sensors

MDPI

Article

An Adaptive, Discrete Space Oriented Wolf Pack Optimization Algorithm for a Movable Wireless Sensor Network

Dongxing Wang [1,2], Huibo Wang [1,*], Xiaojuan Ban [2,*], Xu Qian [1] and Jingxiu Ni [3]

[1] School of Mechanical Electronic & Information Engineering, China University of Mining & Technology, Beijing 100083, China; wangdongxing85@163.com (D.W.); xuqiancumtb@163.com (X.Q.)

[2] Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, Beijing 100083, China

[3] Engineering Integrated Experimental Teaching Demonstration Center, Beijing Union University, Beijing 100101, China; njx1211@163.com

* Correspondence: chinawanghuibo@126.com (H.W.); banxj@ustb.edu.cn (X.B.)

Received: 18 July 2019; Accepted: 1 October 2019; Published: 6 October 2019

Abstract: Recently, many related algorithms have been proposed to find an efficient wireless sensor network with good sustainability, a stable connection, and a high covering rate. To further improve the coverage rate of movable wireless sensor networks under the condition of guaranteed connectivity, this paper proposes an adaptive, discrete space oriented wolf pack optimization algorithm for a movable wireless sensor network (DSO-WPOA). Firstly, a strategy of adaptive expansion based on a minimum overlapping full-coverage model is designed to achieve minimum overlap and no-gap coverage for the monitoring area. Moreover, the adaptive shrinking grid search wolf pack optimization algorithm (ASGS-CWOA) is improved to optimize the movable wireless sensor network, which is a discrete space oriented problem. This improvement includes the usage of a target–node probability matrix and the design of an adaptive step size method, both of which work together to enhance the convergence speed and global optimization ability of the algorithm. Theoretical research and experimental results indicate that compared with the coverage algorithm based on particle swarm optimization (PSO-WSN) and classical virtual force algorithm, the newly proposed algorithm possesses the best coverage rate, better stability, acceptable performance in terms of time, advantages in energy savings, and no gaps.

Keywords: wireless sensor network; coverage rate; wolf pack algorithm; optimization; probability matrix; adaptive step size

1. Introduction

The wireless sensor network (WSN) is one of the most promising technologies for some real-time applications because of its size, cost-effectiveness, and easily deployable nature [1]. For decades, with the rapid development of the Internet of Things, artificial intelligence, and other advanced technologies, the use of wireless networks as a basic technology has been studied a great deal by researchers, such as in [2], in which Dai et al. proposed a novel multichannel network with infrastructure support (called the MC-IS network), which has not been studied in the literature. To solve the problems of large-scale wireless networks, with their wide variety, high volume, real-time velocity data and huge value, which leads to unique research challenges that are different from existing computing systems, in [3], researchers presented a survey of state-of-the-art big data analytics (BDA) approaches for large-scale wireless networks and the technical solutions to challenges in BDA for large-scale wireless networks according to each stage in the life cycle of BDA. They also discussed open research issues and outlined future directions in this promising area.

In particular, WSNs are being used for more and more applications, such as in military reconnaissance, environmental monitoring, smart homes, and medical health, and so on; for example, in [4], Thakur et al. reviewed the applicability of wireless sensor networks in precision agriculture to identify various WSN technologies adopted for precision agriculture and the impact of these technologies on achieving smart agriculture. In [5], Li et al. presented a forest fire detection system framework based on WSN and its implementation scheme, then they discussed some key problems, focusing on forest fire forecast modeling, WSN node deployment, WSN nodes and forest fire positioning, and wireless communication control protocol. In [6], WSN was used to monitor the water environment and automatically collect information on the ecological environment, meteorological environment, water pollution, and so on. In [7], because existing opportunity networks can hardly satisfy the requirements of disaster scenarios for low-latency messaging, Fu et al. developed a new opportunistic network framework, the WSN-assisted opportunistic network (WAON), and proposed a forwarding mechanism, net spray for WAON, which supports mobile-to-mobile, static-to-mobile, mobile-to-static, and static-to-static operations.

However, in a complex and changeable working environment, WSNs need a large number of sensor nodes to cooperate with each other to complete detection and sensing tasks [8]. For various monitoring areas, a reasonable arrangement of sensor nodes is conducive to improving the working efficiency and reducing the energy consumption of WSNs. Mobile nodes are used to dynamically reconstruct the network structure, expand the network coverage rate, and optimize performance; thus, they have become one of the key technologies of WSNs [9].

Accordingly, in recent years, researchers have put forward many effective methods for solving the location optimization problem of wireless sensor nodes from different perspectives. In [10], Zhou et al. proposed an improved algorithm based on virtual force where the WSN was divided into grids and nodes, which were redeployed under the adaptively chosen resultant force to enable WSN to gain the best coverage. In [11], in order to solve the coverage problem, Zhang et al. proposed a coverage algorithm based on virtual forces including an enhanced coverage algorithm, a connectivity-preserved method for neighboring nodes, and a coverage algorithm for the relevant area. In [12], Yarinezhad et al. proposed a fixed parameter tractable (FPT) approximation algorithm with an approximation factor of 1.2 for the load-balanced clustering problem (LBCP) and an energy-efficient, balanced routing algorithm, which effectively solved the load-balanced clustering problem and maximized the network's lifetime by reducing the energy consumption. In [13], Chen et al. proposed a novel wireless sensor network with energy-efficient coverage that achieved a good balance between target coverage and energy consumption by fusing the genetic algorithm (GA) and WSN. In [14], Somaieh et al. proposed the distributed energy-aware hexagon-based clustering algorithm to improve coverage (DEHCIC), which considers energy and topological features such as the number of mobile neighbor nodes and number of neighbor nodes to elect cluster heads and attempts to cover holes as much as possible by static sensor nodes, while the closest mobile node is used to cover holes if this is not possible; furthermore, the proposed algorithm retains sensor nodes in an active mode that covers interest points and puts others into a low-power sleep mode. In [15], Alavi et al. proposed a distributed event-triggered control strategy for DC microgrids, based on the publish–subscribe model over industrial wireless sensor networks, to efficiently stabilize grid voltage and to further balance the energy level of energy storage systems. In [16], Sajwan et al. proposed a novel routing algorithm for wireless sensor networks, which achieved uniform energy depletion across all nodes and, thus, led to a prolonged network lifetime. In [17], to improve network coverage, So-In et al. proposed a novel distributed deployment algorithm, the coverage hole-healing algorithm (CHHA), to maximize area coverage. In [18], Wang et al. proposed "parallel particle swarm optimization-based mobile sensor node deployment in wireless sensor networks" by adopting particle swarm optimization (PSO) in a parallel mechanism to optimize the deployment of mobile sensor nodes.

Unfortunately, all of these algorithms have their own shortcomings, in that some coverage rates are insufficient, some can only be used in specific environments due to constraints, and so on.

In 2007, Yang and co-authors proposed the wolf swarm algorithm [19], which is a new swarm intelligence algorithm. Because of its excellent performance, some special variants of the wolf pack optimization algorithm have been proposed to gain optimal solutions. For example, in [20], Wang et al. proposed "an adaptive distributed size wolf pack optimization algorithm using the strategy of jumping for raids" in order to enhance performance. In [21], by introducing the position order coding method in the initialization phase, Huang et al. studied the path optimization problem in discrete domains and introduced a secondary search in the iterative process to improve the speed and accuracy of the algorithm to determine the optimal solution before the maximum number of iterations. In [22], Wu et al. proposed a discrete wolf pack algorithm by redesigning the position of artificial wolves and intelligent behaviors to solve the traveling salesman problem. In combination with the probabilistic nearest neighbor method, the proposed algorithm preserved the cooperative searching feature based on the job distribution of the wolf pack and was able to balance both the breadth and depth of the searching ability. In [23], a novel, discrete GWO was proposed where a random leader selection was performed, and the probability for the main leader to be selected increased at the detriment of the other leaders across iterations. In [24], a new model based on the discrete wolf pack search (DWPS) algorithm was proposed to maximize the number of satisfied passengers, the total number of transfers, and the total travel time of all served passengers for a transit network design problem.

Consequently, the low coverage of WSN and good performance of the wolf pack optimization algorithm allow us to attempt to fuse the coverage problem between the two. Therefore, on the premise of ensuring adequate coverage and absolute connectivity, this paper proposes a method to find the shortest path to minimize the energy consumption of the whole sensor network.

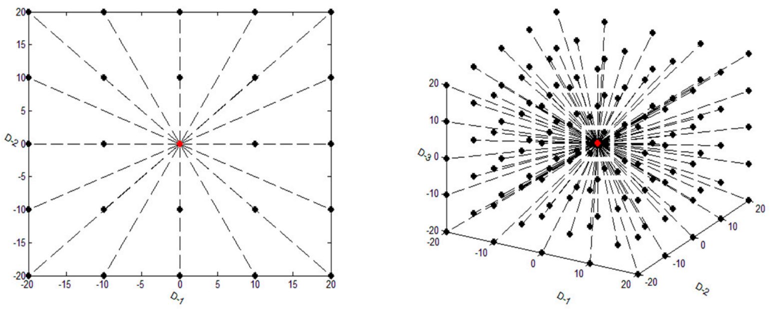
2. Related Works

2.1. *Adaptive Shrinking Grid Search Wolf Pack Optimization Algorithm (ASGS-CWOA) [25]*

The ASGS-CWOA was proposed in 2018 by Wang et al. in [25], and it is an efficient variant of the wolf pack optimization algorithm in the optimization field aimed at solving application problems efficiently. Its key improvements are as follows.

- Method of Adaptive Shrinking Grid Search (ASGS)

According to the principle of ASGS, during the process of migration or siege, the location of the current wolf is taken as the center of the search in D directions at the same time (here, D is the dimension of the solution space). The diagram can be seen in Figure 1.



(**a**) (**b**)

Figure 1. (**a**) The searching situation of wolves in adaptive shrinking grid search (ASGS-CWOA) when the number of dimensions is 2; (**b**) the situation when the number of dimensions is 3, where the step size of migration is as follows: step_a = 10, or the step size of siege, step_c = 10; and the searching number is K = 2 (the red point means the current location of the wolf, while the black ones mean the searching locations of the wolf).

From Figure 1, it is easy to see that a large number of points is distributed around the current wolf in some special way, and that means the best solution should be found in the local neighborhood space centered on the current wolf at the end of this process.

Generally, the strategy of ASGS enhances the performance of exploitation.

- Strategy of Opposite–Middle Raid (OMR)

The main idea of this is that the opposite location from the current wolf relative to the lead wolf should be inspected first; if the opposite wolf has better fitness than the current one, then the current wolf will move to the opposite one. Otherwise, 2D points (D means the dimension number of the solution space) around the current point and the middle point (regarding the current wolf and the leader wolf) should be inspected. Thus, the point with the best local fitness can be found, which becomes the new location of the current wolf at the end of this process.

In short, this enhances the performance of exploration.

- Adaptive Standard Deviation Updating Amount (ASDUA)

ASDUA is a dynamic number that reflects the dynamic situation of the wolf pack during any iteration, which means the population of the wolf pack and the standard deviation of its fitness determine how many wolves there are and who will be eliminated and regenerated. The standard deviation can be obtained by Equation (1):

$$\sigma = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(x_i-\mu)^2}\,, \tag{1}$$

where N means the size of the wolf pack; x_i means the fitness of the i-th wolf; and μ is the mean value of fitness. Then, ASDUA is gained by the following formula.

$$ASDUA\begin{cases} ASDUA+1,\ if\ fitness(xi) < (\mu-\sigma/2) \\ Do\ nothing,\ if\ fitness(xi) \geq (\mu-\sigma/2) \end{cases} \tag{2}$$

Aiming at the problem of solving the maximum, ASDUA is zero when the iteration begins; next, the difference between the mean value and the SD regarding the fitness of the wolf pack should be computed, and if the fitness of the current wolf is less than the difference, ASDUA increases by 1; otherwise, nothing changes. Thus, the value of ASDUA is obtained, and ASDUA wolves are eliminated and regenerated.

In brief, this enhances the performance of the exploration as well as the OMR.

2.2. Some Covering Models

Generally speaking, a wireless sensor network hopes to cover all the monitered area and include a few sensors that are as far apart as possible. Several classical covering models are listed as follows.

A: Inscribed Rectangular – Full Covering Model (IRFCM)

As seen in Figure 2a, in IRFCM, the area of the maximum inscribed rectangle about the sensing circle is the effective covering area of the corresponding sensor, and it is under this model that the covering rate of the wireless sensor network (WSN) is high and is easy to expand; however, the overlapping rate is too high, and its value reaches (π − 2)/π (π means the circular constant). This model requires that Rc is greater than or equal to $\sqrt{2} * $ Rs.

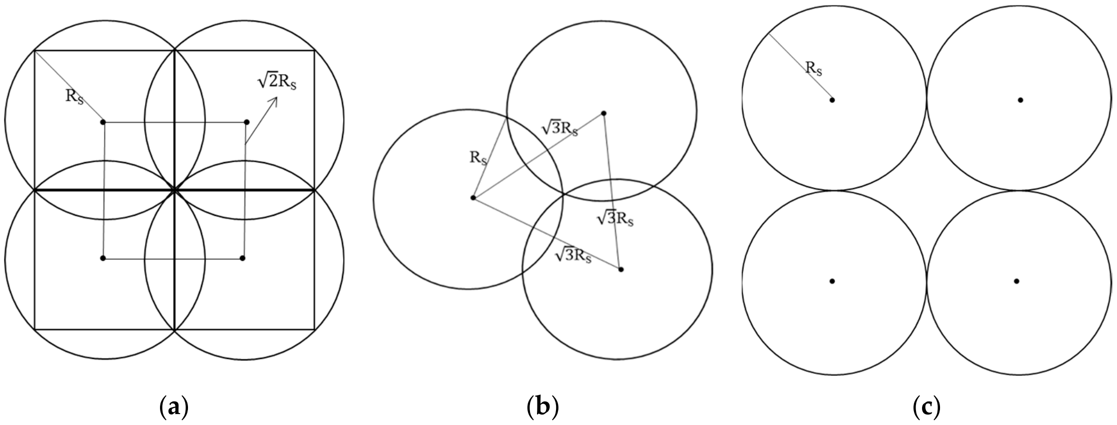


Figure 2. (**a**) The diagram of the inscribed rectangular full covering model; (**b**) the diagram of the minimum overlapping full covering model; and (**c**) the diagram of the tangent covering model of the adjacent boundary (the black point means the current location of the sensor, while the black circle means the boundary of the area sensed by the sensor).

B: Minimum Overlapping Full Covering Model (MOFCM)

As seen in Figure 2b, in MOFCM, the area of the overlapping parts of any two intersecting circles can be calculated according to the radius and the distance between their centers, and the overlapping area is Rs2 $*$ $(2 * \pi - 3\sqrt{3})/6$. Thus, the overlapping rate is $(2 * \pi - 3\sqrt{3})/(6 * \pi)$, which is lower than the value $(\pi - 2)/\pi$ from IRFCM, as detailed in Figure 2 (b). Under this model, the covering rate of the WSN is high, as under IRFCM, and it has the possibility to expand. This model requires that Rc is greater than or equal to $\sqrt{3} *$ Rs. A detailed discussion can be found in [17].

C: Tangent Covering Model of Adjacent Boundary

As seen in Figure 2c, under this model, the overlapping rate is zero, which is the best state hoped for, while the coverage is unsatisfactory, in that there are obvious coverage vulnerabilities. This model requires that Rc is greater than or equal to $2 *$ Rs.

2.3. Assumptions

Put simply, some assumptions are as follows regarding the characteristics of this paper.

A: The wireless sensor network should have an intelligent control and management center that can communicate with all wireless sensor nodes and obtain their positions by some methods, including the BEIDOU (Beidou Navigation Satellite System) and GPS (global positioning system), and so on. This intelligent center should own the map of the whole space which the wireless sensor network is intended to monitor.

B: The intelligent center should have optimized computing power in order to obtain the global optimal solution according to the methods in this paper.

C: Wireless sensor nodes should be movable and able to go to the designated location.

D: All the sensor nodes should have same the sensing radius Rs and communication radius Rc, centered on themselves respectively. The Euclidean distance is adopted in this paper.

3. Proposed Method

To adopt ASGS-CWOA with good performance to address the low coverage of WSN, we made improvements in the following aspects.

3.1. Adaptive Expansion Based on the Minimum Overlapping Full Covering Model (AE-MOFCM)

Compared to the other covering models mentioned above, MOFCM makes wireless sensor networks have high coverage and easy deployment as well as an acceptably low overlapping rate. Thus, this model is selected as the basic covering model for the WSN (Figure 3).

$$Target_1 = [(range_{max} - range_{min})/2,\ (range_{max} - range_{min})/2]; \tag{3}$$

$$\left\{ \begin{array}{l} Target_{i+1} = Target_i + [DISTANCE,\ 0] \\ Target_{i+2} = Target_i + \left[DISTANCE/2,\ DISTANCE * \sqrt{3}/2\right] \\ Target_{i+3} = Target_i + \left[-DISTANCE/2,\ DISTANCE * \sqrt{3}/2\right] \\ Target_{i+4} = Target_i + [-DISTANCE,\ 0] \\ Target_{i+5} = Target_i + \left[-DISTANCE/2,\ -DISTANCE * \sqrt{3}/2\right] \\ Target_{i+6} = Target_i + \left[DISTANCE/2,\ -DISTANCE * \sqrt{3}/2\right] \\ If\ Target_i \notin [range_{min},\ range_{max}],\ Target_i = Target_{i+1},\ \left(\mathrm{i}\ =\ 1, 2, \ldots, N_{Target}\right), \end{array} \right. \tag{4}$$

where $range_{max}$ and $range_{min}$ are the upper and lower limit of the solution space, respectively; *DISTANCE* means √3R, and R is the sensing radius of a single sensor; and N_{Target} means the number of the target locations.

$$\left\{ \begin{array}{ll} Assist_{Target\,j} = & (Target_{current} + [R, 0]),\ If\ (Target_{current} + [DISTANCE, 0]) \notin \\ & [range\_min, range\_max\,]\ and\ (Target_{current} + [R, 0]) \in [range\_min, range\_max\,] \\ Assist_{Target\,j} = & \left(Target_{current} + \left[R\sqrt{2}/2, R\sqrt{2}/2\right]\right),\ If\ \left(Target_{current} + \left[DISTANCE/2,\ DISTANCE * \sqrt{3}/2\right]\right) \\ & \notin [range\_min, range\_max\,]\ and\ \left(Target_{current} + \left[R\sqrt{2}/2, R\sqrt{2}/2\right]\right) \\ & \in [range\_min, range\_max\,] \\ Assist_{Target\,j} = & \left(Target_{current} + \left[-R\sqrt{2}/2, R\sqrt{2}/2\right]\right),\ If\ \left(Target_{current} + \left[-DISTANCE/2,\ DISTANCE * \sqrt{3}/2\right)\right. \\ & \notin [range\_min, range\_max\,]\ and\ \left(Target_{current} + \left[-R\sqrt{2}/2, R\sqrt{2}/2\right]\right) \\ & \in [range\_min, range\_max\,] \\ Assist_{Target\,j} = & (Target_{current} + [-R, 0]),\ If\ (Target_{current} + [-DISTANCE, 0]) \\ & \notin [range\_min, range\_max\,]\ and\ (Target_{current} + [-R, 0]) \in [range\_min, range\_max\,] \\ Assist_{Target\,j} = & \left(Target_{current} + \left[-R\sqrt{2}/2, -R\sqrt{2}/2\right]\right),\ If\ (Target_{current} + [-DISTANCE/2,\ -DISTANCE \\ & * \sqrt{3}/2])\ \notin [range\_min, range\_max\,]\ and\ \left(Target_{current} + \left[-R\sqrt{2}/2, -R\sqrt{2}/2\right]\right) \\ & \in [range\_min, range\_max\,] \\ Assist_{Target\,j} = & \left(Target_{current} + \left[R\sqrt{2}/2, -R\sqrt{2}/2\right]\right),\ If\ (Target_{current} + [DISTANCE/2,\ -DISTANCE* \\ & \sqrt{3}/2])\ \notin [range\_min, range\_max\,]\ and\ \left(Target_{current} + \left[R\sqrt{2}/2, -R\sqrt{2}/2\right]\right) \\ & \in [range\_min, range\_max\,] \\ & (j = 1, 2, 3, \ldots). \end{array} \right. \tag{5}$$

First of all, the first target can be generated by Equation (3), and then the others can be obtained by Equation (4). It is important to note in particular that the array of target locations should not take in the targets that are out of [rangemin, rangemax], which should be processed by another array according to Equation (5). Another array named AssistTarget-locations is related to special treatment of the boundary of the solution space. A demo diagram of the AE-MOFCM is shown in Figure 4.

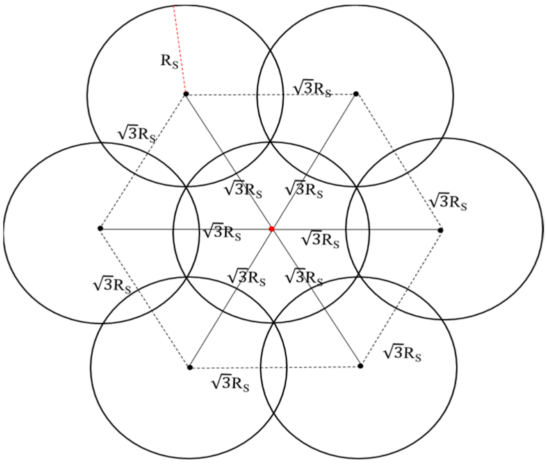
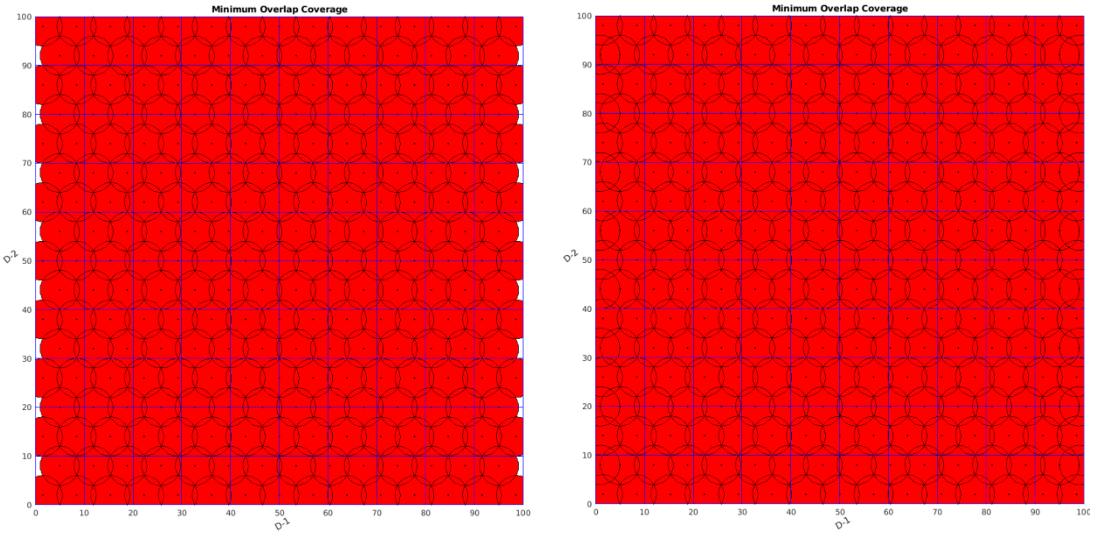


Figure 3. Diagram of the adaptive expansion based on the minimum overlapping full-coverage model (AE-MOFCM). The red point is a single given location; the red dashed line means the sensing radius of sensors, and the value is Rs; while the black points are subsequent points that can be obtained by the corresponding formula.



(**a**) (**b**)

Figure 4. Here, rangemax = 100 and rangemin = 0, while the sensing radius of a single sensor is 4. The black point means a target location, the black circle means the sensing space boundary of a single sensor, and the red area means the whole sensing space of the wireless sensor network, while the very small white areas are vulnerabilities in monitoring. (**a**) The diagram without special treatment of the boundary and (**b**) with special treatment.

3.2. Improvement of ASGS-CWOA

The adaptive shrinking grid search wolf pack optimization algorithm is oriented to the continuous solution space, and the path planning of WSN refers to the discrete solution space, which enables ASGS-CWOA to not be used directly for the coverage problem of WSN. Therefore, we proposed a new variant of the wolf pack optimization algorithm by absorbing some ideas of ASGS-CWOA and making some necessary improvements according to the special conditions of optimization problems oriented to discrete space. The new variant includes two improvements, which are detailed as follows.

3.2.1. Target–Node Probability Matrix

N targets and N sensor nodes are in the WSN, meaning that according to the one-to-one correspondence between sensor nodes and target points, there are (N!) matching options. To choose the optimal, shortest overall path length, some matrices have been designed.

$$Dis_{Matrix-k} = [Distance(target[k], node(1)),\ Distance(target[k], node(2)), \ldots,\ Distance(target[k], node(N\_Node)); \quad (6)$$

$$Dis_{Total_{Reverse}-k} = \sum_{i=1}^{N\_Node} 1/Dis_{Matrix-k}[i]; \quad (7)$$

$$Probability_{Target-k_{disperse}} = [1/Dis_{Matrix-k}[1],\ 1/Dis_{Matrix-k}[2] \ldots, 1/Dis_{Matrix-k}[N\_Node]]/Dis_{Total_{Reverse}-k}; \quad (8)$$

$$Probability_{Target-k}[i] = \sum_{j=1}^{i} Probability_{Target-k_{disperse}}[j]\ (i = 1, 2, \ldots, N\_Node); \quad (9)$$

$$\begin{gathered} Probality_{Matrix}[k] = Target[k] - [Probability_{Target}[1],\ Probability_{Target}[2], \ldots,\ Probability_{Target}[N\_Node] \\ (k = 1, 2, \ldots, N\_Node); \end{gathered} \quad (10)$$

where *N_Node* is the number of wireless sensor nodes, and *N_Target* is the number of destinations that wireless sensor nodes will move to. $Dis_{Matrix\text{-}k}$ indicates the matrix consisting of the distances of each sensor node to the *k*-th target location. $Dis_{Total\_Reverse\text{-}k}$ indicates the reciprocal sum of all elements of the distance matrix about the *k*-th target. $Probability_{Target\text{-}k\_disperse}$ indicates the probability matrix consisting of the possibilities of each node moving to the *k*-th target location. $Probability_{Matrix}[k]$ is another expression of $Probability_{Target\text{-}k\_disperse}$ with a continuous space between 0 and 1, and it is conducive to the use of Roulette for computing.

$$Min - Dis_{Matrix} = sort(Dis_{Matrix},\ ascend), \quad (11)$$

where $Min\text{-}Dis_{Matrix}$ means the ascending index arranged according to values of Dis_{Matrix}; $sort(Dis_{Matrix},$ *ascend)* will return an index, which is arranged as ascending according to values of Dis_{Matrix}.

In order to describe the practical application of the target–node probability matrix in the wolf pack optimization algorithm, an example is shown as follows.

Put simply, the range of the solution space is set as [0, 100], and the sensing radius of a sensor is 30. Thus, the targets can be obtained by AE-MOFCM, and the sensors can be obtained randomly by Equations (13) and (14).

Targets = (50, 50); (75.9808, 95); (24.0192, 95); (24.0192, 5); (75.9808, 5); (80, 50); (20, 50); (97.1940, 73.7868); (2.8060, 73.7868); (2.8060, 26.2132); (97.1940, 26.2132).

Sensors = (9.4925, 68.6594); (77.9807, 98.643); (18.8638, 98.5761); (0.9487, 45.6081); (41.0077, 34.8011); (14.1326, 98.2914); (53.2759, 48.587); (55.776, 31.5094); (69.344, 38.1385); (87.6369, 1.801); (1.9635, 50.3632).

The Dis_{Matrix} can be obtained by Equation (6), $Probability_{Target\text{-}Disperse}$ can be obtained by Equations (7) and (8), $Probability_{Target}$ can be obtained by Equations (9) and (10), and $Min\text{-}Dis_{Matrix}$ can be obtained by Equation (11), detailed as follows.

$$Dis_{Matrix} =$$
$$\begin{matrix}
43.7789 & 51.1520 & 23.4172 & 48.5644 & 41.0088 & 45.2269 & 43.7371 & 59.0807 & 22.8023 & 52.4575 & 52.6837 \\
89.5848 & 102.7955 & 34.1164 & 96.6705 & 82.3390 & 96.4476 & 58.4000 & 110.9124 & 65.4490 & 57.7990 & 75.2081 \\
94.5011 & 93.0703 & 57.3921 & 69.9634 & 50.2953 & 77.6262 & 14.6844 & 93.4426 & 43.7937 & 6.5906 & 28.7570 \\
34.7525 & 8.1325 & 73.0260 & 27.7816 & 44.5392 & 13.7408 & 76.2582 & 8.9160 & 46.4362 & 86.9647 & 72.7660 \\
17.3990 & 44.3956 & 56.5939 & 72.2643 & 78.9547 & 58.8652 & 94.9223 & 60.4114 & 67.2458 & 104.2120 & 100.6190 \\
47.8966 & 67.9433 & 14.7869 & 76.3406 & 70.9614 & 69.5489 & 68.2461 & 80.3224 & 52.7804 & 74.5351 & 80.3976 \\
57.7849 & 49.1607 & 51.7501 & 26.2510 & 11.3107 & 32.4638 & 31.1182 & 48.2635 & 7.3564 & 41.8103 & 29.7889 \\
76.9987 & 97.0132 & 29.4629 & 101.4065 & 92.0827 & 96.7314 & 78.2804 & 108.6867 & 73.4754 & 80.9460 & 93.6696 \\
87.0215 & 77.1456 & 68.8263 & 46.8968 & 27.5488 & 58.3573 & 18.0068 & 72.9921 & 33.1871 & 23.7224 & 1.2631 \\
59.0808 & 37.5188 & 76.2551 & 4.4587 & 21.6885 & 17.8927 & 57.3080 & 27.4667 & 35.0345 & 67.3609 & 48.6140 \\
42.9721 & 69.7695 & 44.1116 & 90.0214 & 90.5024 & 79.1930 & 95.3299 & 85.0845 & 74.3281 & 102.6011 & 105.5259
\end{matrix}$$

$$Probability_{Target\text{-}Disperse} =$$
$$\begin{matrix}
0.0397 & 0.0316 & 0.0307 & 0.0360 & 0.1003 & 0.0295 & 0.4967 & 0.0915 & 0.0781 & 0.0290 & 0.0369 \\
0.0361 & 0.6205 & 0.0451 & 0.0287 & 0.0370 & 0.0416 & 0.0499 & 0.0387 & 0.0450 & 0.0275 & 0.0298 \\
0.0798 & 0.0444 & 0.3827 & 0.0441 & 0.0384 & 0.2305 & 0.0438 & 0.0338 & 0.0330 & 0.0213 & 0.0482 \\
0.0793 & 0.0479 & 0.0553 & 0.1109 & 0.1510 & 0.0552 & 0.0987 & 0.1252 & 0.0923 & 0.0813 & 0.1027 \\
0.0439 & 0.0432 & 0.0369 & 0.0474 & 0.0880 & 0.0361 & 0.0823 & 0.1213 & 0.1197 & 0.3346 & 0.0466 \\
0.0523 & 0.0784 & 0.0489 & 0.0482 & 0.0912 & 0.0467 & 0.1426 & 0.1252 & 0.2393 & 0.0782 & 0.0489 \\
0.1403 & 0.0397 & 0.0618 & 0.1537 & 0.1159 & 0.0618 & 0.0902 & 0.0746 & 0.0592 & 0.0362 & 0.1666 \\
0.0653 & 0.1827 & 0.0699 & 0.0572 & 0.0839 & 0.0663 & 0.1133 & 0.0970 & 0.1269 & 0.0790 & 0.0585 \\
0.3359 & 0.0357 & 0.0958 & 0.1002 & 0.0519 & 0.1048 & 0.0502 & 0.0418 & 0.0375 & 0.0254 & 0.1208 \\
0.0964 & 0.0397 & 0.0559 & 0.2127 & 0.1058 & 0.0568 & 0.0751 & 0.0778 & 0.0613 & 0.0469 & 0.1715 \\
0.0525 & 0.0683 & 0.0480 & 0.0521 & 0.0900 & 0.0465 & 0.1038 & 0.1225 & 0.1689 & 0.1952 & 0.0521
\end{matrix}$$

$$Probability_{Target} =$$
$$\begin{matrix}
0.0397 & 0.0713 & 0.102 & 0.138 & 0.2384 & 0.2678 & 0.7645 & 0.856 & 0.9341 & 0.9631 & 1.0000 \\
0.0361 & 0.6566 & 0.7017 & 0.7304 & 0.7674 & 0.809 & 0.859 & 0.8977 & 0.9427 & 0.9702 & 1.0000 \\
0.0798 & 0.1242 & 0.507 & 0.551 & 0.5894 & 0.8199 & 0.8636 & 0.8975 & 0.9305 & 0.9518 & 1.0000 \\
0.0793 & 0.1273 & 0.1826 & 0.2935 & 0.4445 & 0.4997 & 0.5984 & 0.7237 & 0.816 & 0.8973 & 1.0000 \\
0.0439 & 0.0871 & 0.124 & 0.1714 & 0.2594 & 0.2955 & 0.3778 & 0.4992 & 0.6188 & 0.9534 & 1.0000 \\
0.0523 & 0.1307 & 0.1796 & 0.2278 & 0.319 & 0.3657 & 0.5083 & 0.6335 & 0.8729 & 0.9511 & 1.0000 \\
0.1403 & 0.18 & 0.2419 & 0.3956 & 0.5114 & 0.5732 & 0.6634 & 0.738 & 0.7973 & 0.8334 & 1.0000 \\
0.0653 & 0.248 & 0.3179 & 0.3751 & 0.459 & 0.5253 & 0.6386 & 0.7356 & 0.8624 & 0.9415 & 1.0000 \\
0.3359 & 0.3716 & 0.4675 & 0.5677 & 0.6195 & 0.7244 & 0.7746 & 0.8163 & 0.8538 & 0.8792 & 1.0000 \\
0.0964 & 0.1361 & 0.192 & 0.4047 & 0.5106 & 0.5674 & 0.6424 & 0.7203 & 0.7816 & 0.8285 & 1.0000 \\
0.0525 & 0.1208 & 0.1688 & 0.2209 & 0.3109 & 0.3575 & 0.4613 & 0.5838 & 0.7527 & 0.9479 & 1.0000
\end{matrix}$$

$$Min\text{-}Dis_{Matrix} =$$
$$\begin{matrix}
7 & 5 & 8 & 9 & 1 & 11 & 4 & 2 & 3 & 6 & 10 \\
2 & 7 & 3 & 9 & 6 & 8 & 5 & 1 & 11 & 4 & 10 \\
3 & 6 & 1 & 11 & 2 & 4 & 7 & 5 & 8 & 9 & 10 \\
5 & 8 & 4 & 11 & 7 & 9 & 10 & 1 & 3 & 6 & 2 \\
10 & 8 & 9 & 5 & 7 & 4 & 11 & 1 & 2 & 3 & 6 \\
9 & 7 & 8 & 5 & 2 & 10 & 1 & 11 & 3 & 4 & 6 \\
11 & 4 & 1 & 5 & 7 & 8 & 3 & 6 & 9 & 2 & 10 \\
2 & 9 & 7 & 8 & 5 & 10 & 3 & 6 & 1 & 11 & 4 \\
1 & 11 & 6 & 4 & 3 & 5 & 7 & 8 & 9 & 2 & 10 \\
4 & 11 & 5 & 1 & 8 & 7 & 9 & 6 & 3 & 10 & 2 \\
10 & 9 & 8 & 7 & 5 & 2 & 1 & 4 & 11 & 3 & 6
\end{matrix}$$

3.2.2. Adaptive Step Size

$$\begin{cases} stepa = stepa_{basic} * (1/ceil(3 * t/T)),\ stepa_{basic} = ceil(N_{Node}/2) \\ stepb = stepb_{basic} * (1/ceil(3 * t/T)),\ stepb_{basic} = ceil(N_{Node}/3) \\ stepc = stepc_{basic} * ceil(3 * t/T),\ stepc_{basic} = ceil(N_{Node}/3), \end{cases} \tag{12}$$

where $stepa_{basic}$ is the original step size of migration, $stepb_{basic}$ is the step size of summon, and $stepc_{basic}$ is the step size of siege; t is the number of the current iteration, and T means the maximum number of iterations; and *ceil*() is a function that can round a number up.

During the early stage of optimization, for migration, it is hoped that a search for large areas can be conducted to find potential global optima, while the search for small areas should be conducted to accelerate the speed of convergence in the later stage.

For summon, it is in the former stage that a larger step size is needed to enable other wolves to move quickly towards the location of the best wolf, so that the rate of convergence can be enhanced, while a smaller step size can stop the algorithm from falling into a local optimum in the later stage.

Finally, for siege, a smaller step size makes the algorithm converge quickly in the early stage, while a larger one can prevent the algorithm from falling into the local optimum in the later stage.

4. Main Steps

In order to clearly show the details of the new proposed algorithm, specific implementation steps are given as follows.

Step 1: Initialization

Initially, all the movable sensor nodes will be randomly and unorderly scattered in the monitored area, just as wolves are distributed randomly and unorderly in the solution space. Thus, the locations of sensor nodes can be represented by the wolf population generated by Equation (13).

$$\mathrm{w}i = (\mathrm{w}i1, \mathrm{w}i2, \ldots, \mathrm{w}id, \ldots, \mathrm{w}i\mathrm{D})(i = 1, 2, \ldots, \mathrm{N};\ \mathrm{d} = 1, 2, \ldots, \mathrm{D}), \tag{13}$$

where wid is the location of the *i*-th wolf in the *d*-th dimension. N means the size of the wolf population. D is the maximum dimension number of the solution space. The initial location of each wolf can be produced by Equation (14).

$$\mathrm{w}id = \mathrm{rangemin} + \mathrm{rand}(0,1)*(\mathrm{rangemax} - \mathrm{rangemin}), \tag{14}$$

where rand (0, 1) is a random number distributed uniformly in the interval [0, 1]; rangemax and rangemin are the upper and lower limits of the solution space, respectively.

Step 2: Migration

In this process, each wolf takes its own location as the center to search and moves along D directions, meaning that new locations will be generated, and the best one should be the destination that the current wolf is devoted to. The formula is as follows:

$$wolf_{new} = [wolf(1), wolf(2), \ldots\ wolf(RAND), \ldots,\ wolf(N\_Node)]; \tag{15}$$

$$\begin{cases} wolf(RAND) = & i, Probability_{Matrix}[RAND][i-1] < rand(1,1) < \\ & Probability_{Matrix}[RAND][i], (i = 2, 3, \ldots, N\_Node) \\ wolf(RAND) = & 1,\ rand(1,1) < Probability_{Matrix}[RAND][1]; \end{cases} \tag{16}$$

where *RAND* is a randomly generated positive integer between 1 and *N_Node*, and *rand*(1,1) generates a random number between 0 and 1.

If the new wolf has better fitness than the current wolf, the latter should move to the new location and update the corresponding fitness. Otherwise, it continues to loop until the number is out of *stepa*, which is the step size of migration and is obtained by Equation (12).

Step 3: Summon–Raid

$$\begin{cases} wolf_{new} = & [wolf(1), wolf(2), \ldots wolf(index(i)) \ldots, wolf(RAND-1), bestwolf(RAND), bestwolf(RAND+1), \\ & \ldots bestwolf(RAND+stepb-1), wolf(RAND+stepb), \ldots wolf(N_{Node})) \\ wolf(index) = & bestwolf(RAND:\ (RAND\ +\ stepb\ -1)),\ index(i) \notin [RAND,\ (RAND+\ stepb\ -1)] \\ & (RAND <=\ N_{Node}-stepb\ +\ 1), (i = 1, 2, \ldots, stepb) \end{cases} \tag{17}$$

The current wolf moves towards the lead wolf with the best fitness with the step size "*stepb*" according to Equation (17); if the fitness of the new location is better than the current one, the wolf should move to the new location and update the corresponding fitness. Otherwise, it stays at the old position; here, *stepb* means the step size of summon–raid, which can be obtained by Equation (12).

Step 4: Siege

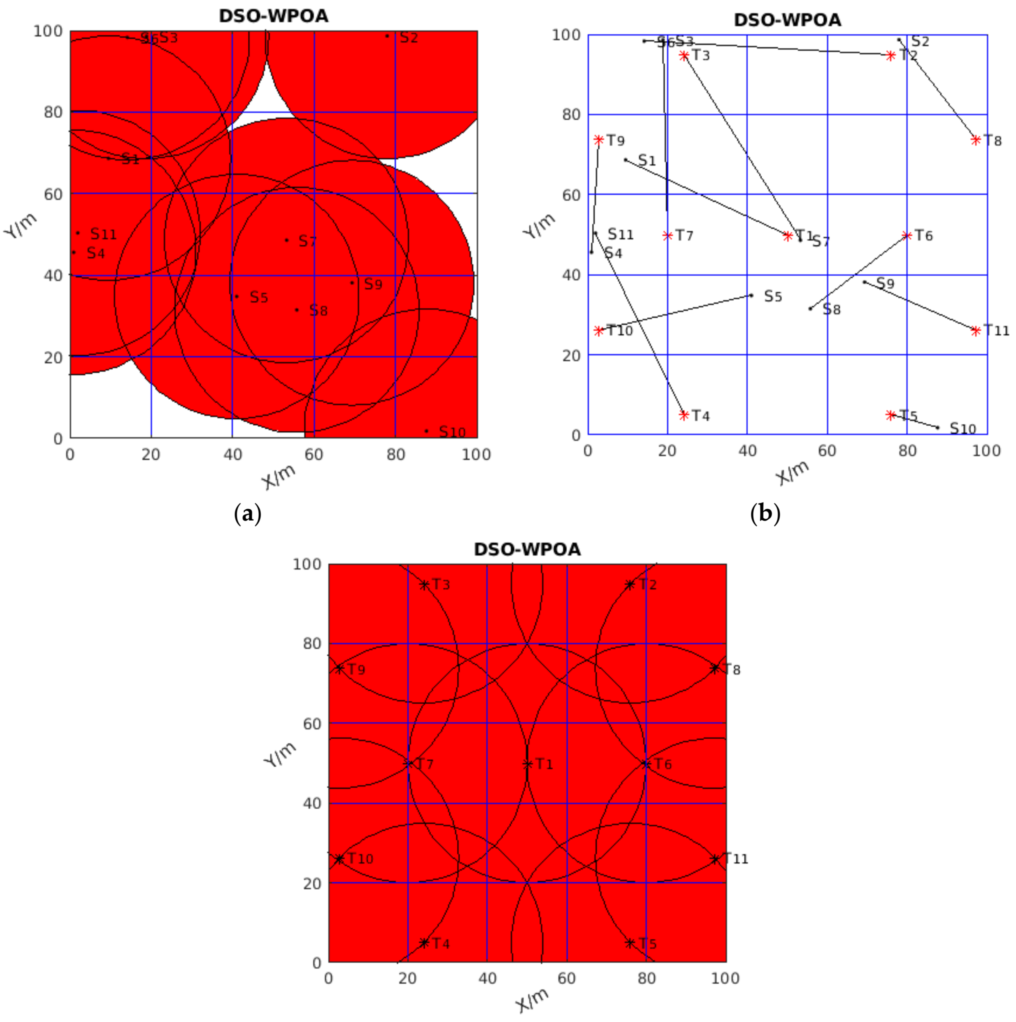
$$\begin{cases} wolf_{new} = [wolf(1),\ wolf(2), \ldots, wolf(i),\ \ldots, wolf(N_{Node})] \qquad (i = 1, 2, \ldots, N_{Node}) \\ wolf(i) = Min-Dis_{Matrix[i][j]} \qquad (j = 1, 2, \ldots, stepc) \end{cases} \tag{18}$$

During this process, each wolf performs small-scale, fine-grained searches for potential global optima; that is, for each dimension of each wolf, several searches will be conducted according to the step size *stepc* (which can be obtained by Equation (12)) and a shorter distance matrix, whose name is *Min-Dis*$_{Matrix}$, as stated above. The new location to which the current wolf tries to move to can be obtained by Equation (18), and if the fitness of the new location is better than the current fitness, the current wolf should move to the new position; otherwise, it stays at the old one. At the end of this process, the new wolf has a new fitness that is no less than its original fitness.

Step 5: Updating Population

After siege, some wolves with poorer fitness will be eliminated, while the same number of wolves will be regenerated, and this amount can be obtained by Equation (2).

Across the whole process of this newly proposed algorithm, the purpose is to find the shortest path under a high coverage rate to reduce energy consumption and improve the sustainability of the whole network as much as possible. The process is shown briefly in Figure 5.



(**a**) (**b**)

(**c**)

Figure 5. (**a**) The initial diagram of wireless sensor network coverage (distributed randomly); (**b**) the solution of how the movable sensors move to the targets one by one; and (**c**) the coverage that has been optimized by DSO-WPOA The red stars mean the locations of targets, while the black points mean the randomly generated initial locations of sensors, and the black lines mean the moving paths. Specifically, the parameters in the diagram correspond to the example in Section 3.

5. Experiments

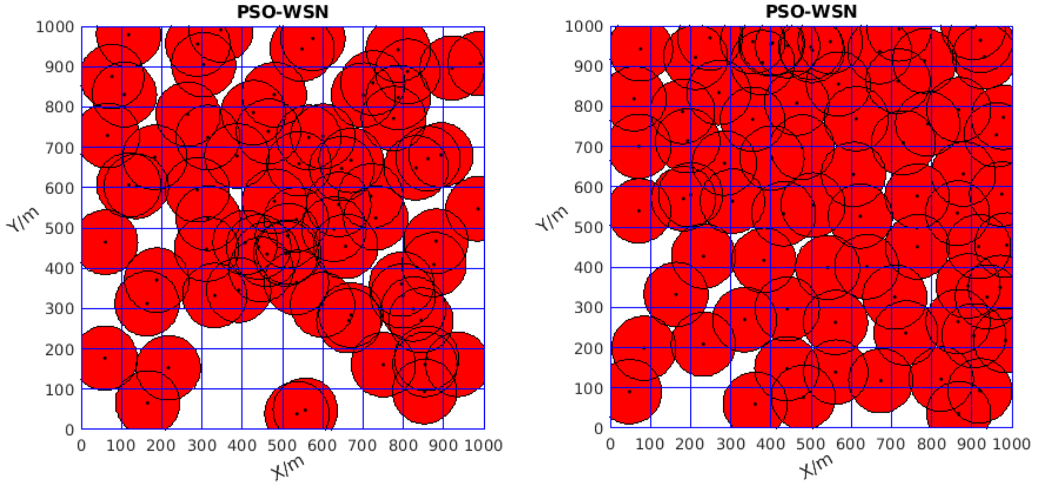
To evaluate the performance of the proposed algorithm, several comparative experiments were conducted with the Ubuntu 16.04.4 operating system, Intel(R) Core(TM) i7-5930K processor, 64G memory and MATLAB-2017b. We also used algorithms such as the random algorithm, virtual force algorithm (VFA), and coverage algorithm based on particle swarm optimization (PSO-WSN). In this paper, the range of the solution space was [0,1000]; for all related algorithms, the number of sensors was 77, the sensing radius of a sensor was 80, and the communication radius of sensors was more than 80*$\sqrt{3}$. The PSO-WSN was conducted based on the idea behind the paper [18], while the VFA was based on the basic concept of VFA [10,11,26,27], and the steps to implement DSO-WPOA followed those described in Section 4. Furthermore, the main parameters of DSO-WPOA were the following: the size of the wolf population was 50, the initial step size of migration was stepa = 39, the initial step size of summon–raid was stepb = 26, and the step size of siege was stepc = 26.

Firstly, focusing on Table 1, seen from the perspective of the best coverage rate, the value of the new algorithm could reach 100%, which means all the monitoring area was covered. Although the two other algorithms had good coverage rates that reached 93.83% and 99.645%, respectively, some white area existed in the monitoring area, as shown in Figures 6b and 7b, which means that some area was uncovered, while no white gap existed in Figure 8b. Thus, DSO-WPOA possessed the best coverage rate.

Table 1. Results of experiments. PSO-WNS: particle swarm optimization-wireless sensor network; VFA: virtual force algorithm.

Algorithm	Initial Coverage Rate	Best Coverage Rate	Mean Coverage Rate	Worst Coverage Rate	Variance	Mean Time	Moving Distance
PSO-WSN	77.71%	93.83%	91.228%	89.74%	12593.36	2124.0355	36575.491
VFA	76.10%	99.645%	99.476%	98.149%	1.7324e-06	23.2982	11094.4339
DSO-WPOA	76.53%	100%	100%	100%	0	57.8892	7662.2987

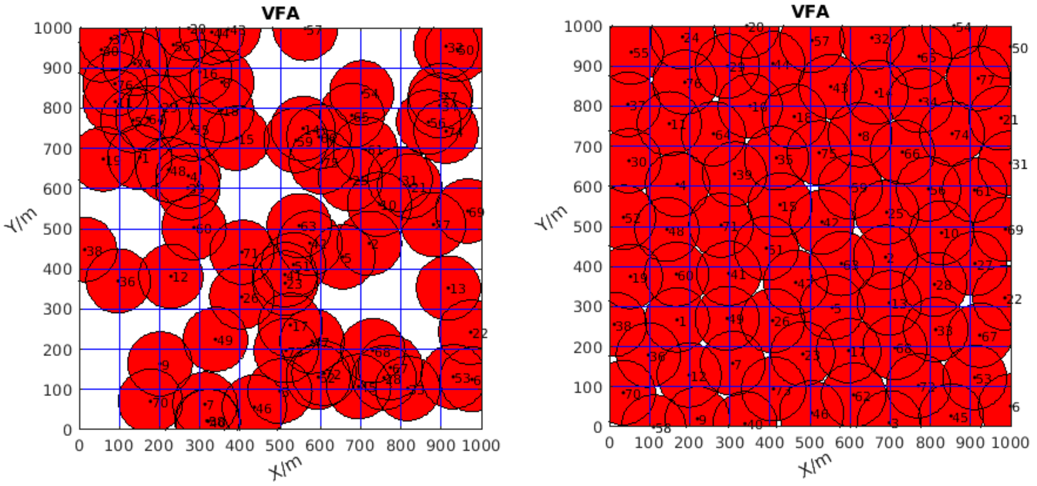
PSO-WSN



(**a**) (**b**)

Figure 6. (**a**) The initial diagram of wireless sensor network coverage (distributed randomly); (**b**) the coverage optimized by the PSO.

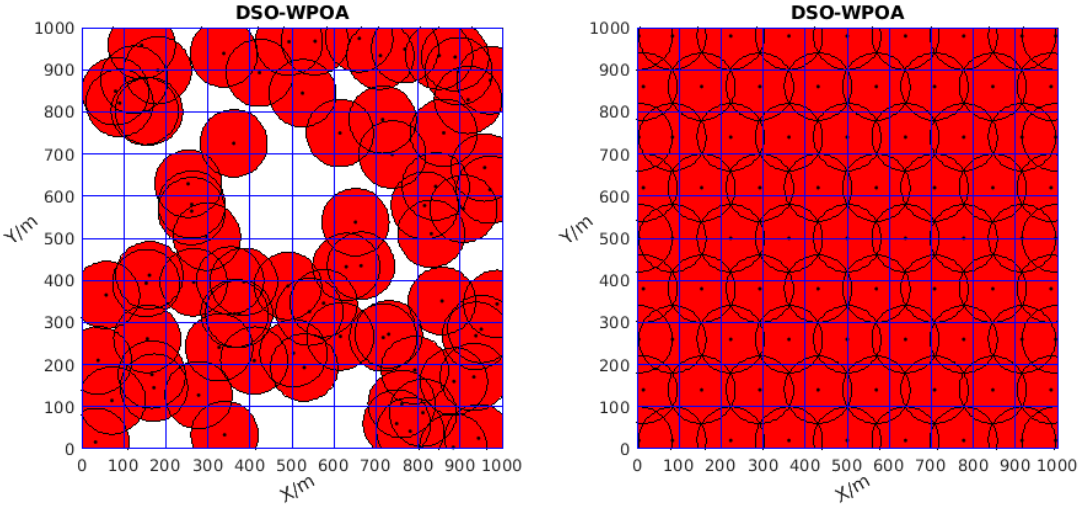
VPA



(**a**) (**b**)

Figure 7. (**a**) The initial diagram of wireless sensor network coverage (distributed randomly); (**b**) the coverage optimized by VPA.

DSO-WPOA



(**a**) (**b**)

Figure 8. (**a**) The initial diagram of wireless sensor network coverage (distributed randomly); (**b**) the coverage optimized by DSO-WPOA.

Moreover, after repeating 100 times, both the worst and mean coverage rates of DSO-WPOA reached 100%, and its variance was zero, while the terms of the two other algorithms were less than the former. Therefore, although the variance of VFA was very small, DSO-WPOA had better stability in general.

Furthermore, focusing on the mean time, the PSO-WSN required the most time, and the time the newly proposed algorithm required was on the same order of magnitude as that of VFA. Thus, DSO-WPOA possessed an acceptable time performance.

Finally, the mean moving distance according to the rules of DSO-WPOA was smallest; the value was 7,662,2987, while the values of the other two algorithms were 3,6575,491 and 1,1094,4339, respectively, which means the new algorithm enabled the WSN to spend less energy. Consequently, DSO-WPOA possessed advantages in saving energy.

In summary, DSO-WPOA had the best coverage rate, better stability, acceptable performance in terms of time, and advantages in saving energy.

6. Discussion and Conclusions

To further improve the coverage rate of a movable wireless sensor network under the condition of guaranteed connectivity, we proposed an adaptive, discrete space oriented wolf pack optimization algorithm for a movable wireless sensor network. Firstly, a strategy of adaptive expansion based on the minimum overlapping full-coverage Model was designed to achieve minimum overlap and no-gap coverage of the monitoring area. Moreover, the ASGS-CWOA was improved to optimize the movable wireless sensor network, which is a discrete space oriented problem, and the improvement included the usage of a target–node probability matrix and the design of an adaptive step size method, both of which worked together to improve the convergence speed and global optimization ability of the algorithm.

Theoretical research and experimental results indicated that, compared to the PSO-WSN algorithm and classical virtual force algorithm, the newly proposed algorithm, named DSO-WPOA, possessed the best coverage rate, better stability, acceptable performance in terms of time, advantages in energy saving, and no gaps. As detailed in Table 1, Figure 4b, and Figure 8b, the newly proposed algorithm enabled the WSN to cover the whole monitored area (no gaps) with the smallest number of sensors as distant as possible from each other. However, it performed weaker in some aspects; for example, DSO-WPOA spent more time on iterations than VFA, as shown in Table 1. The coverage rate of

DSO-WPOA may not be superior to other coverage algorithms when not enough sensor nodes can be supplied, such as in this paper where the number of sensors was 77, which just satisfied AE-MOFCM. Assuming that the number of sensors is less than 77, the advantages of DSO-WPOA may be gone.

Our future work will continue to perfect the performance of DSO-WPOA in all aspects and apply it to specific projects as well to expand its scope of application.

Author Contributions: "D.W. and H.W. conceived and designed the experiments; D.W. performed the experiments; D.W. and X.B. analyzed the data; X.Q. contributed reagents/materials/analysis tools; D.W. and J.N. wrote the paper." Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This study was funded by the National Key Research and Development Program of China (Grant No. 2016YFB1001404) and the National Natural Science Foundation of China (Grant No. 61873299).

Acknowledgments: The authors are grateful to peer experts for their full support of this paper, and they thank China University of Mining & Technology, Beijing, which provided the necessary scientific research environment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, D.P.; Amgoth, T.; Annavarapu, C.S.R. Machine learning algorithms for wireless sensor networks: A survey. *Inf. Fusion* **2019**, *49*, 1–25. [CrossRef]
2. Dai, H.-N.; Wong, R.C.W.; Wang, H. On Capacity and Delay of Multichannel Wireless Networks With Infrastructure Support. *IEEE Trans. Veh. Technol.* **2017**, *66*, 1589–1604. [CrossRef]
3. Dai, H.; Wong Raymond, C.; Wang, H. Big Data Analytics for Large ScaleWireless Networks: Challenges and Opportunities. *ACM Comput. Surv.* **2019**. [CrossRef]
4. Thakur, D.; Kumar, Y.; Kumar, A.; Singh, P.K. Applicability of Wireless Sensor Networks in Precision Agriculture: A Review. *Wirel. Pers. Commun.* **2019**, *107*, 471–512. [CrossRef]
5. Li, G.; Zhao, J.; Wang, Z. Forest Fire Detection System based on Wireless Sensor Network. *Chin. J. Sens. Actuators* **2006**, *19*, 2760–2764.
6. Xu, L.; Nie, J.; Ma, X.; Lv, H.; Yang, J. Design and Implementation of Water Environment Monitoring System Based on Wireless Sensor Network. *Digit. Technol. Appl.* **2018**, *36*, 128–129.
7. Fu, X.; Yao, H.; Postolache, O.; Yang, Y. Message forwarding for WSN-Assisted Opportunistic Network in disaster scenarios. *J. Netw. Comput. Appl.* **2019**, *137*, 11–24. [CrossRef]
8. Yick, J.; Bharathidasan, A.; Pasternack, G.; Mukherjee, B.; Ghosal, D. Optimizing Placement of Beacons and Data Loggers in a Sensor Network—A Case Study. In Proceedings of the IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733), Atlanta, GA, USA, 21–25 March 2004; pp. 2486–2491.
9. Qu, Y.; Zhai, Y.; Lin, Z.; Zhao, B.; Zhang, Y. A Novel Sensor Placement Model in Wireless Sensor Network. *J. Beijing Univ. Posts Telecommun. Telecommun.* **2004**, *27*. [CrossRef]
10. Zhou, F.; Gao, J.; Fan, X.; An, K. Dynamic Covering Algorithm of Node Based on Virtual Force in Wireless Sensor Networks. *J. Syst. Simul.* **2018**, *30*, 97–106.
11. Zhang, C. Coverage Algorithm based on Virtual Forces in Wireless Sensor Networks. *Appl. Res. Comput.* **2019**, *36*, 1854–1857.
12. Yarinezhad, R.; Hashemi, S.N. A routing algorithm for wireless sensor networks based on clustering and an fpt-approximation algorithm. *J. Syst. Softw.* **2019**, *155*, 145–161. [CrossRef]
13. Chen, Y.; Xu, X.; Wang, Y. Wireless sensor network energy efficient coverage method based on intelligent optimization algorithm. *Discret. Contin. Dyn. Syst. S* **2019**, *12*, 887–900. [CrossRef]
14. Somaieh, Z.; Shahram, B. Dehcic: A Distributed Energy-aware Hexagon based Clustering Algorithm to Improve Coverage in Wireless Sensor Networks. *Peer Peer Netw. Appl.* **2019**, *12*, 689–704.
15. Alavi, S.A.; Mehran, K.; Hao, Y.; Rahimian, A.; Mirsaeedi, H.; Vahidinasab, V. A Distributed Event-Triggered Control Strategy for DC Microgrids Based on Publish-Subscribe Model over Industrial Wireless Sensor Networks. *IEEE Trans. Smart Grid* **2019**, *10*, 4323–4337. [CrossRef]
16. Sajwan, M.; Gosain, D.; Sharma, A.K. CAMP: Cluster aided multi-path routing protocol for wireless sensor networks. *Wirel. Netw.* **2018**, *25*, 2603–2620. [CrossRef]
17. So-In, C.; Nguyen, T.; Nguyen, N. An Efficient Coverage Hole-Healing Algorithm for Area-Coverage Improvements in Mobile Sensor Networks. *Peer-Peer Netw. Appl.* **2019**, *12*, 541–552. [CrossRef]

18. Wang, X.; Wang, S.; Ma, J. Parallel Particle Swarm Optimization based Mobile Sensor Node Deployment in Wireless Sensor Networks. *Chin. J. Comput.* **2007**, *30*. [CrossRef]
19. Yang, C.; Tu, X.; Chen, J. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. In Proceedings of the International Conference on Intelligent Pervasive Computing (IPC 2007), Institute of Electrical and Electronics Engineers (IEEE), Jeju City, Korea, 11–13 October 2007; Volume 2007, pp. 462–467.
20. Wang, D.; Qian, X.; Liu, K.; Ban, X.; Guan, X. An Adaptive Distributed Size Wolf Pack Optimization Algorithm Using Strategy of Jumping for Raid (September 2018). *IEEE Access* **2018**, *6*, 65260–65274. [CrossRef]
21. Huang, H.; Ren, Z.; Wei, J. Improved Wolf Group Algorithm for Solving Traveling Salesman Problem. *Appl. Res. Comput.* **2019**, *36*. [CrossRef]
22. Wu, H.; Zhang, F.; Li, H.; Liang, X. Discrete Wolf Pack Algorithm for Traveling Salesman Problem. *Control Decis.* **2015**, *30*, 1861–1867.
23. Martin, B.; Marot, J.; Bourennane, S. Improved Discrete Grey Wolf Optimizer. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 494–498.
24. Wu, R.; Wang, S. Discrete Wolf Pack Search Algorithm based Transit Network Design. In Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 26–28 August 2016.
25. Wang, D.; Qian, X.; Liu, K.; Ban, X. An Adaptive Shrinking Grid Search Chaos Wolf Optimization Algorithm with Adaptive Standard-Deviation Updating Amount. *IEEE Access* **2018**. [CrossRef]
26. Liu, Z.; Liu, Q.; Yuan, Y.; Guan, X. Location Scheme in Wireless Sensor Networks based on Bayesian Estimation, Virtual Force and Genetic Algorithm. *Control Decis.* **2013**, *28*, 899–903.
27. Li, M.; Shi, W. Virtual Force-Directed Differential Evolution Algorithm based Coverage Enhancing Algorithm for Heterogeneous Mobile Sensor Networks. *Chin. J. Sci. Instrum.* **2011**, *32*, 1043–1050. [CrossRef]