij} = \begin{cases} \omega_\alpha(d_{ij} - D_{th}), \alpha_{ij} & R \geq d_{ij} > D_{th} \\ 0 & d_{ij} = D_{th} \text{ or } d_{ij} > R \\ \omega_\gamma/d_{ij}, \alpha_{ij} + \pi & d_{ij} < D_{th} \end{cases} \quad (16)$$

The physical meaning of the variables and parameters in Equation (16) is the same as Equation (13), and the updated equation of the sensor node position is the same as Equations (14) and (15).

### 3. Wireless Sensor Node Deployment with the VFLGWO Algorithm

#### 3.1. Problem Description

##### 3.1.1. Network Model

1. All sensor nodes are mobile.
2. All sensor nodes are omnidirectional sensors, and the perceptual model is a probabilistic detection model.
3. All sensor node location information is known by GPS.
4. All sensor nodes are able to move to the scheduled position, where the scheduled position is within their mobility range.

##### 3.1.2. Coverage Rate

The sensor node in this paper adopts the probabilistic detection model. Compared with the disk model, this model can objectively reflect the real network deployment environment, as shown in Figure 1. In Figure 1, the inside of the solid circle is the determined area, and the area between the solid circle and the outermost dotted circle is an uncertain area. The outside of the outermost dotted circle is an unmeasured area. Assuming that the monitoring area is a two-dimensional planar area, each sensor node can be deployed anywhere within the area. If the coordinates of the sensor node  $s_i$  in the sensor network are  $(x_i, y_i)$  and the coordinates of the monitoring target point  $p$  (pixel point) in the region are  $(x_p, y_p)$ , then the monitoring probability  $C_p(s_i, p)$  of the sensor node  $s_i$  to the target point  $p$  is set as follows:

$$C_p(s_i, p) = \begin{cases} 1 & d(s_i, p) \leq R_s - r_e \\ \exp\left(\frac{-\lambda_1 \alpha_1^{\beta_1}}{\alpha_2^{\beta_2}} + \lambda_2\right) & R_s - r_e < d(s_i, p) < R_s + r_e \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where  $d(s_i, p)$  is the Euclidean distance between the sensor node  $s_i$  and the target point  $p$ ;  $R_s$  is the distance radius that the sensor node can perceive;  $r_e$  ( $0 < r_e < R_s$ ) is the measurement reliability parameter of the sensor node;  $\alpha_1 = r_e - R_s + d(s_i, p)$ ,  $\alpha_2 = r_e + R_s - d(s_i, p)$ , and  $\alpha_1 > 0$ ,  $\alpha_2 > 0$ ;  $\lambda_1, \lambda_2, \beta_1, \beta_2$  are the measurement parameters related to the characteristics of the sensor node. The parameters of the work by Author [22] are used to set  $\lambda_1 = 1$ ,  $\lambda_2 = 0$ ,  $\beta_1 = 1$ ,  $\beta_2 = 1.5$ .

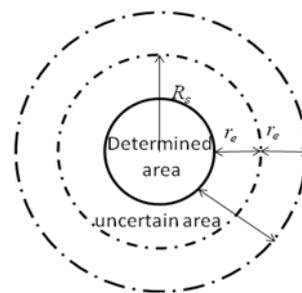


Figure 1. Node sensing model.

Let the probability threshold of target point  $p$  be detected as  $C_{th}$ , and the joint monitoring probability  $C_p(s_{all}, p)$  of the sensor node to target point  $p$  in the whole monitoring area is:

$$C_p(s_{all}, p) = \begin{cases} 1, & \left(1 - \prod_{i=1}^l [1 - C_p(s_i, p)]\right) \geq C_{th} \\ 0, & \left(1 - \prod_{i=1}^l [1 - C_p(s_i, p)]\right) < C_{th} \end{cases} \quad (18)$$

where  $s_{all}$  represents the set of sensor nodes that measure target point  $p$ , and  $l$  is the number of sensor nodes in the monitoring area.

Assuming that the monitoring area is a rectangle of  $m \times n$  ( $m^2$ ), the area to be measured is divided into  $m \times n$  identical grids of equal size and area of  $1 m^2$ . Then, the grids are simplified to pixel points with a discrete precision of 1. In this paper, the WSN node coverage rate is defined as the ratio of the number of grids covered by Equation (19) to the total number of grids in the monitoring area, namely:

$$fitness = \frac{\sum_{x_p=1}^m \sum_{y_p=1}^n C_p(s_{all}, p)}{m \times n} \quad (19)$$

Then, the coverage optimization problem is briefly described as follows:

- Step 1. Calculate the probability of monitoring one-pixel point by one sensor node using Equation (17).
- Step 2. Calculate the joint monitoring probability of all sensor nodes to one-pixel point using Equation (18).
- Step 3. The coverage rate of the monitored area is calculated using Equation (19). Equation (19) is used as the objective function of the coverage optimization algorithm.

### 3.1.3. Uniformity of Sensor Node Distribution

Uniformity is an important indicator for measuring the uniform distribution of sensor nodes in wireless sensor networks. It is used to describe the uniformity of the sensor node distribution. It is usually expressed by the mean of the sum of the standard deviation of all sensor nodes. The wireless sensor network node uniformity indicator  $E$  is calculated as shown in Equation (20):

$$E = \frac{1}{l} \sum_{i=1}^l E_i; E_i = \left[ \frac{1}{k} \sum_{j=1}^k (D_{i,j} - M_i)^2 \right]^{\frac{1}{2}} \quad (20)$$

where  $l$  is the total number of sensor nodes,  $k$  is the number of neighboring nodes of sensor node  $i$ ,  $D_{i,j}$  is the Euclidean distance between adjacent nodes  $i, j$ , and  $M_i$  is the average of the Euclidean distance between the sensor node  $i$  and all its neighboring sensor nodes. It can be found from Equation (20) that the closer the values of  $D_{i,j}$  and  $M_i$  are, the more uniform the distribution is. Therefore, the smaller the  $E$  value, the better the uniformity of the sensor node distribution.

### 3.1.4. Average Moving Distance of Sensor Nodes

The average moving distance of sensor nodes refers to the average distance that each sensor node in the wireless sensor network moves from the initial position to the best position with the largest final coverage rate. It is assumed that the initial position information of each sensor node is known. Through the MATLAB simulation, the VFLGWO algorithm is used to solve the optimal position of each sensor node with the largest coverage rate. Each sensor node is directly moved from the initial position to the optimal position in a straight line. Obviously, for wireless sensor nodes, the shorter the average moving distance is, the smaller the energy consumption is.

## 3.2. The VFLGWO Algorithm

### 3.2.1. The Modified LGWO Algorithm

The LGWO algorithm applies GS strategy to allow the algorithm to retain a better solution for each iteration. For the LGWO algorithm,  $f(\vec{X})$  represents a convergence curve and the smaller the value, the better. However, for wireless sensor node deployment,  $f(\vec{X})$  represents  $fitness(\vec{X})$  (the coverage rate). The larger the  $fitness(\vec{X})$  value that is obtained, the larger the corresponding coverage rate. Therefore, the modified LGWO proposed in this paper is the same as the LGWO, except that

Equation (11) is modified to Equation (21). When the new value  $fitness(\vec{X})$  obtained by a new iteration is not greater than the previous generation value and the satisfaction probability condition is reached, the optimal solution is not updated; otherwise, the obtained superior solution is substituted for the previous generation solution.

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t) & f(\vec{X}_{new}(t)) < f(\vec{X}(t)) \text{ and } r_{new} < p \\ \vec{X}_{new}(t) & \text{otherwise} \end{cases} \quad (21)$$

### 3.2.2. Combination of the Improved Virtual Force Algorithm and the Modified LGWO Algorithm

The performance of the LGWO is significantly better than that of the GWO. However, when the modified LGWO algorithm is applied to WSN node deployment, the problem of non-uniform distribution of sensor nodes still exists, while the Virtual Force algorithm can make the sensor nodes more uniform. Therefore, combined with the modified LGWO algorithm, the improved Virtual Force algorithm is introduced. In this paper, the Virtual Force-Lévy-embedded Grey Wolf Optimization (VFLGWO) algorithm is proposed.

In the modified LGWO algorithm, the position of a wolf represents a possible solution to the optimization problem. Therefore, the deployment of the sensor nodes in the detection area refers to a wolf (a solution) in the algorithm.

The procedure of deploying wireless sensor nodes by the VFLGWO algorithm can be summarized as follows:

$N$  sensor nodes are randomly generated to be the initial position of the real sensor nodes. Initialize each wolf (each set of solutions) by the modified LGWO algorithm as the virtual sensor node position.

Enter the iteration, update the position of each wolf using Equations (9) and (21). Then, apply the improved Virtual Force algorithm to update the sensor nodes' positions for each wolf, calculate the virtual force using Equation (20), and adjust the position of the nodes for each wolf using Equations (14) and (15). Apply the survival of the fittest selection rule using Equation (22), and retain the better solution obtained by the improved virtual force algorithm:

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t) & f(\vec{X}_{new}(t)) < f(\vec{X}(t)) \\ \vec{X}_{new}(t) & \text{otherwise} \end{cases} \quad (22)$$

Through the superior solution of the  $t_{th}$  generation  $\vec{X}'(t)$ , whether to update  $\alpha$  wolf or  $\beta$  wolf is determined; the VFLGWO algorithm completes an iteration. If the number of iterations does not reach the specified number of iterations  $T$ , perform the above iterative process again; otherwise, the process ends. The specific implementation of the VFLGWO algorithm will be described in Section 3.3.

### 3.3. Procedure for the VFLGWO Algorithm

Application of the VFLGWO algorithm in WSN node deployment is shown in Figure 2. The steps of the VFLGWO algorithm are described as follows:

1. Determine the size of the monitoring area and the number of sensor nodes  $N$  to be deployed, and randomly deploy the sensor nodes in the monitoring area at the real sensor node positions, the set of which is  $S_0 = \{s_1, s_2, s_3, \dots, s_N\}$ .
2. Initialize the parameters of the VFLGWO algorithm: population size  $sizepop$ , iteration number  $T$ , related parameters of the improved virtual force algorithm  $\omega_\alpha, \omega_\gamma$  and  $D_{thr}$ , and related parameters of the probability perception model  $\lambda_1, \lambda_2, \beta_1, \beta_2; t = 1$ .
3. Generate the initialized grey wolf population  $\vec{X}(t)$  randomly and calculate the coverage rate  $fitness_t$  of each wolf (each set of solutions) using Equations (17)–(19).

4. Set the solution with the largest coverage rate to  $\alpha$  wolf, set the solution with the second largest coverage rate to  $\beta$  wolf, and record the coverage rate  $fitness(\alpha)$  and  $fitness(\beta)$  of  $\alpha$  wolf and  $\beta$  wolf.
5. Enter the iteration, set  $t = 1$ , calculate the position of each grey wolf after the  $t$ th iteration using Equation (11), calculate the coverage  $fitness_t$  of each wolf, judge the better grey wolf using Equation (21), and then update  $a, p, \vec{A}, \vec{C}$ .
6. Apply the improved virtual force algorithm to calculate the virtual force of each sensor node of each wolf using Equation (16), adjust the position of each sensor node of each wolf using Equations (14) and (15), calculate the coverage  $fitness'_t$  of each wolf, and obtain better grey wolves  $\vec{X}'(t)$  using Equation (22).
7. Compare the  $fitness'_t$  value of the  $t$ th iteration with the  $fitness(\alpha)$  and  $fitness(\beta)$  obtained by the previous iteration to determine whether to update the  $\alpha$  wolf or the  $\beta$  wolf.
8. Determine whether the number of iterations reaches the specified number of iterations  $T$ , "No" then  $t = t + 1$ , return 5); "Yes" then the iteration ends.
9. Output the optimal solution  $\alpha$  wolf as the optimal target position of the real sensor nodes, recorded as  $S_0' = \{S_1', S_2', S_3', \dots, S_N'\}$ .
10. Use the sensor nodes matching algorithm, and match the position of the initial sensor node positions  $S_0 = \{s_1, s_2, s_3, \dots, s_N\}$  and the target position of the optimal solution  $S_0' = \{S_1', S_2', S_3', \dots, S_N'\}$  to complete the deployment of the WSN.

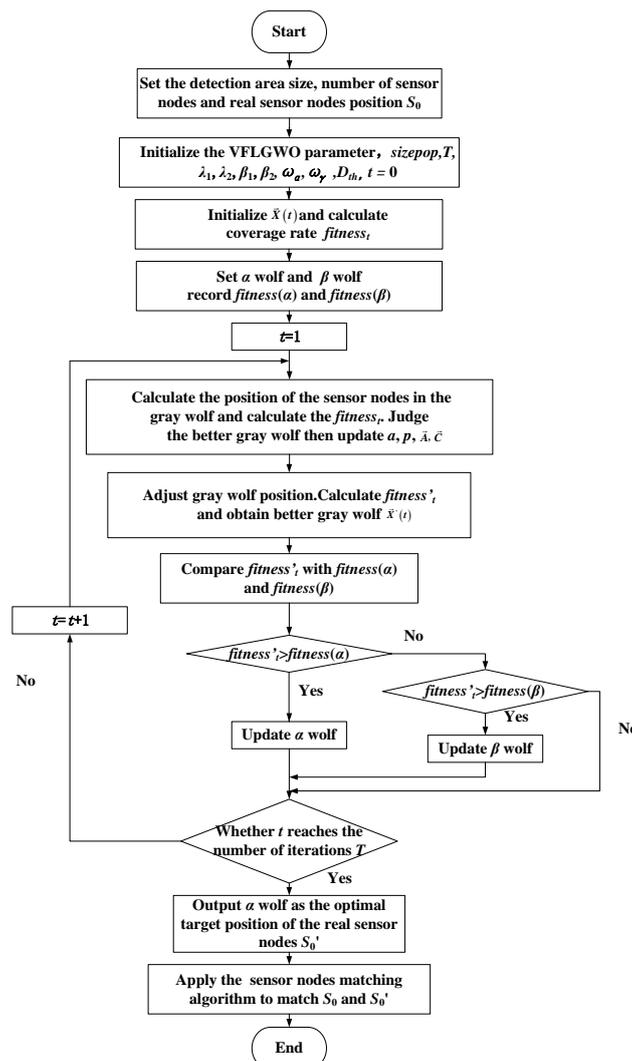


Figure 2. Process of the VFLGWO algorithm.

There are currently two options for WSN node deployment: (1) give the target sensor node position without considering how the sensor nodes move; (2) match the same sequence numbers of the sensor nodes before and after optimization and design the corresponding relationship of node movements.

The pseudo code of the sensor nodes matching algorithm proposed in this paper is shown in Algorithm 2. The initial position set of  $N$  sensor nodes in the region is  $S_0 = \{s_1, s_2, s_3, \dots, s_N\}$ , and the target position for obtaining the optimal solution is  $S_0' = \{S_1', S_2', S_3', \dots, S_N'\}$ . The matching of  $S_0$  and  $S_0'$  is achieved by the sensor nodes matching algorithm.

---

**Algorithm 2** Pseudo code for sensor nodes matching algorithm

---

1. Input initial position set  $S_0$ , optimal solution position set  $S_0'$ , number of nodes  $N$
  2. **For**  $i$  from 1 to  $N$  **do**
  3.     **For**  $j$  from 1 to  $N$  **do**
  4.         Calculate the Euclidean distance between the  $S_0$  node position  $s_i$  and the  $S_0'$  node position  $s_j$  and record
  5.     **end for**
  6. **end for**
  7.      $k = N$
  8. **While** ( $k > 0$ ) and ( $s_i', s_j'$  are unmatched nodes in  $S_0, S_0'$ ) **do**
  9.     Take the shortest distance between the shortest  $s_i$  in  $S_0$  and  $s_j$  in  $S_0'$  to complete position matching and record
  10.      $k = k - 1$
  11. **end while**
  12. Return the match result of  $S_0, S_0'$
- 

The VFLGWO algorithm proposed in this paper applies the LGWO algorithm with a stronger search ability to the deployment of wireless sensor nodes, and combines the improved Virtual Force algorithm to achieve a wireless sensor network with a higher coverage rate and more uniform distribution. In order to achieve the goal of making the average moving distance of the sensor nodes shorter, the randomly generated real node positions are taken as the initial positions of the sensor nodes, and the sensor node position of the optimal solution obtained by the VFLGWO algorithm is taken as the final position of the real node. The sensor nodes matching algorithm proposed in this paper is used to complete the WSN node deployment.

### 3.4. Algorithm Time Complexity Analysis

For the modified LGWO algorithm, initialize the number of sensor nodes  $N$ , and initialize the number of grey wolves  $m$  and the maximum number of iterations  $T$ . The modified LGWO algorithm calculates a wolf (update  $N$  sensor node position) with a time complexity of  $O(N)$ . According to the LGWO algorithm, each iteration needs to update the position of the  $N$  sensor nodes of each wolf in the  $m$  wolf at one time. The time complexity of the algorithm to update the position at one iteration is  $O(m \times N)$ , so the time complexity required in the case of  $T$  iterations is  $O(T \times m \times N)$ . Similarly, compared with the modified LGWO, the CPSO algorithm and CS algorithm only have different search methods, so the iterative process is similar. Therefore, the time complexity of the coverage strategy of the CPSO algorithm and CS algorithm is also  $O(T \times m \times N)$ .

The VFLGWO algorithm proposed in this paper executes the improved Virtual Force algorithm again after the LGWO search phase. The time complexity of the improved Virtual Force algorithm is  $O(N^2)$ . Therefore, according to the VFLGWO algorithm process, it needs to be updated to  $N$  times per iteration. The time complexity is  $O(m \times N^2)$ . In addition, the  $T$  set in this paper is much larger than  $N$ . Therefore, the time complexity of the sensor nodes matching algorithm proposed in this paper is negligible. The time complexity of the total coverage strategy of the VFLGWO algorithm is  $O(T \times m \times N^2)$ .

Although the VFLGWO algorithm increases the amount of computation, the coverage rate and uniformity obtained by the VFLGWO algorithm are significantly higher than other algorithms. The position adjustment of the sensor nodes after random delivery is not very time-sensitive. The position-adjusted coverage rate is significant, therefore it makes sense to increase the amount of computation to achieve higher wireless sensor network deployment performance.

#### 4. VFLGWO Algorithm Simulation

Firstly, the relevant parameters of the VFLGWO algorithm are discussed through simulation experiments. Different simulation experiments are designed to test the performance of the VFLGWO algorithm. The performance indexes of the simulation test in this paper include: coverage rate, uniformity, average moving distance of sensor nodes (m), and running time (s). When testing the average moving distance of sensor nodes, the VFLGWO algorithm applies the sensor nodes matching algorithm proposed in this paper. Other comparison algorithms still use the conventional method in which the  $S_0$  and  $S_0'$  are matched by the sequence number of sensor nodes.

In this paper, the experiments are performed using MATLAB 2014. The simulation environment is an area of 50 m  $\times$  50 m. The number of randomly distributed mobile sensor nodes is  $N = 50$ , and the sensing radius is  $R_s = 5$  m for all sensor nodes; the communication radius is  $R = 2R_s = 10$  m, the detection probability threshold is  $C_{th} = 0.8$ , the measurement reliability parameter is  $r_e = 2.5$  m, the number of basic iterations is  $T = 3000$  generations, and the population number is  $sizepop = 30$  (the algorithm uses the cell array to store the sensor node position coordinates, and each wolf contains 50 sensor node position coordinates).

##### 4.1. VFLGWO Algorithm Parameter Selection

###### 4.1.1. Determination of Gravity and Repulsion Parameters

According to the gravity and repulsion parameters of the VFLGWO algorithm, six groups of experiments are used to obtain relatively optimal  $\omega_\alpha$  and  $\omega_\gamma$ . Each group of experiments is performed 20 times independently, and the average values of each indicator are obtained. The control variable method is applied. Firstly, the *MaxStep* is unified at 1.2 m, and the experimental value of  $\omega_\alpha$  is unchanged at 1. The values of  $\omega_\gamma$  are 600, 800, 1000, 1500 and 2000, respectively. Since this article mainly discusses how to deploy sensor nodes to maximize the coverage rate of the monitoring area, coverage rate is taken as the first indicator. The four optimization indicators calculated by the VFLGWO algorithm are listed in Table 1. As can be seen from Table 1, when the value of  $\omega_\gamma$  is 1000, the coverage rate is the largest. Therefore, in the simulation experiment of the proposed algorithm,  $\omega_\alpha$  is set to 1 and  $\omega_\gamma$  is set to 1000.

**Table 1.** Four indicators of the Virtual Force-Lévy-embedded Grey Wolf Optimization (VFLGWO) algorithm with different  $\omega_\alpha$  values.

$\omega_\gamma$	Coverage Rate	Uniformity	Moving Distance	Running Time
600	0.9189	1.2521	6.79	1184.5
800	0.93	1.2406	6.95	1133.3
1000	0.9427	1.1014	7.52	1085.8
1500	0.933	1.1412	6.99	1104.5
2000	0.9308	1.0597	6.92	1101.1

###### 4.1.2. Determination of the *MaxStep* Parameter

Five groups of experiments are designed for the moving step *MaxStep* of the sensor nodes in the VFLGWO algorithm. Each group of experiments is performed 20 times, and the average value of each indicator is calculated. The *MaxStep* values are 0.6 m, 0.8 m, 1 m, 1.2 m and 1.4 m, respectively.

The experimental results are shown in Table 2. When the *MaxStep* is 1.2 m, the coverage rate is the largest. Therefore, set the *MaxStep* to 1.2 m.

**Table 2.** Comparison of four indicators of the VFLGWO algorithm with different *MaxStep* values.

<i>MaxStep</i>	Coverage Rate	Uniformity	Moving Distance	Running Time
0.6	0.9273	1.3159	7.33	1050.8
0.8	0.9293	1.2298	7.09	1048.4
1.0	0.9319	1.3202	7.61	1035.7
1.2	0.9427	1.1014	7.52	1085.8
1.4	0.9302	1.2457	7.51	1060.8

#### 4.2. VFLGWO Algorithm Effectiveness Simulation Experiment

In order to verify the performance of the VFLGWO algorithm, the VFLGWO is compared with the wireless sensor network coverage optimization algorithm using Chaotic Particle Swarm (CPSO), Cuckoo Search (CS), and LGWO.

##### 4.2.1. Comparison Algorithm Parameter Settings

The parameters of the CPSO, CS, and LGWO are set as shown in Table 3 [7]. The other parameters, such as the size of the population *sizepop* and the number of basic iterations *T*, are the same as those of the VFLGWO algorithm.

**Table 3.** The Chaos Particle Swarm Optimization (CPSO), Cuckoo Search (CS), and Lévy-embedded Grey Wolf Optimization (LGWO) algorithm parameter settings.

Optimizer	Description
CPSO	$c_1, c_2 = 2$
CS	$p_a = 0.25$
LGWO	$a_0 = 2, \beta \sim U(0,2), p \sim U(0,1)$

##### 4.2.2. Deployments of the Sensor Positions after Optimization of Four Algorithms

Figure 3 shows the initial positions of the sensor nodes. Figure 4a–d are the development of the best solutions (each large circle represents a schematic representation of the sensing area of the sensor nodes at its center, with a perceived radius  $R_s = 5$  m) after one optimization by the CPSO, CS, LGWO, and VFLGWO algorithms. The coverage rate and uniformity of the CPSO, CS, LGWO, and VFLGWO algorithms for this experiment are listed in Table 4. It can be seen from Figure 4 that the sensor nodes have the most uniform distribution and the largest coverage rate after the VFLGWO algorithm is optimized. The experimental data in Table 4 also confirmed this conclusion.

**Table 4.** Comparison of algorithm coverage and uniformity.

Optimizer	Coverage Rate	Uniformity
CPSO	0.8376	1.3475
CS	0.8856	1.6637
LGWO	0.9108	1.5954
VFLGWO	0.9452	1.1385

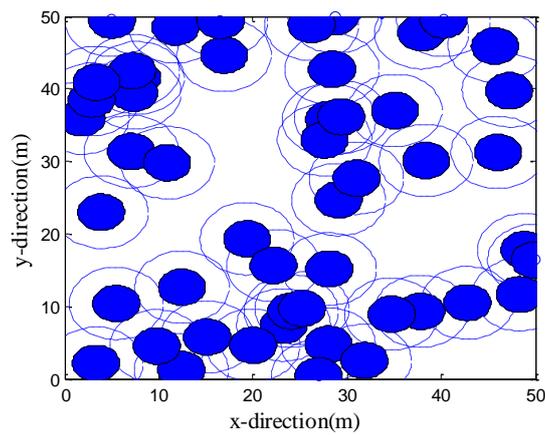


Figure 3. Initial positions of the sensors.

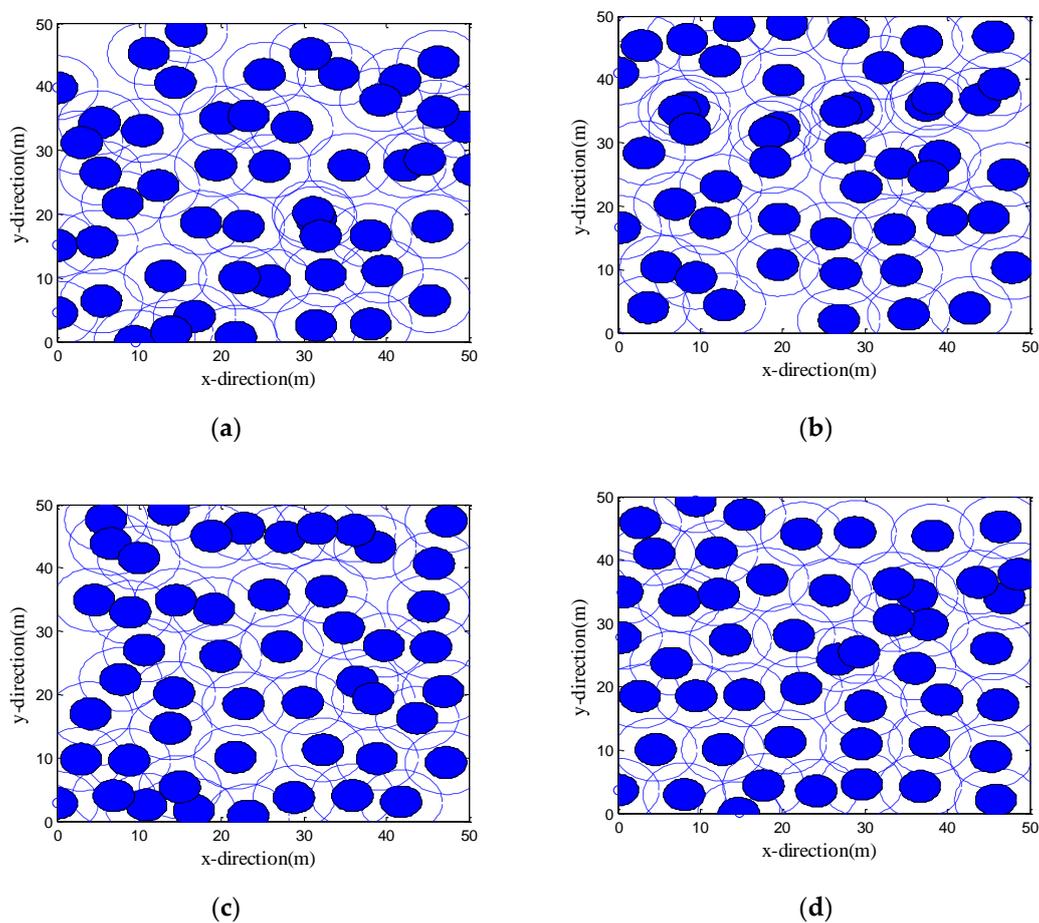


Figure 4. Final deployments of four algorithms. (a) Chaos Particle Swarm Optimization (CPSO) algorithm, (b) Cuckoo Search (CS) algorithm, (c) Lévy-embedded Grey Wolf Optimization (LGWO) algorithm, (d) VFLGWO algorithm.

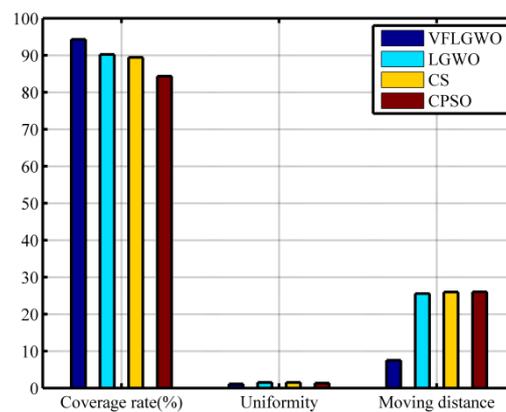
#### 4.2.3. Multiple Performance Index of Four Algorithms

In this paper, 20 independent experiments of 3000 generations are performed on the four algorithms. The average values of the coverage, uniformity, average moving distance of the sensor nodes, and running time are listed in Table 5. In order to facilitate visual comparison, the data of Table 5 is plotted as a histogram, as shown in Figure 5. It can be seen from Table 5 that the coverage rate of the VFLGWO is improved by 4.05%, 4.87% and 9.94% compared with the other three algorithms,

respectively; the uniformity of the VFLGWO algorithm is optimal. The average moving distance of the sensor nodes of the VFLGWO algorithm is significantly shorter than the average moving distance of the sensor nodes of the other three algorithms. In terms of running time of the VFLGWO algorithm, it is about twice that of the other algorithms.

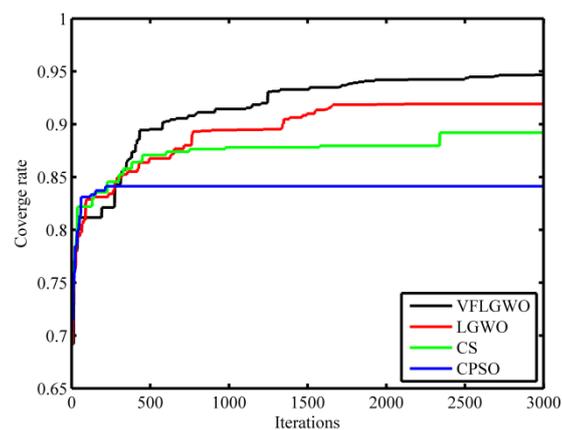
**Table 5.** Indicators of the CPSO, CS, LGWO and VFLGWO algorithms for 3000 generations.

Optimizer	Coverage	Uniformity	Moving Distance	Running Time
CPSO	0.8433	1.3549	26.01	513.2
CS	0.894	1.5171	25.98	530.4
LGWO	0.9022	1.5064	25.54	537.1
VFLGWO	0.9427	1.1014	7.52	1085.8



**Figure 5.** Comparison of the four algorithms.

Figure 6 shows the network coverage rate curves of the four algorithms, CPSO, CS, LGWO and VFLGWO; running 3000 generations. It can be seen from Figure 6 that the coverage rate of the VFLGWO algorithm is significantly larger than that of the other three algorithms.



**Figure 6.** Coverage rate evolution curves of the four algorithms.

### 4.3. Comprehensive Comparison of Experimental Results

#### 4.3.1. Different Numbers of Sensors in the Same Area

The performance index of the coverage rate, uniformity, and average moving distance of the number of sensor nodes placed in the 50 m × 50 m monitoring area from 40 to 60 (each group adds five sensor nodes) are listed in Tables 6–8. In order to facilitate visual comparison, the data of Tables 6–8 are plotted as graphs, as shown in Figure 7. It can be seen from Figure 7 that when the size of the

monitoring area is the same, the coverage rate of the four algorithms increases as the number of sensor nodes increases. This is because the number of sensor nodes per unit area in the monitoring area increases, and the area coverage rate naturally increases. At the same time, the coverage rate and uniformity of the VFLGWO algorithm are better than the other three algorithms, and the average moving distance of the sensor nodes of the VFLGWO algorithm is significantly shorter than the other three algorithms.

**Table 6.** Coverage rate with different numbers of sensor nodes.

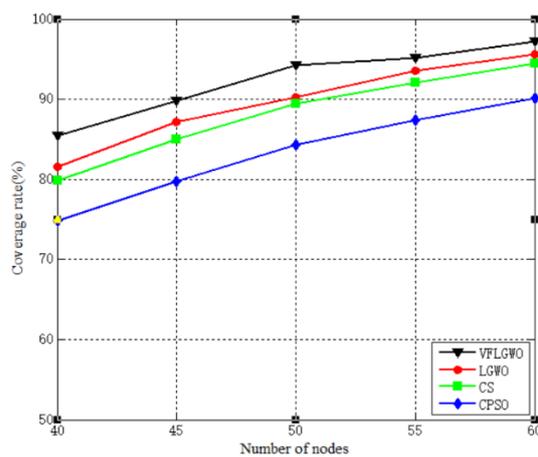
Number of Nodes	40	45	50	55	60
CPSO	0.7478	0.7972	0.8433	0.8744	0.9016
CS	0.7982	0.8501	0.894	0.9201	0.9452
LGWO	0.8153	0.8721	0.9022	0.9358	0.9564
VFLGWO	0.8546	0.8974	0.9427	0.9515	0.9725

**Table 7.** Uniformity with different number of sensor nodes.

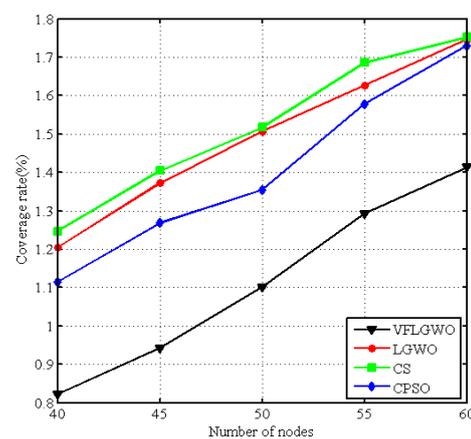
Number of Nodes	40	45	50	55	60
CPSO	1.1141	1.268	1.3549	1.5779	1.7299
CS	1.2464	1.4044	1.5171	1.6857	1.7526
LGWO	1.2041	1.3722	1.5064	1.6265	1.7466
VFLGWO	0.8214	0.9419	1.1014	1.2923	1.413

**Table 8.** Average moving distance with different number of sensor nodes.

Number of Nodes	40	45	50	55	60
CPSO	28.32	26.16	26.01	27.26	28.18
CS	26.07	26.29	25.98	26.25	26.67
LGWO	26.47	26.45	25.54	25.36	26.06
VFLGWO	6.91	6.9	7.52	7.53	7.63

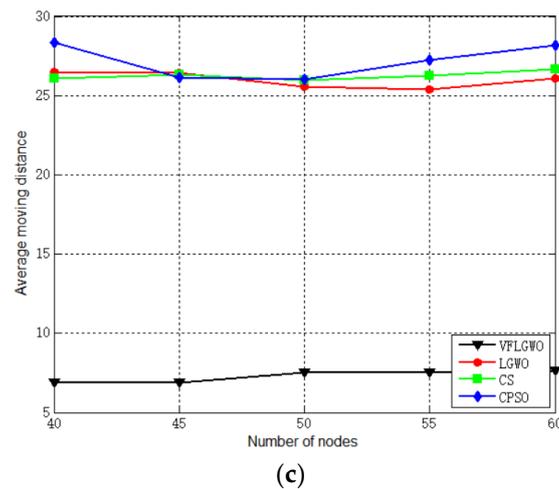


(a)



(b)

**Figure 7.** Cont.



**Figure 7.** Performance index of the four algorithms with different numbers of sensor nodes. (a) Coverage rate of the four algorithms. (b) Uniformity of the four algorithms. (c) Average moving distance of the four algorithms.

#### 4.3.2. Same Sensor Node Density with Different Monitoring Areas

The monitoring area includes 32 sensor nodes in a monitoring area of 40 m × 40 m (50 m<sup>2</sup>/node), and 72 sensor nodes in a monitoring area of 60 m × 60 m (50 m<sup>2</sup>/node). The experimental results are compared with the experimental data of 50 sensor nodes placed in the monitoring area of 50 m × 50 m (50 m<sup>2</sup>/node) in Section 4.3.1. The results are shown in Tables 9–11. In order to facilitate visual comparison, the data of Tables 9–11 are plotted as histograms, as shown in Figure 8. It can be seen from Figure 8 that the VFLGWO algorithm is superior to the other three algorithms in coverage rate, uniformity, and average moving distance of the sensor nodes in different size monitoring areas.

**Table 9.** Coverage rate in different size monitoring areas.

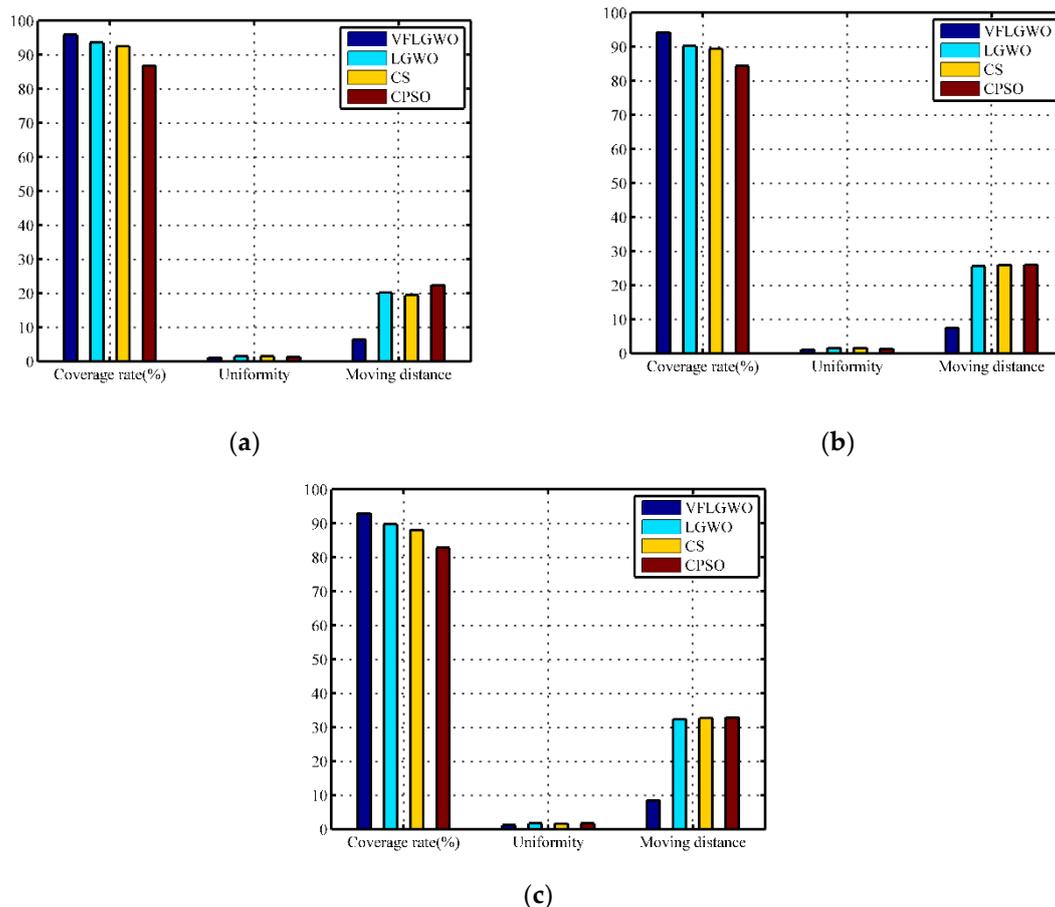
Monitoring Area	40 m × 40 m	50 m × 50 m	60 m × 60 m
CPSO	0.8673	0.8433	0.8279
CS	0.9249	0.894	0.8806
LGWO	0.9361	0.9022	0.8979
VFLGWO	0.959	0.9427	0.9292

**Table 10.** Uniformity in different size monitoring areas.

Monitoring Area	40 m × 40 m	50 m × 50 m	60 m × 60 m
CPSO	1.2449	1.3549	1.6919
CS	1.4877	1.5171	1.5977
LGWO	1.4585	1.5064	1.6436
VFLGWO	1.006	1.1014	1.1228

**Table 11.** Average moving distance of sensor nodes in different size monitoring areas.

Monitoring Area	40 m × 40 m	50 m × 50 m	60 m × 60 m
CPSO	22.34	26.01	32.87
CS	19.41	25.98	32.67
LGWO	20.12	25.54	32.41
VFLGWO	6.36	7.52	8.4955



**Figure 8.** Performance index of four algorithms in different size monitoring areas. (a) 40 m × 40 m, (b) 50 m × 50 m, (c) 60 m × 60 m.

In summary, coverage rate, uniformity, and average moving distance of sensor nodes of the VFLGWO algorithm achieve a better performance compared to the other three algorithms. The sensor nodes adjusted by the VFLGWO algorithm are more uniformly distributed and have shorter sensor node movement distances. This paper also performs the indicators of the VFLGWO algorithm when the working environment changes. The experimental results of the VFLGWO algorithm are still better than the experimental results of the CPSO, CS and LGWO algorithms.

## 5. Conclusions

In this paper, the modified LGWO algorithm is applied to WSN node deployment. In order to obtain a better performance index, a coverage optimization algorithm for wireless sensor networks with the Virtual Force-Lévy-embedded Grey Wolf Optimization (VFLGWO) algorithm is proposed to solve the coverage optimization problem. The VFLGWO algorithm can effectively make sensor node distribution more uniform, the coverage rate higher in the same environment, the average moving distance of the sensor nodes shorter, and it can ensure the expected design goal is obtained. The simulation results show that the wireless sensor network coverage rate, uniformity, and average moving distance of the VFLGWO algorithm are significantly better than those of the CPSO, CS and LGWO, and the VFLGWO algorithm has good environmental adaptability.

After obtaining the optimal position of the wireless sensor nodes through the VFLGWO algorithm, the sensor nodes matching algorithm proposed in this paper can greatly shorten the average moving distance of the sensor nodes. The running time of the VFLGWO algorithm is relatively long, however because there is no need for frequent scheduling during the deployment of wireless sensor nodes, it will not affect the deployment of WSN nodes—and the coverage rate, uniformity, and average

moving distance of the nodes are more important indicators. They affect the regional coverage of the wireless sensor network monitoring area. Our future work will apply the proposed algorithm to the actual scene.

**Author Contributions:** S.W.: Conceptualization, Formal analysis, Methodology, Software, Writing—original draft. X.Y.: Project administration, Supervision, Writing—review & editing. X.W.: Investigation, Writing—review & editing. Z.Q.: Funding acquisition, Project administration, Resources, Writing—review & editing.

**Funding:** This work is supported by the Special Fund Project of Jilin Province School Co-Construction Plan (No. SXGJQY2017-9), the High-Level Technology Innovation Team Building Project of Jilin University (No. 2017TD-19) and the National Natural Science Foundation of China (61771219).

**Conflicts of Interest:** The authors state that there is no conflict of interest.

## References

1. Ni, Q.; Du, H.; Pan, Q.; Cao, C.; Zhai, Y. An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization. *Nat. Comput.* **2017**, *16*, 5–13. [[CrossRef](#)]
2. Zhou, Y.; Zhao, R.; Luo, Q.; Wen, C. Sensor Deployment Scheme Based on Social Spider Optimization Algorithm for Wireless Sensor Networks. *Neural Process. Lett.* **2017**, 1–24. [[CrossRef](#)]
3. Ho, N.L.; Dao, T.-P.; Le, H.G.; Le Chau, N. Optimal design of a compliant microgripper for assemble system of cell phone vibration motor using a hybrid approach of ANFIS and Jaya. *Arab. J. Sci. Eng.* **2019**, *44*, 1205–1220. [[CrossRef](#)]
4. Ho, N.L.; Dao, T.-P.; Le Chau, N.; Huang, S.-C. Multi-objective optimization design of a compliant microgripper based on hybrid teaching learning-based optimization algorithm. *Microsyst. Technol.* **2018**, 1–17. [[CrossRef](#)]
5. Yang, X.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; pp. 210–214.
6. Dao, T.P.; Huang, S.C.; Thang P, T. Hybrid Taguchi-cuckoo search algorithm for optimization of a compliant focus positioning platform. *Appl. Soft Comput.* **2017**, *57*, 526–538. [[CrossRef](#)]
7. Heidari, A.A.; Pahlavani, P. An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Appl. Soft Comput.* **2017**, *60*, 115–134. [[CrossRef](#)]
8. Zhang, G.; You, S.; Ren, J.; Li, D.; Wang, L. Local coverage optimization strategy based on voronoi for directional sensor networks. *Sensors* **2016**, *16*, 2183. [[CrossRef](#)] [[PubMed](#)]
9. Qin, N.; Chen, J. An area coverage algorithm for wireless sensor networks based on differential evolution. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718796734. [[CrossRef](#)]
10. Céspedes-Mota, A.; Castanon, G.; Martínez-Herrera, A.F.; Cárdenas-Barrón, L.E.; Sarmiento, A.M. Differential evolution algorithm applied to wireless sensor distribution on different geometric shapes with area and energy optimization. *J. Netw. Comput. Appl.* **2018**, *119*, 14–23. [[CrossRef](#)]
11. Jia, J.; Chen, J.; Chang, G.-R.; Zhao, L.-L.; Wang, G.-X. Optimal coverage scheme based on genetic algorithm in wireless sensor networks. *Control Decis.* **2007**, *22*, 1289.
12. Cong, C. A Coverage Algorithm for WSN Based on the Improved PSO. In Proceedings of the 2015 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Halong Bay, Vietnam, 19–20 December 2015.
13. Yu, B.; Yuanping, W.; Liang, Z.; Yuan, H.; Aijuan, Z. Relay Node Deployment for Wireless Sensor Networks Based on PSO. In Proceedings of the IEEE International Conference on Computer & Information Technology; Ubiquitous Computing & Communications, Liverpool, UK, 26–28 October 2015.
14. Wang, X.; Zhang, H.; Fan, S.; Gu, H. Coverage control of sensor networks in IoT based on RPSO. *IEEE Internet Things J.* **2018**, *5*, 3521–3532. [[CrossRef](#)]
15. Chau, N.L.; Le, H.G.; Dao, T.P.; Dang, M.P.; Dang, V.A. Efficient Hybrid Method of FEA-Based RSM and PSO Algorithm for Multi-Objective Optimization Design for a Compliant Rotary Joint for Upper Limb Assistive Device. *Math. Probl. Eng.* **2019**. [[CrossRef](#)]
16. Ozturk, C.; Karaboga, D.; Gorkemli, B. Probabilistic Dynamic Deployment of Wireless Sensor Networks by Artificial Bee Colony Algorithm. *Sensors* **2011**, *11*, 6056–6065. [[CrossRef](#)] [[PubMed](#)]

17. Xia, J. Coverage Optimization Strategy of Wireless Sensor Network Based on Swarm Intelligence Algorithm. In Proceedings of the International Conference on Smart City & Systems Engineering, Changsha, China, 11–12 November 2017.
18. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
19. Jia, J.; Chen, J. An energy-balanced self-deployment algorithm based on virtual force for mobile sensor networks. *Int. J. Sens. Netw.* **2015**, *19*, 150–160. [[CrossRef](#)]
20. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization based on virtual forces. In Proceedings of the Joint Conference of the IEEE Computer & Communications IEEE Societies, San Francisco, CA, USA, 30 March–3 April 2003.
21. Li, Y.; Zhang, B.; Chai, S. An energy balanced-virtual force algorithm for Mobile-WSNs. In Proceedings of the IEEE International Conference on Mechatronics & Automation, Beijing, China, 2–5 August 2015.
22. Changxing, L.; Qing, Z.; Long-yao, Z. Research on wireless sensor network coverage based on improved particle swarm optimization algorithm. In Proceedings of the 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 23–25 September 2017; pp. 305–311.



© 2019 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/>).