Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1206129, 12 pages
https://doi.org/10.1155/2017/1206129

Research Article

Solving *k*-Barrier Coverage Problem Using Modified Gravitational Search Algorithm

Yanhua Zhang,[1,2] Xingming Sun,[1] and Zhanke Yu[3]

[1] *Nanjing University of Information Science & Technology, Nanjing, China*
[2] *Jiangsu Lightning Protection Center, Nanjing, China*
[3] *College of Communications Engineering, PLA University of Science and Technology, Nanjing, China*

Correspondence should be addressed to Xingming Sun; sunnudt@163.com

Received 28 December 2016; Revised 11 March 2017; Accepted 30 March 2017; Published 3 May 2017

Academic Editor: Dan Simon

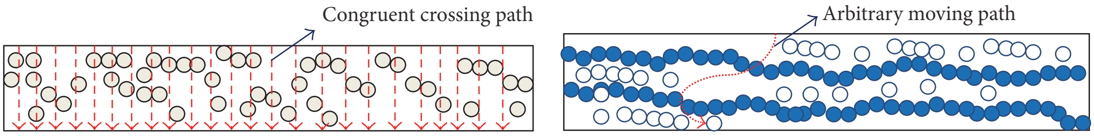
Coverage problem is a critical issue in wireless sensor networks for security applications. The *k*-barrier coverage is an effective measure to ensure robustness. In this paper, we formulate the *k*-barrier coverage problem as a constrained optimization problem and introduce the energy constraint of sensor node to prolong the lifetime of the *k*-barrier coverage. A novel hybrid particle swarm optimization and gravitational search algorithm (PGSA) is proposed to solve this problem. The proposed PGSA adopts a *k*-barrier coverage generation strategy based on probability and integrates the exploitation ability in particle swarm optimization to update the velocity and enhance the global search capability and introduce the boundary mutation strategy of an agent to increase the population diversity and search accuracy. Extensive simulations are conducted to demonstrate the effectiveness of our proposed algorithm.

1. Introduction

Recently, interest in wireless sensor networks (WSNs) in numerous applications has increased considerably. The connection of physical things to the Internet makes it possible to access remote sensor data and to control the physical world from a distance [1]. This capability is a critical issue for security applications, such as border surveillance, forest fires monitoring, and intruding enemy planes detection [2–6]. Three categories of the coverage problem exist in the literature [7, 8]: target coverage [9–15], area coverage [16–23], and barrier coverage [24–30]. Unlike the full coverage, barrier coverage does not necessarily cover the whole region in the WSNs. Barrier coverage of the WSNs aims to detect the intruders attempting to cross the regions of interest. It requires forming a chain of overlapping sensors across the whole regions of interest from left to right.

In [31], barrier coverage is classified into two categories: weak barrier coverage and strong barrier coverage. Weak barrier coverage only requires detecting intruders moving along congruent crossing paths, and strong barrier coverage requires detecting intruders with arbitrary moving paths. Figure 1 shows the two kinds of barrier coverage. In this paper, we address the barrier coverage formation problem for strong barrier coverage.

The k-barrier coverage refers to all crossing paths through the region that is k-covered. A crossing path can be covered by at least k distinct sensors [32]. The WSNs are said to be fault tolerant if they remain functional after a failure of up to $k-1$ sensors. Therefore, k-barrier coverage is an effective measure to ensure robustness. To achieve k-barrier coverage, the k disjoint chain of overlapping sensors must be formulated in the WSNs. However, after the initial random deployment, in general, the k-barrier coverage cannot be satisfied. Recent technological advances in practical mobile sensors allowed sensor nodes to have the ability to improve barrier coverage performance after sensor networks had been deployed. Meanwhile, the development of detection technology improves the detection level of sensor in WSNs [33–35].



(a) Weak barrier coverage

(b) Strong barrier coverage

Figure 1: Two kinds of barrier coverage.

There are many challenging issues in the k-barrier coverage because WSNs have their own characteristics [25]. First, there are a large number of sensor nodes in WSNs, and building large-scale models has always been a technical difficulty. Second, sensor nodes have limited energy and processing capacities and are influenced by the surrounding environment and irregular terrain. Third, moving the mobile sensors to build the k-barrier coverage costs a lot of energy; therefore, we aim to move the sensor nodes as few as possible. Considering the above reasons, we study the k-barrier coverage problem in hybrid sensor networks, which consist of both stationary and mobile sensors. After initial deployment with stationary sensors, barrier coverage generally cannot be satisfied. In this paper, we move mobile sensors to repair barrier gaps and form k-barrier coverage. In WSNs, relocating the mobile sensors to repair such gaps requires considerable energy. An issue to be addressed is ensuring that the number of relocated sensors is as few as possible. Therefore, we investigated how to form k-barrier coverage with a minimum number of mobile sensors that need to be moved. In this paper, we formulate the k-barrier coverage problem as a constrained optimization problem in large-scale WSNs.

For large-scale constrained optimization problems, the classical optimization algorithms cannot provide a suitable solution because the search space is increasing exponentially with problem size. Many new types of algorithms, such as intelligence algorithm and heuristic algorithm, have been proposed. Gravitational search algorithm (GSA) is one of the latest heuristic optimization algorithms, which is based on Newton's law of gravity and mass interactions [36]. GSA has been proven to have high-quality performance in solving different optimization problems [37–39]. It can speed up the solution process by adjusting the accuracy of the search with gravitational constant. Furthermore, GSA is memoryless and works efficiently similar to algorithms with memory. However, in solving complex constrained problems, GSA may be easily trapped into local optimums. To enhance the global optimum ability and escape from local optimal solution, a newly modified GSA called hybrid particle swarm optimization and gravitational search algorithm (PGSA) is developed in this paper. PGSA integrates the ability to exploit in particle swarm optimization (PSO) with the ability to explore in GSA to synthesize both algorithms' strength. In PGSA, the velocity updating is adjusted to improve the search accuracy, and agent boundary mutation strategy is introduced to increase the agent diversity and improve the convergence of the algorithm significantly. Simulation experiments were performed to evaluate the proposed algorithm performance. The numerical results demonstrate that the proposed algorithm provides very remarkable results for solving k-barrier coverage problem.

Our main contributions are summarized as follows:

(i) To the best of our knowledge, we are the first to apply the GSA algorithm to solve the k-barrier coverage problem. At the same time, we are the first to propose the encoding strategy based on probability and k-barrier coverage generation strategy by introducing the vertex-splitting method.

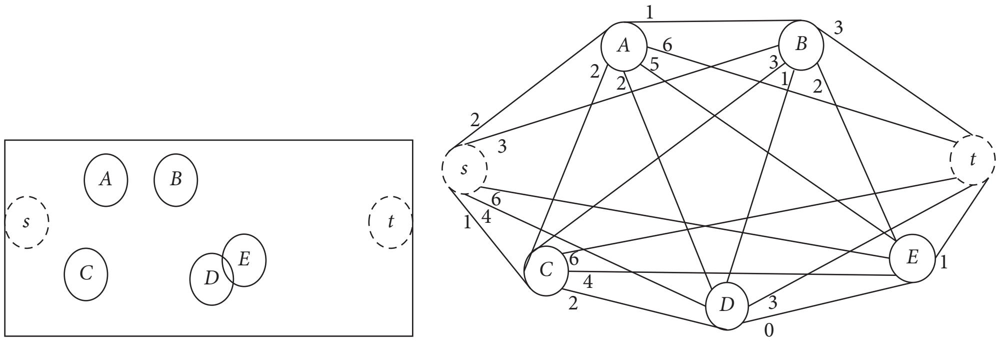
(ii) We formulate the k-barrier coverage into constraint optimization problem and first propose the energy constraint of sensor node to prolong the lifetime of the k-barrier coverage.

(iii) We propose the newly modified GSA (PGSA), which adjusts the velocity updating by integrating the ability to exploit in PSO to enhance the global search capability and introduce agent boundary mutation strategy to increase population diversity and search accuracy.

The rest of the paper is organized as follows: in Section 2, we briefly introduce the existing related works. In Section 3, preliminaries and the standard GSA algorithm are introduced. Section 4 presents our algorithm design. The simulation and comparison study are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Related Works

Kumar et al. [31] first defined the notion of k-barrier coverage and proposed two notions of barrier coverage in a belt region: weak barrier coverage and strong barrier coverage. They also presented a critical condition for weak barrier coverage. Liu et al. [40] derived critical conditions and presented a solution for strong barrier coverage when sensor nodes are distributed according to a Poisson point process. Bereg and Kirkpatrick [41] studied the redundant properties of k-barrier coverage and defined two notions of thickness and resilience. The thickness of the barrier counts the minimum number of sensor region intersections. The resilience of the barrier counts the minimum number of sensors whose removal permits a path with no sensor region intersections. In [42], Zhang et al. studied strong barrier coverage problem in wireless sensor networks with directional sensors. They presented an efficient centralized algorithm and a distributed algorithm to solve the barrier coverage problem. Sun et al. [43] proposed a novel algorithm to solve the multiobjective optimization coverage problem. The novel algorithm improves the quality of



(a) Sensor deployment

(b) Weighted barrier graph (WBG)

Figure 2: Deployment and WBG of sensors.

network coverage and extends the life cycle of the network significantly. In [44], they presented a novel linear programming optimization coverage scheme in WSNs. This algorithm improves the coverage and network quality of service and extends the network lifetime effectively. Silvestri and Goss [45] proposed an autonomous deployment algorithm named MobiBar to construct k-barrier coverage with mobile sensors. In [46], they studied the question of barrier coverage of a line-based sensor deployment strategy and concentrated on the efficiency of improving barrier using sensor mobility. Du et al. [47]. aim at prolonging the barrier lifetime under a novel k-discrete barrier coverage model in terms of the sensor mobility of the mobile sensors participated.

Recently, many evolutionary and heuristic algorithms have been employed to solve coverage-related problems [48–59]. Zhang et al. [48] proposed an integer linear programming (ILP) formulation for the barrier coverage problem. They investigated how mobile sensors can be efficiently relocated to achieve k-barrier coverage and construct an ILP model with a totally unimodular constraint coefficient matrix to solve the barrier coverage problem. Gupta et al. [53] proposed a genetic algorithm (GA) to solve the coverage and connectivity issues in k-barrier coverage problem. They attempted to find the potential positions for placing sensor nodes so that it will fulfill the k-coverage of the targets and m-connectivity of the sensor nodes. A learning automata-based method had been introduced in [54]. They formulated a barrier coverage model based on stochastic coverage graph and then proposed a learning automata-based method to find a near-optimal solution to the stochastic barrier coverage problem. A neural network algorithm is proposed to solve the sensor node fault detection issues and coverage problems in [55]. They presented a dynamic model of WSNs and its application to sensor node fault detection. Liao et al. [56] used the ant colony optimization (ACO) algorithm to solve the coverage problem and maximize the lifetime of the network. They modeled the deployment problem as multiple knapsack problems and evaluated the improvement of the energy use problem in sensor networks using the area coverage. The simulations showed that the algorithm was effective in the coverage problem and could prolong the lifetime of the network. Huang and Li [57] focused on the problems of the coverage optimization method based on artificial fish swarm algorithm. They established a mathematical model to solve the optimization coverage problems and used the artificial fish swarm algorithm to search an optimal solution in the solution space by simulating fish swarm behaviors. Maleki et al. [59] presented a PSO algorithm to solve the optimized coverage problem using a hybrid PSO and differential evolution (DE) in WSNs. The results of the simulation show that PSO algorithm is efficient in the lifetime of the network.

3. Preliminaries and Problem Formulations

3.1. Preliminaries. In this section, some definitions, assumptions, and preliminaries for the proposed method to guarantee the k-barrier coverage in the network are introduced. We assumed that the N sensors with omnidirectional motion are randomly deployed in a two-dimensional rectangular belt area with a size of $L \times W$. The location information of each sensor can be known through a localization mechanism. We assume that each mobile sensor can move anywhere in the belt area without being limited by energy constraints.

Definition 1 (weighted barrier graph (WBG)). Wang et al. [25] introduced a novel graph model of the weighted barrier graph (WBG). They defined WBG as a triple $\langle V, E, W \rangle$, where V is a set of sensors nodes $V = \{s, v_1, v_2, \ldots, v_n, t\}$. In the set V, s and t are two virtual nodes and correspond to the original node and definition node, respectively (Figure 2(b)). $E = e(u, v)$, $u, v \in V$, is the link from sensor u to v. $W = \{w(u, v)\}$ is the set of weight of each edges. $d(u, v)$, $u, v \in V$, is the distance between two sensors u and v.

Definition 2 (the weight of edges $w(u, v)$). We define the weight of edges $w(u, v)$ as the minimum number of sensors that need to be relocated to connect the two vertices. $w(u, v)$ can be calculated as follows:

$$w(u, v) = \left\lceil \frac{d(u, v)}{l_r} \right\rceil, \quad (1)$$

where $d(u, v)$ is the distance between two sensors u and v. l_r is the largest coverage range of a sensor.

Figure 2(a) shows sensors randomly deployed in the two-dimensional belt region. Figure 2(b) shows the weighted barrier graph. In Figure 2(b), each pair of sensors is linked except s and t. $w(D,E) = 0$ presents that sensors D and E intersect with each other. $w(C,D) = 2$ presents that sensors C and D are disjoint from each other, and two sensors should be added to repair the barrier gap.

Lemma 3. *Any path from the original node s to the definition node t on the WBG is a barrier composed of the initially deployed stationary sensors and mobile sensors. The length of the barrier is the number of mobile sensors required to form the barrier coverage.*

Proof. Suppose we choose a path from s to t in WBG and place the exact number of mobile sensors to fill the gaps of the path. Then, the initially deployed sensors are connected to the mobile sensors; therefore, a barrier is formed. The length of the barrier is equivalent to the sum of the weights on the path.

To explain better Lemma 3, the path in Figure 2(b) is used as an example. The length of the path $s \rightarrow C \rightarrow D \rightarrow E \rightarrow t$ in Figure 2(b) is 4, which means that 4 mobile sensors are required to form the barrier along the path. There are three gaps on the path: $s \rightarrow C$, $C \rightarrow D$, and $E \rightarrow t$, which require 1, 2, and 1 mobile sensors to fill, respectively. □

Lemma 4. *The minimum number of mobile sensors required to be moved to form the k sensor-disjoint barriers is equivalent to finding the minimum total length of k-sensors disjoint paths on the WBG.*

Proof. Based on Lemma 3, the length of a barrier is the number of mobile sensors required to form the barrier coverage. Therefore, finding the minimum number of mobile sensors to form the k sensor-disjoint barriers is equivalent to finding the minimum total length of k-sensors disjoint paths on the WBG. □

3.2. Problem Formulation. Based on Lemma 4, after deploying the stationary sensors, the problem to find a minimum number of mobile sensors to form the k sensor-disjoint barriers is transformed to find the minimum total length of the k sensor-disjoint paths on the WBG. Considering a topology in WBG, each link is denoted by $w(u,v)$. Introducing the variable y_{uvl}, let $y_{uvl} = 1$ if $w(u,v)$ is on the lth barrier; otherwise, $y_{uvl} = 0$. The problem can be defined as a constrained optimization problem as follows [48]:

$$\min \quad \sum_{l=1}^{k} \sum_{(u,v)\in E} w(u,v)\, y_{uvl} \tag{2}$$

Subject to for $l = 1, 2, \ldots k$, $\forall u \in V$.

$$\sum_{\{v \mid (u,v)\in E\}} y_{uvl} - \sum_{\{v \mid (v,u)\in E\}} y_{vul} = 1, \quad \text{for } u = s \tag{3}$$

$$\sum_{\{v \mid (u,v)\in E\}} y_{uvl} - \sum_{\{v \mid (v,u)\in E\}} y_{vul} = 0, \quad \forall u \in V \setminus \{s,t\} \tag{4}$$

$$\sum_{\{v \mid (u,v)\in E\}} y_{uvl} - \sum_{\{v \mid (v,u)\in E\}} y_{vul} = -1, \quad \text{for } u = t \tag{5}$$

$$\begin{aligned} & E_i \geq E_{\min}, \quad i \in V \\ & y_{uvl} \in \{0,1\}, \quad (u,v) \in E \end{aligned} \tag{6}$$

The objective function of (2) denotes the minimum number of mobile sensors required to form the sensor-disjoint k-barrier coverage. Every sensor-disjoint barrier from a source node s to a destination node t should satisfy the constraints in (3)–(5). As sensor nodes operate on limited battery power, energy usage is a very important concern in WSNs. When one of the sensors in the barrier is depleted of energy, the barrier will be broken. Therefore, when we choose the stationary sensors to construct the barrier, the energy of each stationary sensor E_i must satisfy the constraint of (6), where $E_{\min}$ denotes the minimum energy at a stationary sensor node battery for it to be operational.

3.3. GSA Algorithm. GSA is a novel stochastic search algorithm developed by Rashedi et al. [36]. In this paper, a new optimization algorithm based on the law of gravity and mass interactions is introduced. In the GSA, agents have been considered as objects whose performance is measured by their masses. All these agents attract every other mass with a force, which is the "gravitational force," and this force causes the agents to be attracted by agents with heavier masses. Specifically, the heaviest agent presents the optimum solution and other agents will be attracted by it. The GSA can be considered in a system with N agents as follows:

$$X_i = \left(X_i^1, \ldots, X_i^d, \ldots, X_i^D\right) \quad \text{for } i = 1, 2, \ldots, N, \tag{7}$$

where X_i^d presents the position of the ith agent in the dth dimension.

At a specific time, the "gravitational force" acts on the ith agent from the jth agent. We can denote this force as follows:

$$F_{ij}^d(t) = G(t)\,\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij} + \varepsilon} \times \left(X_j^d(t) - X_i^d(t)\right), \tag{8}$$

where $G(t)$ is the gravitational constant at the special time t. $R_{ij}(t)$ is the Euclidean distance between agents at time t, and ε is a small constant to ensure that the value of the denominator is not zero. M_{aj} and M_{pi} are the active gravitational mass of the jth agents and the passive gravitational mass of the ith agents, respectively.

Based on [36], in GSA, the mass of each agent is calculated after computing the current population fitness. The equality

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node probability	P_1	P_2	P_3																	

(a) Agent encoding method based on probability

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node probability		29	78	19	33	49	28	65	17	6	9	31	22	11	21	38	36	44	48	52

(b) Probabilities of the nodes of the agent

Figure 3: Agent encoding strategy for barrier coverage problem.

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20_1	20_2	20_3
Node probability		29	78	19	33	49	28	65	17	6	9	31	22	11	21	38	36	44	48	52	36	22

Figure 4: Vertex-splitting technique.

of the gravitational and inertial masses has been assumed by GSA; hence, we can update (9) as follows:

$$M_{ai} = M_{pi} = M_{ii} = M_i$$
$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{n} m_j(t)}, \qquad (9)$$

where $m_i(t)$ is measured as

$$m_i(t) = \frac{\text{fit}(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}. \qquad (10)$$

In (10), fit(t) represents the fitness value of the ith agent at time t. worst(t) and best(t) are given as follows:

$$\text{best}(t) = \min_{j\in\{1,2,\dots,N\}} \text{fit}_j(t) \qquad (11)$$

$$\text{worst}(t) = \max_{j\in\{1,2,\dots,N\}} \text{fit}_j(t). \qquad (12)$$

To compute the acceleration of an agent, the total forces from a set of heavier masses applied on the ith agent should be considered based on a combination of the law of gravity and the second law of Newton [36] on motion:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \qquad (13)$$

where the total force $F_i^d(t)$ applied on the ith agent could be calculated as (14), and rand$_j$ is uniform random in the interval $[0, 1]$.

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} \text{rand}_j \times F_{ij}^d(t). \qquad (14)$$

Furthermore, the next velocity of an agent is calculated as a fraction of its current velocity added to its acceleration (see (15)), where rand$_i$ is a random variable in the interval $[0, 1]$ and can ensure the random characteristic to the GSA.

$$V_i^d(t+1) = \text{rand}_i \times V_i^d(t) + a_i^d(t). \qquad (15)$$

Finally, the agent's position could be updated using

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1). \qquad (16)$$

4. The Proposed Hybrid PSO and GSA Algorithm (PGSA)

GSA is one of the metaheuristic algorithms that search for the global optimum in large-scale networks. However, the k-barrier coverage problem can be formulated as a constrained optimization problem for its constraint and particular structure. Given this issue, it cannot be solved by the GSA for its own limitations, such as easy trapping into local optima and slow convergence. Thus, we introduce a new GSA called PGSA to solve this problem. We discuss the modified steps of the GSA in following sections subsequently.

4.1. Initial Population

4.1.1. Agent Encoding Strategy. The most difficult problem in applying GSA to the k-barrier coverage problem is how to encode a barrier into an agent in GSA. We investigated relevant studies and proposed the agent encoding method based on the probability for the barrier coverage problem as follows. Figure 3 illustrates the agent encoding strategy. Each node in the agent represents a sensor. We assume that 20 sensors of the agent are randomly deployed in the monitoring area, and nodes 1 and 20 are the original node and definition node, respectively. In Figure 3(a), $\{P_1, P_2, P_3, \dots\}$ are the probabilities of the nodes to be chosen to generate the barrier, which are randomly generated in $[1, 100]$. Given that each barrier starts with the initial node, the selection probability is not set for the initial node.

4.1.2. Vertex-Splitting Technique. To construct the k-barrier coverage, we introduce the vertex-splitting technique. We split the destination node t into k splitting node, that is, $\{t_1, t_2, \dots, t_k\}$. Then, we generate the probability value of the splitting t, $\{P_1, P_2, \dots, P_k\}$. As shown in Figure 4, we

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20_1	20_2	20_3
Node probability		29	78	19	33	49	28	65	17	6	9	31	10	11	21	38	30	44	48	52	36	22

(a) Barrier 1: 1 → 3 → 8 → 20_1

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20_1	20_2	20_3
Node probability		29		19	33	49	28		17	6	9	31	10	11	21	38	30	44	48		36	22

(b) Barrier 2: 1 → 6 → 19 → 18 → 16 → 20_2

Node ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20_1	20_2	20_3
Node probability		29		19	33		28		17	6	9	31	10	11	21		30					22

(c) Barrier 3: 1 → 12 → 17 → 2 → 7 → 20_3

Figure 5: Agent k-barrier coverage generation.

split the destination node 20 into 3 splitting nodes, that is, $\{20_1, 20_2, 20_3\}$, and generate the probability value of the splitting node 20, $\{52, 36, 22\}$.

4.1.3. Barrier Coverage Generation Strategy. The barrier is constructed by the node selecting strategy beginning with the source node and terminating at the destination node. Nodes with high probability are selected in turn until reaching the destination. In Figure 3(b), the node with maximum probability, that is, node 3 (probability value is 78), is selected to be added to the barrier. Then, node 3 is removed from Figure 3(b) and node 8 is selected for its highest probability. The steps are repeated until destination node 20 is reached; then, a complete barrier {1 → 3 → 8 → 20} is obtained.

4.1.4. k-Barrier Coverage Generation Strategy. To construct the k-barrier coverage, we first split the destination node t into k splitting node using the vertex-splitting technique in Section 4.1.2. Second, we construct the k-barrier coverage using the barrier coverage generation strategy in Section 4.1.4.

To explain further the k-barrier coverage generation strategy, consider the 3-barrier coverage of the agent in Figure 5 as an example. Using the barrier coverage generation strategy, the node with maximum probability, that is, node 3 (probability value is 78), is selected to be added to barrier 1. Then, node 3 is removed from Figure 5(a) and node 8 is selected for its highest probability. The steps are repeated until the destination node 20_1 is reached. Then, barrier 1 {1 → 3 → 8 → 20_1} is obtained. After constructing barrier 1, node 3, node 8, and node 20_1 are already removed from Figure 5(b). Repeating the steps in constructing barrier 1, the node with maximum probability, that is, node 6 (probability value is 49), is selected to be added to barrier 2 in Figure 5(b). Then, node 6 is removed from Figure 5(b) and node 19 is selected for its highest probability. The steps are repeated until the destination node 20_2 is reached. Then, barrier 2 {1 → 6 → 19 → 18 → 16 → 20_2} is constructed. In the same manner, in Figure 5(c), barrier 3 {1 → 12 → 17 → 2 → 7 → 20_3} is constructed by the node selecting strategy, beginning with the source node 1 and terminating at the destination node 20_3.

4.2. Fitness Function. The length of the k-barrier can be used as the fitness function, where the smaller the total length of the k-barrier coverage, the better the solution. Considering the model of WBG, each link is denoted by $w(u, v)$. Introducing the variable y_{uvl}, let $y_{uvl} = 1$ if $w(u, v)$ is on the lth barrier; otherwise, $y_{uvl} = 0$. The fitness function can be calculated as follows:

$$\text{fit} = \sum_{l=1}^{k} \sum_{(u,v)\in E} w(u, v)\, y_{uvl}. \tag{17}$$

The fitness function (17) denotes that the smaller the fitness value, the less the number of sensors needed to relocate, the better the solution.

4.3. Gravitational Constant. The gravitational constant $G(t)$ is a function of the initial value (G_0) and decreases as time passes. $G(t)$ is formulated as follows:

$$G(t) = G_0\left(e^{-\beta(t/T)}\right), \tag{18}$$

where G_0 and β are constant values, and G_0 and β are set to 100 and 20, respectively. T and t are the current and total number of iterations.

4.4. Updating the Agent's Velocity. In solving complex problems, GSA may be easily trapped into local optima. To enhance the global optimum ability to escape from the local optimal solution, the idea of saving previous local optimum solution and global optimum solution from PSO is adopted into GSA. Our k-barrier coverage problem uses a new set of equations for updating the agent velocities. Therefore, (15) is modified as (19). Where c_1 and c_2 are positive constants, c_1 adjusts the step-size of the particle flying to local optimal position and c_2 adjusts the step-size of the particle flying to global optimum position; $pbest_i$ and $gbest$ represent the best previous position of the ith agent and the best previous position among all particles in the population, respectively. The hybrid PSO and GSA algorithm combine the local search ability of GSA with the social thinking ability of PSO. Through the simulation and comparative performance evaluation in Section 5, we can conclude that the hybrid algorithm successfully escapes from the local optimal solution and obtain a solution close to the global optimum in the large-scale WSNs.

$$\begin{aligned} V_i^d(t+1) = {} & \text{rand}_0 \times V_i^d(t) + c_1 \times \text{rand}_1 \\ & \times \left(pbest_i^d(t) - x_i^d(t) \right) \\ & + c_2 \text{rand}_2 \left(gbest^d(t) - x_i^d(t) \right) + a_i^d(t) . \end{aligned} \tag{19}$$

4.5. Agent Boundary Mutation. GSA is an optimization method based on Newton's second law; therefore, the position of the agent may be beyond the scope of $[x_{\min}, x_{\max}]$. In these circumstances, the GSA will force the agent to be pulled back to its boundaries at $x = x_{\min}$ or $x = x_{\max}$. However, the final optimal results will be skewed. To solve this problem, we introduce the boundary mutation strategy in the PGSA. The mutation strategy can be described as follows:

$$\begin{aligned} \text{if} \quad & x_i \geq x_{\max} \\ & \text{or } x_i \leq x_{\min}, \\ \text{then} \quad & x_i = \text{rand}_i \times (x_{\max} - x_{\min}) + x_{\min}. \end{aligned} \tag{20}$$

After boundary mutation, the agent beyond the boundaries is pulled back to the feasible range of $[x_{\min}, x_{\max}]$. Through the simulation and comparative performance in Section 5, we can prove that the boundary mutation strategy can increase the agent diversity and improve the convergence of the algorithm significantly.

4.6. Termination Condition. In GSA, the maximum number of iterations is a condition of the termination. However, in some cases, GSA could not improve the optimal value. To solve the problem, we employ a termination condition to terminate the algorithm in advance. If the algorithm could not get its optimum for a large number of stages (θ), the algorithm may be subject to early termination. We determine the variable θ by trial and error. If we select min_ineration/10 $\leq \theta \leq$ max_ineration/5, then, the performance of the algorithm is optimal.

4.7. Proposed Algorithm Description. The algorithm of PGSA is described as follows.

Step 1 (initialization). The initial population size is $N = 50$, then the maximum iterate number is max_iteration = 500, and min_iteration = 200. The initial gravitational constant is $G_0 = 100$ and $\beta = 20$. Remove sensor nodes with energy less than $E_{\min}$.

Step 2 (initial population). First, randomly generate a group of agents at a size of 50, with each node in the agent representing the location of a sensor. Randomly generate the sensors probabilities in the range $[1, 100]$. Second, split the destination node t into k splitting node, and assign the probability value to the splitting node of each agent. Third, construct the k-barrier coverage using the strategy in Section 4.1.4.

Step 3 (fitness). Evaluate the fitness for each agent by (17).

Step 4 (constant G). Update the gravitational constant G by (18).

Step 5 (best solution and worst solution). Update the best solution and the worst solution of the population by (11) and (12), respectively.

Step 6 (mass). Calculate the mass for each agent by (6).

Step 7 (acceleration). Calculate the acceleration for each agent by (13).

Step 8 (velocity). Update the velocity for each agent by (19).

Step 9 (position). Update the position for each agent by (16).

Step 10 (termination). If max_iteration or termination condition is reached, then return the solution; otherwise, Steps 3–9 are repeated.

4.8. k-Barrier Coverage with Minimum Cost Problem (KCMC). Through the solution of the PGSA algorithm, we obtain the minimum number and target location of the mobile sensor to be moved to fill the gaps. The KCMC problem mainly involves determining how to move the mobile sensors to the target locations at the minimum cost. The KCMC problem could be formulated as a 0-1 ILP and solved rapidly by commercial quality CPLEX package in [48].

5. Performance Evaluation

In this section, we conduct extensive experiments to evaluate and testify the proposed algorithms (PGSA). The sensors are randomly deployed in a belt region of $L = 100$ m and $W = 20$ m. The largest range of the sensor is $l_r = 2$. The maximum iteration number is 500, the number of agents (N) is 50, and the initial gravitational constant (G_0) is 100. The constant is $\beta = 20$. The algorithm was conducted with MATBLAB 2009 and implemented on a CPU with an Intel Corei9, 3.06 GHz, and 4 GB RAM running on windows 7.

In this paper, PGSA and GSA are compared to verify whether the PGSA can overcome the shortcomings of traditional GSA and escape from a local optimum effectively.

The PGSA is proposed to find the approximate optimal solution in a short time. Therefore, the evaluation mainly focuses on two performance metrics: running time and the optimal value of the PGSA. GSA is one of the latest heuristic optimization algorithms, and it has been proven that GSA has high-quality time performance in solving different optimization problems [37–39]. PGSA is a modified algorithm of GSA; therefore, we compare the average iteration times of PGSA and GSA to verify the time performance of the PGSA. When evaluating the performance of the optimal value, we compare the optimal value of the PGSA with that of RSMN. RSMN is an efficient algorithm for k-barrier coverage based on optimal value [48], and it gets the optimal value but is not suitable for

Table 1: Comparison of statistical results between GSA and PGSA.

Metrics		GSA				PGSA			
		Optimal number of mobile sensors			Average iteration number	Optimal number of mobile sensors			Average iteration number
		Best	Worst	Average		Best	Worst	Average	
$L = 100$ m, $W = 20$ m, $l_r = 20$, $N = 50$, $G_0 = 100$	$n = 200,\ k = 4$	57	98	73	262	40	69	49	115
	$n = 150,\ k = 4$	78	121	94	278	53	82	62	123
	$n = 120,\ k = 3$	45	83	65	235	34	62	45	98
	$n = 100,\ k = 3$	42	79	58	251	29	58	38	105
	$n = 80,\ k = 2$	44	76	59	217	30	56	39	94
	$n = 60,\ k = 2$	48	84	63	223	32	62	44	107

Table 2: Comparison of statistical results on different number of barriers (k).

Metrics	Barrier number	RSMN		PGSA	
		Optimal number of mobile sensors	Running time (s)	Optimal number of mobile sensors	Running time (s)
$L = 100$ m $W = 20$ m $n = 100$ $l_r = 2$ $N = 50$ $G_0 = 100$	$k = 2$	26	4.84	26	2.08
	$k = 3$	29	38.18	29	2.12
	$k = 4$	37	69.36	41	2.14
	$k = 5$	65	112.44	72	2.33
	$k = 6$	79	171.2	88	2.76

large-scale networks. Therefore, we evaluate the performance of the optimal value by comparing two algorithms.

The comparison results of GSA and PGSA are given in Table 1. We generate six random datasets with 200, 150, 120, 100, 80, and 60 nodes and 4, 3, and 2 barriers. For all six test datasets, the best, the worst, and average solutions are gained among 20 independent runs. In addition, the average iteration number was also reported.

From the numerical results, PGSA performs better with respect to the best, the worst, the average solutions, and average iteration number compared to GSA. The results show that PGSA can explore the search space more effectively and overcome the premature convergence.

Figure 6 plots the comparison of convergence curve with $n = 120$ and $k = 3$ between PGSA and GSA. GSA converges to optimal solution 68 at iteration 230, while PGSA converges to 37 at iteration 92. The result shows superior performance of PGSA over that of GSA. The average iteration number of PGSA is less than that of the GSA algorithm, and the time performance of the algorithm is greatly improved.

To verify further the performance of PGSA, we compare it with RSMN algorithm. In the RSMN algorithm, the k-barrier coverage problem was formulated as a 0-1 ILP, and then the commercial software, such as CPLEX or Lingo, was used to solve the problem directly.

Table 2 and Figure 7 show the performance of PGSA and RSMN on a different number of barriers. Figure 7(a) shows the comparison of the average running time of the two algorithms. With the increase of the barriers, the average running time of the two algorithms also increases. Figure 7(a) shows that the PGSA algorithm increases gradually and smoothly. However, the RSMN algorithm increases remarkably. The numerical results show that PGSA has better time performance than the RSMN algorithm with the increase of the barrier number. Figure 7(b) shows a comparison of the optimal values obtained by the two algorithms. The results show that some optimal values are the same, whereas others are just close. The PGSA algorithm obtains the near-optimal solution in a very short amount of time and, in some cases, obtains the optimal solution.

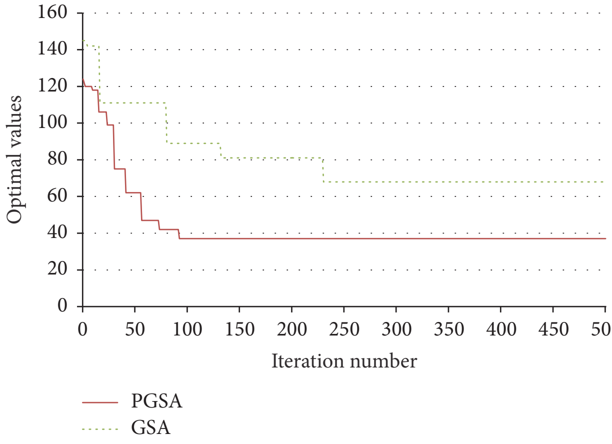
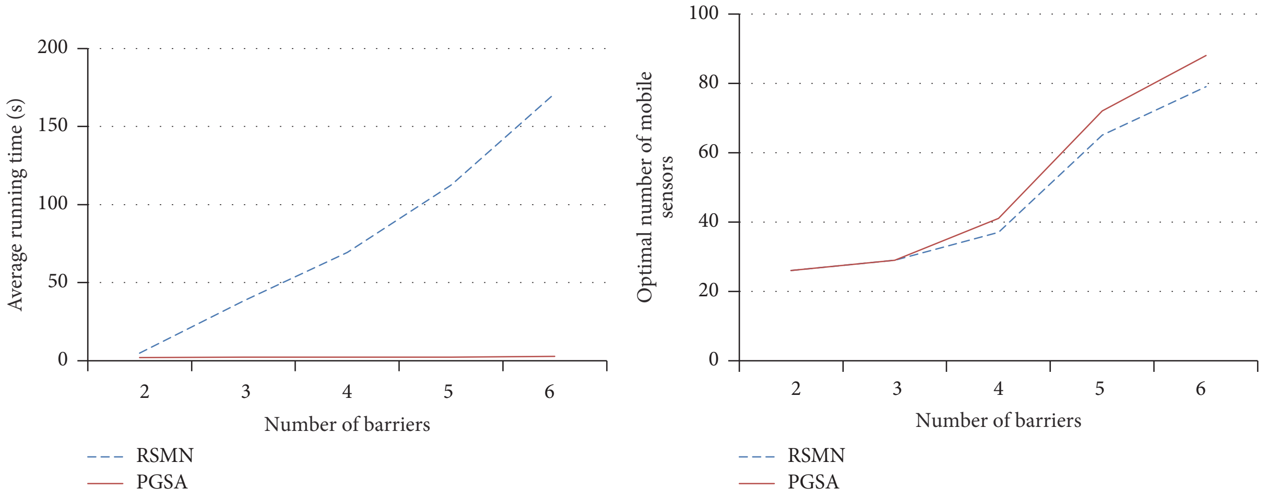


Figure 6: Comparison of convergence curve between GSA and PGSA ($n = 120$, $k = 3$).

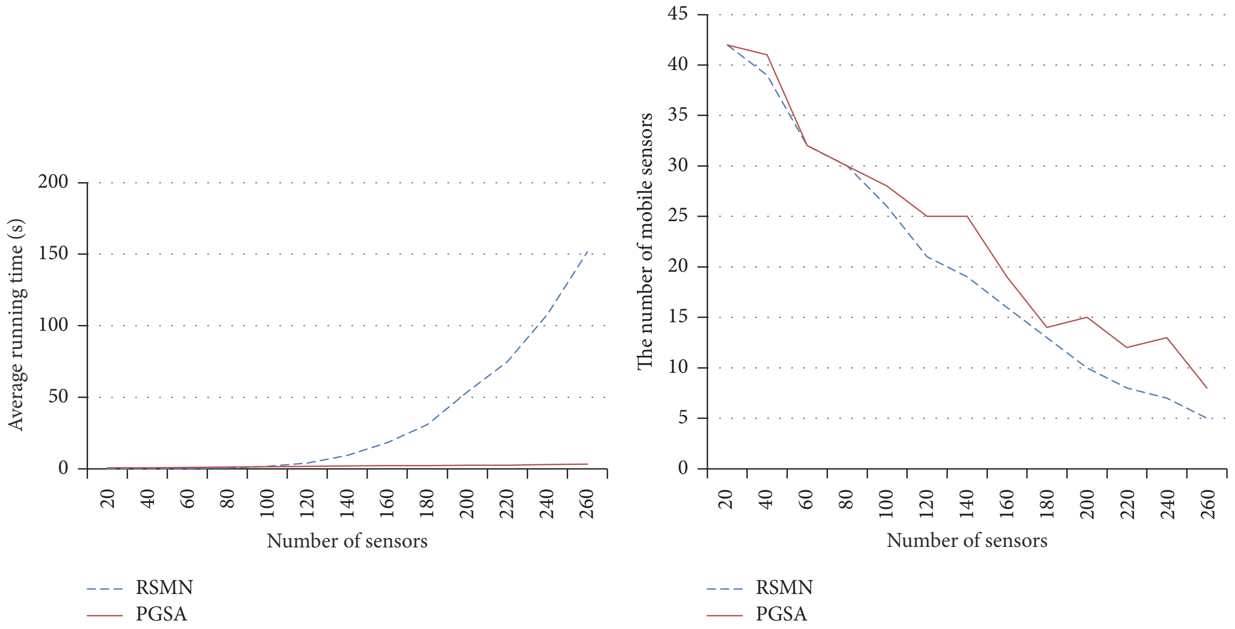
Figure 8 and Table 3 show the optimal value and the average running time against the number of sensor nodes. In



(a) Average running time

(b) Optimal number of mobile sensors

Figure 7: Comparison of average running time and optimal number of mobile sensors.



(a) Average running time

(b) Optimal number of mobile sensors

Figure 8: Comparison of average running time and optimal number of mobile sensors on different number of sensors.

Figure 8(a), we compare the average running time of the two algorithms. The average running time for PGSA is worse when the number of sensors is less than 80. However, with the increase of scale, PGSA performs better than RSMN. With an increase in the number of nodes, the average running time of both algorithms also increases. However, the increase rate of PGSA is far less than RSMN. The average running time of PGSA was not significantly affected by the number of sensor nodes. Figure 8(b) shows the comparison of the optimal values obtained by the two algorithms. The results show that the optimal values of the two algorithms are the same in some cases and close in others.

6. Conclusion

In this paper, we investigated the GSA measures for constructing the k-barrier coverage, which is aimed to detect intruders attempting to cross the regions of interest with at least k distinct sensors. By proposing a new method based on GSA, many problems of the previous studies, such as finding

Table 3: Comparison of statistical results on different number of sensors (n).

Metrics	Sensor number	RSMN		PGSA	
		Optimal number of mobile sensors	Running time (s)	Optimal number of mobile sensors	Running time (s)
$L = 100$ m $W = 20$ m $k = 2$ $l_r = 2$ $N = 50$ $G_0 = 100$	$n = 20$	42	0.042	42	0.778
	$n = 40$	39	0.08	41	0.907
	$n = 60$	32	0.289	32	1.098
	$n = 80$	30	0.51	30	1.29
	$n = 100$	26	1.887	28	1.569
	$n = 120$	21	4.009	25	1.761
	$n = 140$	19	9.557	25	2.157
	$n = 160$	16	18.458	19	2.263
	$n = 180$	13	31.089	14	2.362
	$n = 200$	10	53.886	15	2.639
	$n = 220$	8	74.892	12	2.667
	$n = 240$	7	108.045	13	2.991
	$n = 260$	5	151.594	8	3.31

an inaccurate solution or the unreasonably low speed, are solved. Finally, we obtained a solution close to the global optimum in a short period and successfully applied the GSA algorithm to the WSNs.

Conflicts of Interest

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

Acknowledgments

This work is supported by the NSFC (U1536206, 61232016, U1405254, 61373133, 61502242, 61672294, 61602253, and 71401176), BK20150925, KYLX16_0926, and PAPD fund.

References

[1] S. Jeschke, C. Brecher, S. H. Song et al., *Industrial Internet of Things: Foundations, Principles and Applications*, Springer, Cham, Switzerland, 2017.

[2] M. P. Durišić, Z. Tafa, G. Dimić, and V. Milutinović, "A survey of military applications of wireless sensor networks," in *Proceedings of the 1st Mediterranean Conference on Embedded Computing (MECO '12)*, pp. 196–199, June 2012.

[3] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, "Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 63–72, 2014.

[4] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, "A framework for use of wireless sensor networks in forest fire detection and monitoring," *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 614–625, 2012.

[5] X. Sun, S. Yan, B. Wang, L. Xia, Q. Liu, and H. Zhang, "Air temperature error correction based on solar radiation in an economical meteorologicalwireless sensor network," *Sensors*, vol. 15, no. 8, pp. 18114–18139, 2015.

[6] Z. F. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. 98, no. 1, pp. 190–200, 2015.

[7] B. Wang, "Coverage problems in sensor networks: a survey," *ACM Computing Surveys*, vol. 43, no. 4, article 32, 2011.

[8] D. G. Costa and L. A. Guedes, "The coverage problem in video-based wireless sensor networks: a survey," *Sensors*, vol. 10, no. 9, pp. 8215–8247, 2010.

[9] Q. Zhao and M. Gurusamy, "Lifetime maximization for connected target coverage in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1378–1391, 2008.

[10] J.-M. Gil and Y.-H. Han, "A target coverage scheduling scheme based on genetic algorithms in directional sensor networks," *Sensors*, vol. 11, no. 2, pp. 1888–1906, 2011.

[11] K.-P. Shih, H.-C. Chen, C.-M. Chou, and B.-J. Liu, "On target coverage in wireless heterogeneous sensor networks with multiple sensing units," *Journal of Network and Computer Applications*, vol. 32, no. 4, pp. 866–877, 2009.

[12] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2014.

[13] M. Naderan, M. Dehghan, and H. Pedram, "Sensing task assignment via sensor selection for maximum target coverage in WSNs," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 262–273, 2013.

[14] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing movement for target coverage and network connectivity in mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1971–1983, 2015.

[15] Z. Lu, W. W. Li, and M. Pan, "Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 714–727, 2015.

[16] V. Zalyubovskiy, A. Erzin, S. Astrakov et al., "Energy-efficient area coverage by sensors with adjustable ranges," *Sensors*, vol. 9, no. 4, pp. 2446–2460, 2009.

[17] J. A. Torkestani, "An adaptive energy-efficient area coverage algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1655–1666, 2013.

[18] S. Rizvi, H. K. Qureshi, S. A. Khayam, V. Rakocevic, and M. Rajarajan, "A1: an energy efficient topology control algorithm for connected area coverage in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 597–605, 2012.

[19] S. Sengupta, S. Das, M. D. Nasir, and B. K. Panigrahi, "Multi-objective node deployment in WSNs: in search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 405–416, 2013.

[20] Y.-G. Fu, J. Zhou, and L. Deng, "Surveillance of a 2D plane area with 3D deployed cameras," *Sensors*, vol. 14, no. 2, pp. 1988–2011, 2014.

[21] M. Vecchio and R. López-Valcarce, "Improving area coverage of wireless sensor networks via controllable mobile nodes: a greedy approach," *Journal of Network and Computer Applications*, vol. 48, pp. 1–13, 2015.

[22] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, "Energy-efficient probabilistic area coverage in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 367–377, 2015.

[23] H. Zhu and Y. Shi, "Brain storm optimization algorithm for full area coverage of wireless sensor networks," in *Proceedings of the 8th International Conference on Advanced Computational Intelligence (ICACI '16)*, pp. 14–20, February 2016.

[24] S. He, J. Chen, X. Li, X. S. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1268–1282, 2014.

[25] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Weng, "Achieving k-barrier coverage in hybrid directional sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1443–1455, 2014.

[26] S. Dobrev, S. Durocher, M. Eftekhari et al., "Complexity of barrier coverage with relocatable sensors in the plane," *Theoretical Computer Science*, vol. 579, pp. 64–73, 2015.

[27] D. Tao and T.-Y. Wu, "A survey on barrier coverage problem in directional sensor networks," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 876–885, 2015.

[28] Y. Wu and M. Cardei, "Distributed algorithms for barrier coverage via sensor rotation in wireless sensor networks," *Journal of Combinatorial Optimization*, pp. 1–22, 2016.

[29] X. Gong, J. Zhang, D. Cochran, and K. Xing, "Optimal Placement for Barrier Coverage in Bistatic Radar Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 259–271, 2016.

[30] B. Wang, J. Chen, W. Liu, and L. Yang, "Minimum cost placement of bistatic radar sensors for belt barrier coverage," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 577–588, 2016.

[31] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 284–298, ACM, Cologne, Germany, September 2005.

[32] H. M. Ammari and S. K. Das, "Fault tolerance measures for large-scale wireless sensor networks," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 1, article 2, 2009.

[33] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[34] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[35] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[36] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[37] S. R. Balachandar and K. Kannan, "A meta-heuristic algorithm for set covering problem based on gravity," *International Journal of Computational and Mathematical Sciences*, vol. 4, no. 5, pp. 223–228, 2010.

[38] S. Duman, U. Güvenç, and N. Yörükeren, "Gravitational search algorithm for economic dispatch with valve-point effects," *International Review of Electrical Engineering*, vol. 5, no. 6, pp. 2890–2895, 2010.

[39] H. R. Hassanzadeh and M. Rouhani, "A multi-objective gravitational search algorithm," in *Proceedings of the 2nd International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN '10)*, pp. 7–12, IEEE, Liverpool, UK, July 2010.

[40] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 411–419, ACM, May 2008.

[41] S. Bereg and D. Kirkpatrick, "Approximating barrier resilience in wireless sensor networks," in *Proceedings of the International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pp. 29–40, Springer, Wroclaw, Poland, 2009.

[42] L. Zhang, J. Tang, and W. Zhang, "Strong barrier coverage with directional sensors," in *Proceedings of the Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, IEEE, Honolulu, Hawaii, USA, 2009.

[43] Z. Sun, Y. Zhang, Y. Nie, W. Wei, J. Lloret, and H. Song, "CASMOC: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks," *Wireless Networks*, pp. 1–22, 2016.

[44] Z. Sun, Y. Shu, X. Xing et al., "LPOCS: A Novel Linear Programming Optimization Coverage Scheme in Wireless Sensor Networks," *Adhoc & Sensor Wireless Networks*, vol. 33, 2016.

[45] S. Silvestri and K. Goss, "MobiBar: an autonomous deployment algorithm for barrier coverage with mobile sensors," *Ad Hoc Networks*, vol. 54, pp. 111–129, 2017.

[46] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage with line-based deployed mobile sensors," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1381–1391, 2013.

[47] J. Du, K. Wang, H. Liu, and D. Guo, "Maximizing the lifetime of k-discrete barrier coverage using mobile sensors," *IEEE Sensors Journal*, vol. 13, no. 12, pp. 4690–4701, 2013.

[48] Y. Zhang, X. Sun, and B. Wang, "Efficient algorithm for k-barrier coverage based on integer linear programming," *China Communications*, vol. 13, no. 7, pp. 16–23, 2016.

[49] B. Gu, X. Sun, and V. S. Sheng, "Structural minimax probability machine," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2016.

[50] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Computing*, pp. 1–18, 2017.

[51] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[52] B. Gu and V. S. Sheng, "A robust regularization path algorithm for ν-support vector classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–8, 2016.

[53] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," *Computers & Electrical Engineering*, vol. 56, pp. 544–556, 2016.

[54] H. Mostafaei, "Stochastic barrier coverage in wireless sensor networks based on distributed learning automata," *Computer Communications*, vol. 55, pp. 51–61, 2015.

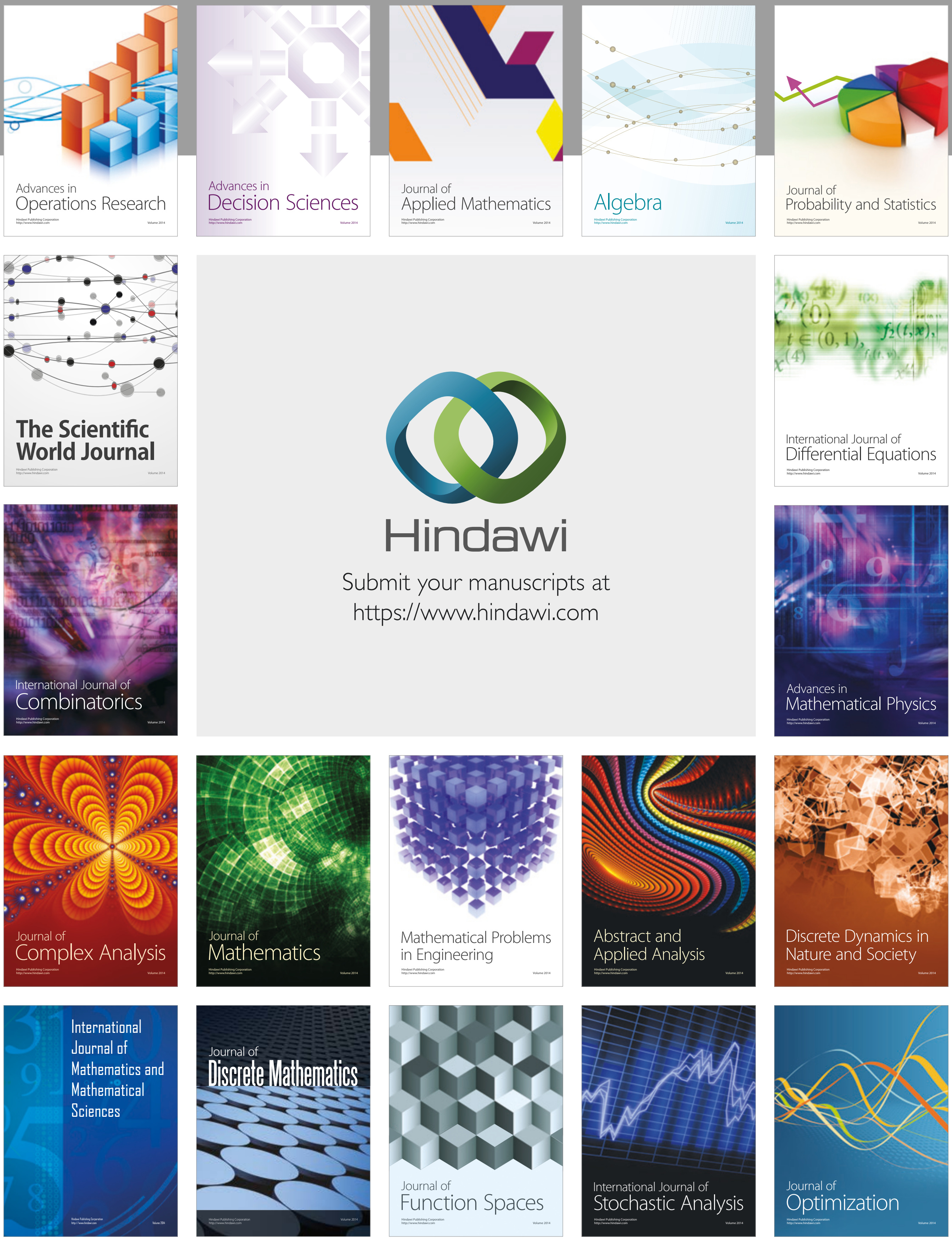
[55] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[56] W.-H. Liao, Y. Kao, and R.-T. Wu, "Ant colony optimization based sensor deployment protocol for wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6599–6605, 2011.

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

[58] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding," *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 166–176, 2015.

[59] I. Maleki, S. R. Khaze, M. M. Tabrizi et al., "A new approach for area coverage problem in Wireless Sensor Networks with hybrid particle swarm optimization and differential evolution algorithms," *International Journal of Mobile Network Communications & Telematics (IJMNCT)*, vol. 3, 2013.


Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization