Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7815257, 8 pages
https://doi.org/10.1155/2018/7815257

Research Article

An Adaptive Fish Swarm-Based Mobile Coverage in WSNs

Ningning Qin [1,2] and Jian Xu [2]

[1] *Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China*
[2] *School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China*

Correspondence should be addressed to Ningning Qin; ningning801108@163.com

Received 4 August 2018; Revised 8 October 2018; Accepted 17 October 2018; Published 1 November 2018

Academic Editor: Mohammad Shojafar

Swarm intelligent algorithms are embedded into sensor networks to achieve perfect coverage with minimal cost. However, these methods are often highly complex and easily fall into the local optimum when balancing coverage and resource consumption. We introduce adaptive improved fish swarm optimization (AIFS) that extricates each node from the local optimum and reduces overlap and overflow coverage. Drawing on the habits of fish, AFIS ensures node mobility with respect to the food concentration at a certain point. Node dispersion shows good compromise under coordination by two presented parameters, namely, food concentration and crowd density. In addition to inheriting properties from traditional fish swarm, the initial random nodes become dispersed without overflow in assisting the proposed jumping and dodging behavior. The resulting network avoids potential local optima and improves the network boundary coverage efficiency. The convergence speed and efficiency of AIFS are verified. Extensive simulation experiments reveal that an improved coverage gain is obtained, and computation cost and overflow waste are reduced.

1. Introduction

In wireless sensor networks (WSNs), numerous sensor nodes are randomly scattered in a monitoring region. Low coverage efficiency is likely to arise. Several overlapping portions are caused by redundant nodes with high density. Meanwhile, nodes are eager to cover blank areas. We can start from two ways, such as adding and moving nodes to improve coverage performance. The former method moves nodes to these weak coverage areas. The latter is able to improve coverage quality by adding more nodes. These additional nodes are directly deployed into the weak area (i.e., decisive coverage compensation) or dispersed with the given random distribution probability (i.e., random coverage compensation), which is referred to as the complex system [1], especially the small-world theory [2]. Considering investment in network equipment, our study focuses on selecting and moving redundant nodes to blank spaces that are not covered by nodes. This scenario is the key reason network coverage has been an enduring topic for researchers.

Instead of adding nodes, WSNs can move several nodes to maximize coverage, and this method is effective. Such mobile coverage uses inordinate amounts of energy, and many scholars [3–5] have presented important constructive achievements in this area. Andrea [6] examined node mobility and posited that dynamic mobile coverage is valuable in improving efficiency in event-triggered WSNs, especially in extending node lifetime and reducing the number of nodes that sense the same area. Similar to [6], [7] regarded coverage quality and lifetime as two key parameters for mobile WSNs. Reasonable coverage density was discussed with the presented threshold expressions as nodes moved. The results showed that coverage quality and lifetime improved by more than 10%.

Moving nodes expand the coverage area. Thus, the next issue to address is the optimal coverage that provides moving positions for nodes. In [8], horizontal and vertical sampling lines were set in a monitoring area, and a coverage optimization algorithm based on sampling for homogeneous WSNs (COSH) was established based on the relationship between the node-sensed circular boundary and sampling line. The algorithm simulates the optimal coverage of the entire network with the optimal coverage effect of multiple sampling lines and effectively improves coverage performance. However, the frequency of sampling lines is manually configured without the introduction of any intelligent method.

Consequently, the universality of the algorithm is unsatisfactory. Another study [9] designed Virtual Force Diminishing Particle Swarm Optimization (VFDPSO) to address the limitations of nonintelligent configuration through network-wide optimization. VFDPSO achieves automatic coverage optimization, but its stability is imperfect because Particle Swarm Optimization (PSO) easily falls into local optimization [10]. Huang [11] introduced the Artificial Fish Swarm Algorithm (AFSA) into mobile coverage. AFSA can jump out of local optimization. The objective function is promoted to jump out of local optimization by following the behavior of some fish [12], thus preventing rear-end collision. Although AFSA solves local optimization, its time consumption is high. Therefore, AFSA was combined with virtual force in [13] to increase its working efficiency. The speed in jumping out of local optimization is improved by introducing this type of fusion with virtual force. However, the algorithm still has drawbacks, such as late response to serious local optimization and inconspicuous improvement in coverage performance.

The energy consumption in the mobile process has also been fully considered in many studies [14, 15]. By adopting an intelligent algorithm, [14] was able to schedule partial nodes to participate in coverage to save energy. The algorithm presented in [15] disposed a small number of beacon nodes to improve coverage accuracy and reduce the overall energy consumption.

Our study fully utilizes mobile features to optimize coverage performance. An Adaptive Improved Fish Swarm (AIFS) algorithm is developed to explore the practical application of intelligent algorithms in jumping out of local optimization in random sensor networks. AIFS designs a type of jumping behavior on the basis of the K-level mean [16], which not only allows rapid jumping out of local optimization but also reduces the probability of falling into local optimization in advance. The calculation and implementation of behavior are executed separately, such that the generated best moving means reduce energy wastage caused by invalid movement. Additional wall-dodging behavior is used to optimize the boundary problem and inhibit overlap and overflow coverage. Convergence speed is accelerated by recording the speed variation trend of convergence and adaptively adjusting the field of view and step length of movement.

The remainder of the paper is organized as follows. In Section 2, we review AFSA, essential models, and notions. We investigate the details of AIFS in Section 3. The results of simulations are introduced in Section 4. In Section 5, we present the conclusion and several directions for future work.

2. Mathematical Model

2.1. Network Model. N homogeneous sensor nodes, $S = \{S_1(x_1, y_1), \ldots, S_k(x_k, y_k), \ldots, S_N(x_N, y_N)\}$, are randomly scattered within a given monitoring area $I = l \times l$, and all nodes adopt a Boolean sensing model with radius r, where $k = 1, 2, 3, \ldots, N$. For illustration, area I is discretized into $l \times l$ pixels [17]$M = \{M_{(1,1)}, M_{(1,2)}, \ldots, M_{(x,y)}, \ldots, M_{(l,l)}\}$, where $x, y = 1, 2, \ldots, l$, and randomly deployed nodes are assumed to be located on a certain pixel.

2.2. Basic Definition. Network coverage is an important index for measuring the coverage performance of sensor networks. The expressions of several concepts related to network coverage are as follows.

Definition 1 (pixel-sensing ratio). When the distance between pixel $M_{(x,y)}$ and sensor node S_k is less than r, the sensing ratio of S_k at $M_{(x,y)}$ is considered to be 1; otherwise, it is 0. That is,

$$P\left(M_{(x,y)}, S_k\right) = \begin{cases} 1, & \left(x - x_k\right)^2 + \left(y - y_k\right)^2 \leq r^2 \\ 0, & \text{others.} \end{cases} \quad (1)$$

Definition 2 (pixel coverage). Pixel coverage shows the coverage performance of network S at a certain pixel $M_{(x,y)}$. When the distance between pixel $M_{(x,y)}$ and any $S_k \in S$ is less than r, $M_{(x,y)}$can be covered by S, i.e.,

$$P\left(M_{(x,y)}, S\right) = \begin{cases} 0, & \sum_{k=1}^{N} P\left(M_{(x,y)}, S_k\right) = 0 \\ 1, & \text{others.} \end{cases} \quad (2)$$

Definition 3 (network coverage). The ratio sum of the coverage of all pixels to the total number of pixels is defined as network coverage Y, i.e.,

$$Y = \frac{\sum_{x=1}^{l} \sum_{y=1}^{l} P\left(M_{(x,y)}, S\right)}{l^2} \times 100\%. \quad (3)$$

Equations (2) and (3) show that the greater the number of pixels that satisfy $P(M_{(x,y)}, S) = 1$ is, the more effectively the network coverage Y of area I can be improved. Therefore, the goal of coverage optimization is to move nodes with overlapped coverage to the blank place to increase the number of pixels that meet $P(M_{(x,y)}, S) = 1$ as much as possible.

2.3. Basic Fish Swarm Algorithm: AFSA. AFSA is an intelligent algorithm that can jump out of local optimization. This algorithm exhibits good robustness and global convergence and does not have high requirements for initial values. AFSA is applied to the coverage of sensor networks, where a sensor node S_k is thought of as a fish S_k. The food concentration at each pixel is positively correlated with its attraction to a sensor node (or a fish).

In accordance with the behavior habits of a fish swarm [18], AFSA includes four behavioral patterns, namely, foraging, clustering, repulsing, and random walking, to solve the movement problem for every fish. Before presenting these types of behavior, the basic parameters are described as follows.

Field of view (*visual*): each fish has a circular range of observation called field of view, and its radius is represented as *visual*.

Fish spacing threshold: key distances that stimulate the fish swarm to perform some given behavior are provided. α and β are presented as clustering and repulsing thresholds, respectively.

Reference and moving steps: reference step $step_0$ is used to indicate the unit moving step. Moving step indicates the moving length of a movement performed by a fish. It satisfies $step = step_0 \times rand$, where $rand \in (0, 1)$ is a random value.

Foraging: given a pixel $M_{(x,y)}$within the *visual* region of fish S_k, the food concentration at point $M_{(x,y)}$ may be higher than that at point S_k if $\min(\mathrm{D}(M_{(x,y)}, \overline{S_k} \mid S)) < \min(\mathrm{D}(S_k, \overline{S_k} \mid S)$ exists. Then, fish S_k should move *step* toward $M_{(x,y)}$. If point $M_{(x,y)}$ is not found within the *visual* region of fish S_k, then fish S_k begins random walking. $\mathrm{D}(\bullet)$ represents the Euclidean distance between two points.

Clustering: if the given fish S_k satisfies $\min(\mathrm{D}(S_k, \overline{S_k} \mid S) > \alpha$, then S_k moves *step* toward the current nearest fish.

Repulsing: if the given fish S_k satisfies $\min(\mathrm{D}(S_k, \overline{S_k} \mid S) < \beta$, then S_k moves *step* toward the direction opposite to the current nearest fish.

Random walking: the given fish S_k moves *step* toward a random direction.

The mentioned foraging behavior motivates a node (also called fish) to expand the covered region. With the help of clustering behavior, an isolated node can effectively move close to the center. Repulsing behavior can avoid crowded nodes, and random walking can increase the randomness of node movement. The combination of repulsing behavior and random walking ensures that the fish swarm jumps out of local optimization.

3. AIFS Algorithm

3.1. Algorithm Idea. Considering the natural pursuit for full coverage and efficient movement to save energy, an appropriate method should provide a rapid and effective response that faces the local optimization and boundary problem. Therefore, we propose an AIFS algorithm based on AFSA. The speed and efficiency of fish jumping out of local optimization are improved by additional jumping behavior. Combined with the K-level means idea [16], a new food concentration at each pixel is redefined to reduce the probability of falling into local optimization. Wall-dodging behavior is a valuable addition due to the presented critical strategies for the nearby boundaries of the nodes. Our significant parameters, *step* and *visual*, are adaptively adjusted to ensure convergence stability.

3.2. Key Definitions. The food concentration of a pixel directly determines the mobile attraction of this position to a fish (i.e., sensor node). Basic experiences indicate that covering a large blank area can provide high coverage quality. Thus, food concentration is related not only to the coverage ratio of the pixel's location but also to that of adjacent pixels. Therefore, the relevant parameters of AIFS are updated as follows.

Definition 4 (food concentration $T_{(x,y)}$). Given parameter K, the food concentration $T_{(x,y)}$ of pixel $M_{(x,y)}$ is related to all the pixels' coverage within the K-level neighborhood of $M_{(x,y)}$ and defined as follows:

$$T_{(x,y)} = -\sum_{t=0}^{K} \frac{2^{K-t}}{2^{K+1}} \sum_{f=y-t}^{y+t} \sum_{e=x-t}^{x+t} P\left(M_{(e,f)}, S\right). \quad (4)$$

The K-level means method is applied to calculate $T_{(x,y)}$, in which the pixels' coverage ratios within the K-level neighborhood around $M_{(x,y)}$ are included in the calculation process. The difference among the levels around $M_{(x,y)}$ is characterized by setting the weight coefficient $2^{K-t}/2^{K+1}$ for each layer of the neighborhood.

Definition 5 (crowding density). The sum of the coverage of pixels within r sensing range of S_k is defined as the crowding density of S_k.

$$Density_k = \sum_{f=y_k-\lfloor r}^{y_k+\lfloor r} \sum_{e=x_k-\lfloor r}^{x_k+\lfloor r} P\left(M_{(e,f)}, S_k\right) \quad (5)$$

$M_{(e,f)}$ is a pixel that satisfies $\sqrt{(e-x)^2 + (f-y)^2} \leq \underline{r}\rfloor$ in area I, where $\underline{r}\rfloor$ represents rounded-down numbers of r. $Density_k$ embodies the crowding density around S_k. Thus, it is an important parameter for evaluating whether redundant nodes exist within the sensing range of S_k or not.

3.3. Updated Behavior. In addition to the four basic types of behavior included in AFSA, two new behavior types, namely, jumping and wall dodging, properly address the restrictions originating from local optimization and the boundary problem. An attenuation factor, $0 < \theta < 1$, is introduced to adjust the values of *visual* and *step* to address the lack of convergence, which is helpful in enhancing the adaptability [19] of our AIFS. Several key parameters, such as variable jumping factor p_k, jumping threshold P_{th} $(0 < P_{th} < 1)$, and crowding threshold Q_{th}, are introduced.

Jumping: if node S_k satisfies $Density_k > Q_{th}$ and $p_k > P_{th}$, the jumping behavior is executed, and S_k jumps to the point of the pixel $\arg\max(T_{(x,y)})$, where $\sqrt{(x-x_k)^2 + (y-y_k)^2} \leq \underline{r}\rfloor$. S_k prefers to randomly select one as its jumping destination when more than one pixel fits in with $\arg\max(T_{(x,y)})$, where $\arg\max(T_{(x,y)})$ represents the coordinate of maximum$T_{(x,y)}$.

Wall dodging: if $\mathrm{D}(S_k, \boxed{I}) < r/2$, then node S_k moves $rand \times (r - \mathrm{D}(S_k, \boxed{I}))$ toward the opposite direction of the boundary, where $\boxed{I}$ means the boundaries of I.

Updating *visual* and *step*: after C_{th} rounds of optimization, the values of *visual* and *step* are updated to overcome the weak convergence of Y. If Y does not achieve a sufficient inclement for itself after optimization, *visual* and *step* should be reduced as follows:

$$visual = visual \times \theta, \quad (6)$$

$$step = step \times \theta. \quad (7)$$

The smaller *visual* and *step* are, the more accurate the moving destination is searched.

3.4. AIFS Algorithm Flow. As a virtual fish swarm, WSNs use a distributed schedule in the initial stage of AIFS. In each round of optimization, each node implements five types of behavior (i.e., foraging, clustering, repulsing, jumping, and wall dodging) and records its updated location in a

Step 1: ***Foraging*** and to update **Billboard**.
Step 2: if min $(D(S_k, \overline{S_k} \mid S) > \alpha$, ***Clustering*** *and to update* **Billboard**.
Step 3: if min $(D(S_k, \overline{S_k} \mid S) < \beta$, ***Repulsing*** *and to update* **Billboard**.
Step 4: if $p_k \geq P_{th}$ & $Density > Q_{th}$, ***Jumping*** *and to update* **Billboard**.
Step 5: ***Dodging*** and to update **Billboard**.

Pseudocode 1: The pseudo-code-steps for distributed node (PCS for DN) S_k.

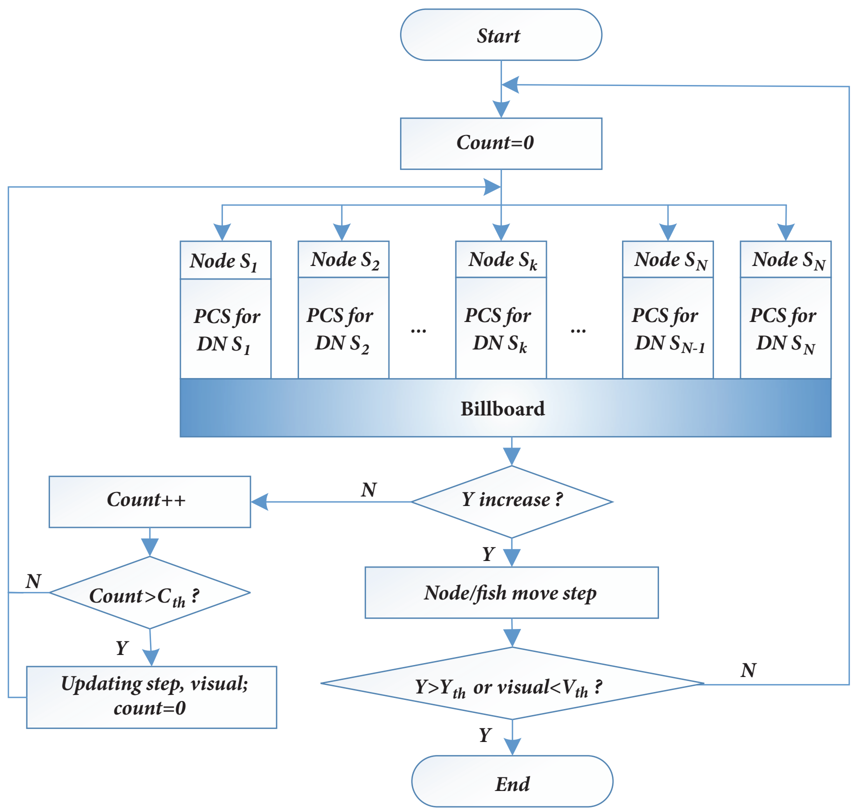

Figure 1: AIFS flowchart.

bulletin board (**Billboard**) after every behavioral instance. The pseudo-code-steps executed by each distributed node are shown in Pseudocode 1.

Each distributed node performs PCS to move itself to an appropriate position without overlap and overflow, thereby exhibiting an efficient use of sensing resources. However, such local optimal coverage sourcing from a single node does not necessarily converge to the global optimum for the entire network. All of the nodes' integrative behavior determines their mobility.

A central scheduling mechanism is needed to achieve optimal network coverage and efficient motion. As a measurement of network coverage, Y is calculated and appraised after each optimization round. Although the current Y is above those before, S_k moves to the location recorded by **Billboard**; otherwise, a new round of optimization is performed after updating the parameters.

In the factor synthesis, AIFS (shown in Figure 1) is involved in the distributed and central scheduling mechanisms at the same time. Each node independently exhibits five types of behavior. All nodes have to work together only for calculating Y. This distributed phase tells each node how to move, and the central phase verifies the movement quality. Worthless movement can be avoided, and behavior availability is evaluated before actual movement. With the repetition of the two phases, nodes can move to the appropriate positions. Here, two terminal thresholds [20], namely, the main threshold Y_{th} and the threshold field of view V_{th}, are set.

During the distributed phase, the complexity of Min or Max in ***Jumping*** is $o(N)$. In the central phase, the main complexity is $o(l^2)$ obtained by calculating Y. Not more than $\log_\theta^{(V_{th}/visual)}$ round is carried out in AIFS. In sum, computational complexity can be counted as $\log_\theta^{(V_{th}/visual)}[o(l^2)+o(N)]$. Ordinarily, we think $N << l^2$ and $\log_\theta^{(V_{th}/visual)} << +\infty$. Thus, $\log_\theta^{(V_{th}/visual)}[o(l^2) + o(N)] \longrightarrow o(l^2)$. Furthermore, $\log_\theta^{(V_{th}/visual)} \in (1, 5)$ in our simulation experiments.

Table 1: Algorithm parameters.

Parameter	PSO	COSH	VFDPSO	AIFS/AFSA
r	8m	8m	8m	8m
N	50	50	50	50
visual	26m	26m	26m	26m
-	$G_{\max} = 200$	$\omega_T = 10$m	$step_0 = 8$m	$C_{th} = 10$
-	$c_1 = c_2 = 3$m	$N_T = 10$	$c_1 = c_2 = c_3 = 3$	$V_{th} = 6$m
-	$\omega_{ini} = 0.9$		$\lvert F_{th}\rvert = 0.4$m	$Q_{th} = 200$
-	$\omega_{end} = 0.4$		$d_{VFDPSO} = 8m$	$P_{th} = 0.5$
-				$Y_{th} = 0.9$

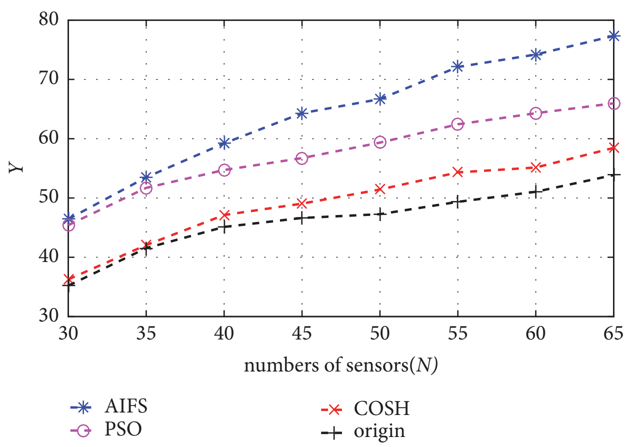

Figure 2: Influence of coverage performance Y with various network sizes N.

4. Simulations

4.1. Scene Description. We use the MATLAB R2016b platform to conduct simulation experiments and set up the experimental scenario as follows: $I = 100\text{m}\times 100\text{m}$, $r = 8$m, $step_0 = 8$m, and $visual = 26$m. The value of α is the distance between two nodes when their sensing boundaries are tangent, i.e., $\alpha = 2r = 16$m. The repulsing threshold β is adjustable according to experimental requirements. In this experiment, $\beta = r \times \sqrt{3}/2 = 6.93$m.

The proposed AIFS is compared with four other algorithms, namely, COSH [8], VFDPSO [7], AFSA [11], and PSO [10], to verify its performance. The other parameters are provided in Table 1. The basic parameters of AIFS are similar to those of AFSA. Given the randomness of the experiments, all resulting data are the mean of 30 independent experiments to derive universal conclusions.

4.2. Network Coverage Analysis. The coverage performances of the three algorithms (AIFS, PSO, and COSH) are compared and analyzed in the various scenarios in Figure 2. Network size is adjusted by various $N \in [30, 65]$, and Y is calculated to evaluate the coverage performance of the algorithms. The algorithm "origin" represents Y in the original environment, where original Y is used as the coverage reference.

The results of the comparative experiments shown in Figure 2 indicate that the growth of Y proves that the coverage performance of all algorithms is optimized with increasing N. When $N > 40$, the growth of Y is significant. Our AIFS shows the best coverage performance among all of the compared algorithms, followed by PSO. The nonintelligent algorithm COSH cannot match the growth rate of the two previous intelligent algorithms. With $N > 50$ increasing further, AIFS becomes more pronounced than PSO because the proposed jumping and repulsing behavior types prevent the nodes from crowding or gathering and expand the covered area. With increasing N, crowding density becomes serious, and AIFS becomes likely to exhibit its advantage of jumping out of crowding.

4.3. Analysis of Local Optimization. Given node sets S with various crowding densities, we analyze their capability of jumping out of local crowding for three algorithms (the proposed AIFS, AFSA, and VFDPSO). Sets S are randomly arranged in a square with side length *Test* at the center of I, where $Test = 50$m, 40m, 30m, and 20m. In each move for nodes, the corresponding Y is investigated. The experimental results are presented in Figure 3, where *times* indicates the number of node moves.

The simulation results indicate that AIFS, AFSA, and VFDPSO can improve Y after the nodes move; thus, they can bring sensor networks out of the local optimum. However, unlike the faint progress of VFDPSO, Y results in smooth growth in AIFS and AFSA. Therefore, AIFS and AFSA are qualified in terms of working stability. Based on the rising speeds of Y, we find that AIFS is superior to AFSA in terms of speed in jumping out of the local optimum. This superiority is pronounced with deteriorating local optimization ($Test \downarrow$).

We conduct a comprehensive analysis for AIFS. We let coverage gain ΔY indicate the speed of Y relative to the previous movement. The statistics ΔY are extracted at five time points ($times = 2, 4, 6, 8$, and 10), as shown in Figure 4. Our AIFS can obtain $\Delta Y = 7{\sim}14\%$ at an early time ($times = 2$), indicating that AIFS can quickly jump out of the local optimum in the early stage of the algorithm even when it is in the worst local optimum ($Test = 20m$). Thus, our AIFS exhibits strong adaptability.

4.4. Boundary Analysis. This experiment investigates the performance of three algorithms (AIFS, AFSA, and VFDPSO) on

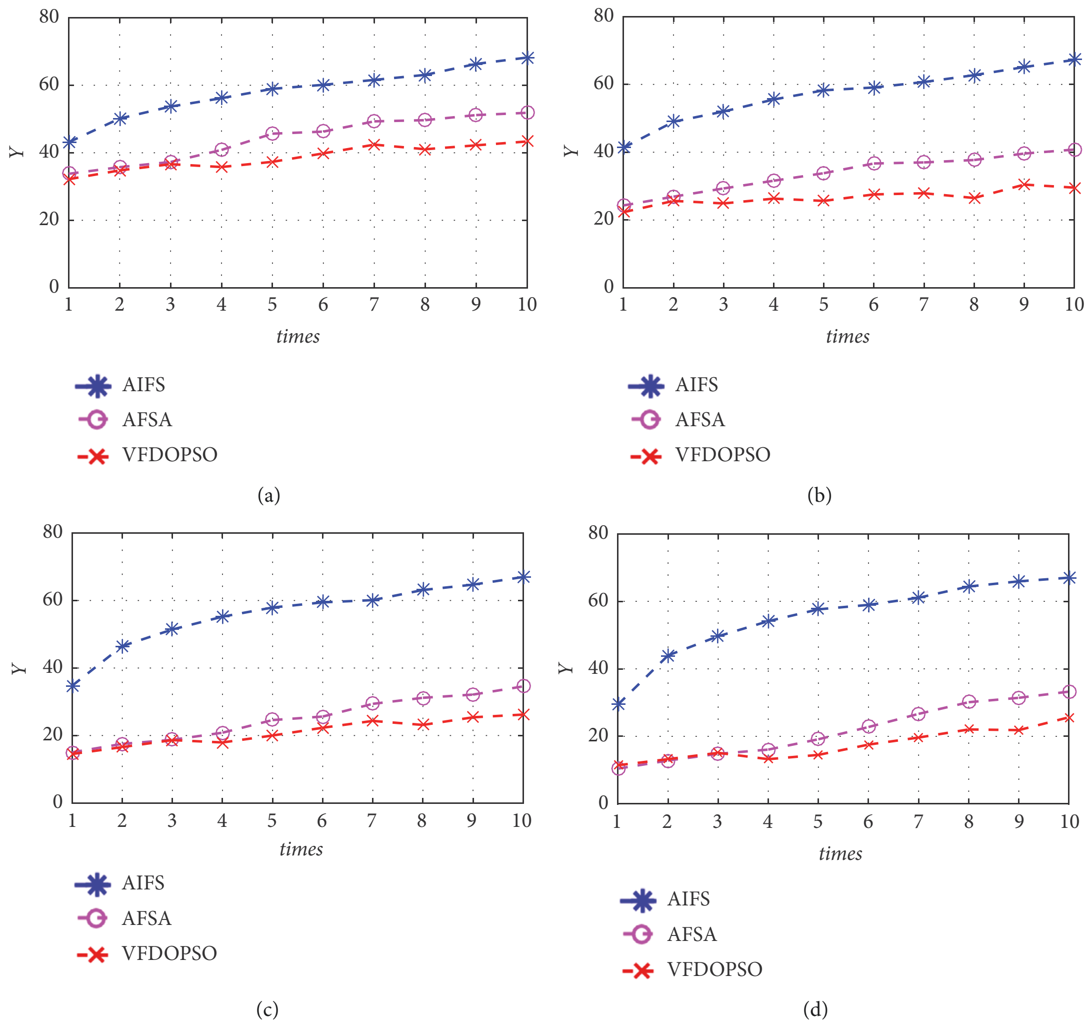

Figure 3: Analysis of coverage performance Y with movements and various local optimizations. (a) *Test*=50 m, (b) *Test*=40 m, (c) *Test*=30 m, and (d) *Test*=20 m.

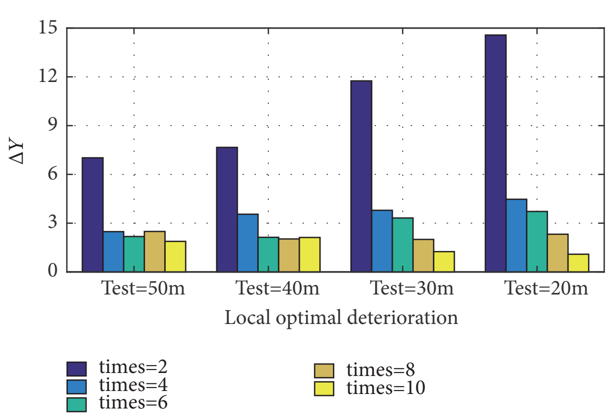

Figure 4: Analysis of ΔY with various *Test* for AIFS.

boundary problems by adjusting network size. The boundary problem can be effectively solved if region I is covered with less overflowing coverage. Thus, boundary loss is defined as follows:

$$waste = \frac{(\textit{the covered area by } S) \cup I - I}{N}. \qquad (8)$$

When *waste* is small, the average covered area outside I is also small. The variations in *waste* at each move are shown in Figure 5.

Numerous nodes that are located near the boundaries may not always appear in the initial phase ($times \leq 2$), where $waste < 0.5$. When algorithms are implemented, nodes tend to be evenly distributed, and several nodes gradually move to the vicinity of the boundary. For AFSA and VFDPSO, the variation curves of *waste* exhibit steep upward trends with increasing *times*. Until $times > 6$, the values of *waste* are stable with small fluctuations, indicating that the covered area outside I has stopped growing. Hopefully, the boundary loss *waste* of AIFS is always controlled within the region of $waste < 0.5$ because of the provided valuable behavior of wall

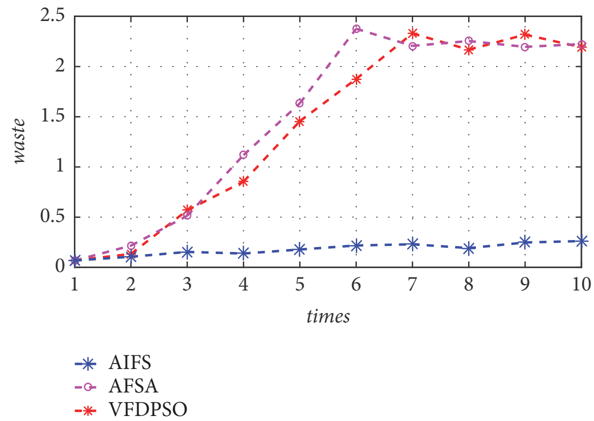

Figure 5: Boundary problem analyses with movements.

dodging. Thus, AIFS can work stably and efficiently on the boundary problem.

5. Conclusions and Future Work

The balance between coverage and cost is given full consideration in this study. The proposed AIFS solves two problems. One is the local optimum problem, to which AIFS proposes jumping behavior to enhance the speed of escaping from the local optimum. The other is the boundary problem, in which sensing resources are triggered by overflow coverage. Such a dodging behavior is designed to reduce the wasted area. Given such adaptive attenuation factors that attempt to characterize positional relations among nodes and pixels, the blindness and ineffectiveness in global search are avoided. All introduced techniques guide networks to effectively cover the monitoring area.

Although negative factors are disregarded in this study because optimal assumptions can make the research feasible, highly realistic environment settings and network modules remain important. Specifically, heterogeneous properties on sensing, communication, and processing should be regarded as part of our extensive research. And, preserving energy should be considered throughout the entire process of design. Distribution is a fundamental reason why sensor networks can be applied in many applications. Occasional dependency on global information is required in our AIFS, although each node performs its behavior independently in the initial process. Thus, future research should focus on combining our resulting work and a realistic model to achieve coverage performance. The present schedule is limited to a mixed methodology consisting of distributed elements combined with the central mode, which should be further refined and enhanced in future studies.

Data Availability

Our experiments have random scenarios with random values in a given range. The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

We are grateful to Le Yan for helping in editing the grammar of this paper. All the authors appreciate the supports from the National Natural Science Foundation of China, Nos. 61702228, 61803183, and 61304264, the Natural Science Foundation of Jiangsu Province, Nos. BK20170198, and BK20180591, the 11th Batch of Jiangsu "Six Level Talents Peak" Project, No. DZXX-026, Jiangsu Planned Projects for Postdoctoral Research Funds, No. 1601012A, and PAPD of Jiangsu Higher Education Institutions and the Fundamental Research Funds for the Central Universities, No. JUSRP1805XNC.

References

[1] Z. Jiao, H. Wang, and K. Ma, "The connectivity measurement in complex directed networks by motif structure," *International Journal of Sensor Networks*, vol. 21, no. 3, pp. 197–204, 2016.

[2] Z. Q. Jiao and K. Ma, "Adaptive synchronization in small-world networks with Lorenz chaotic oscillators," *International Journal of Sensor Networks*, vol. 24, no. 2, pp. 90–97, 2017.

[3] Q. Zhang, K. Q. Li, H. Dai et al., "Coverage optimization strategy based on co-evolution bee colony algorithm," *Computer Engineering and Design*, vol. 35, no. 4, pp. 1142–1146, 2014.

[4] J. Naskath, K. Srinivasagan, and S. Pratheema, "Coverage Maintenance using Mobile Nodes in Clustered Wireless Sensor Networks," *International Journal of Computer Applications*, vol. 21, no. 4, pp. 6–12, 2011.

[5] J. Gui, L. Hui, and N. Xiong, "Enhancing Cellular Coverage Quality by Virtual Access Point and Wireless Power Transfer," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9218239, 19 pages, 2018.

[6] A. V. Gonzalez, L. Brisolara, and P. R. Ferreia, "Efficiency Evaluation of Strategies for Dynamic Management of Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 5618065, 13 pages, 2017.

[7] Y. Lu and N. Sun, "Controllable Effective Threshold Based Fusion Coverage Algorithm in Mobile Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1529084, 8 pages, 2018.

[8] X. Y. Du, L. J. Sun, J. Guo et al., "Coverage Optimization Algorithm for Heterogeneous WSNs," *Journal of Electronics and Information Technology*, vol. 36, no. 3, pp. 696–702, 2014.

[9] J. B. Zhou, H. L. Liu, and K. Xu, "Dynamic sensor deployment strategy to combine particle swarm and virtual force," *Computer Engineering and Applications*, vol. 52, no. 10, pp. 118–123+156, 2016.

[10] F. Y. Jiang, S. L. Li, and X. R. Bao, "Multi-Objective Coverage Algorithm for WSN Based on Particle Swarm Optimization," *Advanced Materials Research*, vol. 765-767, pp. 508–513, 2013.

[11] Y. Y. Huang and K. Q. Li, "Coverage optimization of wireless sensor networks based on artificial fish swarm algorithm," *Application Research of Computers*, vol. 30, no. 2, pp. 554–556, 2013.

[12] L. M. Zhou, K. H. Yang, P. Zhou et al., "Optimal coverage configuration based on artificial fish swarm algorithm in WSNs,"

Application Research of Computers, vol. 27, no. 6, pp. 2276–2279, 2010.

[13] F. Qi and Y. R. Sun, "Wireless Sensor Network Coverage Efficiency Optimization Simulation," *Computer Simulation*, vol. 34, no. 08, pp. 297–301, 2017.

[14] N. Qin and J. Chen, "An area coverage algorithm for wireless sensor networks based on differential evolution," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, pp. 1–11, 2018.

[15] X. Liu and C. Liu, "Wireless Sensor Network Dynamic Mathematics Modeling and Node Localization," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1082398, 8 pages, 2018.

[16] P. Sui, Y. Guo, K.-F. Zhang, and H. Li, "Frequency-Hopping Transmitter Fingerprint Feature Classification Based on Kernel Collaborative Representation Classifier," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 9403590, 9 pages, 2017.

[17] K. X. Chen, W. C. Jiang, and Z. H. Long, "Energy Balance Coverage Control Algorithm for Wireless Sensor Networks," *Journal of Chinese Computer Systems*, vol. 38, no. 07, pp. 1448–1452, 2017.

[18] X. M. Mang and N. Liu, "Adaptive visual field artificial fish swarm algorithm for shortest path problem," *Journal on Communications*, vol. 35, no. 1, pp. 1–6, 2014.

[19] L. Yong and S. Na, "Controllable Effective Threshold Based Fusion Coverage Algorithm in Mobile Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1529084, 8 pages, 2018.

[20] Z. Li and P. Xin, "Evidence-Efficient Multihop Clustering Routing Scheme for Large-Scale Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 1914956, 14 pages, 2017.