Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 7080249, 11 pages
https://doi.org/10.1155/2019/7080249

Research Article

Sensing Coverage Algorithm of Sparse Mobile Sensor Node with Trade-Off between Packet Loss Rate and Transmission Delay

Kehua Zhao ,[1] Yourong Chen ,[1,2] Siyi Lu ,[2] Banteng Liu,[1] Tiaojuan Ren ,[1] and Zhangquan Wang[1]

[1] *College of Information Science and Technology, Zhejiang Shuren University, Hangzhou, Zhejiang 310015, China*
[2] *School of Information Science & Engineering, Changzhou University, Changzhou, Jiangshu 213164, China*

Correspondence should be addressed to Yourong Chen; jack_chenyr@163.com

Received 25 March 2019; Revised 16 June 2019; Accepted 23 June 2019; Published 8 July 2019

Guest Editor: Maurizio Casoni

To solve the problem of sensing coverage of sparse wireless sensor networks, the movement of sensor nodes is considered and a sensing coverage algorithm of sparse mobile sensor node with trade-off between packet loss rate and transmission delay (SCA_SM) is proposed. Firstly, SCA_SM divides the monitoring area into several grids of same size and establishes a path planning model of multisensor nodes' movement. Secondly, the social foraging behavior of Escherichia coli in bacterial foraging is used. A fitness function formula of sensor nodes' moving paths is proposed. The optimal moving paths of all mobile sensor nodes which can cover the entire monitoring area are obtained through the operations of chemotaxis, replication, and migration. The simulation results show that SCA_SM can fully cover the monitoring area and reduce the packet loss rate and data transmission delay in the process of data transmission. Under certain conditions, SCA_SM is better than RAND_D, HILBERT, and TCM.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes and are deployed manually or randomly in a given monitoring area. The sensor node recognizes and collects the state of the monitoring environment (such as temperature, humidity, and brightness) and sends this information to devices (sink nodes or base stations) that are suitable for data processing, visualization, analysis, and decision-making [1, 2]. Because sensing range and energy of sensor nodes are limited, the analysis of large amounts of local data is impossible when the monitoring area is huge and the environment is complex. Moreover, large-scale deployment requires a huge cost investment. Therefore, a sparse mobile sensor network with a small number of sensor nodes is considered; in particular, a device such as a drone or robot can be used in the sensor node to move to each monitoring position. The mobile monitoring node can sense the entire monitoring area. It reduces the system application cost [3].

However, sparse mobile sensor network may generate a large packet loss rate and a long data transmission delay. Therefore, mobile path selection and sensing coverage of mobile sensor nodes need be studied. At present, some scholars focus on the path selection of mobile sensor nodes installed on UAVs (unmanned aerial vehicle) and achieve certain results. For example, a rasterized motion model is established based on the environmental information within the scope of the UAV in the literature [4], and A$*$ algorithm is used to obtain the UAV's moving path. The literature [5] constructs the model of UAV flight path, quantitatively describes the threat area on the path, and uses improved artificial potential field method to obtain a sufficiently smooth flight path. A three-layer framework consisting of sensor layer, cluster head layer, and moving collector layer is proposed in the literature [6]. The mobile collector calculates its moving trajectory planning, uses the divide and conquer method to determine the turning point on the path, and obtains its efficient mobile data collection scheme. In the literature [7], each mobile sensor node is assigned a unique ID to find its potential moving position. The greedy algorithm is used to select the optimal stayed position from these potential moving positions. Finally, its optimal path is constructed. The literatures [4–7] mainly implement the path of mobile sensor node from starting position to target position, but how to efficiently cover the entire monitoring area is not taken into consideration. The literature [8] uses HILBERT curve as moving path of mobile aggregation node to cover the entire network. The literature [9] proposes a temporal coverage mechanism (TCM) for mobile sensor nodes in a sparse network environment. TCM divides the monitoring area into several grids so that mobile sensor nodes can cover the entire monitoring area as quickly as possible. Meanwhile, to achieve efficient coverage of the monitoring area under the sparse WSNs, some scholars focus on using artificial intelligence algorithms to obtain the moving paths of mobile sensor nodes. For example, the literature [10] proposes an improved artificial bee colony algorithm to solve the traveling salesman problem of neighborhood search in sparse WSNs and obtains the optimal data collection and movement paths of mobile sensor nodes. The literature [11] proposes autonomous movement model of mobile sensor node, which can be solved by bacterial foraging optimization algorithm (BFOA). The autonomous mobile solution of each mobile sensor node can be obtained. However, the literatures [8–11] do not consider that the storage space of mobile sensor nodes is limited and have large packet loss rate and high data transmission delay.

Therefore, a sensing coverage algorithm of sparse mobile sensor node with trade-off between packet loss rate and transmission delay (SCA_SM) is proposed. The monitoring area is divided into a finite number of square virtual grids by SCA_SM. Each mobile sensor node takes neighbor grid center as next stayed position, proposes a mathematical expression of moving paths of mobile sensor nodes, and establishes a moving path planning model for multiple sensor nodes. To quickly solve the optimization model, the processes of chemotaxis, replication, and migration in the BFOA are used to solve the model. In accordance with the optimization goal of the WSNs, fitness function is proposed to reduce the packet loss rate and data transmission delay as much as possible. Mobile sensor node simulates two basic actions of Escherichia coli, such as advancement and flipping in the search space. The mobile sensor node first moves a step in any random direction. If the fitness value of current moving path in this direction is improved compared with that value of previous moving path, then it will continue to advance in this direction. Instead, it will flip to find a new random direction and a position that can improve the fitness value. When all mobile sensor nodes cover the entire monitoring area or reach the end condition, SCA_SM outputs an optimal solution. Therefore, optimal moving paths of all mobile sensor nodes which cover the entire monitoring area are obtained, and the packet loss rate and data transmission delay in data transmission process are reduced.

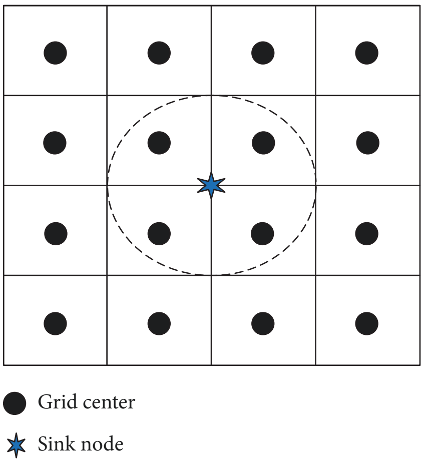


Figure 1: Principle of SCA_SM.

2. Algorithm Assumptions and Basic Principles

In SCA_SM, WSNs are assumed to be evenly distributed in the monitoring area of two dimensions. Multiple mobile sensor nodes and one static sink node exist in the network. Mobile sensor node has a fixed communication radius and data storage capacity. The position coordinates of the nodes are obtained by satellite positioning modules, such as GPS, Beidou, or other positioning modules. Only one grid center is allowed to pass once in one round of data collection. Static sink node collects data at the center of the monitoring area and can only listen to the data reported by mobile sensor nodes in the single communication range of sink node.

As shown in Figure 1, the monitoring area is divided into square grids of uniform size, and some mobile sensor nodes are randomly placed in the monitoring area. When the network is running, all mobile sensor nodes move to the neighbor grid center's positions which are not stayed before and sense the data of grid area. If a mobile sensor node is within the communication range of sink node, then the data are sent to sink node in single-hop manner; otherwise, the data are stored in the mobile sensor node. If storage space of mobile sensor node is full, the oldest data are deleted, and the received data are stored. However, SCA_SM still needs to solve the following two problems. The first is how to establish an optimization model of multisensor nodes' mobile sensing coverage through mathematical model. The second is how to use the improved bacterial foraging algorithm to solve the optimization model and obtain optimal moving path scheme of multiple mobile sensor nodes that can fully cover the monitoring area. The specific solutions to these two problems are as follows.

3. Establishment of the Optimization Model

The optimization model of multisensor nodes' mobile sensing coverage can be converted into

$$\min \quad \left(x_1 D_R + \frac{x_2 T_{average}}{T_{yu}} \right) \tag{1}$$

$$\text{s.t.} \quad P^k = \left\{ p_1^k, p_2^k, p_3^k, \ldots, p_j^k \right\} \tag{1a}$$

$$p_j^k \neq p_i^k, \quad \forall j \text{ and } j \neq i, \ \forall k \tag{1b}$$

$$C_{t+1}^k = \begin{cases} C_{\max} & \left(C_t^k + SD_{t+1}^k \right) \geq C_{\max} \text{ and } d_{ks} > d_{\max} \\ C_t^k + SD_{t+1}^k & \left(C_t^k + SD_{t+1}^k \right) < C_{\max} \text{ and } d_{ks} > d_{\max} \ , \quad \forall k \\ 0 & d_{ks} \leq d_{\max} \end{cases} \tag{1c}$$

$$D_{t+1}^k = \begin{cases} D_t^k + C_t^k + SD_{t+1}^k - C_{\max}, & \left(C_t^k + SD_{t+1} \right) \geq C_{\max} \text{ and } d_{ks} > d_{\max} \\ D_t^k & d_{ks} \leq d_{\max} \end{cases} , \quad \forall k \tag{1d}$$

$$\cup \, Cover_k = 1, \quad \forall k \tag{1e}$$

$$D_R = \frac{\sum_k D_{t+1}^k}{N_{total}} \tag{1f}$$

$$T_{average} = \frac{\left(\sum_m \left(T_m - t_m \right) + \sum_n \left(t - t_n \right) \right)}{N_{pack}} \tag{1g}$$

$$x_1 + x_2 = 1 \tag{1h}$$

where P^k represents the moving path of mobile sensor node k. p_j^k represents the jth stayed grid center's position of mobile sensor node k. C_t^k represents the amount of data stored in the mobile sensor node k at time t. $C_{\max}$ represents the maximum storage space of mobile sensor node. d_{ks} represents the distance from mobile sensor node k to sink node. $d_{\max}$ represents the maximum communication radius of sink node. SD_{t+1}^k represents the amount of data received and sensed by mobile sensor node k at the next moment. D_t^k represents the total number of lost packets of mobile sensor node k from network start time to time t. $Cover_k$ represents the sensing coverage of mobile sensor node k, that is, the ratio of the number of grid center positions in the moving path of mobile sensor node k and the total number of grids. D_R represents the packet loss rate. N_{total} represents the total number of packets generated by all mobile sensor nodes. $T_{average}$ represents the current data transmission delay. T_m represents the time when the packet m is successfully sent to sink node. t_m represents the generation time of packet m. t_n represents the generation time of packet n stored in mobile sensor node or discarded. N_{pack} represents the total number of packets generated by mobile sensor nodes after time t. T_{yu} represents the threshold of data transmission delay. x_1 represents the weight factor of packet loss rate. x_2 represents the weight factor of data transmission delay.

Equation (1) represents the objective function of optimization model, namely, minimizing the packet loss rate and data transmission delay. Formula (1a) represents that the moving path of mobile sensor node k is composed positions of grid centers. Formula (1b) represents that each mobile sensor node does not repeat through the same grid center during one round of movement. Formula (1c) represents that when mobile sensor node k is within the communication range of sink node, the data are directly sent to sink node; otherwise, the data are stored in the cache space. If the cache space is full, then the oldest data will be discarded. Formula (1d) represents an updated calculation formula for the total number of lost packets of mobile sensor node. Formula (1e) represents that the moving paths of all mobile sensor nodes contain all grid centers; namely, the sensing coverage equals 1. Formula (1f) is a formula for calculating the packet loss rate. Formula (1g) is a formula for calculating the data transmission delay.

4. Algorithm Solving

BFOA is a distributed parallel and random global optimization algorithm that is widely used in image processing, machine learning, pattern recognition, workshop scheduling, and other fields [12]. BFOA simulates the foraging behavior of Escherichia coli in human intestines and seeks optimal population through four operations: chemotaxis, aggregation, replication, and migration. The solving process for the optimal moving paths of mobile sensor nodes is the process of Escherichia coli searching for the area of abundant food. However, the solution of bacterial fitness value needs to be solved and BFOA needs to be improved in the solution process. Therefore, each bacterium is moving paths of all mobile sensor nodes which sense and cover the entire monitoring area. $data(S_x, S_y, k, t)$ represents the data sensed by mobile sensor node k on the grid (S_x, S_y) at time t. L_{now} represents the length of current moving path of mobile sensor node. G_{rest} represents the number of grids that mobile sensor node has not passed. $Cover_{sum}$ represents the sum of coverage of all mobile sensor nodes. N_m represents the total number of mobile sensor nodes. S represents the size of bacterial population. N_{ed}, N_{re}, and N_c represent the number of migration, replication, and chemotaxis, respectively. P_{ed} represents the probability of migration. N_m represents the number of mobile sensor nodes. $J(i, g, h, l)$ represents fitness value of bacterium i at the gth chemotaxis, hth replicas and lth migration. NG_i^k represents the neighbor grid set of mobile sensor node k in the bacterium i. $S^i = \{S_1^i, S_2^i, S_3^i, ..., S_k^i\}$ represents the movement scheme of current bacterium i after migration, and it is initially an empty set. S_k^i represents the moving path of mobile sensor node k in the bacterium i. The specific content of the model solution is as follows.

4.1. Calculation of Bacterial Fitness Value. Because the setting of bacterial fitness value affects the solution of optimization model (1), the chemotaxis operation leads the bacteria to preferentially move to a nutrient-rich environment. But if the fitness value of the bacterium i is directly calculated by model (1), mobile sensor node can easily preferentially move to the grid within the communication range of sink node. It results in a large packet loss rate and data transmission delay. Therefore, different fitness value formulas are adopted in the actual path finding process and after the path is searched. The specific implementation steps of bacterial fitness value calculation are as follows.

Step 1. Initialize the parameters and the storage space of all mobile sensor nodes. Initialize the storage space of sink node. t=1, k=1.

Step 2. At the current time t, the data of mobile sensor node k at the position (S_x, S_y) are sensed, stored, and recorded as $data(S_x, S_y, t)$. The data storage amount C_{t+1}^k at next moment according to formula (1c) is updated. If the data storage amount C_{t+1}^k at next moment is greater than maximum storage space $SC_{\max}$, then the oldest data are deleted. According to formula (1d), the deleted data are recorded in the number of lost packets, and the number of lost packets is updated.

Step 3 (t=t+1). Mobile sensor node k moves to next stayed position. If mobile sensor node k is within the single-hop range of sink node, all data in the storage space are directly sent to sink node. The data received by sink node are recorded, and the data storage capacity is updated. Then whether a neighbor mobile sensor node exists is determined. If it exists, then it communicates with the neighbor mobile sensor node, obtains coverage rate of the neighbor sensor node, and calculates the sum $Cover_{sum}$ of sensing coverage of the known mobile sensor nodes.

Step 4. If $t \le L_{now}$, then skip to Step 2; otherwise, k=k+1, t=1. If $k \le N_m$, then skip to Step 2; otherwise, skip to Step 5.

Step 5. If $Cover_{sum}$=1, then packet loss rate D_R is calculated by formula (1f). Skip to Step 7; otherwise, the mobile sensor node is still looking for the path. The number S_{num} of grids in the single-hop communication range of sink node is calculated. The number N_{loss} of discarded packets is calculated by formula (2).

$$N_{loss} = G_{rest} + C_t - S_{num}(C_{\max} + 1) \tag{2}$$

where G_{rest} represents the number of grids which is not passed before. C_t represents the sum of data stored by all mobile sensor nodes.

Step 6. When $N_{loss} \le 0$, the grids in the single-hop communication range of sink node is allocated reasonably, and the other grids do not generate the number of lost packets, $N_{loss} = 0$. Otherwise, N_{loss} is recorded in the number of lost packets, and the packet loss rate D_R' is calculated by formula (3) on the basis of received data packets of sink node and the number of lost packets stored in mobile sensor nodes.

$$D_R' = \frac{(N_t + N_{loss})}{N_{total}} \tag{3}$$

where N_t represents number of lost packets of mobile sensor node at previous moment and N_{total} represents total number of packets generated by all mobile sensor nodes.

Step 7. The estimate value $T_{average}$ of data transmission delay is calculated by formula (1g), and the fitness values of all bacteria are calculated by formula (4).

$$fitness = x_1 \overline{D}_R + \frac{x_2 T_{average}}{T_{yu}} \tag{4}$$

where packet loss rate $\overline{D}_R$ is

$$\overline{D}_R = \begin{cases} D_R', & Cover_{sum} < 1 \\ D_R, & Cover_{sum} = 1. \end{cases} \tag{5}$$

4.2. Chemotaxis of Bacteria. Mobile sensor node simulates two basic actions of Escherichia coli, such as advancement and flipping in the path planning process. First, each mobile sensor node in a bacterium moves a step in any random direction. If the fitness value of current moving path in this direction is improved compared with the fitness value of previous moving path, then it will continue to advance in this direction. Otherwise, it will flip to find a new random direction and a position that can improve the fitness value. If neighbor grid has been accessed, the mobile sensor node is easy to fall into dead end. At this time, the nearest grid should be selected to meet the constraint condition (1b), thereby achieving the chemotaxis operation. As shown in Figure 2, the specific implementation steps are as follows.

Step 1. Obtain all current bacterial information and replication parameter h, and let g=1, l=1, i=1, k=1.

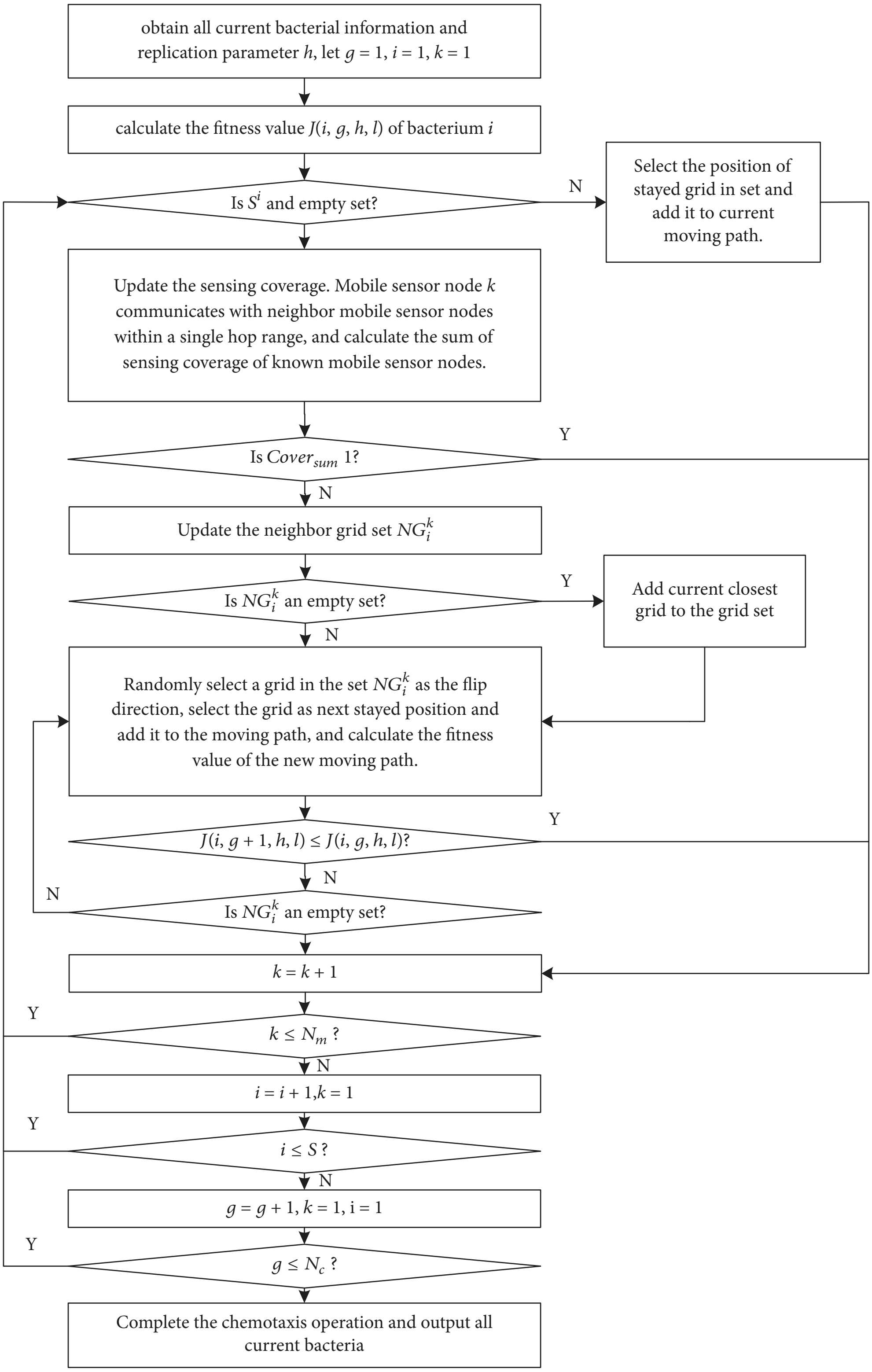

obtain all current bacterial information and replication parameter h, let $g = 1$, $i = 1$, $k = 1$
calculate the fitness value $J(i, g, h, l)$ of bacterium i
Is S^i and empty set?
N
Select the position of stayed grid in set and add it to current moving path.
Update the sensing coverage. Mobile sensor node k communicates with neighbor mobile sensor nodes within a single hop range, and calculate the sum of sensing coverage of known mobile sensor nodes.
Y
Is $Cover_{sum}$ 1?
N
Update the neighbor grid set NG_i^k
Y
Is NG_i^k an empty set?
Add current closest grid to the grid set
N
Randomly select a grid in the set NG_i^k as the flip direction, select the grid as next stayed position and add it to the moving path, and calculate the fitness value of the new moving path.
Y
$J(i, g+1, h, l) \leq J(i, g, h, l)$?
N
N
Is NG_i^k an empty set?
$k = k + 1$
Y
$k \leq N_m$?
N
$i = i + 1, k = 1$
Y
$i \leq S$?
N
$g = g + 1, k = 1$, i = 1
Y
$g \leq N_c$?
Complete the chemotaxis operation and output all current bacteria

Figure 2: Flow chart of chemotaxis.

Step 2. Calculate the fitness value $J(i, g, h, l)$ of the bacterium i. It is judged whether the set S^i is an empty set. If it is an empty set, then skip to Step 3. Otherwise, let $S^i_g = g + 1 + N_c(h-1)$. Select the S^i_gth stayed grid in the set S^i_k. Let it be the next grid of mobile sensor node k. Add it to current moving path, and obtain the new moving path. Skip to Step 6.

Step 3. Update the sensing coverage $Cover_k$, and mobile sensor node k communicates with neighbor mobile sensor nodes in the single-hop range to calculate the sum $Cover_{sum}$ of sensing coverage of known mobile sensor nodes. If $Cover_{sum}$ is 1, then skip to Step 6; otherwise, skip to Step 4.

Step 4. Update the neighbor grid set NG^k_i, and determine whether the set NG^k_i is an empty set. If it is, add current closest gird to the grid set NG^k_i; otherwise, neighbor grid set NG^k_i is not updated. Randomly select a grid in the set NG^k_i as the flip direction, delete the grid in the set NG^k_i, add the next stayed grid to moving path, and calculate the fitness value $J(i, g+1, h, l)$ of the new moving path.

Step 5. If $J(i, g+1, h, l) \leq J(i, g, h, l)$, it indicates that the bacteria are closer to optimal solution, so skip to Step 6. Otherwise, if fitness value is not improved, then judge whether the set NG^k_i is an empty set. If it is, skip to Step 6; otherwise, skip to Step 4 and reselect the random direction for chemotaxis.

Step 6 (k=k+1). If $k \leq N_m$, then skip to Step 2; otherwise, $i = i + 1$, k=1. If $i \leq S$, then skip to Step 2; otherwise, g=g+1, k=1, i=1. If $g \leq N_c$, then skip to Step 2; otherwise, the chemotaxis operation is completed.

4.3. Improved BFOA. As shown in Figure 3, SCA_SM calculates bacterial fitness value through formula (4) and uses chemotaxis, replication, and migration of improved bacterial foraging algorithm to solve the model (1). The specific implementation steps are as follows.

Step 1. Initialize the bacterial population and algorithm parameters: the initial positions of all mobile sensor nodes in bacteria are randomly generated, and neighbor grid set is updated. Let F_{best} be infinite, g=1, h=1, l=1, i=1, k=1.

Step 2. Use chemotaxis to obtain moving paths of mobile sensor nodes in all bacteria.

Step 3. Calculate the fitness values of S bacteria in the population and arrange them in ascending order. Remove half of the bacteria with high fitness value and copy half of the bacteria with small fitness value. h=h+1. If $h \leq N_{re}$, i=1, g=1, then skip to Step 2; otherwise, skip to next step.

Step 4. Select the moving paths and fitness value of the bacterium with smallest fitness value F_{local}. If $F_{local} < F_{best}$, $F_{best} = F_{local}$, then the moving paths of mobile sensor nodes in the bacterium are the optimal moving paths. Otherwise, the bacterium is eliminated and the optimal moving paths of mobile sensor nodes are replicated to the bacterium.

Step 5. Calculate the adaptive probabilities P^i_{self} of all S bacteria via formula (6).

$$P^i_{self} = \frac{J^i_{num} - J^{\min}_{num}}{J^{\max}_{num} - J^{\min}_{num}} P_{ed} \tag{6}$$

where J_{num} represents the fitness function and P_{ed} represents the migration probability.

Step 6. Generate a random number between 0 and 1. If the adaptive probability P^i_{self} is greater than the random number, initial positions of N_m mobile sensor nodes are randomly generated as current grid positions and current moving paths in the bacterium i. Let the set S^i be an empty set; otherwise, the moving paths of bacterium i are saved in set S^i.

Step 7 (l=l+1). If $l \leq N_{ed}$, then skip to Step 2; otherwise, output the optimal moving paths of all mobile sensor nodes and end the algorithm.

5. Algorithm Simulation

5.1. Simulation Parameter Selection. To verify the algorithm performance, the simulation experiment is performed and SCA_SM is compared with RAND_D, HILBERT [8], and TCM [9]. Assume that the 200 m∗200 m monitoring area is divided into several virtual square grids. The sink node is in the center of the monitoring area. The positions of mobile sensor nodes are random, and the communication radius of each nodes is 50 m. In the simulation, the main parameters are set to the following values. The maximum storage space $C_{\max}$ is 6 kbit. The amount of data received and sensed by the node in each uncovered grid is 2 kbit. The threshold T_{yu} of data transmission delay is 10^2. The weight factor x_1 of packet loss rate and weight factor x_2 of data transmission delay are 0.5. The chemotaxis number N_c is 10; the replication number N_{re} is 20. The migration number N_{ed} is 50. The migration probability P_{ed} is 0.6. The number of bacteria S is 50 when square grid number is less than or equal to 150, and the number of bacteria S is 100 when square grid number is greater than 150.

5.2. Analysis of Simulation Results. Because multiple mobile sensor nodes are considered and randomly placed within the monitoring area, initial positions of mobile sensor nodes are randomly generated. To illustrate the effectiveness of the algorithms, random initial positions with (25, 25), (25, 175), and (175, 175) are selected, and the number of mobile sensor nodes is 2, 3, and 4. Then the parameters in Section 4.1 are selected and Figures 4–7 are obtained. As shown in Figure 4, all mobile sensor nodes in RAND_D randomly select the closest grid which is not stayed before as next stayed position. As shown in Figure 5, the number of stayed positions of HILBERT is a power of 4, and each mobile sensor node moves following HILBERT path until the entire monitoring area is covered. As shown in Figure 6, each mobile sensor node in

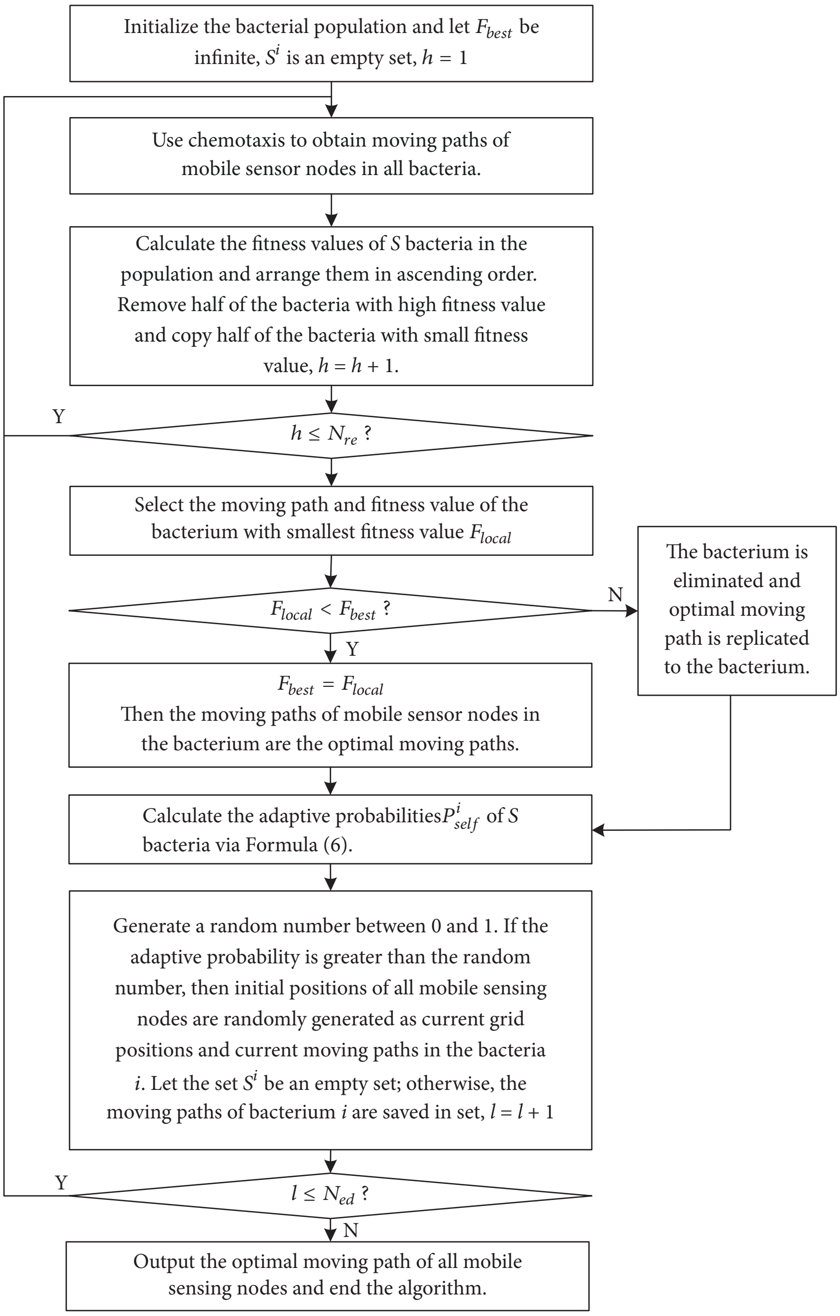

Initialize the bacterial population and let F_{best} be infinite, S^i is an empty set, $h = 1$
Use chemotaxis to obtain moving paths of mobile sensor nodes in all bacteria.
Calculate the fitness values of S bacteria in the population and arrange them in ascending order. Remove half of the bacteria with high fitness value and copy half of the bacteria with small fitness value, $h = h + 1$.
Y
$h \le N_{re}$?
Select the moving path and fitness value of the bacterium with smallest fitness value F_{local}
$F_{local} < F_{best}$?
N
Y
The bacterium is eliminated and optimal moving path is replicated to the bacterium.
$F_{best} = F_{local}$
Then the moving paths of mobile sensor nodes in the bacterium are the optimal moving paths.
Calculate the adaptive probabilitiesP^i_{self} of S bacteria via Formula (6).
Generate a random number between 0 and 1. If the adaptive probability is greater than the random number, then initial positions of all mobile sensing nodes are randomly generated as current grid positions and current moving paths in the bacteria i. Let the set S^i be an empty set; otherwise, the moving paths of bacterium i are saved in set, $l = l + 1$
Y
$l \le N_{ed}$?
N
Output the optimal moving path of all mobile sensing nodes and end the algorithm.

Figure 3: Flow chart of SCA_SM algorithm.

TCM follows the shortest moving path in the literature [10] which fully covers the monitoring area. As shown in Figure 7, no matter the number of mobile sensor nodes, SCA_SM can establish a path planning model of multisensor nodes' movement, calculate bacterial fitness value of each bacterium, solve the optimization model (1) by improving the bacterial foraging algorithm, and obtain optimal moving paths. Every mobile sensor node passes through several grids and moves to the vicinity of sink node to send data. Then SCA_SM reduces the packet loss rate and data transmission delay.

The bacterial population completes chemotaxis operation and replication operation of all mobile sensor nodes, then it implements and completes a migration operation. After each migration operation, mobile sensor nodes have their current optimal moving paths with different steps. The fitness value of optimal moving paths is current optimal fitness value. As shown in Figure 8, the abscissa represents the number of migrations, and the ordinate represents the optimal fitness value. The fitness value of SCA_SM decreases almost vertically after previous migration. After five migrations, the fitness value of optimal moving path drops to a lower value. As migration operation can help the algorithm jump out of local optimum and improve the convergence accuracy, the fitness value converges to 0.13424 after about 16 migrations.

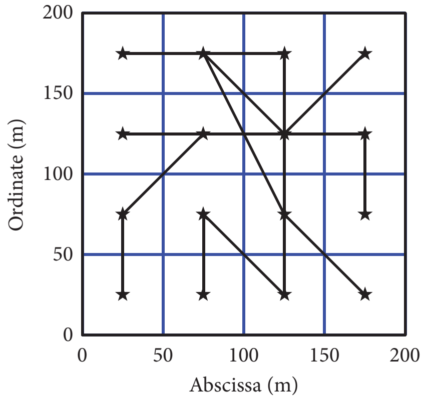


Figure 4: Moving paths of RAND_D.

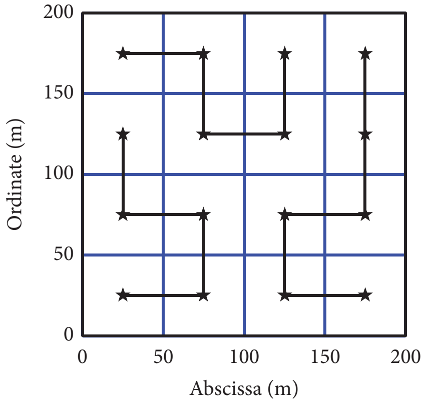


Figure 5: Moving path of HILBERT.

This fitness value is the optimal solution of optimization model (1). Therefore, SCA_SM is convergent.

Simulation Results for Different Maximum Storage Spaces. In the section, the parameters in Section 5.1 are selected. Side length of the monitoring area is 400 m. Number of mobile sensor nodes is 3. Maximum storage space is 20, 22, 24, 26, 28, and 30 kbit. 10 different initial position distributions of the nodes are randomly generated. The packet loss rates and data transmission delays of RAND_D, HILBERT, TCM, and SCA_SM under each initial position distribution of nodes are calculated separately, and the average values are taken as simulation results.

As shown in Figure 9, in RAND, although maximum storage space of the node is increased, mobile sensor node randomly selects moving path. It leads that some moving paths are away from sink node and improves the packet loss rate. Therefore, packet loss rate of RAND fluctuates with the increase in the maximum storage space. Packet loss rates of HILBERT, TCM, and SCA_SM decrease with the increase in the maximum storage space. Especially, packet loss rate of SCA_SM decreases faster. Because HILBERT and TCM do not consider the data loss problem when storage space is full, packet loss rates of HILBERT and TCM are larger than that of SCA_SM. The SCA_SM uses bacterial foraging algorithm to solve the optimization model (1) and finds moving paths that weigh packet loss rate and data transmission delay through multiple migrations. In the moving path selection, when the storage space is almost full, the node moves to the communication range of sink node as much as possible, sends the stored data to sink node, and continues to perform mobile coverage. It can reduce the packet loss rate. In summary, when maximum storage space increases, packet loss rate of SCA_SM is always lower than that of RAND_D, HILBERT, and TCM. When maximum storage space reaches 34 kbit, the packet loss rate is close to 0%.

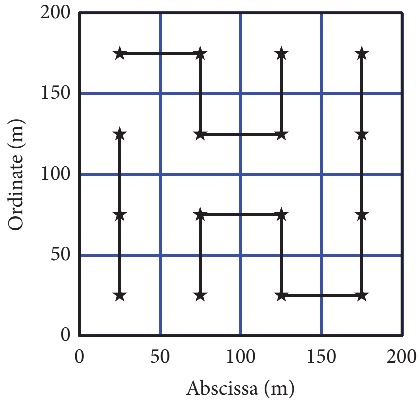
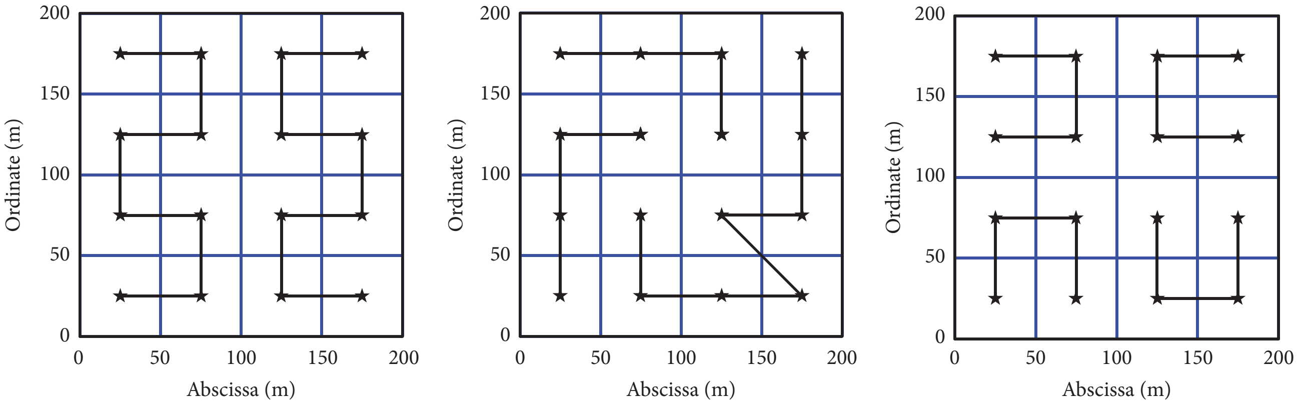


Figure 6: Moving paths of TCM.

As shown in Figure 10, RAND_D has randomness, and data transmission delays of HILBERT and TCM decrease with the increase in the maximum storage space. However, they do not consider the data transmission delay in the moving path selection. SCA_SM establishes an optimization model, which is solved by improved bacterial foraging algorithm, and obtains the optimal moving path of each mobile sensor node which is through the grid centers in the communication range of sink node. When mobile sensor node moves along its optimal path, it can send its own data to sink node in time. Therefore, the data transmission delay of SCA_SM is always lower than the data transmission delays of RAND_D, HILBERT, and TCM.

Simulation Results in Different Monitoring Areas. Considering the size of the monitoring area, the number of square grids divided in the monitoring area is different. To analyze the impact of different sizes of monitoring area on performance parameters, number of mobile sensor nodes is 3, and maximum storage space is 1/4 of the number of grids. Square monitor area 1 has a side length of 800 m. Monitor area 2 has a side length of 700 m. Monitor area 3 has a side length of 500 m. Monitor area 4 has a side length of 400 m. Monitor area 5 has a side length of 200 m. The parameters in 4.1 are selected, and 10 different initial position distributions of nodes are randomly generated. The packet loss rate and data transmission delay of each algorithm are calculated under each initial position distribution of sensor nodes, and the average values are used as simulation results.

As shown in Figures 11 and 12, in the different monitoring areas, RAND_D has a large number of loss packets and



(a) Moving paths of two sensor nodes

(b) Moving paths of three sensor nodes

(c) Moving paths of four sensor nodes

Figure 7: Moving paths of SCA_SM.

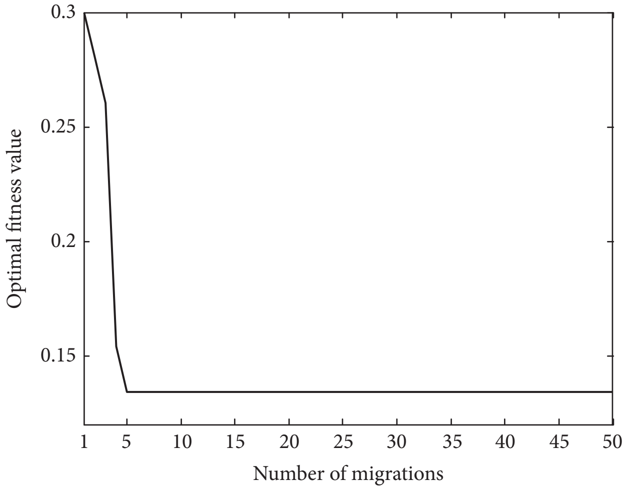


Figure 8: Convergence graph of SCA_SM.

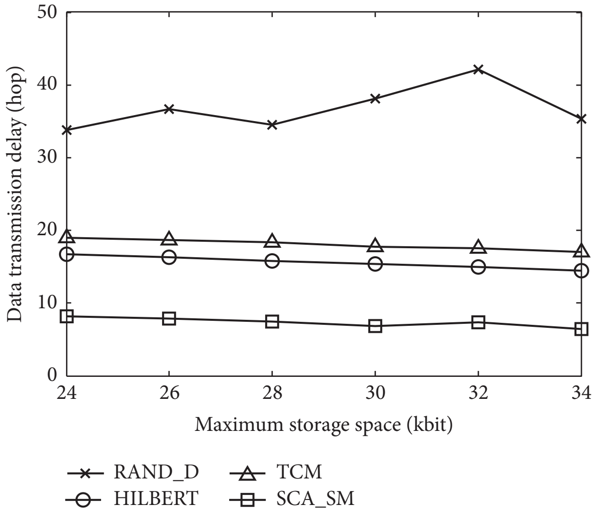


Figure 10: Comparison of data transmission delays for different maximum storage spaces.

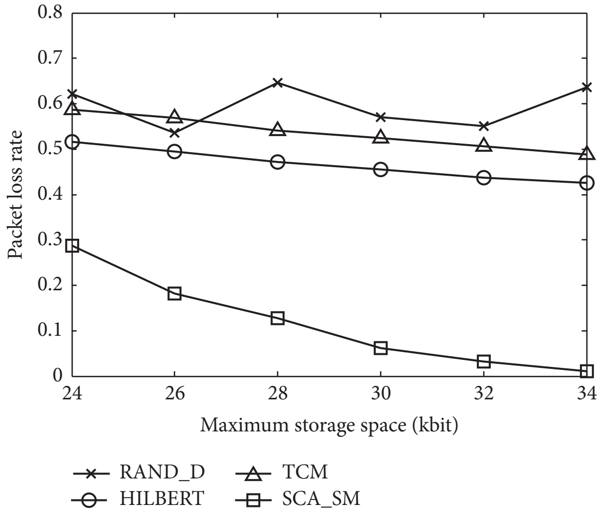


Figure 9: Comparison of packet loss rates for different maximum storage spaces.

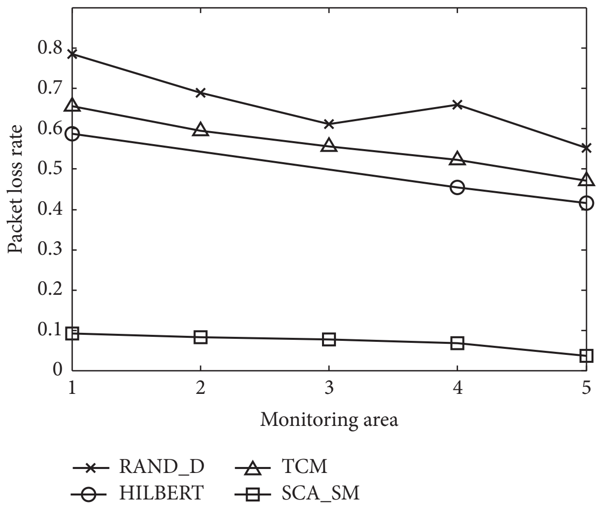


Figure 11: Comparison of packet loss rates in different monitoring areas.

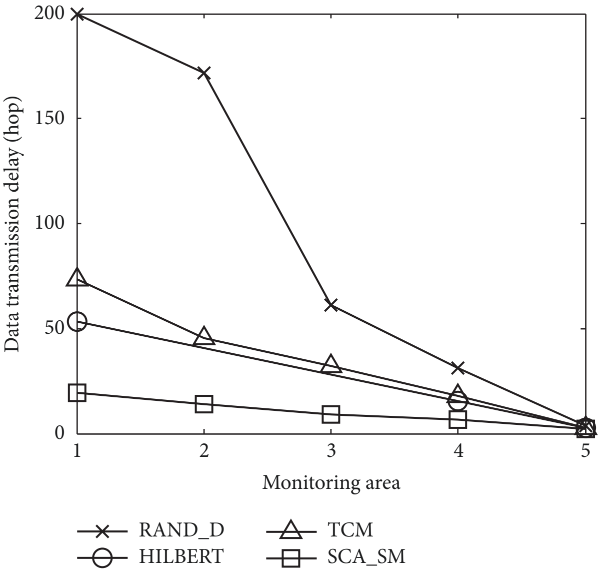


Figure 12: Comparison of data transmission delays in different monitoring areas.

a long moving path; therefore, its packet loss rate and data transmission delay are relatively large. The number of stayed grid of HILBERT must be a power of 4, so only the monitoring areas 1, 4, and 5 are valid and the moving path of each mobile sensor node is fixed. Its perceptual coverage problem of mobile sensor node is not considered. TCM only covers the entire monitoring area as short as possible and does not consider the packet loss rate and data transmission delay as HILBERT. SCA_SM establishes a reasonable optimization model and fitness function due to comprehensive consideration of packet loss rate and data transmission delay. Moreover, it uses improved bacterial foraging algorithm to obtain the optimal moving paths of all mobile sensor nodes. The moving optimal paths can maintain low packet loss rate and data transmission delay, and it is significantly better than RAND_D, HILBERT, and TCM.

6. Conclusion

A sensing coverage algorithm of sparse mobile sensor node with trade-off between packet loss rate and transmission delay (SCA_SM) is proposed in this paper. First, the algorithm assumptions and basic principles are proposed. Second, path selection constraint, data capacity update, packet loss rate, data transmission delay, and other related parameters and conditions are considered. An optimization model for fully covering the monitoring area and balancing the packet loss rate and data transmission delay is established. Bacterial fitness value calculation, bacterial chemotaxis operation, and improved bacterial foraging algorithm are proposed to solve the optimization model. Optimal moving paths of multiple mobile sensor nodes are obtained. Finally, simulation parameters of the algorithm are given. The simulation results show that the SCA_SM algorithm can find the optimal moving path of multiple mobile sensor nodes, has low data packet loss rate and data transmission delay, and is superior to RAND_D, HILBERT, and TCM. However, the algorithmic complexity of SCA_SM is relatively high. Therefore, the next goal is to study the optimization model and heuristic method of each mobile sensor node to solve the optimization model based on local information. Then optimization solution of each mobile sensor node is obtained, and it reduces the complexity of the algorithm.

Data Availability

The simulation data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 6 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

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

Acknowledgments

This work was supported by the Public Welfare Technology Application and Research Projects of Zhejiang Province of China under Grants no. LGF19F010006, no. LGG19F010011, no. LGF19F010005, and no. LGF18F010005, the National Natural Science Foundation of China under Grant no. 61501403, Education Department Project of Zhejiang Province of China under Grant no. Y201840757, and Science and Technology Project of Zhejiang Water Resources Department of China under Grant no. RC1850.

References

[1] W. H. Wang, T. Wang, Q. Wu et al., "Survey of delay-constrained data collection with mobile elements in WSNs," *Journal of Computer Research and Development*, vol. 54, no. 3, pp. 474–492, 2017.

[2] J. Habibi, H. Mahboubi, and A. G. Aghdam, "A gradient-based coverage optimization strategy for mobile sensor networks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 477–488, 2017.

[3] S. Rashed and M. Soyturk, "Effects of UAV mobility patterns on data collection in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Communication, Networks and Satellite (COMNESTAT '15)*, pp. 74–79, IEEE, Indonesia, December 2015.

[4] Q. R. Zhang, R. X. Wei, R. K. He et al., "Path planning for unmanned aerial vehicle in urban space crowdedwith irregular obstacles," *Control Theory & Applications*, vol. 32, no. 10, pp. 1407–1413, 2015.

[5] J. R. Ding, C. P. Du, Y. Zhao et al., "Path planning algorithm for unmanned aerial vehicles based on improved artificial potential field," *Journal of Computer Applications*, vol. 36, no. 1, pp. 287–290, 2016.

[6] M. Zhao, Y. Yang, and C. Wang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 770–785, 2015.

[7] A. Kaswan, P. K. Jana, and M. Azharuddin, "A delay efficient path selection strategy for mobile sink in wireless sensor networks," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '17)*, pp. 168–173, IEEE, 2017.

[8] X. Li and Y. J. Wang, "WSN mobile aggregation node track design based on Hilbert space filling curve," *Modern Electronics Technique*, vol. 39, no. 23, pp. 17–21, 2016.

[9] L.-H. Hung, Y.-W. Huang, and C.-C. Lin, "Temporal coverage mechanism for distinct quality of monitoring in wireless mobile sensor networks," *Ad Hoc Networks*, vol. 21, no. 5, pp. 97–108, 2014.

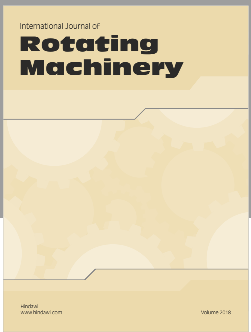
[10] Y. Yue, J. Li, H. Fan, and Q. Qin, "Optimization-based artificial bee colony algorithm for data collection in large-scale mobile wireless sensor networks," *Journal of Sensors*, vol. 2016, Article ID 7057490, 12 pages, 2016.

[11] A. A. Ari, I. Damakoa, A. Gueroui et al., "Bacterial foraging optimization scheme for mobile sensing in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 24, no. 3, pp. 254–267, 2017.

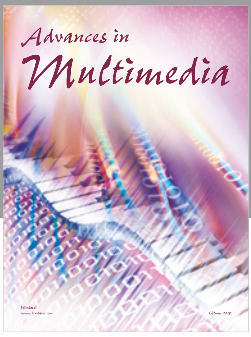
[12] Z. D. Wang, E. L. Chen, and Z. D. Hu, "Sensor node deployment strategy of chaotic optimization of bacterial foraging algorithm," *Chinese Journal of Sensors and Actuators*, vol. 31, no. 1, pp. 110–118, 2018.

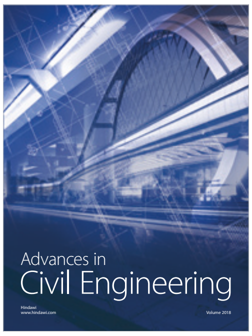
Journal of
Engineering

The Scientific
World Journal

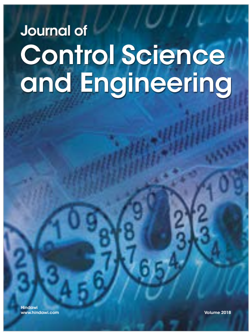

International Journal of
Rotating
Machinery

Journal of
Sensors

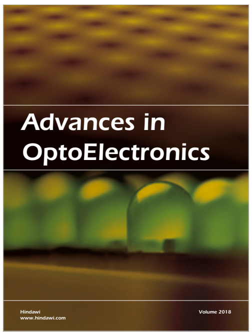

Advances in
Multimedia


Advances in
Civil Engineering

Journal of
Robotics

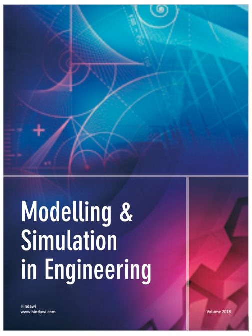

Journal of
Control Science
and Engineering

Journal of
Electrical and Computer
Engineering

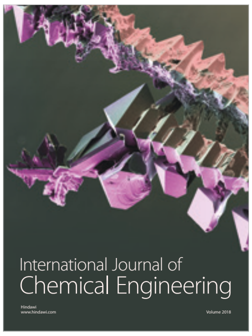

Advances in
OptoElectronics

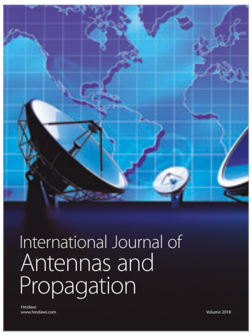
VLSI Design

International Journal of
Navigation and
Observation

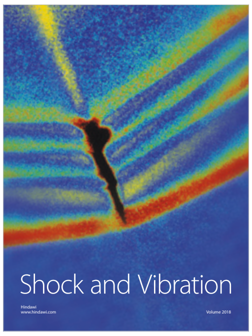

Modelling &
Simulation
in Engineering

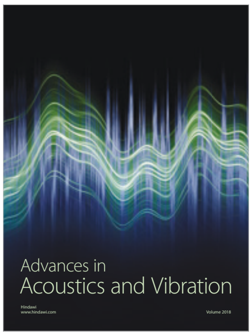
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration