

## Research Article

# Optimal Mobile Sensor Scheduling for a Guaranteed Coverage Ratio in Hybrid Wireless Sensor Networks

Zhixin Fu<sup>1,2</sup> and Keyou You<sup>3</sup>

<sup>1</sup> College of Energy and Electrical Engineering, Hohai University, Nanjing, Jiangsu 211100, China

<sup>2</sup> Research Center for Renewable Energy Generation Engineering of Ministry of Education, Hohai University, Nanjing, Jiangsu 210098, China

<sup>3</sup> Department of Automation, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Zhixin Fu; zhixinfu@hhu.edu.cn

Received 17 July 2012; Revised 1 April 2013; Accepted 7 April 2013

Academic Editor: Donggang Liu

Copyright © 2013 Z. Fu and K. You. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an optimal mobile sensor-scheduling algorithm for recovering the failure sensors in hybrid wireless sensor networks (WSNs). To maintain a guaranteed coverage over the area of interest, spare mobile sensors in WSNs will be activated to replace the failure sensors. The optimal scheduling problem is formulated into two optimization problems, one of which precisely determines the minimum value of the largest distance required to travel for mobile sensors, while the other one gives the optimal dispatch for mobile sensors to minimize the total travel distance. Furthermore, a distributed suboptimal scheduling, which only requires the local matching information of mobile sensors, is developed as well. Both regular and random network topologies are provided to illustrate the proposed algorithms in the simulation.

## 1. Introduction

Wireless sensor networks (WSNs) have received intensive research interest in recent years due to their potential applications as computing platforms for monitoring and controlling various environments [1]. Hybrid WSNs consisting of static and mobile sensors are more robust and fault tolerant to the changing environment conditions than conventional WSNs which only contain static resource constrained sensors [2, 3]. By incorporating mobile sensors to WSNs, the network capability can be further improved in many aspects, such as automatic sensor deployment [2–12], dynamic event coverage [13, 14], mobiles target tracking [15–17], flexible topology adjustment [18, 19], and efficient data collection and processing [20–22].

The coverage ability is one of the most important applications of WSNs, which is closely related to the sensor deployment issue. There are two main factors to affect the coverage performance. One is inappropriate sensor placement at the initial deployment of the network and the other is sensor

failure due to energy depletion or hostile attack during the operation of the network. Observing that mobile sensors can physically react to the events of the environment, support self-configuration mechanisms, and guarantee adaptability and optimal performance, they are utilized to improve the coverage performance if coverage holes are generated at the initial deployment or in the running course due to the failed sensors. Considering the previously mentioned, how to schedule spare mobile sensors to obtain an optimal coverage performance has gained considerable attention, see: for example, [2–19] and references therein.

Solutions to the former issue can be divided into two types in conformity with different methods of sensor deployment. One is to scatter all sensors in the area of interest. For example, [5, 6, 11, 12] discuss how scattered sensors interact to achieve the optimal coverage through rules of the virtual force algorithm (VFA). In [7], Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms are adopted to solve Minimum Weight Sensor Cover Problem (MWSCP). The other issue can be solved by deploying

spare mobile sensors to designated locations. A natural question is how to schedule mobile sensors to the designated locations with the minimum moving distance. For example, [8] compares different dispatch methods for mobile sensors, such as the Voronoi-based (VOR) method, the mini-max method, the grid-quorum and cascade movement method, and the k-coverage method to achieve multilevel coverage of the area of interest. In [9], the problems of sensor placement and sensor dispatch are addressed in the area of interest for an arbitrary polygon shape with the consideration of possible existence of obstacles. [13, 14] discuss the coverage problem of changing events by mobile sensors. Concretely, the motion energy consumption for mobile sensors is considered in [13]. It analyzes the expected information captured Per unit of Energy consumption (IPE) as a function of the event type, the event dynamics, and the speed of the mobile sensor. In [14], it is concerned with the coordinated deployment of a high density of randomly placed sensors in a geographical area for event monitoring. The sensors implement a coordinated sleep algorithm to eliminate the redundancy of coverage while balancing their energy use.

Although efficient schemes can be applied to obtain the optimal coverage performance at the initial deployment, detrimental factors causing sensor failures are inevitable in the course of running and may degrade the coverage performance. In the case of sensor failures, the coverage problem is to schedule spare mobile sensors to maintain a guaranteed coverage performance with minimum cost. Related works can be found in [8, 10]. In these papers, the network is composed of static and mobile sensors, where static sensors are used for sensing, while mobile sensors can move to event locations to conduct more advanced tasks. Both papers transform the sensor dispatch problem to a maximum-matching problem in a weighted bipartite graph. However, they do not consider the coverage performance of the network. Inspired by [10] but also motivated by its limitation, we limit the maximum moving distance for each individual mobile sensor since for homogeneous mobile sensors, it is better to reduce the maximum energy consumption of a single sensor to prolong its lifespan though it may induce a larger overall energy consumption of all mobile sensors.

In our configuration, the network is composed of static and mobile sensors. Static sensors are failure prone and deployed for sensing, while mobile sensors are considered as mobile robots and can keep sleeping until activated and dispatched to recover failed sensors, for example, charge failed sensors or replace them with new sensors. Note that the sleeping sensors are assumed to be capable of receiving necessary activation information.

While most of the existing works are to achieve the maximum network coverage capability, the maintenance of full coverage over the monitored area is extremely expensive. Bearing in mind that the coverage performance of the network still can fulfill the application requirement by allowing a certain number of failed sensors, we first characterize the relationship between the number of failed sensors and the coverage ratio. If the number of failed sensors is detected to exceed a critical value, mobile sensors will be dispatched to locations of failed sensors. Thus, we explicitly derive the

minimum value of  $d$  which is the largest distance moved by all the dispatched sensors and give an optimal sensor dispatch strategy using the knowledge of the minimum number of failed sensors required to be recovered, which is generally motivated by the application requirement for the coverage ratio. Noting that the optimal scheduling requires a centralized decision center and that it may be impractical in some network configurations, a distributed suboptimal scheduling, which only requires the local matching information of mobile sensors, is developed as well.

The rest of the paper is organized as follows. The problem of interest is formulated in Section 2. In Section 3, we give the centralized optimal sensor scheduling by solving two optimization problems. A distributed suboptimal scheduling is discussed at the end of this section. The simulation results are shown in Section 4. Concluding remarks are drawn in Section 5.

## 2. Problem Formulation

The sensing field considered in this paper is described in Figure 1, where static sensors (SSs) denoted as “•” are deployed in the area of interest to collect information from the physical environment. Due to unforeseen factors, some SSs may break down and the coverage ability may significantly decrease. The coverage performance is measured by the ratio of the area covered by active sensing SSs to the monitoring area of interest. To maintain a guaranteed coverage ratio for an application requirement, part (even all) of the failed SSs needs to be recovered by spare mobile sensors denoted as “o”, which are randomly deployed in the area. We shall refer to this type of sensors as backup sensors (BSs), which is designed in the sleeping mode until activated to save energy.

*Remark 1.* Although we describe the problem with a special regular network topology in Figure 1, it is worth mentioning that the result in this paper is still applicable to networks with any topology, which will be verified by simulation results in Section 4. Of course, the computational load for an arbitrary topology is generally higher than that of a regular network topology.

The problem setup assumes that there is a centralized fusion center (FC) to process information sent from SSs. The FC generally has abundant resource, that is, sufficient energy and computing power. Another assumption is that the FC can access the global information, including locations of sensors, the state of sensors (active, sleep, or failed), and the online coverage ratio. Once the FC detects that the coverage ratio falls below the guaranteed value, it will activate a necessary number of BSs to recover the failed SSs. The FC will be removed in our distributed scheduling.

Roughly, we can assume that each SS equally contributes to the coverage ratio since the sensing range of the homogeneous SSs is the same in our case. Thus, the first question is how many failed SSs are required to be recovered to maintain a guaranteed coverage ratio. This is automatically determined by the FC.

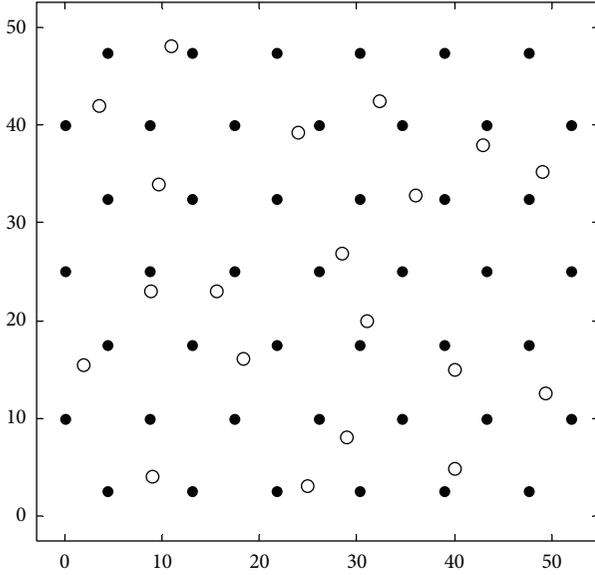


FIGURE 1: Network configuration.

Noticing that each BS usually has limited onboard battery power, it is important to limit its moving distance, which will reduce the number of the failed SSs to be successfully recovered. This raises the first problem: what is the maximum moving distance of BSs in order to recover the minimum number of the failed SSs? Moreover, how to schedule the activated BSs to recover the failed SSs? The main contribution of this paper is to propose algorithms to solve the previous problems.

In the rest of this section, we shall formulate the previous problems into two optimization problems and their solutions will be given in Section 3.

Denote the set of the failed SSs at the time  $t$  by  $K$  and the cardinality by  $|K|$ . Also, denote the set of BSs by  $M$ . For simplicity of presentation and without loss of generality, we set  $|M| = |K| := n$ . To maintain the guaranteed coverage ratio, at least  $q \leq n$  failed SSs need to be recovered. Let  $d$  be the maximum moving distance of the activated BSs.

The first optimization (P1) precisely determines the minimum  $d$ , denoted by  $d^*$ , which is required to travel to recover at least  $q$  failed SSs. Namely, we consider the following optimization:

$$d^* = \min_X d, \quad (1)$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^n x_{ij} = 1, & j = 1, \dots, n; \\ \sum_{j=1}^n x_{ij} = 1, & i = 1, \dots, n; \\ \sum_{ij} x_{ij} 1_{\{w_{ij} \leq d\}} \geq q; \\ x_{ij} = 0 \text{ or } 1, & i, j = 1, \dots, n. \end{cases} \quad (2)$$

Here  $X = (x_{ij}) \in \mathbb{R}^{n \times n}$  is an indicator matrix and  $x_{ij} = 1$  if there is a matching between the  $i$ th BS and the  $j$ th failed SS.  $w_{ij}$  is the corresponding distance. If  $w_{ij} \leq d^*$  and  $x_{ij} = 1$ , the  $i$ th BS will be activated and moves to the location of the  $j$ th failed SS. The requirement that at least  $q$  failed SSs should be recovered to maintain the guaranteed coverage ratio is addressed in the third condition of (2).

*Remark 2.* Under the condition that BSs can move at a constant velocity and the velocity of each BS is the same, the previous problem can be interpreted as the minimization of the recovery time which may improve the responsiveness of the network to dynamic environments. Also, it is of great importance to reduce the maximum energy consumption of a single BS to prolong the lifetime of the network.

The second optimization (P2) is to find the optimal sensor scheduling strategy to minimize the total moving distances of BSs based on  $d^*$ , which is closely related to the total energy consumption of all BSs. More specifically, suppose that  $d^*$  has been obtained in the optimization (P1), the optimization (P2) is given by

$$X^* = \arg \min_X \sum_{1 \leq i, j \leq n} x_{ij} w_{ij} 1_{\{w_{ij} \leq d^*\}}, \quad (3)$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^n x_{ij} = 1, & j = 1, \dots, n; \\ \sum_{j=1}^n x_{ij} = 1, & i = 1, \dots, n; \\ x_{ij} = 0 \text{ or } 1, & i, j = 1, \dots, n, \end{cases} \quad (4)$$

where the standard indicator function  $1_{\{w_{ij} \leq d^*\}}$  is defined as  $1_{\{w_{ij} \leq d^*\}} = 1$  if  $w_{ij} \leq d^*$ , otherwise 0.

In the previous two optimization problems, the first one involves the minimization of the energy of a single BS, while the second one is to minimize the total energy consumption of all BSs during the recovering process.

### 3. Optimal Sensor Scheduling

To maintain a guaranteed network coverage ratio, some spare BSs should be relocated to recover a necessary number of failed SSs. The optimal sensor scheduling problem will be studied according to the following steps. We will first investigate the relationship between the coverage ratio and the number of failed SSs, which can be readily used to determine the number of recovered sensors under different requirements of the coverage ratio  $c_r$ . Then, an algorithm to solve the optimization (P1) will be proposed to evaluate the minimum moving distance of BSs,  $d^*$ , in (P1) if there are  $q$  failed SSs required to be recovered. Finally, the optimal scheduling strategy is found by solving the optimization (P2).

*3.1. Coverage Ratio.* The coverage ability measured by coverage ratio is one of the most important performances of WSNs. Specifically, the coverage ratio is defined as the ratio of

covered area of active sensing SSs to the whole area of interest, that is,

$$c_r = 1 - \frac{c_l}{a}, \quad (5)$$

where  $c_l$  is the total area of losing coverage due to all failure SSs and  $a$  is the total area of interest. For example, the relationship between the coverage ratio and the number of failed SSs in Figure 1 can be clearly established and shown in Figure 2, which gives a guideline to determine the number of failed SSs to be recovered to maintain a certain coverage ratio.

**3.2. Heading Scheduling Algorithm.** Before proceeding further, we first provide the solution to optimization (P1).

Define  $d_{\min} = \min_{1 \leq i, j \leq n} w_{ij}$  and  $d_{\max} = \max_{1 \leq i, j \leq n} w_{ij}$ . Given an arbitrary  $d \in [d_{\min}, d_{\max}]$ , if we restrict the maximum moving distance of BSs to  $d$ , we need to determine the maximum number of failed sensors that can be recovered. In view of the solution to the assignment problem in operation research [23], it can be easily solved. More precisely, it can be formulated as a simple version of the assignment problem (AP):

$$q(d) = \max \sum_{1 \leq i, j \leq n} x_{ij} 1_{\{w_{ij} \leq d\}},$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^n x_{ij} = 1, & j = 1, \dots, n, \\ \sum_{j=1}^n x_{ij} = 1, & i = 1, \dots, n, \\ x_{ij} = 0 \text{ or } 1, & i, j = 1, \dots, n. \end{cases} \quad (6)$$

To get the optimal  $d^*$  in P1, we reorder the  $w_{ij}$  in an increasing order. For simplicity of presentation, assume that  $w_{i_1 j_1} \leq w_{i_2 j_2}$  if  $1 \leq i_1 \leq i_2 \leq n$  and for all  $j_1, j_2 \in \{1, \dots, n\}$  while  $w_{ij_1} \leq w_{ij_2}$  if  $1 \leq j_1 \leq j_2 \leq n$  and for all  $i \in \{1, \dots, n\}$ , that is,  $w_{11} \leq \dots \leq w_{1n} \leq w_{21} \leq \dots \leq w_{nn}$ . Suppose that it is required to recover  $q$  SSs; then we choose an initial  $d = w_{1j}$  satisfying that  $q = j$ , where  $1 \leq j \leq n$  and get a  $q(d)$  by solving an AP. Obviously,  $q(d) \leq q$ . If  $q(d)$  is strictly less than the number of failed SSs to be recovered, we enlarge  $d$  step by step, that is,  $d = w_{1(j+1)}$  and continue to compute a new  $q(d)$  by solving an AP. Repeating the previous process in a finite time, there must exist  $d^* = w_{i^* j^*}$  such that  $q(d^*)$  is the exact minimum number of failed SSs required to be recovered, that is,  $q(d^*) = q$ . In the previously mentioned, if  $j = n$ , then  $w_{i(j+1)} := w_{(i+1)1}$ . Thus, the  $d^*$  in optimization (P1) can be efficiently derived; see Figure 3 for illustration where the block "initialization" decides the minimum number ( $q$ ) of failed SSs to be recovered and "AP" represents solving a simple version of the assignment problem in (6).

Concerning with the second optimization (P2), the authors in [10] transform the dispatch problem into maximum-matching problem and solve it by using a message passing algorithm [24]. Different from [10], the sensor dispatch problem in our work is a minimum matching problem. Further, the objective function in the optimization

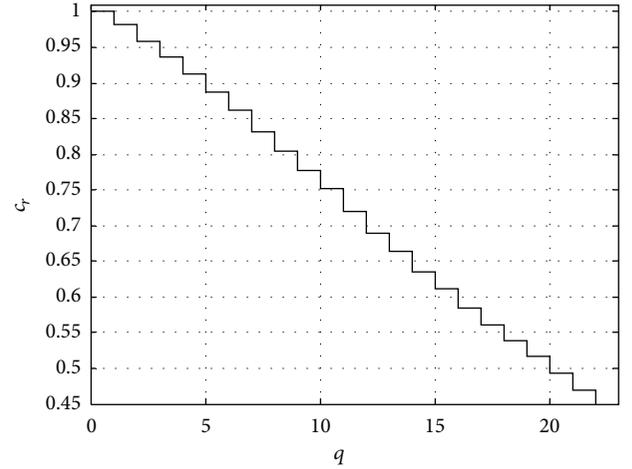


FIGURE 2: The coverage ratio  $c_r$  versus the number of failed sensors  $q$ .

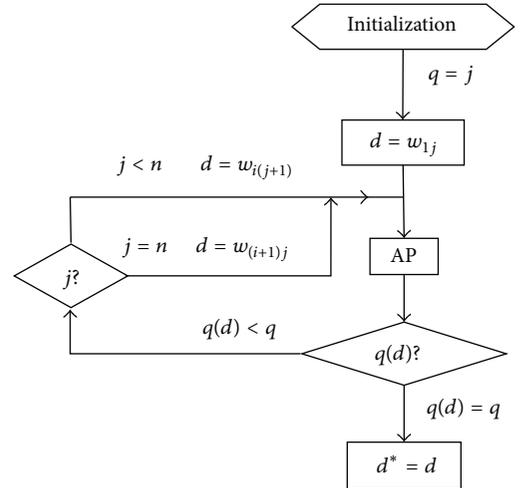


FIGURE 3: The computation flow.

(P2) only deals with the distance that is less than  $d^*$ . As such, our method is to modify the message passing algorithm by letting

$$w'_{ij} = \begin{cases} -w_{ij}, & \text{if } w_{ij} \leq d^* \\ -d^* - 1, & \text{otherwise.} \end{cases} \quad (7)$$

The rationale lies in the fact that if  $w_{ij} > d^*$ , the link  $x_{ij} = 1$  is not applicable since we limit the moving distance of BS by  $d^*$ . Thus, we let the corresponding cost  $w_{ij}$  be a sufficient large number, for example,  $w_{ij} = d^* + 1$ , that is,  $w'_{ij} = -d^* - 1$ . For completeness, we review the maximum weighted matching algorithm mentioned later. Note that [24] proves that this algorithm converges to the optimal solution after  $k > 3nw^*/\epsilon$ , where  $w^* = \max\{|w'_{ij}|\}$  and  $\epsilon$  is the difference between the maximum weight matching and the second largest weight matching.

(i) Initialization:

$$\begin{aligned} m_{T_i \rightarrow x_{ij}}^{(0)} &= -\max_{m \neq i} w'_{mj}, \\ m_{B_j \rightarrow x_{ij}}^{(0)} &= -\max_{l \neq j} w'_{il}. \end{aligned} \quad (8)$$

(ii) At  $k$ th iteration:

$$\begin{aligned} m_{T_i \rightarrow x_{ij}}^{(k)} &= -\max_{m \neq j} \{m_{B_m \rightarrow x_{im}}^{(k-1)} + w'_{im}\}, \\ m_{B_j \rightarrow x_{ij}}^{(k)} &= -\max_{l \neq i} \{m_{T_l \rightarrow x_{lj}}^{(k-1)} + w'_{lj}\}. \end{aligned} \quad (9)$$

(iii) At  $k$ th iteration, we compute:

$$M_{ij}^{(k)} = m_{T_i \rightarrow x_{ij}}^{(k)} + m_{B_j \rightarrow x_{ij}}^{(k)} + w'_{ij}. \quad (10)$$

Estimation of  $x_{ij}$  based on  $k$ th iteration is  $\hat{x}_{ij}^{(k)} = 1$  if  $M_{ij}^{(k)} > 0$  and  $\hat{x}_{ij}^{(k)} = 0$ , otherwise.

To the moment, the two optimization problems formulated in Section 2 can be solved, suggesting that our optimal sensor scheduling is feasible.

**3.3. Discussion on Distributed Algorithm.** Adopting the previous scheduling algorithm, we can get the optimal dispatch for BSs under the constraint that the minimum value of the largest moving distance is limited to  $d^*$ . However, they may be inapplicable to some network configurations since the optimal scheduling is generated by using the global information and requires that the FC should be significantly powerful to process an extensive amount of sensing data. It may be impractical or infeasible to access the global information in a large network. As such, a distributed suboptimal algorithm based on the local matching information of BSs is proposed at the risk of losing the global optimality. In this case, BSs are considered as local FCs to collect the failure information of all failed SSs. They will match and recover the failed SSs on their own without the involvement of the FC. To save energy for BSs to move to locations of failed SSs, it is critical to limit their transmission power which in turn limits their communication range. Thus, BSs can only communicate with their neighbor BSs.

In the sequel, we define the neighbor BSs of  $u$ th BS by  $S_u$  as follows:

$$S_u = \{v \in M \mid D(u, v) \leq R, v \neq u\}, \quad (11)$$

where  $D(u, v)$  denotes the distance between the  $u$ th BS and the  $v$ th BS and  $R$  is the communication range of BSs. Clearly, the connectivity of the communication topology is closely related to the value of  $R$ , and the better connectivity will lead to a better matching. However,  $R$  is obviously constrained by the available battery of BSs.

In this setting, each BS receives information from SSs and simultaneously detects the number of failed SSs. It is sensible since energy expenditure in data processing or message reception is much less than that in transmission [1]. Once

the number of failed SSs is greater than the critical number for a guaranteed coverage ratio, each BS will cooperatively implement the following distributed suboptimal scheduling. Specifically, at the initialization stage, a BS is firstly selected in a random fashion and will choose its closest failed SS as the initial matching. Next, another BS will be randomly chosen and the closest failed SS in the remaining SSs is considered as its matching. The same rule sequentially applies to the other BSs. Finally, each BS has a unique matching SS and vice versa. Then, they communicate the matching information with their neighbors and swap their matchings if the swapping will lead to a lower cost.

This process can be precisely formulated as follows. Let us focus on a specific  $u$ th BS. Assume that the  $u$ th BS is assigned to recover the  $j_u$ th SS and its neighbor  $v$ th BS to the  $j_v$ th SS. They will swap their matchings if the following condition is satisfied:

$$w_{u, j_u} + w_{v, j_v} > w_{u, j_v} + w_{v, j_u}, \quad (12)$$

otherwise, they keep their current matchings. Proceeding with the same way, the  $u$ th BS sequentially communicates its matching with its next neighbor. Likewise, all the BSs will do the same manner until the cost cannot be reduced through the neighbor swapping.

*Remark 3.* The previous proposed initialization algorithm appears to require a centralized unit to notify the completion of the initialization for all the BSs since each BS only accesses neighbors' matching information. In practice, that is unnecessary because the swapping within two neighbor BSs can start regardless of other BSs' matching states, which means that there is no need to wait for all the BSs to obtain their initial matchings.

It is clear that each swap of the matching within two BSs will result in a lower cost, implying that the total cost is a decreasing function of the times of communication. Obviously, the total cost should be greater than zero, that is, lower bounded by zero. Thus, the total cost will converge to a limiting point after a sufficient large number of the communication times. Theoretically, although each communication may induce a lower cost, whether the limit is the minimum one among all possible matchings is unclear.

In the next section, some demonstrating examples would be examined to verify all the proposed algorithms on MATLAB 7.1 platform.

## 4. Simulations

**4.1. Regular Network Topology.** The sensing range of a sensor is set to  $r = 5$  m and the area of interest is  $(10r \times 10r) \text{ m}^2$ , as shown in Figure 1. Assume that there are 20 failed SSs at the time  $t$ ; see Figure 4, in which failed sensors are denoted as "x".

**4.1.1. Centralized Optimal Scheduling.** According to Figure 2, we can easily decide that how many failed sensors need to be recovered to guarantee the corresponding coverage ratio.

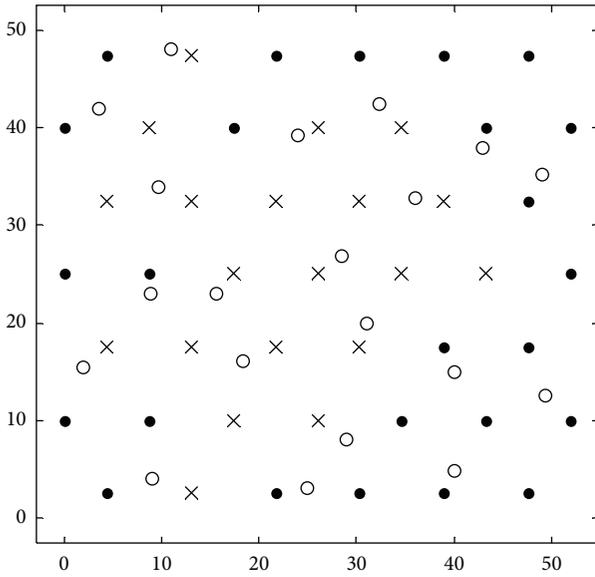


FIGURE 4: The regular network condition: failed sensors are denoted as "x".

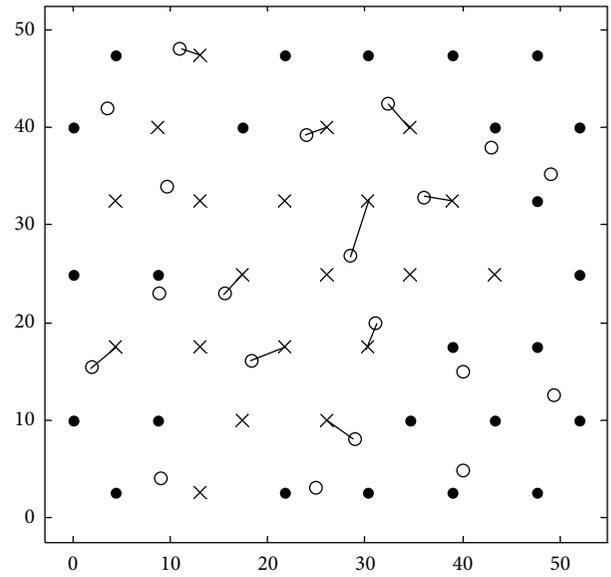


FIGURE 6: The matching result to recover 10 failed sensors by the proposed centralized algorithm.

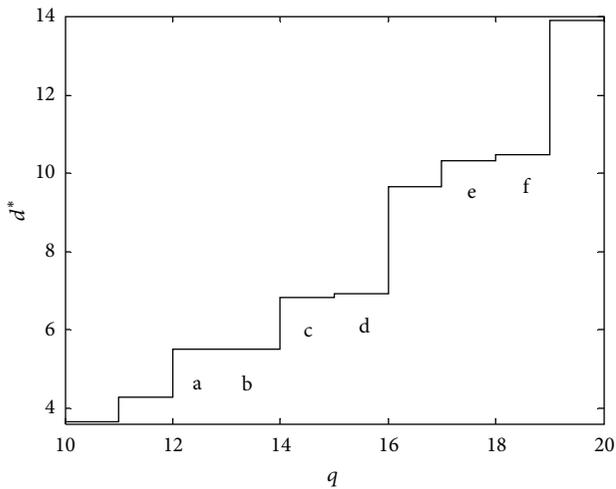


FIGURE 5: The number of failed sensors to be recovered  $q$  versus the minimum largest moving distance  $d^*$ .

Based on the optimization (P1), we get the relationship of  $d^*$  under different numbers of recovered sensors  $q$ , as shown in Figure 5. From the figure, we can see that the larger  $d^*$ , the more number of failed sensor will be recovered. In Figure 5, segments of a-b, c-d, and e-f increase slowly, which means that more failed sensors can be recovered with less additional cost.

Based on the optimization (P2), the optimal sensor scheduling result is obtained under a certain  $d^*$ . For illustration, we give the simulation results of the cases of 10 recovered sensors ( $d^* = 3.6385$  m) and 20 recovered sensors ( $d^* = 14.9533$  m), respectively. In Figure 6, we can see that every failed SS is recovered by the closest BS, which is certainly consistent with our intuition, showing that our algorithm can lead to the correct result.

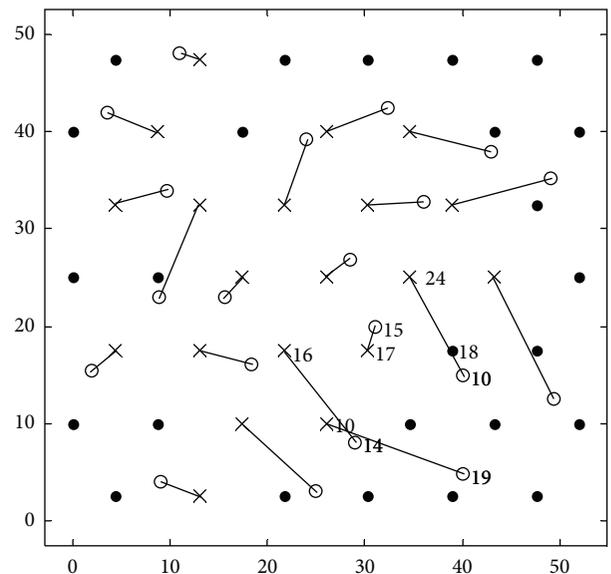


FIGURE 7: The matching result to recover 20 failed sensors by the proposed centralized algorithm.

The more complicated case (20 recovered sensors) is illustrated in Figure 7. In this case, not every failed SS would be recovered by its closest BS. For instance, let us consider the matchings between the 10th SS and the 19th BS, the 16th SS and the 14th BS, and the 17th SS and the 15th BS. In Figure 7, the total cost of these dispatch matchings is  $w_{19,10} + w_{15,17} + w_{14,16} = 29.5549$  m, which is larger than  $w_{14,10} + w_{19,17} + w_{15,16} = 29.2730$  m. The result seems incorrect because our sensor scheduling does not obtain the minimum total cost. However, our priority is to limit the maximum moving distance of each BS and then seek the optimal sensor scheduling under the constraint. Thus, using the matchings

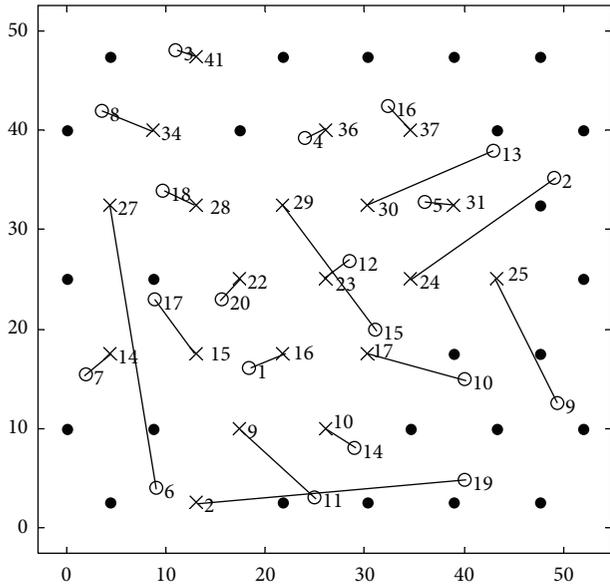


FIGURE 8: The matching result by the proposed distributed initialization algorithm.

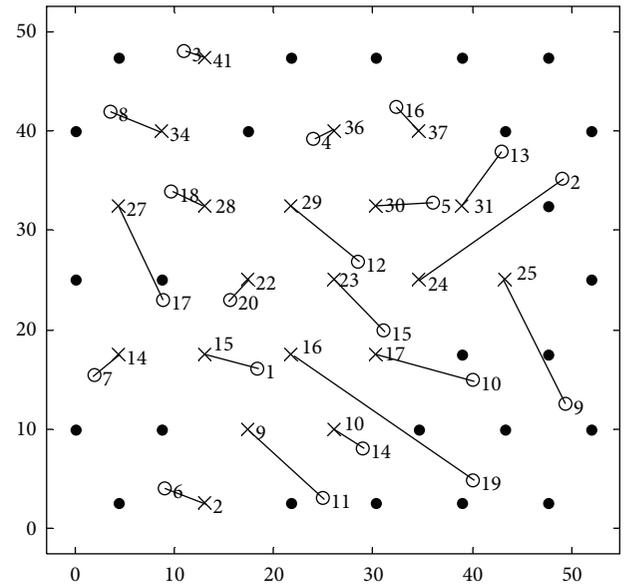


FIGURE 10: The matching result to recover 20 failed sensors by the proposed distributed algorithm ( $R = 4r$ ).

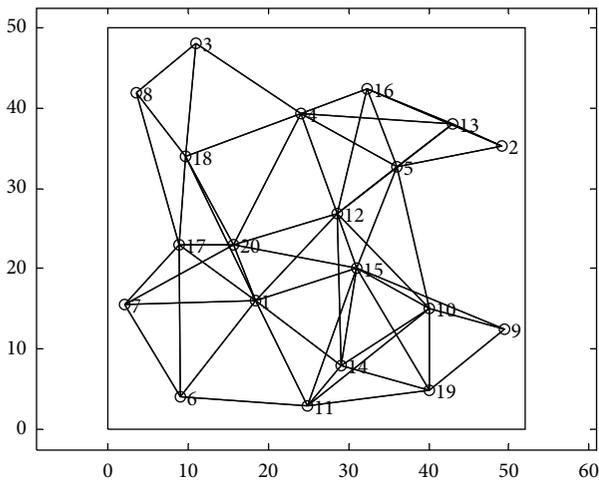


FIGURE 9: The communication topology of BSs for  $R = 20\text{ m}$  ( $4r$ ).

(the 14th SS and the 10th BS, the 19th SS and the 17th BS, and the 15th SS and the 16th BS), the maximum moving distance would be at least  $w_{19,17} > w_{19,10} = d^*$ , which suggests that the energy consumption of 19th BS would be higher or more time would be taken for sensor scheduling. Our scheduling has advantages under the scenario that every BS has the same builtin limited energy as it can reduce the possibility of a single BS breaking down too early. Thus, under the minimum  $d^*$ , we get the optimal result shown in Figure 7. Also from the figure, we can see that the 10th SS is the nearest failed sensor of the 14th BS and the 19th BS and  $w_{14,10} < w_{19,10} = d^*$ . Intuitively, the 10th SS should be recovered by the 14th BS. But our method suggests that it is not that case since it will induce a larger total cost.

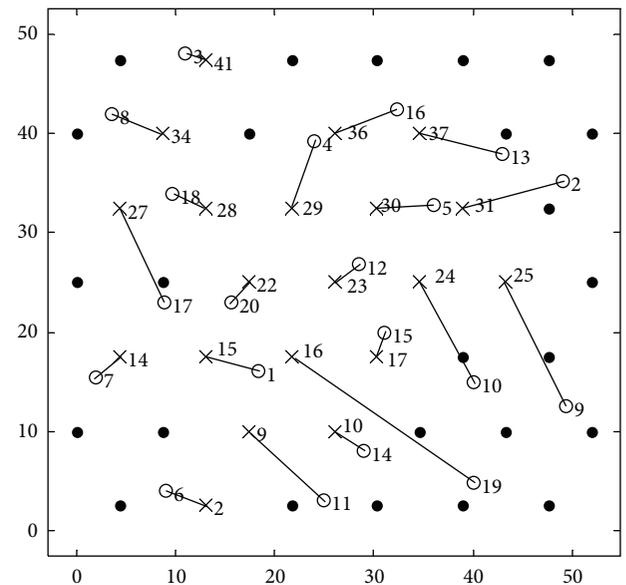


FIGURE 11: The matching result to recover 20 failed sensors by the optimal centralized algorithm ( $d^* = \infty$ ).

The simulation results show that our algorithms can find an optimal solution for sensor scheduling. As indicated, the straight lines in Figures 6 and 7 are not the practical trajectory of moving BSs. If all activated BSs move along in a straight line, some of them would collide with working SSs and other moving BSs. For instance, in Figure 7, the 10th BS may collide with the 18th SS on its way to the location of the 24th failed SS. The 14th BS and the 19th BS may also collide with each other during the moving process. However, the focus of this paper is on the sensor dispatch, and path planning problem for moving sensors will be discussed in our future work.

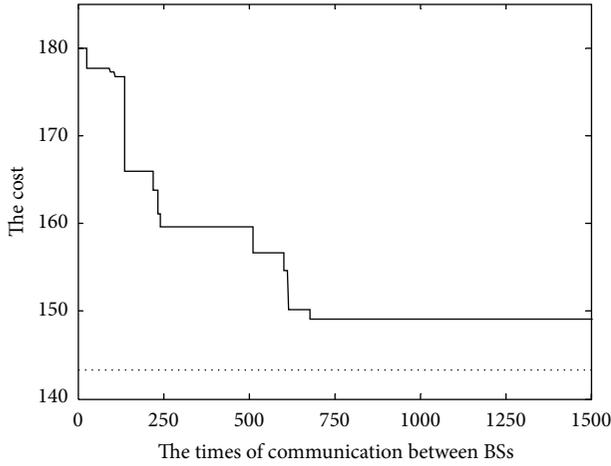


FIGURE 12: The cost function with respect to the communication times.

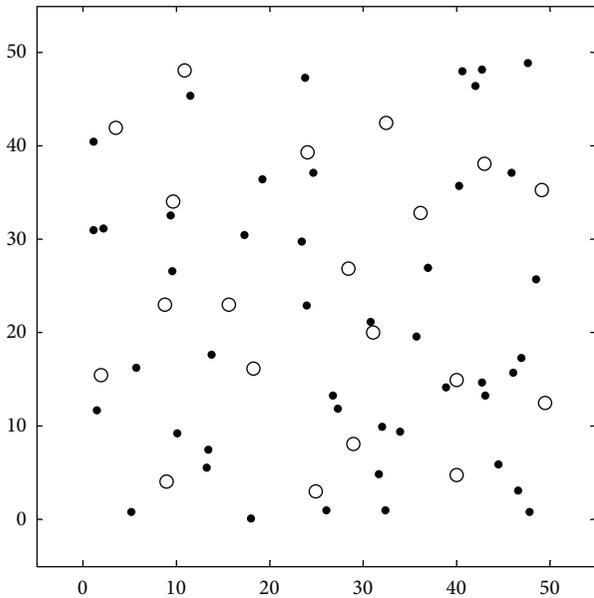


FIGURE 13: The random network condition.

**4.1.2. Distributed Suboptimal Scheduling.** According to the previous distributed suboptimal scheduling algorithm, the initial matching results of the case with 20 BSs and 20 SSs are illustrated in Figure 8. Here the initial matching randomly starts at the 14th BS, and it chooses the closest failed 10th SS as its initial matching. Proceeding with our distributed initialization algorithm, the next six BSs would obey the order of {17, 4, 20, 9, 18, 3}, respectively, choosing the failed SSs indexed by {15, 36, 22, 25, 28, 41}. From Figure 8, it is clear that the previous BSs have their closest failed SSs as the initial matchings. Then, the 11th BS follows and considers the closest 10th failed SS as the candidate. Because the 10th SS has already been occupied by the 14th BS, the 11th BS has to choose the second closest 9th SS as its matching. The same case occurs to other BSs {2, 13, 19, 15, 6}.

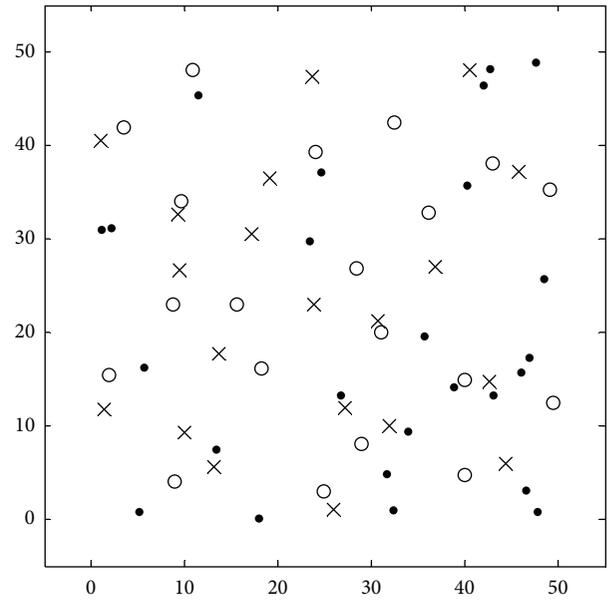


FIGURE 14: Failed sensors are denoted as “x”.

As mentioned previously, the connectivity of the communication topology is closely related to the value of  $R$ . Intuitively, the value of  $R$  is larger, the connectivity is better, which will lead to a better matching. However,  $R$  is obviously constrained by the available battery of BSs, while the near optimal matching can be achieved. Thus, the connectivity and the matching result under different values of  $R$  are discussed as follows.

Let  $R$  be  $2r$ ,  $3r$ ,  $4r$ ,  $5r$ , and  $6r$ , respectively. Based on the initial matching result as shown in Figure 8, the comparison result of the connectivity and the matching result under different values of  $R$  is listed in Table 1. From the table we can see that, along with the increase of  $R$ , the number both of the neighbor BSs and the times of communication between BSs becomes larger. But after  $R > 4r$ , their increases are very slight. Correspondingly,  $d^*$  decreases obviously when  $R = 4r$  and keeps 22.3162 m after  $R > 4r$ . Actually, when  $R$  first starts to increase, the number of the neighbor BSs of each BS becomes larger. So, BSs have more opportunities to swap their matchings, which will lead to the decreases of  $d^*$  and the total cost. However, when  $R$  is large enough,  $d^*$  and the total cost will not decrease any more due to the certain network topology and the determined initial matching result. Above all, we will set  $R = 4r$  during the following simulations considering the balance between the coverage performance and the energy consumption of all BSs.

Setting  $R = 4r$ , the communication topology of BSs is shown in Figure 9. Under the previous initial setup and the communication range, the best performance that the proposed distributed scheduling can achieve (see Figure 10) is compared to that of the centralized optimal scheduling without the consideration of maximum moving distance ( $d^* = \infty$ ) (see Figure 11). In order to show the efficiency of the distributed algorithm, the decrease of the cost due to each communication is shown in Figure 12, where the

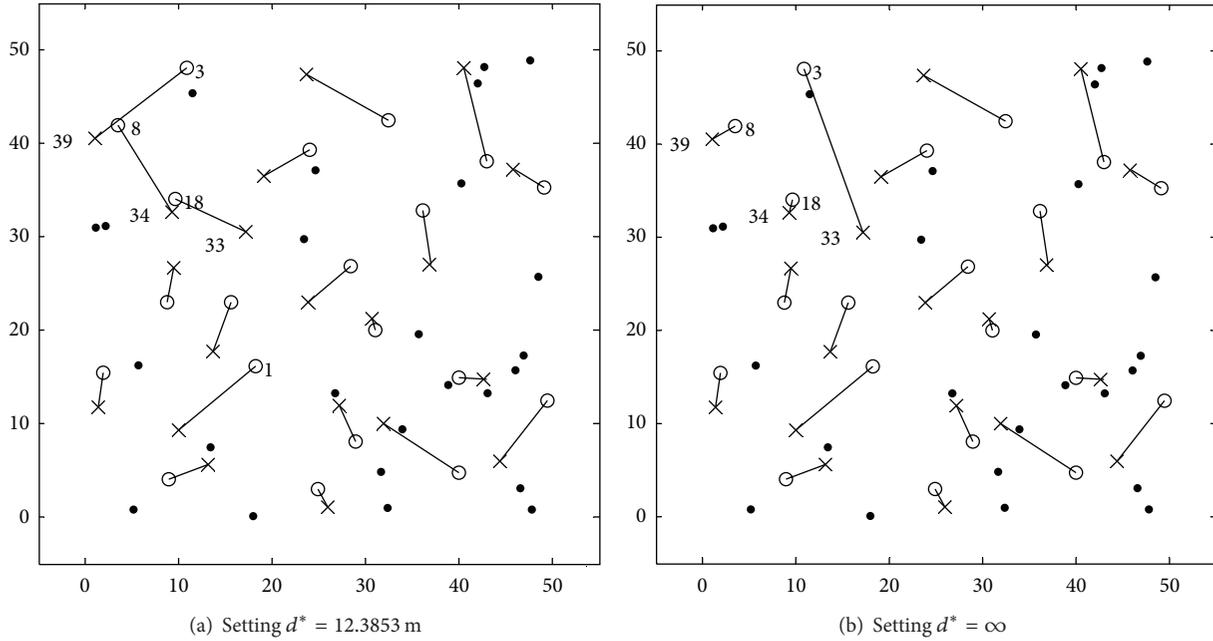


FIGURE 15: The matching result to recover 20 failed sensors by the optimal centralized algorithm.

TABLE 1: The comparison of the connectivity and the matching result under different values of  $R$ .

$R$	$2r$	$3r$	$4r$	$5r$	$6r$
The minimum number of the neighbor BSs	0	1	3	4	8
The maximum number of the neighbor BSs	2	6	9	14	19
The average number of the neighbor BSs	1	3.4	5.7	8.7	11.8
The communication times between BSs during the matching	236	514	680	688	692
$d^*$	Before swapping (m)		28.8712		
	After swapping (m)	28.8712	28.8712	22.3162	22.3162
The total cost	Before swapping (m)		179.9511		
	After swapping (m)	173.1040	161.8864	149.1501	148.3304

dot line represents the global minimum cost obtained by the centralized algorithm. As expected, we can see that the cost of the distributed algorithm converges to a limit which is slightly larger than that of the centralized optimal algorithm. Actually, the costs associated with the centralized and distributed algorithms are 143.3229 m and 149.1501 m, respectively. Thus, although the distributed algorithm cannot reach the global minimum cost due to the lack of a centralized FC, its performance is still acceptable.

**4.2. Random Network Topology.** As shown in Figure 13, the sensing range of a sensor is also set to  $r = 5$  m and the area of interest is  $(10r \times 10r) \text{ m}^2$ , in which BSs are denoted as “o”. Here, the locations of all BSs are the same as that of the regular network for simplicity of deciding the value of  $R$  and without loss of generality. Assume that there are 20 failed SSs at the time  $t$ , see Figure 14, in which failed sensors are denoted as “x”.

**4.2.1. Centralized Optimal Scheduling.** The best performance that the centralized optimal scheduling can achieve under the limitation of maximum moving distance (see Figure 15(a)) is compared to that of the centralized optimal scheduling without the consideration of maximum moving distance ( $d^* = \infty$ ) (see Figure 15(b)). Apparently, the differences between two figures are only the matchings between the 33th SS, the 34th SS, the 39th SS, the 18th BS, the 8th BS, and the 3th BS. In Figure 15(a), the matchings results are the 33th SS and the 18th BS, the 34th SS and the 8th BS, and the 39th SS and the 3th BS. The total cost of these dispatch matchings is  $w_{33,18} + w_{34,8} + w_{39,3} = 8.3066 + 10.9339 + 12.3853 = 31.6258$  m. In Figure 15(b), the matchings results are the 33th SS and the 3th BS, the 34th SS and the 18th BS, and the 39th SS and the 8th BS. The total cost of these dispatch matchings is  $w_{33,3} + w_{34,18} + w_{39,8} = 18.5558 + 1.4552 + 2.7792 = 22.7902$  m. Actually, the costs associated with Figures 15(a) and 15(b) are 129.9015 m and 121.0659 m, respectively. Although Figure 15(a) cannot reach the global

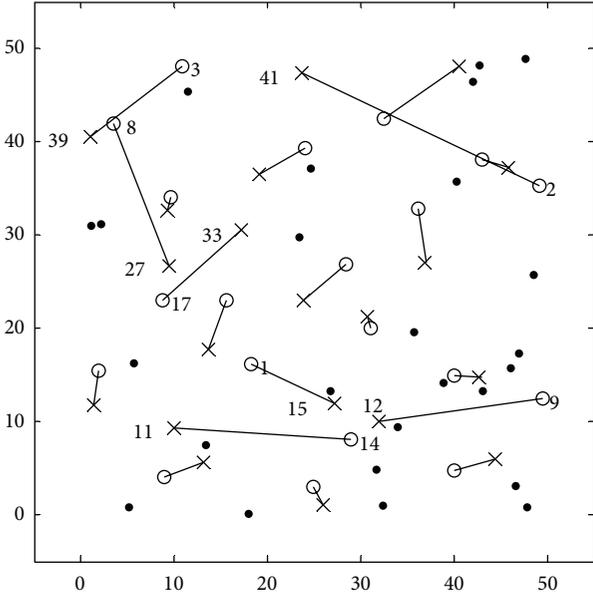


FIGURE 16: The initialization matching result.

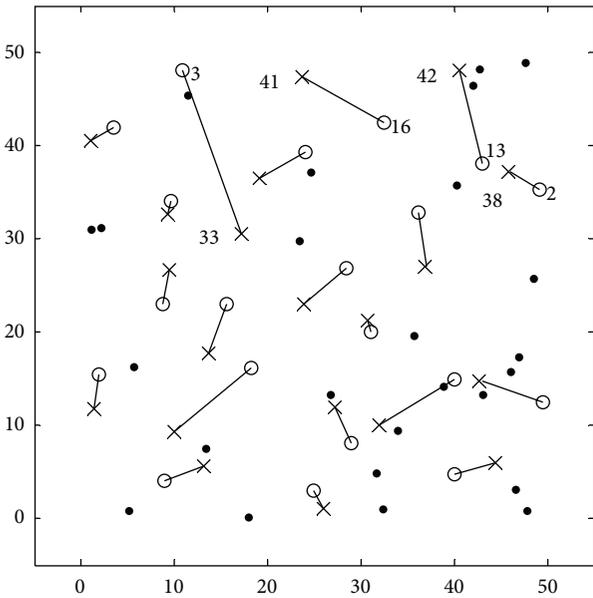


FIGURE 17: The final matching result.

minimum cost, its performance is still acceptable. Moreover, the moving distance of the 3th BS is shorter.

**4.2.2. Distributed Suboptimal Scheduling.** The initial matching results of the case with 20 BSs and 20 SSs are illustrated in Figure 16. Here the initial matching randomly starts at the 3th BS, and it chooses the closest failed 39th SS as its initial matching. It is clear that most of BSs have their closest failed SSs as the initial matchings. However, because the closest failed SSs of BSs {1, 2, 8, 9, 14, 17} have already occupied by other BSs, the distances of their initial matchings are much longer than those of their neighbor BSs.

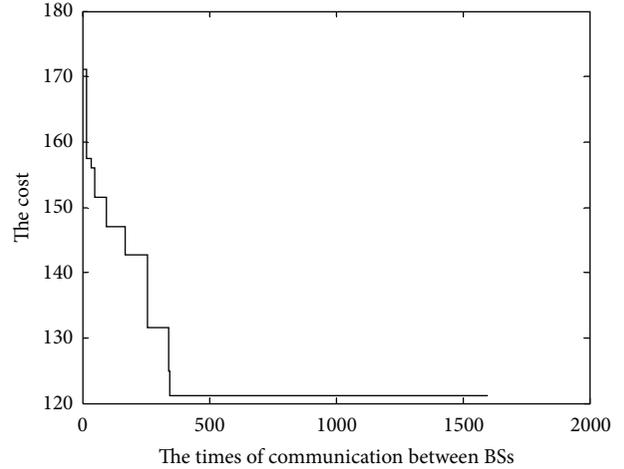


FIGURE 18: The cost function with respect to the communication times.

Also setting  $R = 4r$ , all BSs swap matchings between their neighbors. The best performance that the proposed distributed scheduling can achieve is shown in Figure 17, in which most of distances between BSs and SSs are shorter than the initial results. At the same time, the decrease of the cost due to each communication is shown in Figure 18. Actually, the cost associated with the distributed algorithm is 121.1482 m which is very close to the centralized algorithm result, that is, 121.0659 m ( $d^* = \infty$ ). Thus, the performance of the distributed algorithm is still extremely good. Moreover, due to without a centralized FC, the distributed algorithm is easier to implement with low communication and computation costs from the application point of view.

### 5. Conclusions

We have proposed an optimal mobile sensor scheduling for WSNs with possible static sensor failures due to limited energy and other unknown factors. To maintain the coverage performance of the network, some spare mobile sensors should be deployed to recover failed sensors. Using a centralized FC, we formulated the sensor dispatch problem into two optimization problems and proposed two corresponding algorithms for their solutions. Our scheduling scheme first limited the maximum moving distance of the mobile sensors so as to reduce the maximum energy consumption of a single mobile sensor and then sought an optimal dispatch method to minimize the total moving distance. Removing the centralized FC, we also considered a distributed suboptimal sensor scheduling method which only depended on the local matching information of the neighbor mobile sensors. Simulation results verified the proposed algorithms under regular and random network topologies.

### Acknowledgments

This work is supported by the National Science and Technology Support Program “study on the integration technology of large scale battery energy storage system” (no.

2011BAA07B07), the Natural Science Foundation of Jiangsu Province of China (no. BK2012409), and the Special Fund for Basic Scientific Research of Central Colleges, Hohai University (no. 2011B09014).

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective," *Computer Journal*, vol. 54, no. 3, pp. 455–470, 2011.
- [3] E. Yang, A. T. Erdogan, T. Arslan, and N. H. Barton, "Multi-objective evolutionary optimizations of a space-based reconfigurable sensor network under hard constraints," *Soft Computing*, vol. 15, no. 1, pp. 25–36, 2011.
- [4] Y. Stergiopoulos, Y. Kantaros, and A. Tzes, "Distributed control of mobile sensor networks under RF connectivity constraints," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 741821, 7 pages, 2012.
- [5] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 41, no. 2, pp. 262–267, 2011.
- [6] N. Bartolini, T. Calamoneri, T. F. la Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 753–766, 2011.
- [7] J. Chen, J. Li, S. He, Y. Sun, and H. H. Chen, "Energy-efficient coverage based on probabilistic sensing model in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 9, pp. 833–835, 2010.
- [8] Y. C. Wang, F. J. Wu, and Y. C. Tseng, "Mobility management algorithms and applications for mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 1, pp. 7–21, 2012.
- [9] T. M. Cheng and A. V. Savkin, "Decentralized control for mobile robotic sensor network self-deployment: barrier and sweep coverage problems," *Robotica*, vol. 29, no. 2, pp. 283–294, 2011.
- [10] Y. C. Wang, W. C. Peng, and Y. C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1836–1850, 2010.
- [11] J. Li, B. H. Zhang, L. G. Cui, and S. C. Chai, "An extended virtual force-based approach to distributed self-deployment in mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 417307, 14 pages, 2012.
- [12] X. Z. Han, S. Li, and X. Pang, "Coverage optimization algorithm of wireless sensor network," *Advances in Future Computer and Control Systems*, vol. 159, pp. 33–39, 2012.
- [13] S. He, J. Chen, Y. Sun, D. K. Y. Yau, and N. K. Yip, "On optimal information capture by energy-constrained mobile sensors," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2472–2484, 2010.
- [14] S. B. He, J. M. Chen, D. K. Y. Yau, H. Y. Shao, and Y. X. Sun, "Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1090–1102, 2012.
- [15] J. H. Chen, M. B. Salim, and M. Matsumoto, "A single mobile target tracking in voronoi-based clustered wireless sensor network," *Journal of Information Processing Systems*, vol. 7, no. 1, pp. 17–28, 2011.
- [16] B. Y. Liu, O. Dousse, P. Nain, and D. Towsley, "Dynamic coverage of mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 301–311, 2013.
- [17] T. A. Wettergren and C. M. Traweek, "The search benefits of autonomous mobility in distributed sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 797040, 11 pages, 2012.
- [18] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 1–31, 2011.
- [19] C. Y. Chang, C. Y. Lin, and C. H. Kuo, "EBDC: an energy-balanced data collection mechanism using a mobile data collector in WSNs," *Sensors*, vol. 12, pp. 5850–5871, 2012.
- [20] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks," *Information Fusion*, vol. 12, no. 3, pp. 202–212, 2011.
- [21] W. Almobaideen, K. Hushaidan, A. Sleit, and M. Qatawneh, "A Cluster-based approach for supporting QoS in Mobile Ad Hoc networks," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 1, pp. 1–9, 2011.
- [22] S. K. Zhang, Y. Sun, J. X. Fan, and H. Huang, "Cooperative data processing algorithm based on mobile agent in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 182561, 9 pages, 2012.
- [23] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2009.
- [24] Y. S. Cheng, M. Neely, and K. M. Chugg, "Iterative message passing algorithm for bipartite maximum weighted matching," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1934–1938, Seattle, Wash, USA, July 2006.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

