

# A Generic Framework for Optimal Mobile Sensor Redeployment

Zhong Shen, Yilin Chang, Hai Jiang, *Member, IEEE*, Yanling Wang, and Zhongjiang Yan

**Abstract**—Wireless sensor networks (WSNs) are widely used for various applications. In some applications, it is very likely that the number of sensors is limited, the initial deployment is random, and deployment “holes” (i.e., subareas where there is no sensor) may exist. Therefore, a major challenge in WSNs is to improve the network coverage. A viable solution is to enhance some sensors with mobility capability so that they can move inside the WSNs after the initial deployment, which is referred to as *mobile sensor redeployment*. In this paper, we target at a generic framework for the optimal mobile sensor redeployment problem in WSNs. In particular, in this paper, the area of a WSN is partitioned into a number of grids, and the *gap* of each grid is defined as the difference of the number of sensors in the grid from the desired number of sensors. Then, the mobile sensor redeployment problem is formulated as an optimization problem with three optimization requirements: to minimize the sum of gaps of all grids, to minimize the  $L_p$ -norm ( $1 \leq p \leq \infty$ ) of the gap vector (where  $L_p$ -norm represents a family of optimization objectives when  $p$  takes different values), and to minimize the total movement cost of all mobile sensors. Bipartite matching-based redeployment (BMRD) algorithms are provided to solve the optimization problems with different values of  $p$ . In this paper, the framework is *generic* for mobile sensor redeployment because 1) the mobility capabilities of the sensors, the initial sensor distribution, the movement cost measures, and the sensor types are all *arbitrary*; 2) the coverage requirements in different grids can be heterogeneous; and 3) the BMRD algorithms are *generic* for different  $p$  values. The effectiveness of the BMRD algorithms is validated through theoretical proofs and extensive simulations.

**Index Terms**—Bipartite graph, coverage, flow network, mobility, wireless sensor networks (WSNs).

## I. INTRODUCTION

WIRELESS sensor networks (WSNs) are widely used in various applications [1]–[4]. In some applications (such as military sensing and tracking, fire/earthquake search and rescue, and data collections in hazardous environments), the WSNs may only have a random initial deployment of the sensors due to the probability of high cost or high risk in

sensor deployment in these applications. Some subareas may be insufficiently covered by sensors, and some subareas may even have no sensors (i.e., deployment “holes” may exist). To improve coverage, a large amount of sensors can be deployed so that the desired coverage requirement can be met with a high probability [5], [6]. This approach, however, might be costly and inefficient due to the normally uncontrollable random deployment. Another solution is to enhance some sensors with mobility capability. These *mobile sensors* can move inside the WSNs after the initial deployment. This procedure is referred to as *mobile sensor redeployment*, which is the focus of this paper.

The problem of mobile sensor redeployment has received much attention recently. A number of movement schemes have been proposed in the literature, typically based on three basic techniques [7]: virtual force [8]–[10], coverage pattern [11]–[13], or grid architecture [14]–[20]. Virtual force movement schemes use repulsive and attractive forces to make nodes distributed evenly. In particular, when two nodes are too close (or far away), they expel (or attract) each other. The main idea of coverage pattern movement schemes is to divide the sensor area into some regular shapes (e.g., equilateral triangle, regular hexagon, etc.) and then to move sensors to the vertices of the shapes such that the coverage target is achieved. On the other hand, the sensor area in a grid architecture-based scheme is partitioned into a number of small grids. A grid is said to be covered if it has one or more sensors in it, and the whole sensor area is said to be covered if each grid is covered. In addition, some applications may require each grid be covered by multiple sensors for the purposes of improved sensing accuracy, balanced sensor load, and/or prolonged network lifetime. In this paper, we focus on grid architecture-based mobile sensor redeployment.

Several movement algorithms have been proposed in the literature based on grid architecture. In [15], an optimal movement solution is proposed, and the objective is to make each grid covered by the same number of sensors and minimize the movement cost. All sensors are assumed to be mobile, and any sensor has the capability to move to any grid in the area. However, it is likely in practical systems that only a portion of sensors are mobile and with limited mobility capabilities. Taking limited mobility capability into account, the mobile sensor redeployment problem is studied in [16] and [17]. A flip-based network scenario is considered, and a mobile sensor can only hop from one grid to another. In [16], a movement algorithm is proposed, which can maximize the number of covered grids by using the minimum number of hops. A more general algorithm is given in [17], focussing on the minimum number of hops and the minimum variance of the numbers

Manuscript received February 7, 2010; revised May 21, 2010; accepted July 14, 2010. Date of publication July 29, 2010; date of current version October 20, 2010. This work was supported in part by 111 Project under Grant B08038 with Xidian University, Xi’an, China, and in part by the Alberta Innovates—Technology Futures, Alberta, Canada, under an Alberta Ingenuity New Faculty Award. The review of this paper was coordinated by Dr. T. Taleb.

Z. Shen, Y. Chang, and Z. Yan are with the State Key Laboratory of Integrated Service Network, Xidian University, Xi’an 710071, China (e-mail: z.shen.xidian@gmail.com; ylchang@gmail.com; zhjyan@xidian.edu.cn).

H. Jiang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: hai.jiang@ece.ualberta.ca).

Y. Wang was with Xidian University, Xi’an 710071, China. She is now with the China Mobile (Shijiazhuang Office), Shijiazhuang, Hebei 050000, China (e-mail: wyling429@163.com).

Digital Object Identifier 10.1109/TVT.2010.2062203

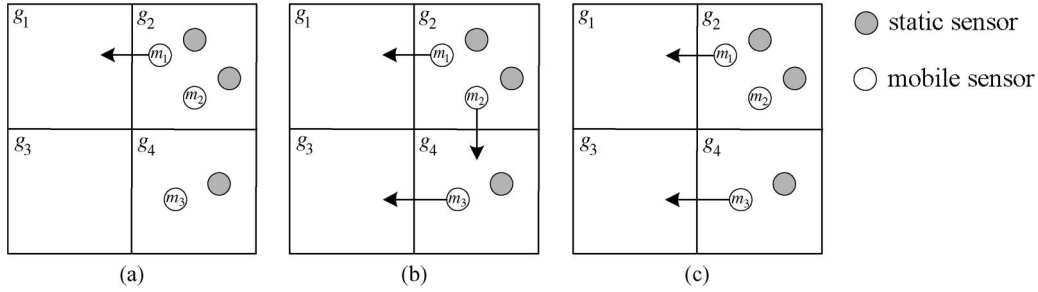


Fig. 1. Redeployment examples.

of sensors in the grids. The above two algorithms can only be applied to flip-based networks. Different from the existing works in the literature, we target at a generic framework for the optimal mobile sensor redeployment in WSNs.

In this paper, the grid architecture-based mobile sensor redeployment problem is generalized with the following three requirements. First, mobile sensors are the *resources* for improving network coverage. Therefore, it is desired to make the maximal use of these resources. More specifically, redundant mobile sensors in a grid should be moved as many as possible into uncovered or less-covered grids. Second, besides satisfying the first requirement, it is also desired to achieve some sensor distribution property. For instance, in some applications, it is preferred that the sensors are distributed as evenly as possible such that the work load of the sensors can be balanced, whereas in other applications whose emphasis is on network survivability, it is preferred to maximize the number of sensors in the worst covered grid(s). Third, the total movement cost of mobile sensors for meeting the above two requirements should be minimized. It can be seen that the first requirement has the highest priority, while the third requirement has the lowest.

Fig. 1 shows a simple example, where a WSN has four grids, denoted as  $g_1, g_2, g_3,$  and  $g_4$ . The target of each grid is to have at least three sensors through a movement plan of mobile sensors. In the initial deployment, grid 1 or 3 has no sensor; grid 2 has two static sensors (denoted by the gray color) and two mobile sensors (i.e.,  $m_1$  and  $m_2$ ; denoted by the white color); and grid 4 has a static sensor and a mobile sensor (i.e.,  $m_3$ ). It is assumed that 1) a mobile sensor can move at most once and can move only to one of its neighboring grids, and 2) the movement cost is measured by the total number of movements. It is worth noting that these assumptions are just to simplify the example, and they are actually not necessary in our algorithm. Apparently, in this example, it is not feasible to make each grid covered by at least three sensors. In this context, we may have different movement plans according to different requirements.

- When our target is to move as many redundant mobile sensors as possible into less-covered grids (the first requirement) and to minimize the movement cost (the third requirement): In this example, only  $g_2$  has one redundant sensor, and thus, moving a mobile sensor from  $g_2$  to  $g_1$  (or to  $g_4$ ) can meet the first requirement, as shown in Fig. 1(a). This movement scheme also has the minimum movement cost (one hop) for fulfilling the first requirement, which satisfies the third requirement.

- When our target is to move as many redundant mobile sensors as possible into less-covered grids (the first requirement), to make the sensors distributed as evenly as possible (the second requirement), and to minimize the movement cost (the third requirement): One solution is to move the two mobile sensors in grid  $g_2$  to grids  $g_1$  and  $g_4$ , respectively, and move mobile sensor  $m_3$  to grid  $g_3$ , as shown in Fig. 1(b). In this solution, no grid has redundant sensors, and thus, the first requirement is satisfied. The numbers of sensors in the four grids are 1, 2, 1, and 2, respectively, which apparently satisfies the second requirement. It can also be seen that the minimum movement cost (three hops) is used, and thus, the third requirement is satisfied.
- When our target is to move as many redundant mobile sensors as possible into less-covered grids (the first requirement), to maximize the number of sensors in the worst covered grid(s) (the second requirement), and to minimize the movement cost (the third requirement): One solution is to move a mobile sensor from grid  $g_2$  to grid  $g_1$  and move mobile sensor  $m_3$  to grid  $g_3$ , as shown in Fig. 1(c). In this solution, no grid has redundant sensors, and thus, the first requirement is satisfied. Either of the two worst covered grids (grids  $g_1$  and  $g_3$ ) has a sensor node, which satisfies the second requirement. It can also be seen that the minimum movement cost (two hops) is used, and thus, the third requirement is satisfied.

To solve problems with different requirements, we present in this paper a generic framework for the optimal mobile sensor redeployment in WSNs. First, the *gap* of each grid is defined as the difference of the number of sensors in the grid from the desired number of sensors. Then, the general mobile sensor redeployment problem is formulated as an optimization problem with three optimization requirements: to minimize the sum of gaps of all grids, to minimize the  $L_p$ -norm ( $1 \leq p \leq \infty$ ) of the gap vector (where  $L_p$ -norm represents a family of optimization objectives when  $p$  takes different values), and to minimize the total movement cost of all mobile sensors. These three optimization objectives, respectively, correspond to the three requirements of the general redeployment problem. Bipartite matching-based redeployment (BMrD) algorithms are provided to solve the optimization problems with different values of  $p$ . The framework in this paper is generic because 1) a family of optimization objectives are represented in a generic form; 2) the mobility capabilities of the sensors, the initial sensor distribution, the movement cost measures, and the sensor

TABLE I  
SUMMARY OF IMPORTANT SYMBOLS USED

Symbols	Meaning
$C(\mathcal{P})$	cost of the flows corresponding to movement plan $\mathcal{P}$
$c_g(g_i, t)$	gap cost of the edge from $g_i$ to the sink $t$
$c_m(m_j, g_i)$	movement cost for $m_j$ moving to grid $g_i$
$c_m(\mathcal{P})$	movement cost of movement plan $\mathcal{P}$
$D$	the constant to compute the gap cost
$f(m_j, g_i)$	the amount of flows through edge $(m_j, g_i)$
$\mathbb{G}$	the set of grids needed to be considered
$\mathbf{G}_b$	bipartite graph
$\mathbf{G}_f$	flow network
$g_i$	the $i$ th grid
$h_\alpha(x)$	gap cost function of the grid gap $x$ for achieving $\alpha$ -coverage
$\mathbb{M}$	the set of mobile sensors
$m_j$	the $j$ th mobile sensor
$(m_j, g_i)$	the edge from $m_j$ to $g_i$
$\mathcal{P}$	movement plan
$r_i$	the gap of the $i$ th grid
$\mathbf{r}$	the gap vector
$\ \mathbf{r}\ _p$	$L_p$ -norm of gap vector $\mathbf{r}$
$u(m_j, g_i)$	the capacity of edge $(m_j, g_i)$

types are all arbitrary; 3) the coverage requirements in different grids can be heterogeneous; and 4) the algorithms to solve the optimal redeployment problems are also generic for different optimization objectives.

The rest of this paper is organized as follows. The problem statement is given in Section II. Optimal algorithms to solve the redeployment problems are provided in Section III. Some properties of the optimal algorithms are discussed in Section IV. Performance evaluation is provided in Section V. Related works are discussed in Section VI, followed by conclusions in Section VII. Important symbols used in this paper are summarized in Table I.

## II. PROBLEM STATEMENT

A hybrid WSN is considered, which includes static sensors and mobile sensors. A mobile sensor can move inside the WSN. It is assumed that the movement capabilities of the mobile sensors may be different from each other. Here, the movement capability of a mobile sensor means the largest distance that the mobile sensor can move. Similar to [14]–[17], it is assumed that each sensor has a sensing range denoted  $R_s$  and a transmission range denoted  $R_t$ . The deployment region of the WSN is a square with the side length being  $W$ . The deployment region is partitioned into  $n$  smaller square-shaped grids denoted  $g_1, g_2, \dots, g_n$ , each with the side length being  $\min\{R_s/\sqrt{2}, R_t/\sqrt{5}\}$ . Thus,  $n = \lceil W/\min\{R_s/\sqrt{2}, R_t/\sqrt{5}\} \rceil^2$ , where  $\lceil \cdot \rceil$  is the ceiling function. By this setting, any sensor in a grid is able to sense each corner of the grid and is also able to communicate with any sensor in any of its neighboring grids (i.e., the top-side, bottom-side, left-hand side, and right-hand side grids).

If a grid has  $x$  sensors, we say that the grid is  $x$ -covered. In this paper, our objective is to design a movement plan, which is denoted  $\mathcal{P}$ , of the mobile sensors to make each grid at least  $k$ -covered. Here,  $k$  ( $\geq 1$ ) is a predefined constant. For a grid  $g_i$  ( $i \in \{1, 2, \dots, n\}$ ), let  $|g_i|$  denote the number of sensors in it. The nonnegative-valued *gap* of grid  $g_i$ , which is

denoted  $r_i$ , is defined as the difference of  $|g_i|$  from  $k$ . That is,  $r_i = \max\{k - |g_i|, 0\}$ . For instance, after the movements shown in Fig. 1(a), the gap of  $g_1, g_2, g_3$ , and  $g_4$  is 2, 0, 3, and 1, respectively.

Recall that the general redeployment problem has three requirements. The first requirement is to move redundant mobile sensors as many as possible into uncovered or less-covered grids. When a redundant mobile sensor moves into an uncovered or less-covered grid, then the gap of the destination grid will decrease by one. Therefore, the first requirement is equivalent to minimizing the sum of the gaps of all grids. For example, all of the three movement schemes shown in Fig. 1 meet the first requirement. They all have the minimum sum of gaps (i.e., six).

The second requirement is on sensor distribution subject to satisfying the first requirement. In some cases, it is required to make a balanced sensor distribution, which is equivalent to minimizing the gap variance, whereas in other cases, it is required to make the “worst” grid have the maximum coverage, which is equivalent to minimizing the maximum gap of the grids. In this paper, these requirements are generalized as minimizing the  $L_p$ -norm of the gap vector  $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$ , which is defined as  $\|\mathbf{r}\|_p = (r_1^p + r_2^p + \dots + r_n^p)^{1/p}$ ,  $1 \leq p \leq \infty$ . Typical values of  $p$  are  $p = 1$ ,  $p = 2$ , and  $p = \infty$ . It can be seen that minimizing  $\|\mathbf{r}\|_1$  means minimizing the sum of gaps of all the grids, minimizing  $\|\mathbf{r}\|_2$  means minimizing the gap variance, and minimizing  $\|\mathbf{r}\|_\infty$  means minimizing the maximum gap.

The third requirement is to minimize the total movement cost subject to satisfying the above two requirements. In a movement plan  $\mathcal{P}$ , a movement cost is associated with a mobile sensor, which can be the moving distance, or the consumed energy, or the number of hops,<sup>1</sup> etc. Therefore, the third requirement is to minimize the movement cost of the movement plan  $\mathcal{P}$ , denoted  $c_m(\mathcal{P})$ , which is the sum of the movement costs of all mobile sensors.

As a summary, we need to find a movement plan  $\mathcal{P}$  with three optimization requirements, i.e., to minimize the sum of the gaps, the  $L_p$ -norm of the gap vector, and the movement cost. In particular, a three-step optimization problem is formulated as follows.

Step 1) Minimizing the sum of gaps, i.e.,

$$\underset{\mathcal{P}}{\text{minimize}} \quad \sum_{i=1}^n r_i$$

subject to moving capability of each mobile sensor.

Let  $\mathbb{F}_1$  denote the set of all optimal movement plans in step 1.

Step 2) Minimizing the  $L_p$ -norm of the gap vector, i.e.,

$$\underset{\mathcal{P} \in \mathbb{F}_1}{\text{minimize}} \quad \|\mathbf{r}\|_p. \quad (1)$$

Let  $\mathbb{F}_2$  denote the set of all optimal movement plans in step 2.

<sup>1</sup>A *hop* is defined as a movement from a grid to one of its neighboring grids.

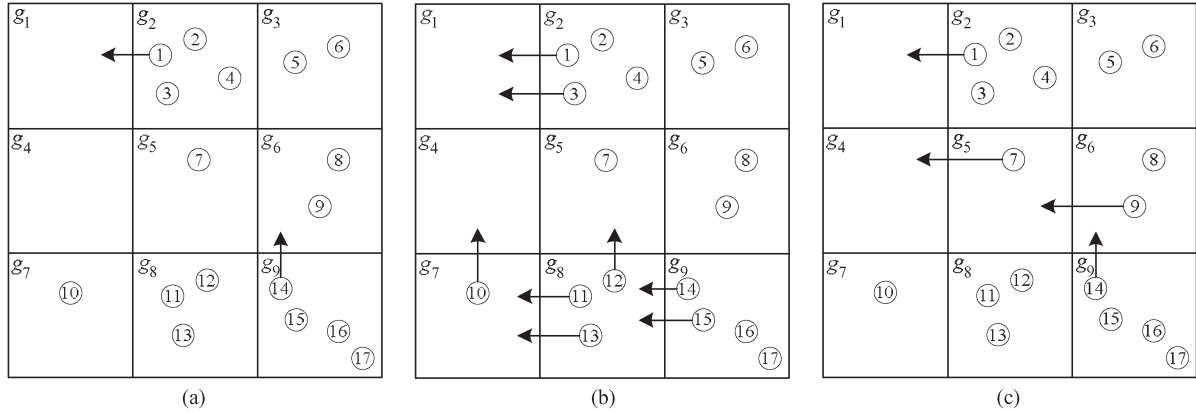


Fig. 2. Optimal movement plan for (a)  $p = 1$ , (b)  $p = 2$ , and (c)  $p = \infty$ , when the WSN has nine grids and 17 mobile sensors.

Step 3) Minimizing the movement cost, i.e.,

$$\underset{\mathcal{P} \in \mathbb{F}_2}{\text{minimize}} \quad c_m(\mathcal{P}). \quad (2)$$

It can be seen that, in the preceding formulation, minimizing the sum of gaps has the highest priority, whereas minimizing the movement cost has the lowest priority. Note that, when  $p = 1$ , steps 1 and 2 are the same.

When  $p = 2$ , we have an interesting observation: Step 2 is equivalent to maximizing the Jain's fairness index [21] of the grid gaps, which is defined as

$$FI = \frac{(\sum_{i=1}^n r_i)^2}{n \sum_{i=1}^n r_i^2}. \quad (3)$$

A higher fairness index value means better fairness performance. When  $r_i$ 's are all equal to each other, the fairness index achieves its maximum value, which is 1. In step 2, all the feasible movement plans are from  $\mathbb{F}_1$ . They have the same (minimum) value of  $\sum_{i=1}^n r_i$  and, thus, have the same numerator in (3). Therefore, minimizing the  $L_2$ -norm of the gap vector is equivalent to minimizing the denominator in (3) and, thus, is equivalent to maximizing the fairness index.

Different values of  $p$  will result in different optimal movement plans for the three-step optimization problem. The three examples in Fig. 1 correspond to  $p = 1$ ,  $p = 2$ , and  $p = \infty$ , respectively. It can be clearly seen that the optimal movement plan depends on the value of  $p$ . Another example is given in Fig. 2 in which a WSN has 9 grids and 17 mobile sensors. In this example, the coverage task is  $k = 3$ , and each mobile sensor can move at most once and can move only to one of its neighboring grids.

Note that when it is feasible to meet the coverage requirements of all grids (i.e., each grid is at least  $k$ -covered), then the optimal values of steps 1 and 2 will both be 0, regardless of the value of  $p$ . In this case, the three-step optimization will find a solution that meets the coverage requirements of all grids (thereby achieving the optimality in steps 1 and 2) and has the minimum movement cost. When it is infeasible to meet the coverage requirements of all grids, the three-step optimization will find a solution that achieves the optimization objectives in the three steps.

Generally, it is challenging to solve a multistep optimization problem due to the complexity that is involved in each step (e.g., we need to get all the optimal solutions in steps 1 and 2 of the aforementioned three-step optimization problem). To address this challenging issue, in Section III, bipartite matching-based algorithms are provided to find optimal solutions for the three-step optimization problems with different values of  $p$ .

### III. BIPARTITE MATCHING-BASED REDEPLOYMENT ALGORITHMS

Here, optimal algorithms are presented to solve the aforementioned three-step optimization problems with different  $p$  values. In particular, a bipartite graph is first constructed to model the movement capabilities of the mobile sensors. Then, based on the bipartite graph, a flow network is formed. An optimal matching of the flow network determines an optimal movement plan for the three-step optimization problem. This algorithm is referred to as the *Bipartite Matching-based Redeployment (BMrD)* algorithm, denoted  $\text{BMrD}_{k,p}$  when the target of each grid is to be at least  $k$ -covered, and the  $L_p$ -norm of the gap vector is to be minimized in step 2 of the three-step optimization problem. In the following two subsections, the  $\text{BMrD}_{k,p}$  algorithm with  $1 \leq p < \infty$  is presented first, followed by the  $\text{BMrD}_{k,\infty}$  algorithm. Note that a bipartite graph, a flow network, and matching are usually used to solve machine-scheduling problems and personnel-assignment problems [22]. Here, the methods are adopted to solve our three-step optimization problem.

#### A. $\text{BMrD}_{k,p}$ Algorithm ( $1 \leq p < \infty$ )

The pseudocode of the  $\text{BMrD}_{k,p}$  ( $1 \leq p < \infty$ ) algorithm is shown in Algorithm 1 in Table II. Three phases are included in the algorithm: bipartite graph construction, flow network formation, and optimal matching computation.

1) *Bipartite Graph Construction (Lines 1–5 in Algorithm 1)*: Let  $\mathbb{M} = \{m_j : m_j \text{ is a mobile sensor}\}$  denote the set of mobile sensors, and let  $\mathbb{G} = \{g_i : \text{the number of static sensors in } g_i \text{ is less than } k\}$  denote the set of grids that are needed to be considered, in which any grid has less than  $k$  static sensors. Note that any grid with  $k$  or more static sensors is not considered, since it is already at



TABLE II  
ALGORITHM 1:  $\text{BMrD}_{k,p}$ ,  $k \geq 1$ ,  $1 \leq p < \infty$

**Algorithm 1**  $\text{BMrD}_{k,p}$ ,  $k \geq 1$ ,  $1 \leq p < \infty$

---

```

1: Construct a bipartite graph  $\mathbf{G}_b = (\mathbb{V}_b, \mathbb{E}_b)$ , where  $\mathbb{V}_b = \mathbb{M} \cup \mathbb{G}$ ,  $\mathbb{M} = \{m_j : m_j \text{ is a mobile sensor}\}$ ,  $\mathbb{G} = \{g_i : \text{the number of static sensors in } g_i \text{ is less than } k\}$ , and  $\mathbb{E}_b = \emptyset$ 
2: for each vertex  $m_j \in \mathbb{M}$  do
3:   for each vertex  $g_i \in \mathbb{G}$  do
4:     if mobile sensor  $m_j$  can reach grid  $g_i$  then
5:        $\mathbb{E}_b \leftarrow \mathbb{E}_b \cup \{(m_j, g_i)\}$ 
6: Construct a flow network  $\mathbf{G}_f = (\mathbb{V}_f, \mathbb{E}_f)$  based on  $\mathbf{G}_b$  by adding a source  $s$  and a sink  $t$ , where  $\mathbb{V}_f = \mathbb{V}_b \cup \{s, t\}$ , and  $\mathbb{E}_f = \mathbb{E}_b \cup \{(s, m_j) | m_j \in \mathbb{M}\} \cup \{(g_i, t) | g_i \in \mathbb{G}\}$ 
7:  $\triangleright$  For each edge in  $\mathbb{E}_f$ , assign capacity and cost
8: for each edge  $(s, m_j)$ ,  $m_j \in \mathbb{M}$  do
9:   capacity  $u(s, m_j) \leftarrow 1$ , cost of  $(s, m_j) \leftarrow 0$ 
10: for each edge  $(m_j, g_i)$ ,  $m_j \in \mathbb{M}$  and  $g_i \in \mathbb{G}$  do
11:   capacity  $u(m_j, g_i) \leftarrow 1$ 
12:   cost  $c_m(m_j, g_i) \leftarrow$  the movement cost for  $m_j$  to move to grid  $g_i$ 
13:  $D \leftarrow (|\mathbb{M}| + 1) \times$  (the maximal possible movement cost of a mobile sensor)
14: for edge  $(g_i, t)$ ,  $g_i \in \mathbb{G}$  do
15:   capacity  $u(g_i, t) \leftarrow k -$  (the number of static sensors in  $g_i$ )
16:   cost  $c_g(g_i, t) \leftarrow D \cdot [u(g_i, t) - f(g_i, t)]^p$ 
17: find a feasible flow set with maximum-flow minimum-cost from  $s$  to  $t$  in  $\mathbf{G}_f$ , and find the associated optimal matching
18: convert the optimal matching to a movement plan

```

---

least  $k$ -covered. Then, a bipartite graph  $\mathbf{G}_b = (\mathbb{V}_b, \mathbb{E}_b)$  can be constructed such that

- for the vertex set  $\mathbb{V}_b : \mathbb{V}_b = \mathbb{M} \cup \mathbb{G}$ ;
- for the edge set  $\mathbb{E}_b$ : if a mobile sensor  $m_j \in \mathbb{M}$  is capable of moving to grid  $g_i \in \mathbb{G}$ , then there is a directed edge  $(m_j, g_i)$  in  $\mathbb{E}_b$ .

Clearly, the bipartite graph is used to represent the moving capabilities of the mobile sensors. The WSN with  $k = 3$  in Fig. 1 is used as an example, and its initial deployment is repeated in Fig. 3(a). The associated bipartite graph is shown in Fig. 3(b). The mobile sensor set (i.e.,  $\mathbb{M}$ ) is listed on the left-hand side, whereas the set of grids that are needed to be considered (i.e.,  $\mathbb{G}$ ) is listed on the right-hand side. Note that, although grid  $g_2$  has 4 ( $> k = 3$ ) sensors in the initial deployment, it still needs to be included in  $\mathbb{G}$  since the two mobile sensors (i.e., sensors  $m_1$  and  $m_2$ ) may move to other grids. Consider mobile sensor  $m_1$  as an example to demonstrate how to determine the edges in the bipartite graph.  $m_1$  may move to grid  $g_1$  or  $g_4$ , or stay in grid  $g_2$ . Therefore, in the bipartite graph, there are directed edges from  $m_1$  to  $g_1$ ,  $g_2$ , and  $g_4$ .

In the bipartite graph, the value beside a grid is the *mobile sensor demand* of the grid, which is defined as the number of mobile sensors that are needed in the grid to make the grid  $k$ -covered. Therefore, the mobile sensor demand is equal to the difference of the number of static sensors in the grid from  $k$ . In Fig. 3(a), grids  $g_1$  and  $g_3$  have no static sensors and, thus, have the mobile sensor demand being 3. Grids  $g_2$  and  $g_4$  have two and one static sensors and, thus, have mobile sensor demand being 1 and 2, respectively.

2) *Flow Network Formation (Lines 6–16 in Algorithm 1)*: Based on the bipartite graph  $\mathbf{G}_b$ , a flow network can be formed, with the graph denoted  $\mathbf{G}_f = (\mathbb{V}_f, \mathbb{E}_f)$ . In particular, we have the following.

- $\mathbb{V}_f = \mathbb{V}_b \cup \{s, t\}$ , i.e., two special nodes are added, which are a source node  $s$  and a sink node  $t$ .

- $\mathbb{E}_f = \mathbb{E}_b \cup \{(s, m_j) | m_j \in \mathbb{M}\} \cup \{(g_i, t) | g_i \in \mathbb{G}\}$ , i.e., an edge is added from source node  $s$  to each mobile sensor, and an edge is added from each grid to the sink node  $t$ .

For the example WSN in Fig. 3(a), the graph of the associated flow network is shown in Fig. 3(c).

In the flow network, each edge is associated with a capacity. The capacity assignment rule is as follows.

- The capacity of edge  $(s, m_j)$ , which is denoted  $u(s, m_j)$ , is equal to a unit.
- The capacity of edge  $(m_j, g_i)$ , which is denoted  $u(m_j, g_i)$ , is equal to a unit.
- The capacity of edge  $(g_i, t)$ , which is denoted  $u(g_i, t)$ , is equal to the mobile sensor demand of grid  $g_i$ .

In the flow network shown in Fig. 3(c), the values on edges  $(g_i, t)$ 's mean the edge capacities, whereas the capacity of any other edge is a unit (which is not shown in the figure).

Consider the cases when *flows* are sent out from the source node  $s$  to the sink node  $t$ . Here, a flow follows a path from  $s$  to  $t$  and has a unit amount. Therefore, if there are multiple flows on an edge in  $\mathbb{E}_f$ , then the amount of flows on the edge is an integer larger than 1. If there is no flow on an edge, then the amount of flows on the edge is said to be zero. With this setting, a set of flows from  $s$  to  $t$  is considered *feasible* if 1) for each vertex in  $\mathbb{V}_f$  other than  $s$  or  $t$ , the amount of its incoming flows is equal to the amount of its outgoing flows, and 2) the flow amount on any edge in  $\mathbb{E}_f$  does not exceed the capacity of the edge.

Define a *matching* as a subset of the edge set  $\{(m_j, g_i) | m_j \in \mathbb{M}, g_i \in \mathbb{G}\}$ . Therefore, from a feasible flow set, a matching can be determined as follows: An edge  $(m_j, g_i)$  is included in the matching if and only if it has a nonzero amount of flows in the feasible flow set. Since the capacity of an edge  $(m_j, g_i)$  is a unit, any edge in a matching has a unit amount of flows or, equivalently, has a flow. Then, interestingly, a matching can be used to represent a feasible movement plan of the mobile sensors. In particular, if edge  $(m_j, g_i)$  is in a matching, then it means that mobile sensor  $m_j$  moves to grid  $g_i$  in the associated movement plan. Based on this observation, it can be seen that the physical meaning of the unit capacity of edge  $(s, m_j)$  and edge  $(m_j, g_i)$  is to specify that each mobile sensor can move to only one grid (or equivalently, in a matching, there is at most one edge that originates from a vertex  $m_j$ ), and the physical meaning of the capacity of edge  $(g_i, t)$  is to specify grid  $g_i$ 's maximum demand for mobile sensors. Due to the mapping of a feasible set of flows to a matching, and to a feasible movement plan, the three terms are interchangeably used in the sequel and are all denoted by  $\mathcal{P}$ .

Each edge in the flow network is also associated with a cost, as follows.

- The cost of edge  $(s, m_j)$  does not have a physical meaning, and thus, its value is assigned as zero.
- The cost of edge  $(m_j, g_i)$ , which is denoted  $c_m(m_j, g_i)$ , is the movement cost of mobile sensor  $m_j$  if it moves to grid  $g_i$ .
- The cost of edge  $(g_i, t)$ , which is denoted  $c_g(g_i, t)$ , is defined as  $D \cdot [u(g_i, t) - f(g_i, t)]^p$ , where  $f(a, b)$  means

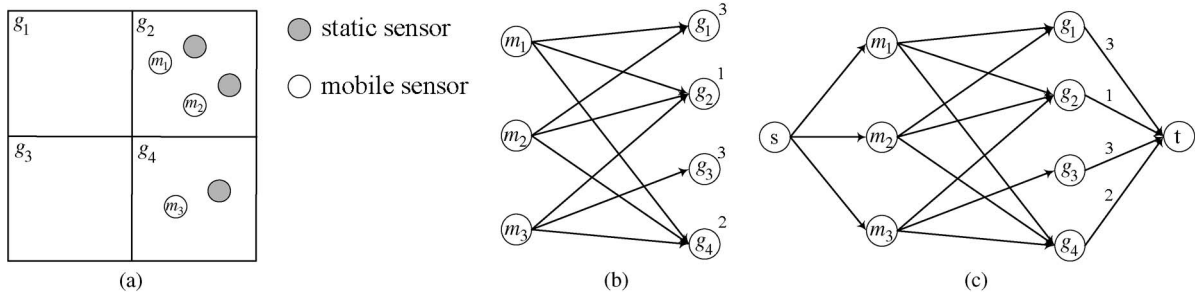


Fig. 3. WSN when the target of each grid is to be at least 3-covered. (a) Initial deployment; (b) bipartite graph; (c) flow network.

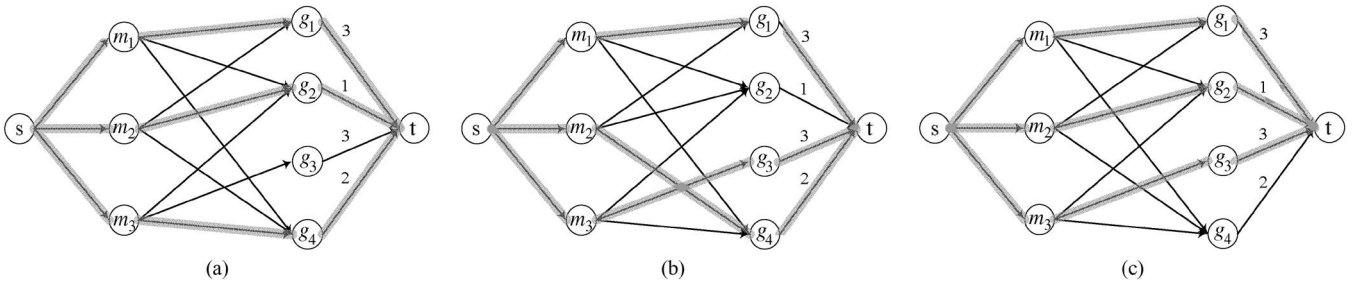


Fig. 4. Optimal set of flows in the flow network in Fig. 3(c) when (a)  $p = 1$ , (b)  $p = 2$ , and (c)  $p = \infty$ .

the amount of flows on an edge  $(a, b) \in \mathbb{E}_f$ , and  $D$  is a constant given by

$$D = (|\mathbb{M}| + 1) \cdot (\text{the maximal possible movement cost of a mobile sensor}) \quad (4)$$

where  $|\mathbb{M}|$  is the number of elements in  $\mathbb{M}$ .

We refer to the cost of edge  $(g_i, t)$ , i.e.,  $c_g(g_i, t)$ , as the *gap cost* since it is related to the gap of grid  $g_i$ , as shown in the following. Recall that the capacity of edge  $(g_i, t)$  is the difference of the number of static sensors in grid  $g_i$  from  $k$  and that a unit amount of flows into grid  $g_i$  means moving a sensor to the grid. Therefore, we have

$$u(g_i, t) - f(g_i, t) = r_i. \quad (5)$$

Note that, here,  $r_i$  means the gap of grid  $g_i$  after the associated movement plan is implemented. Therefore, the cost of edge  $(g_i, t)$  is equal to  $D \cdot r_i^p$ , which is related to the gap of grid  $g_i$ .

The motivation of designing the gap cost is to achieve the objective of step 2 in the three-step optimization problem, that is, minimizing the  $L_p$ -norm of the gap vector, i.e.,  $\|\mathbf{r}\|_p$ . Since  $\|\mathbf{r}\|_p = (r_1^p + r_2^p + \dots + r_n^p)^{1/p}$ , for  $1 \leq p < \infty$ , minimizing  $\|\mathbf{r}\|_p$  is equivalent to minimizing  $(r_1^p + r_2^p + \dots + r_n^p)$ . Considering that the gap cost of grid  $g_i$  is  $D \cdot r_i^p$  and that  $D$  is a constant (will be explained later), a feasible set of flows that can minimize the total gap cost of all grids will also minimize  $\|\mathbf{r}\|_p$ ,  $1 \leq p < \infty$ . Next, we will discuss how to find such a feasible set of flows.

3) *Optimal Matching Computation (Lines 17 and 18 in Algorithm 1)*: For the flow network, we can find an opti-

mal feasible set of flows from  $s$  to  $t$ , which is denoted  $\mathcal{P}^*$ , which maximizes the total amount of flows from  $s$  to  $t$  (given by  $\sum_{m_j \in \mathbb{M}} f(s, m_j)$ ) and also minimizes the following cost function:

$$C(\mathcal{P}^*) = \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f(m_j, g_i) \cdot c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} c_g(g_i, t). \quad (6)$$

The cost of the flows (or the corresponding movement plan) given in (6) consists of two components: the movement cost and the gap cost. It can be seen that the gap cost and the movement cost correspond to the objective functions in steps 2 and 3 of the three-step optimization problem in Section II, respectively. This means that  $C(\mathcal{P}^*)$  manages to combine the two objectives in the three-step optimization problem in a single form. However, according to the three-step optimization problem, step 2 should have a higher priority over step 3, that is, it is required to minimize first the  $L_p$ -norm of the gap vector and then to minimize the movement cost. To guarantee this priority, we have introduced in the gap cost function a constant  $D$  [defined in (4)] that is a strict upper bound of the total movement cost, which makes the flow computation first minimize the total gap cost and then minimize the total movement cost.

As an example, Fig. 4 gives the optimal feasible set of flows in the flow network in Fig. 3(c) when  $p = 1$ ,  $p = 2$ , or  $p = \infty$ . Based on the optimal feasible set of flows, the optimal matching and the optimal movement plan can be determined. For instance, when  $p = 1$ , as shown in Fig. 4(a), the optimal matching is the set  $\{(m_1, g_1), (m_2, g_2), (m_3, g_4)\}$ , and the optimal movement plan is to move mobile sensor  $m_1$  to grid  $g_1$  and keep mobile sensors  $m_2$  and  $m_3$  stay in grids  $g_2$  and  $g_4$ , respectively.

Since the flow cost of each edge is a convex function of the amount of flows through it,<sup>2</sup> the *capacity scaling algorithm* given in [22] can be used to find the optimal flow set with the maximum amount of flows and minimum cost, with computation complexity  $O((|\mathbb{E}_f|^2 + |\mathbb{E}_f| \cdot |\mathbb{V}_f| \cdot \log |\mathbb{V}_f|) \log k)$ . This is also the complexity of the  $\text{BMrD}_{k,p}$  algorithm, since the complexity of bipartite graph construction and flow network formation is negligible.

*Theorem 1:* The movement plan that is determined in  $\text{BMrD}_{k,p}$  ( $1 \leq p < \infty$ ) is an optimal movement plan for the three-step optimization problem defined in Section II.

*Proof:* See the Appendix.

### B. $\text{BMrD}_{k,\infty}$ Algorithm

When  $p = \infty$ , the objective of step 2 is to minimize  $\|\mathbf{r}\|_\infty$ , i.e., minimizing the maximum gap of all grids. This objective is equivalent to finding the maximal possible coverage, for example,  $\alpha^*$ , that each grid can at least be. In other words, there exists a movement plan that can make each grid at least  $\alpha^*$ -covered, but there does not exist a movement plan that can make each grid at least  $(\alpha^* + 1)$ -covered. Then, when the maximum gap is minimized, it is equal to  $k - \alpha^*$ .

Because  $\alpha^*$  is an integer value between 0 and  $k$ , we can use a binary search method to find this value. At each search value, for example,  $\alpha$ , we need to determine whether there exists a movement plan that can make each grid at least  $\alpha$ -covered. To do this, we still use the method in the preceding subsection. The difference from the procedure in the preceding subsection lies in that we use a convex function  $h_\alpha(x)$  (to be defined) with parameter  $\alpha$  as the gap cost function. Based on the new convex gap cost function, we can determine whether each grid can be at least  $\alpha$ -covered.

The function  $h_\alpha(x)$  (in which  $x \in [0, k]$  means the grid gap,<sup>3</sup> and  $\alpha \in [0, k]$  is an integer parameter) is defined as any continuous, increasing, and convex function that satisfies the following:

$$\begin{cases} h_\alpha(x) = Ax, & \text{if } 0 \leq x \leq k - \alpha \\ h_\alpha(x) \geq Ax + 1, & \text{if } x = k - \alpha + 1 \end{cases} \quad (7)$$

where  $A$  is a constant with a positive value. The definition is demonstrated in Fig. 5. We can see that, when  $0 \leq x \leq k - \alpha$ ,  $h_\alpha(x)$  is a line from  $(0, 0)$  to  $(k - \alpha, A(k - \alpha))$ . When  $k - \alpha + 1 \leq x \leq k$ , the above definition implies that  $h_\alpha(x)$  should be in the shaded area with the left lower corner being  $(k - \alpha + 1, A(k - \alpha + 1) + 1)$ .

A lot of functions can satisfy the above condition (7) of  $h_\alpha(x)$ . For instance, a realization of  $h_\alpha(x)$  is the modified Huber function [23] given by

$$h_\alpha(x) = \begin{cases} (k - \alpha)x, & 0 \leq x \leq k - \alpha \\ x^2, & k - \alpha \leq x \leq k \end{cases} \quad (8)$$

as demonstrated in Fig. 6 when  $k = 4$  and  $\alpha = 2$ .

<sup>2</sup>The flow cost of edge  $(g_i, t)$  is  $D \cdot [u(g_i, t) - f(g_i, t)]^p$ , which is a convex function of the amount of flows. The flow cost of edge  $(m_i, g_i)$  is equal to  $f(m_j, g_i) \cdot c_m(m_j, g_i)$ , which is a linear function of the amount of flows, as well as a convex function of the amount of flows.

<sup>3</sup>Note that, although  $x$  is used to stand for the integer-valued grid gap, the function  $h_\alpha(x)$  is still defined on continuous  $x$  values.

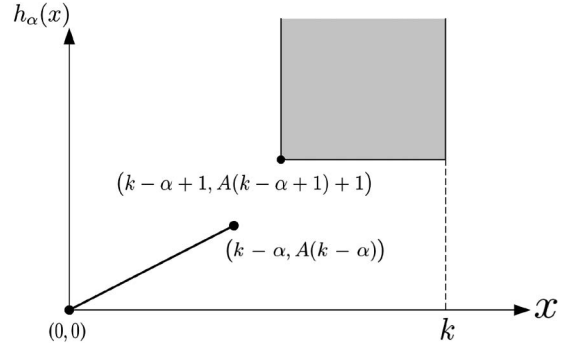


Fig. 5. Function  $h_\alpha(x)$ .

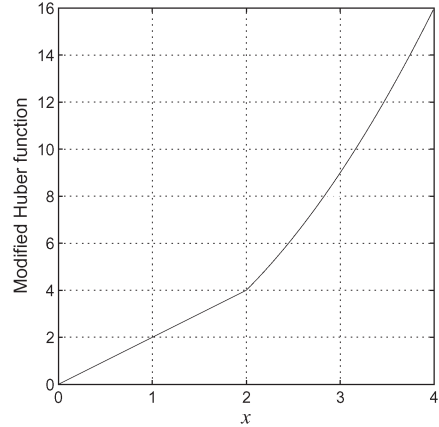


Fig. 6. Modified Huber function when  $k = 4$  and  $\alpha = 2$ .

TABLE III  
ALGORITHM 2:  $\text{BMrD}_{k,\infty}$

#### Algorithm 2 $\text{BMrD}_{k,\infty}$

- 1:  $L \leftarrow 0, U \leftarrow k, \alpha^* = 0$
- 2: Construct a bipartite graph  $\mathbf{G}_b = (\mathbb{V}_b, \mathbb{E}_b)$
- 3: Construct the flow network  $\mathbf{G}_f = (\mathbb{V}_f, \mathbb{E}_f)$  without setting  $c_g(g_i, t)$ ,  $g_i \in \mathbb{G}$
- 4: **while**  $U > L$  **do**
- 5:    $\alpha \leftarrow \lceil (L + U)/2 \rceil$
- 6:   assign  $c_g(g_i, t) \leftarrow D \cdot h_\alpha(u(g_i, t) - f(g_i, t))$
- 7:   find the optimal matching  $\mathcal{P}$
- 8:   **if**  $\mathcal{P}$  can make all grids at least  $\alpha$ -covered **then**
- 9:      $L \leftarrow \alpha; \alpha^* \leftarrow \alpha; \mathcal{P}^* \leftarrow \mathcal{P}$
- 10:   **else**
- 11:      $U \leftarrow \alpha - 1$
- 12: The optimal matching  $\mathcal{P}^*$  leads to the optimal movement plan of the three-step optimization problem. And  $\alpha^*$  is the maximum value of  $\alpha$  such that each grid can be at least  $\alpha$ -covered.

Based on the convex function  $h_\alpha(x)$ , in the  $\text{BMrD}_{k,\infty}$  algorithm, we use a binary search method to find the maximum value of  $\alpha$  such that each grid can be at least  $\alpha$ -covered. The algorithm is detailed in Algorithm 2 in Table III.

In the initialization phase of the algorithm (lines 1–3), similar to  $\text{BMrD}_{k,p}$  ( $1 \leq p < \infty$ ), the bipartite graph and the flow graph are constructed, except that the gap cost of edge  $(g_i, t)$  is to be determined. A lower bound and an upper bound of  $\alpha$  in the binary search are initially set up as  $L = 0$  and  $U = k$ , respectively.

In the loop of the algorithm (lines 4–11), a binary search is conducted to find the maximum  $\alpha$ , which is denoted  $\alpha^*$ , such that each grid can be at least  $\alpha^*$ -covered. The binary search

starts with  $\alpha = \lceil k/2 \rceil$ . Then, the gap cost of an edge  $(g_i, t)$  in the flow network is assigned as  $c_g(g_i, t) = D \cdot h_\alpha(u(g_i, t) - f(g_i, t))$ , and the optimal matching of the flow network can be obtained. If the associated movement plan can make each grid at least  $\alpha$ -covered, we set  $L \leftarrow \alpha$  and  $\alpha^* \leftarrow \alpha$ ; otherwise, we set  $U \leftarrow \alpha - 1$ . This procedure is repeated until  $U = L$ . Then, the optimal matching when  $h_{\alpha^*}(x)$  is used as the gap cost function<sup>4</sup> will devise the optimal movement plan of the three-step optimization problem with  $p = \infty$ , to be proved.

In the BMrD $_{k,\infty}$  algorithm, the binary search has  $\log k$  rounds. The computation complexity in each round is  $O((|\mathbb{E}_f|^2 + |\mathbb{E}_f| \cdot |\mathbb{V}_f| \cdot \log |\mathbb{V}_f|) \log k)$ . Therefore, the total computation complexity of the BMrD $_{k,\infty}$  algorithm is  $O((|\mathbb{E}_f|^2 + |\mathbb{E}_f| \cdot |\mathbb{V}_f| \cdot \log |\mathbb{V}_f|) \log^2 k)$ .

Before we prove the optimality of the above algorithm, we provide some insights into the gap cost function  $h_\alpha(x)$ . We use the modified Huber function in (8) as an example. Recall that our objective in each round of the binary search is to find a movement plan that can make each grid at least  $\alpha$ -covered. From (8), it can be seen that the gap cost function linearly increases with grid gap  $x$  when the grid is more than  $\alpha$ -covered (i.e., the gap is smaller than  $k - \alpha$ ) and increases faster when the grid is less than  $\alpha$ -covered (i.e., the gap is larger than  $k - \alpha$ ). Because there is more penalty for less than  $\alpha$ -covered grids, the algorithm tends to make each grid at least  $\alpha$ -covered.

*Theorem 2:* The movement plan determined by the BMrD $_{k,\infty}$  algorithm is an optimal movement plan for the three-step optimization problem with  $p = \infty$ .

*Proof:* See the Appendix.

### C. Extension to the Case With Heterogeneous Sensors

In our problem statement in Section II, except for the different mobility capabilities, all sensors are homogeneous. However, in some scenarios, there might be different types of sensors (for example, to sense smoke, temperature, pressure, etc.). Suppose that there are  $L$  types of sensors. Each grid is desired to be  $k_l$ -covered by type  $l$  sensors,  $l \in \{1, 2, \dots, L\}$ . Let  $r_{i,l}$  denote the gap of grid  $g_i$  in terms of type  $l$  sensors, which is defined as the difference of the number of type  $l$  sensors in grid  $g_i$  from  $k_l$ . The overall gap of grid  $g_i$  is given by  $r_i = \sum_{l=1}^L r_{i,l}$ . Then, similar to Section II, a three-step optimization problem can also be formulated, in which step 2 is to minimize the  $L_p$ -norm of the overall gap vector  $(r_1, r_2, \dots, r_n)$ . The BMrD algorithms can still be used to solve the problems for different  $p$  values, with the following modifications.

In the bipartite graph, a column of subgrids is inserted between the column of the mobile sensors and the column of grids. Here, subgrids are virtual representation of grids. Each grid (for example, grid  $g_i$ ) is represented by a number  $L$  of subgrids (for example, subgrids  $g_{i,1}, g_{i,2}, \dots, g_{i,L}$ ), each for a sensor type. If a mobile sensor  $m_j$  is of type  $l$  and is capable of moving to grid  $g_i$ , then there is a directed edge from  $m_j$  to

subgrid  $g_{i,l}$ . For each grid  $g_i$ , there is a directed edge from each of its subgrids to itself.

In the bipartite graph, the mobile sensor demand of a subgrid  $g_{i,l}$  is the difference of the number of type- $l$  static sensors in grid  $g_i$  from the value  $k_l$ . Furthermore, the mobile sensor demand of a grid is the summation of the mobile sensor demand values of its subgrids in the bipartite graph.

After source  $s$  and sink  $t$  are added, a flow network can be constructed, and the capacity and cost assignment rule is as follows.

- The capacity and cost-assignment rule for edges from  $s$  to mobile sensors and for edges from grids to  $t$  is the same as the rule in the BMrD $_{k,p}$  algorithms.
- For an edge from  $m_j$  to  $g_{i,l}$ : The capacity has a unit value, and the cost is the movement cost of sensor  $m_j$  if it moves to grid  $g_i$ .
- For an edge from  $g_{i,l}$  to  $g_i$ : The capacity is equal to the mobile sensor demand of subgrid  $g_{i,l}$ ; the cost does not have a physical meaning and, thus, has a value of zero.

After the flow network is constructed, an optimal matching in the flow network will lead to an optimal movement plan.

For presentation simplicity, in the discussion in Section IV and performance evaluation in Section V, we consider only the case with homogeneous sensors.

### D. Extension to the Case With Heterogeneous Coverage Requirements

In our problem statement in Section II, it is assumed that the coverage requirement of each grid is the same: to be at least  $k$ -covered. In some applications, the coverage requirement of the grids may be different. Some crucial grids need more sensors, whereas some unimportant grids need fewer sensors (even no sensors). BMrD algorithms can be extended to deal with the case with heterogeneous coverage requirements. Assume that the target of grid  $g_i$  ( $1 \leq i \leq n$ ) is to be at least  $k_i$ -covered. Therefore, for grid  $g_i$ , the gap is  $r_i = \max\{k_i - |g_i|, 0\}$ . Define the unsatisfaction level of grid  $g_i$  as  $e_i = r_i/k_i$ , and define  $\mathbf{e} = \{e_1, e_2, \dots, e_n\}$  as the unsatisfaction level vector. Then, we have the following three-step optimization problem.

Step 1) Minimizing the sum of gaps, i.e.,

$$\underset{\mathcal{P}}{\text{minimize}} \quad \sum_{i=1}^n r_i$$

subject to moving capability of each mobile sensor.

Let  $\mathbb{F}_1$  denote the set of all optimal movement plans in step 1.

Step 2) Minimizing the  $L_p$ -norm of the unsatisfaction level vector, i.e.,

$$\underset{\mathcal{P} \in \mathbb{F}_1}{\text{minimize}} \quad \|\mathbf{e}\|_p. \quad (9)$$

Let  $\mathbb{F}_2$  denote the set of all optimal movement plans in step 2.

<sup>4</sup>For presentation simplicity, when we say “ $h_\alpha(x)$  is used as the gap cost function,” it means “ $D \cdot h_\alpha(u(g_i, t) - f(g_i, t))$  is used as the gap cost function for edge  $(g_i, t)$ .”



Step 3) Minimizing the movement cost, i.e.,

$$\underset{\mathcal{P} \in \mathbb{F}_2}{\text{minimize}} \quad c_m(\mathcal{P}). \quad (10)$$

The three-step optimization problem can be solved by similar BMrD algorithms. Here, we only list the differences of the BMrD algorithms from those in Sections III-A and B.

- When  $1 \leq p < \infty$ : Line 16 of Algorithm 1 is changed to cost  $c_g(g_i, t) \leftarrow D \cdot [w_i \cdot (u(g_i, t) - f(g_i, t))]^p$ , where  $w_i = M/k_i$ , and  $M$  is the least common multiple of  $k_1, k_2, \dots, k_n$ .
- When  $p = \infty$ : For grid  $g_i$ , define  $(M/k_i)(k_i - r_i)$  and  $(M/k_i)r_i$  as the *normalized satisfaction level* and the *normalized unsatisfaction level*, respectively, which are integer values. For a specific target value  $\alpha$  of the normalized satisfaction level, let  $h_\alpha^i(x)$  define any continuous, increasing, and convex function that satisfies the following:

$$\begin{cases} h_\alpha^i(x) = Ax, & \text{if } 0 \leq (M/k_i)x \leq M - \alpha \\ h_\alpha^i(x) \geq Ax + 1, & \text{if } (M/k_i)x = M - \alpha + 1 \end{cases} \quad (11)$$

where  $x$  stands for the grid gap,  $(M/k_i)x$  stands for the normalized unsatisfaction level, and  $A$  is a constant with a positive value. Then, in Algorithm 2, line 6 should be as follows: assign  $c_g(g_i, t) \leftarrow D \cdot h_\alpha^i(u(g_i, t) - f(g_i, t))$ . In Algorithm 2, the search range for  $\alpha$  is from 0 to  $M$ , and the objective is to find the maximal value of  $\alpha$  such that each grid has at least  $\alpha$ -normalized coverage (i.e., the normalized satisfaction level is at least  $\alpha$ ).

It can be proved that the above BMrD algorithms can find an optimal movement plan for the three-step optimization problem define at the beginning of this subsection. The proofs are similar to those in Theorems 1 and 2 and, thus, are omitted here.

For presentation simplicity, in the discussion in Section IV and performance evaluation in Section V, we consider only the case when the coverage requirement of each grid is to be at least  $k$ -covered.

#### E. Features of the Framework

It can be seen that our framework for the optimal mobile sensor redeployment problem is generic in the following aspects.

- A generic system model: A hybrid WSN with both static and mobile sensors is considered; the initial sensor distribution, the moving capabilities of the mobile sensors, the measures of the movement costs, and the sensor types are all arbitrary, and the coverage requirements in different grids can be heterogeneous.
- A generic cost function: The  $L_p$ -norm represents a family of cost functions with different design objectives.
- A generic method: The BMrD algorithms are used to solve problems with different  $p$  values.

### IV. PROPERTIES IN SPECIAL SCENARIOS

Here, some interesting properties of the BMrD algorithms are discussed in special scenarios. The following fact is used in the proofs of the properties:  $\mathbb{F}_1$  and  $\mathbb{F}_2$  are the sets of the

optimal solutions in steps 1 and 2 of the three-step optimization problem as defined in Section II, respectively.

*Lemma 1:* If there exists a feasible movement plan that can make each grid at least  $k$ -covered, then BMrD $_{k,p}$  algorithms are independent of the value of  $p$  (i.e., for whatever value of  $p$ , BMrD $_{k,p}$  results in a movement plan that makes all grids at least  $k$ -covered and has the minimum movement cost).

*Proof:* Consider a movement plan  $\mathcal{P}$  that can make each grid at least  $k$ -covered. After  $\mathcal{P}$  is implemented, each grid has a zero-valued gap, which minimizes the sum of gaps. Therefore, it can be concluded that  $\mathbb{F}_1$  is the set of movement plans that can make each grid at least  $k$ -covered. Recall that  $\mathbb{F}_2$  is the set of movement plans from  $\mathbb{F}_1$  such that the  $L_p$ -norm of the gap vector is minimized. Since any movement plan in  $\mathbb{F}_1$  makes each grid have a zero-valued gap, it also minimizes the  $L_p$ -norm of the gap vector for whatever value of  $p$  (i.e.,  $\|\mathbf{r}\|_p = 0$ ). Therefore, we have  $\mathbb{F}_2 = \mathbb{F}_1$  for whatever value of  $p$ . Then, the optimal solution to the three-step optimization problems is to select a movement plan in  $\mathbb{F}_1$  (or, equivalently,  $\mathbb{F}_2$ ) whose movement cost is the minimum. Hence, BMrD $_{k,p}$  algorithms are independent of  $p$ , and the optimal solution is a movement plan that makes all grids at least  $k$ -covered and has the minimum movement cost. ■

*Lemma 2:* If there does not exist a movement plan such that each grid is at least 1-covered, then the BMrD $_{k,1}$  and BMrD $_{k,\infty}$  algorithms are equivalent.

*Proof:* In step 2 of the three-step optimization problem, we have the following.

- When  $p = 1$ : Apparently, the set of optimal solutions in step 2 is the same as  $\mathbb{F}_1$ .
- When  $p = \infty$ : Since there does not exist a movement plan such that each grid is at least 1-covered, any solution in  $\mathbb{F}_1$  will lead to the maximum gap being  $k$ . Therefore, the set of optimal solutions in step 2 is the same as  $\mathbb{F}_1$ .

Then, it can be concluded that the three-step optimization problems with  $p = 1$  and  $p = \infty$  are equivalent. As their optimal solutions, the BMrD $_{k,1}$  and BMrD $_{k,\infty}$  algorithms are also equivalent. ■

*Lemma 3:* An optimal movement plan from the BMrD $_{k,1}$  algorithm has the minimum movement cost among those optimal movement plans from the BMrD $_{k,p}$  algorithms when  $1 \leq p \leq \infty$ .

*Proof:* In step 2 of the three-step optimization problem, we have the following.

- When  $p = 1$ : Apparently, the set of optimal solutions in step 2 is the same as  $\mathbb{F}_1$ .
- When  $p \geq 1$ : The set of optimal solutions in step 2 is a subset of  $\mathbb{F}_1$ .

In step 3 of the three-step optimization problem, a movement plan with the minimum movement cost is found in  $\mathbb{F}_1$  when  $p = 1$  or in a subset of  $\mathbb{F}_1$  when  $p \geq 1$ . Therefore, an optimal movement plan of the three-step optimization problem with  $p = 1$  has the minimum movement cost. Equivalently, an optimal movement plan from the BMrD $_{k,1}$  algorithm has the minimum movement cost among those optimal movement plans from the BMrD $_{k,p}$  algorithms when  $1 \leq p \leq \infty$ . ■

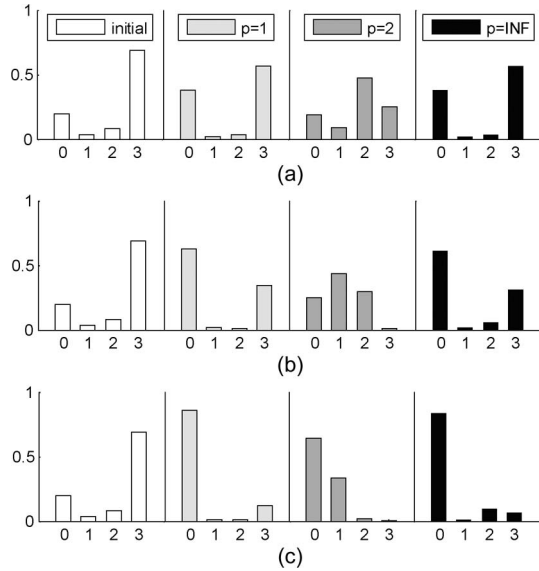


Fig. 7. Gap distribution for BMrD algorithms when  $k = 3$  and  $\sigma = 10$  m. (a)  $\lambda = 20\%$ ; (b)  $\lambda = 50\%$ ; (c)  $\lambda = 80\%$ .

V. PERFORMANCE EVALUATION

Extensive simulations are carried out over a customized C++ simulator to evaluate the performance of the BMrD algorithms when  $p$  takes one of the three typical values: 1, 2, and  $\infty$ . When  $p = \infty$ , the modified Huber function given in (8) is used as the gap cost. A WSN is considered where the target of each grid is to be at least 3-covered, i.e.,  $k = 3$ . The WSN is to be deployed in a  $100 \text{ m} \times 100 \text{ m}$  square area, which is further partitioned into 100 square-shaped grids, each with the side length being 10 m. Three hundred sensors are deployed. In the initial deployment, both the horizontal and vertical coordinates of each sensor follow a normal distribution with a mean value being 50 m and a standard deviation being  $\sigma = 10, 25, \text{ or } 50$  m. A larger  $\sigma$  means that the sensors are relatively uniformly distributed. Among all the sensors, a percentage  $\lambda$  of them are mobile sensors, and the value of  $\lambda$  varies from 20%, 50%, to 80%.  $\lambda$  is also termed the *mobile sensor percentage*. Mobile sensors are randomly selected from all the sensors. The maximum moving distance values of the mobile sensors are independent and randomly selected from the range [10 m, 50 m]. If a mobile sensor moves to a grid, it moves to the center point of the grid,<sup>5</sup> and the movement cost is measured by the moving distance. In the following, each statistic is collected in 50 simulation runs.

First, we fix the value  $\sigma$  at 10 m. Fig. 7 shows the gap distribution of all the grids after the BMrD algorithm is implemented, when  $p$  is 1, 2, or  $\infty$  (“INF”), and  $\lambda$  is 20%, 50%, or 80%. In Fig. 7, the values of 0, 1, 2, and 3 in the horizontal axis mean the gap values, whereas the vertical axis means the probabilities of the gap values. The gap distribution in the initial deployment (denoted by “initial”) is also given for reference.

From Fig. 7, it is observed that the BMrD <sub>$k,1$</sub>  algorithm tends to have the maximum number of 3-covered (with gap value 0)

<sup>5</sup>This requirement is to simplify the simulations. In fact, in our framework, a mobile sensor can move to any position in the grid according to the requirement of specific applications.

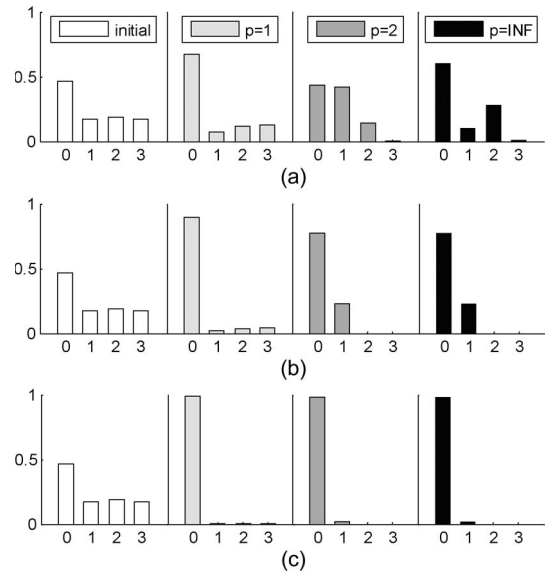


Fig. 8. Gap distribution for BMrD algorithms when  $k = 3$  and  $\sigma = 25$  m. (a)  $\lambda = 20\%$ ; (b)  $\lambda = 50\%$ ; (c)  $\lambda = 80\%$ .

grids, among all BMrD <sub>$k,p$</sub>  algorithms with different  $p$ . This is reasonable as follows. The BMrD <sub>$k,2$</sub>  algorithm minimizes the gap variance (or, equivalently, maximizes the Jain’s fairness index) and, thus, will lead to a fair distribution of the gaps. Therefore, the number of 3-covered grids is likely to be less (than the number in BMrD <sub>$k,1$</sub> ). The BMrD <sub>$k,\infty$</sub>  algorithm needs to minimize the maximum gap. Thus, sensors from initially 3-covered grids may move to the grids that initially have the smallest number of sensors. Therefore, when  $p = \infty$ , the number of 3-covered grids is also very likely to be less.

In Fig. 7, it is also observed that the BMrD <sub>$k,1$</sub>  and BMrD <sub>$k,\infty$</sub>  algorithms have the same gap distribution when  $\lambda = 20\%$ . This is because the value of  $\sigma$  is a very small value, i.e., 10 m. A very small  $\sigma$  means that a very large portion of the sensors are located around the center of the WSN. When only a small percentage (e.g.,  $\lambda = 20\%$ ) of the sensors are mobile sensors, it is very likely that there does not exist a movement plan that makes each grid at least 1-covered. Then, from Lemma 2, the BMrD <sub>$k,1$</sub>  and BMrD <sub>$k,\infty$</sub>  algorithms are equivalent.

From Fig. 7, it can also be seen that, with the increase in the mobile sensor percentage  $\lambda$ , more grids are with higher coverage, i.e., are 3-covered or 2-covered.

To evaluate the impact of the initial deployment of the sensors, Figs. 8 and 9 show the gap distribution of the BMrD <sub>$k,p$</sub>  algorithms when  $\sigma = 25$  m (i.e., when sensors are initially deployed less uniformly) and when  $\sigma = 50$  m (i.e., when sensors are initially deployed more uniformly), respectively.

- When the mobile sensor percentage is low [e.g.,  $\lambda = 20\%$ , as shown in Figs. 8(a) and 9(a)]: The BMrD <sub>$k,1$</sub>  algorithm still tends to make the largest number of 3-covered grids, and BMrD <sub>$k,2$</sub>  still makes the gap distribution more fair. On the other hand, BMrD <sub>$k,\infty$</sub>  makes a good tradeoff between the largest number of 3-covered grids and fairness.
- When the mobile sensor percentage increases, e.g., to  $\lambda = 50\%$ , as shown in Figs. 8(b) and 9(b): The BMrD <sub>$k,\infty$</sub>  algorithm tends to be similar to BMrD <sub>$k,2$</sub>  since both the BMrD <sub>$k,2$</sub>  and BMrD <sub>$k,\infty$</sub>  algorithms tend to

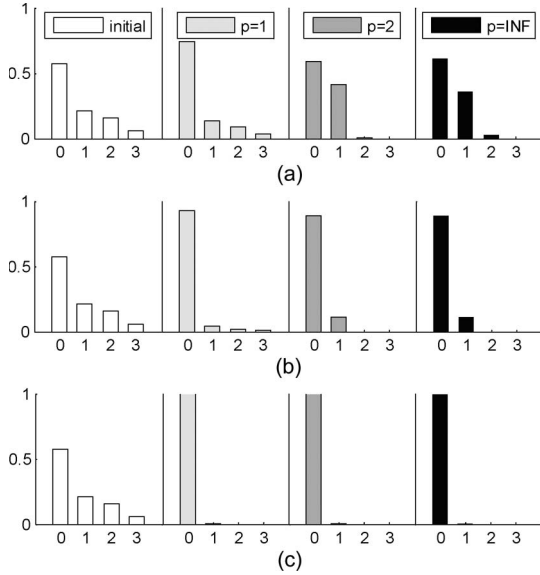


Fig. 9. Gap distribution for BMrD algorithms when  $k = 3$  and  $\sigma = 50$  m. (a)  $\lambda = 20\%$ ; (b)  $\lambda = 50\%$ ; (c)  $\lambda = 80\%$ .

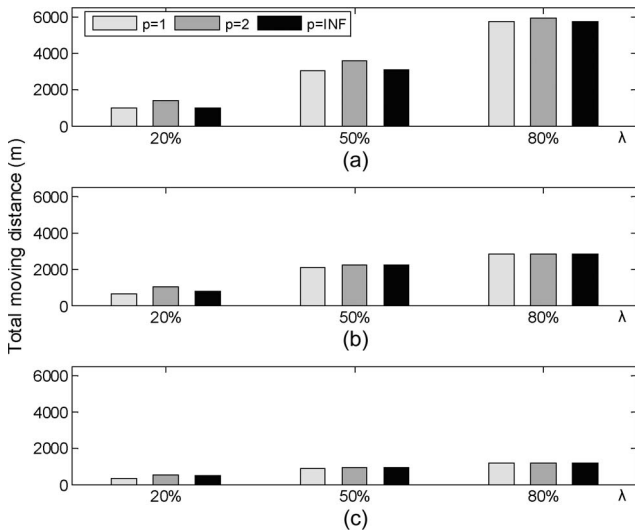


Fig. 10. Total moving distance of the movement plans decided by BMrD algorithms. (a)  $\sigma = 10$  m; (b)  $\sigma = 25$  m; (c)  $\sigma = 50$  m.

decrease the number of 0-covered grids. Interestingly, when all the grids can be at least 2-covered, the  $\text{BMrD}_{k,2}$  and  $\text{BMrD}_{k,\infty}$  algorithms are equivalent, as shown in Figs. 8(b) and 9(b). This is because when  $\alpha = 2$ , the gap cost in the  $\text{BMrD}_{k,\infty}$  algorithm is 0, 1, 4, and 9 when the gap is 0, 1, 2, and 3, respectively, based on (8). This is exactly the same as the gap cost in the  $\text{BMrD}_{k,2}$  algorithm.

- When the mobile sensor percentage increases to a very large value, e.g.,  $\lambda = 80\%$ : It is very likely that all the grids can be at least 3-covered. Then, based on Lemma 1, it is very likely that the  $\text{BMrD}_{k,1}$ ,  $\text{BMrD}_{k,2}$ , and  $\text{BMrD}_{k,\infty}$  algorithms are equivalent, as shown in Figs. 8(c) and 9(c).

The total moving distance of the mobile sensors in the three BMrD algorithms is shown in Fig. 10, when  $\sigma = 10$ , 25, or 50 m, and  $\lambda = 20\%$ , 50%, or 80%. Apparently, the  $\text{BMrD}_{k,1}$  algorithm has the lowest movement cost, which is consistent with Lemma 3. On the other hand, the  $\text{BMrD}_{k,2}$

algorithm has the highest movement cost. This is because the algorithm needs to distribute the gaps among the grids in a fairer manner, and thus, it is likely that more mobile sensors need to move.

As a summary, in our generic redeployment framework,  $\text{BMrD}_{k,1}$  has the lowest movement cost and tends to have the maximum number of  $k$ -covered grids;  $\text{BMrD}_{k,2}$  can be applied to achieve an even distribution of sensors, which is helpful to prolong the network lifetime or balance the workload of sensors, at the cost of the highest movement cost; and  $\text{BMrD}_{k,\infty}$  lies in between  $\text{BMrD}_{k,1}$  and  $\text{BMrD}_{k,2}$  in terms of the number of  $k$ -covered grids and even sensor distribution.

## VI. RELATED WORKS

As mentioned in Section I, most mobile sensor redeployment algorithms are designed based on three basic techniques: virtual force, coverage pattern, or grid architecture. Since our work belongs to the category of grid architecture, here, we mainly discuss related works in this category.

The work in [18] investigates how to deal with sensor failure events. Once sensor failure events occur, redundant sensors are found and are cooperatively moved to fix the problem. When designing the movement plan of the sensors, the responding time is used as a constraint.

In [14], a load-balanced state is achieved by mobile sensor redeployment, in which the difference between the numbers of sensors in any two grids is upper bounded by one. A scan-based redeployment algorithm named SMART is proposed, which has two scan processes (for rows and for columns). Based on the two scan processes, sensors in overloaded grids are moved to underloaded grids such that a balanced sensor distribution can be eventually reached. To further improve SMART, the Hungarian method is used in [15] to minimize the movement cost. A complete bipartite graph is used to model the load balance problem. The graph has two sets of vertices: overloaded grids and underloaded grids. An edge exists from an overloaded grid to any underloaded grid, and the cost of the edge is the moving distance from the overloaded grid to the underloaded grid. After a perfect matching of the bipartite graph is found, the optimal movement plan can be obtained. Note that in [14] and [15], all sensors are assumed to be mobile and have the capability to move to any grid in the area (i.e., with unlimited mobility capability).

Redeployment with limited mobility capability is studied in [16] and [17]. The work in [16] targets at minimum movement cost to deal with the coverage holes and to achieve 1-coverage. The work in [17] is more general, since it studies the  $k$ -coverage problem, and its objectives are the minimum variance of the numbers of sensors in the grids and the minimum movement cost. The problem is modeled by a *virtual graph*. In particular, each grid is modeled by three basic vertices (a base vertex that represents the source of sensors in the grid, and output and input vertices responsible for sending and receiving mobile sensors, respectively) and  $k$  sink vertices denoting the coverage requirement of the grid. The objective of minimizing the variance of the numbers of sensors in the grids, which is a nonlinear objective, is converted into a linear objective by

assigning different weights for the  $k$  sinks. A minimum-cost maximum weighted flow can be found on the virtual graph, and the optimal movement strategy of mobile sensors can be obtained accordingly.

Although we also use the network flow technique, there are several key differences between our work and the work in [17].

- We use a bipartite graph to describe the mobile sensor redeployment problem. Compared with a virtual graph, our model is more general. For instance, since the virtual graph is based on the grids, it implicitly assumes that all mobile sensors in one grid have the same mobility capabilities. However, in our model, mobile sensors in one grid can have different mobility capabilities. The advantage of using a virtual graph model over a bipartite graph model is that the virtual graph may have fewer edges, which helps decrease the computation complexity. To be specific, assume that there are  $m$  mobile sensors and  $n$  grids, and the maximum mobility capability of a mobile sensor is  $H$  hops. The virtual graph has  $O(kn)$  vertices and  $O(nH)$  edges, whereas the bipartite graph has  $O(m+n)$  vertices and  $O(mH)$  edges. Since both methods use the minimum-cost flow algorithm (e.g., the capacity scaling algorithm) to compute the movement schemes, they have the same order of computation complexity when  $m = O(n)$ . However, when  $m$  is far greater than  $n$  [e.g.,  $m = O(n \log n)$  or  $m = O(n^2)$ , etc.], the virtual graph method will have lower computation complexity than that of the bipartite graph method. However, as pointed out before, the virtual graph method can only deal with cases when 1) the sensor network is a flip-based network; 2) the movement cost is measured by hops; and 3) the mobile sensors in one grid have the same mobility capabilities.
- In this paper, we use a family of optimization objectives ( $p = 1$ ,  $p = 2$ , and  $p = \infty$ ), whereas the work in [17] considers only one objective, which is the  $p = 2$  case in this paper.
- The method to solve the problems with different optimization objectives is also generic in this paper. Based on the bipartite graph, the movement plans with different objectives can be computed by using corresponding convex cost functions.

Therefore, it can be concluded that this paper provides a more generic framework, which is complementary to existing work efforts in the literature.

## VII. CONCLUSION

In this paper, we have provided a generic framework for the mobile sensor redeployment in WSNs. The framework consists of a generic system model, a generic objective function definition, and a generic method based on bipartite matching to solve the problems with different design objectives. The effectiveness of the BMrD algorithms has been validated through theoretical proofs and extensive simulations. This paper should provide helpful insights to the redeployment of mobile sensors in WSNs.

## APPENDIX PROOFS OF THEOREMS 1 AND 2

### *Proof of Theorem 1*

Without loss of generality, we consider the case when  $1 < p < \infty$ . The proof for  $p = 1$  is similar. The proof consists of three parts: to prove that the BMrD $_{k,p}$  algorithm minimizes the sum of gaps (i.e., optimizes step 1 of the three-step optimization problem), minimizes the  $L_p$ -norm of the gap vector (i.e., optimizes step 2 of the three-step optimization problem), and minimizes the total movement cost (i.e., optimizes step 3 of the three-step optimization problem).

*To prove that the BMrD $_{k,p}$  algorithm minimizes the sum of gaps:*

From the procedure of the BMrD $_{k,p}$  algorithm, it can be seen that a flow from  $s$  to  $t$  means that a mobile sensor moves to a grid that has less than  $k$  static sensors. Thus, a flow means that the sum of grid gaps decreases by one. Therefore, for each feasible set of flows, the sum of grid gaps after the associated movement plan is implemented is given by

$$\sum_{i=1}^n r_i = \sum_{g_i \in \mathbb{G}} u(g_i, t) - \sum_{m_j \in \mathbb{M}} f(s, m_j). \quad (12)$$

On the right-hand side of (12), the first term (i.e., the sum of mobile sensor demands of all the grids) is a fixed value depending on the initial deployment of the static sensors. Since the BMrD $_{k,p}$  algorithm maximizes  $\sum_{m_j \in \mathbb{M}} f(s, m_j)$ , apparently, it also minimizes  $\sum_{i=1}^n r_i$ , which is the sum of the gaps of all grids after the mobile sensor redeployment.

*To prove that the BMrD $_{k,p}$  algorithm minimizes the  $L_p$ -norm of the gap vector:*

We use proof by contradiction. Assume that the BMrD $_{k,p}$  algorithm determines a set of flows or, equivalently, a movement plan denoted  $\mathcal{P}$ , that has the maximum amount of flows and minimum cost given in (6). In movement plan  $\mathcal{P}$ , the flow amount on an edge  $(a, b)$  is denoted  $f(a, b)$ . After the movement plan is implemented, the gap vector is  $\mathbf{r} = \{r_1, r_2, \dots, r_n\}$ . Suppose that there exists a set of flows or, equivalently, a movement plan  $\mathcal{P}'$  with a resulted gap vector being  $\mathbf{r}' = \{r'_1, r'_2, \dots, r'_n\}$ , such that  $\sum_{i=1}^n r'_i = \sum_{i=1}^n r_i$ , and  $\|\mathbf{r}'\|_p < \|\mathbf{r}\|_p$ . In the movement plan  $\mathcal{P}'$ , the flow amount on an edge  $(a, b)$  is denoted  $f'(a, b)$ . Then, we will show that the cost of movement plan  $\mathcal{P}'$  is less than that of movement plan  $\mathcal{P}$ , which contradicts the fact that  $\mathcal{P}$  has the minimum cost.

It follows from  $\|\mathbf{r}'\|_p < \|\mathbf{r}\|_p$  that  $\sum_{i=1}^n r_i^p > \sum_{i=1}^n (r'_i)^p$ . As  $r_i$ ,  $r'_i$ , and  $p$  are all nonnegative integers, we have

$$\sum_{i=1}^n r_i^p \geq \sum_{i=1}^n (r'_i)^p + 1. \quad (13)$$

By multiplying both sides of (13) by the constant  $D$  defined in (4), we have

$$\sum_{i=1}^n D \cdot r_i^p \geq \sum_{i=1}^n D \cdot (r'_i)^p + D. \quad (14)$$



Among the  $n$  grids (i.e., grids from  $g_1$  to  $g_n$ ), if a grid is not included in  $\mathbb{G}$ , it means that the grid has  $k$  or more static sensors (or, equivalently, the gap of the grid is zero). Thus, from (14), we have

$$\sum_{g_i \in \mathbb{G}} D \cdot r_i^p \geq \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p + D. \quad (15)$$

Recall that the cost of flows corresponding to a movement plan consists of two parts: movement cost and gap cost. Therefore, from (6), the cost of movement plan  $\mathcal{P}$  is

$$\begin{aligned} C(\mathcal{P}) &= \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f(m_j, g_i) \cdot c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} c_g(g_i, t) \\ &\geq \sum_{g_i \in \mathbb{G}} c_g(g_i, t) = \sum_{g_i \in \mathbb{G}} D \cdot [u(g_i, t) - f(g_i, t)]^p \\ &\stackrel{(a)}{=} \sum_{g_i \in \mathbb{G}} D \cdot r_i^p \stackrel{(b)}{\geq} \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p + D \end{aligned} \quad (16)$$

where (a) follows from (5), and (b) follows from (15).

For movement plan  $\mathcal{P}'$ , we have

$$\begin{aligned} C(\mathcal{P}') &= \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f'(m_j, g_i) \cdot c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} c_g(g_i, t) \\ &\stackrel{(c)}{=} \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f'(m_j, g_i) \cdot c_m(m_j, g_i) \\ &\quad + \sum_{g_i \in \mathbb{G}} D \cdot [u(g_i, t) - f'(g_i, t)]^p \\ &\stackrel{(d)}{=} \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f'(m_j, g_i) \cdot c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \\ &\leq \left[ \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f'(m_j, g_i) \right] \\ &\quad \cdot \max_{\substack{m_j \in \mathbb{M} \\ g_i \in \mathbb{G}}} c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \\ &= \left[ \sum_{m_j \in \mathbb{M}} \sum_{g_i \in \mathbb{G}} f'(m_j, g_i) \right] \\ &\quad \cdot \max_{\substack{m_j \in \mathbb{M} \\ g_i \in \mathbb{G}}} c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \\ &\stackrel{(e)}{\leq} \left[ \sum_{m_j \in \mathbb{M}} 1 \right] \cdot \max_{\substack{m_j \in \mathbb{M} \\ g_i \in \mathbb{G}}} c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \\ &= |\mathbb{M}| \cdot \max_{\substack{m_j \in \mathbb{M} \\ g_i \in \mathbb{G}}} c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \\ &\stackrel{(f)}{<} D + \sum_{g_i \in \mathbb{G}} D \cdot (r'_i)^p \end{aligned} \quad (17)$$

where (c) follows from the definition of the gap cost, (d) follows from (5), (e) follows from the fact that the total flow

amount out of  $m_j$  should not exceed its incoming capacity, which is a unit, and (f) follows from (4).

From (16) and (17), we have  $C(\mathcal{P}) > C(\mathcal{P}')$ , which contradicts the fact that the movement plan  $\mathcal{P}$  has the minimum cost.

To prove that the total movement cost is minimized by the  $BMrD_{k,p}$  algorithm:

We need to consider only the feasible sets of flows that result in the minimum sum of gaps and the minimum  $L_p$ -norm of the gap vector (i.e., satisfying steps 1 and 2 in the three-step optimization problem). We repeat (6) for such a feasible flow set  $\mathcal{P}$ , i.e.,

$$C(\mathcal{P}) = \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f(m_j, g_i) \cdot c_m(m_j, g_i) + \sum_{g_i \in \mathbb{G}} c_g(g_i, t). \quad (18)$$

On the right-hand side of (18), the second summation is equal to  $D \cdot (r_1^p + r_2^p + \dots + r_n^p)$ , which corresponds to the objective of step 2 and, thus, is minimized. Recall that the  $BMrD_{k,p}$  algorithm minimizes the cost  $C(\mathcal{P})$ . Thus, it also minimizes the first summation on the right-hand side of (18), which is exactly the movement cost of a movement plan.

This completes the proof. ■

### Proof of Theorem 2

The proof consists of three parts: to prove that the  $BMrD_{k,\infty}$  algorithm minimizes the sum of gaps of all grids (i.e., optimizes step 1 of the three-step optimization problem), minimizes the  $L_\infty$ -norm of the gap vector (i.e., optimizes step 2 of the three-step optimization problem), and minimizes the total movement cost (i.e., optimizes step 3 of the three-step optimization problem).

To prove that the  $BMrD_{k,\infty}$  algorithm minimizes the sum of gaps of all grids:

This is obvious because with any gap cost function, an optimal matching of the flow network always achieves the property of “maximum amount of flows” and, thus, always minimizes the sum of gaps of all the grids.

To prove that the  $BMrD_{k,\infty}$  algorithm minimizes the  $L_\infty$ -norm of the gap vector:

This is equivalent to proving that the  $BMrD_{k,\infty}$  algorithm minimizes the maximum gap or, equivalently, maximizes the minimum number of sensors in a grid. We use proof by contradiction.

Let the movement plan determined by the  $BMrD_{k,\infty}$  algorithm be denoted  $\mathcal{P}^*$ . From the procedure of the  $BMrD_{k,\infty}$  algorithm,  $\mathcal{P}^*$  makes any grid have at least  $\alpha^*$  sensors. Suppose that there exists a movement plan  $\mathcal{P}'$  such that the amount of flows in the flow network is maximized, and each grid has at least  $\alpha^\dagger (> \alpha^*)$  sensors. Then, in the binary search of the  $BMrD_{k,\infty}$  algorithm, an  $\alpha$  value in  $(\alpha^*, \alpha^\dagger]$  should have been checked, but the resulted movement plan fails to make each grid at least  $\alpha$ -covered. For presentation simplicity, the checked value in the  $BMrD_{k,\infty}$  algorithm is assumed to be  $\alpha^\dagger$ , and the resulted movement plan (i.e., the optimal movement plan when  $h_{\alpha^\dagger}(x)$  is used as the gap cost function) is denoted  $\mathcal{P}^\dagger$ . It follows that movement plan  $\mathcal{P}^\dagger$  fails to make each grid at least

$\alpha^\dagger$ -covered. In movement plans  $\mathcal{P}'$  and  $\mathcal{P}^\dagger$ , the amount of flows on an edge  $(a, b)$  in the flow network is denoted  $f'(a, b)$  and  $f^\dagger(a, b)$ , respectively.

Recall that  $\mathcal{P}^\dagger$  is the minimum-cost flow at the round when  $\alpha = \alpha^\dagger$ , and  $h_{\alpha^\dagger}(x)$  is used as the gap cost function. Since  $\mathcal{P}'$  is also a feasible set of flows at this round, we have

$$C(\mathcal{P}^\dagger) \leq C(\mathcal{P}'). \quad (19)$$

Since movement plan  $\mathcal{P}^\dagger$  fails to make each grid at least  $\alpha^\dagger$ -covered, we use  $\hat{\mathbb{G}}$  to denote the subset of  $\mathbb{G}$  in which each grid is less than  $\alpha^\dagger$ -covered after movement plan  $\mathcal{P}^\dagger$  is implemented. Therefore,  $\hat{\mathbb{G}}$  is not a null set. Furthermore, in  $\mathbb{G} \setminus \hat{\mathbb{G}}$ , each grid is at least  $\alpha^\dagger$ -covered after movement plan  $\mathcal{P}^\dagger$  is implemented. Then, for the cost function of movement plan  $\mathcal{P}^\dagger$  at round  $\alpha = \alpha^\dagger$ , we have

$$\begin{aligned} C(\mathcal{P}^\dagger) &= \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f^\dagger(m_j, g_i) \cdot c_m(m_j, g_i) \\ &\quad + \sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^\dagger}(u(g_i, t) - f^\dagger(g_i, t)) \\ &\geq \sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^\dagger}(u(g_i, t) - f^\dagger(g_i, t)) \\ &\stackrel{(g)}{\geq} \sum_{g_i \in \mathbb{G} \setminus \hat{\mathbb{G}}} D \cdot A \cdot [u(g_i, t) - f^\dagger(g_i, t)] \\ &\quad + \sum_{g_i \in \hat{\mathbb{G}}} D \cdot \{A \cdot [u(g_i, t) - f^\dagger(g_i, t)] + 1\} \\ &= \sum_{g_i \in \mathbb{G}} D \cdot A \cdot [u(g_i, t) - f^\dagger(g_i, t)] + \sum_{g_i \in \hat{\mathbb{G}}} D \\ &\stackrel{(h)}{\geq} D \cdot A \sum_{g_i \in \mathbb{G}} u(g_i, t) - D \cdot A \sum_{g_i \in \mathbb{G}} f^\dagger(g_i, t) + D \end{aligned} \quad (20)$$

where (g) follows from (7), and (h) follows from the fact that  $\hat{\mathbb{G}}$  is not a null set.

On the other hand, the cost of movement plan  $\mathcal{P}'$  at round  $\alpha = \alpha^\dagger$  is

$$\begin{aligned} C(\mathcal{P}') &= \sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f'(m_j, g_i) \cdot c_m(m_j, g_i) \\ &\quad + \sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^\dagger}(u(g_i, t) - f'(g_i, t)) \\ &\stackrel{(i)}{<} D + \sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^\dagger}(u(g_i, t) - f'(g_i, t)) \\ &\stackrel{(j)}{=} D + \sum_{g_i \in \mathbb{G}} D \cdot A \cdot [u(g_i, t) - f'(g_i, t)] \\ &= D + D \cdot A \sum_{g_i \in \mathbb{G}} u(g_i, t) - D \cdot A \sum_{g_i \in \mathbb{G}} f'(g_i, t) \end{aligned} \quad (21)$$

where (i) is similar to the proof of (17), and (j) follows from the assumption that movement plan  $\mathcal{P}'$  makes each grid at least  $\alpha^\dagger$ -covered [and, therefore,  $u(g_i, t) - f'(g_i, t) \leq k - \alpha^\dagger$ ].

Since both movement plans  $\mathcal{P}'$  and  $\mathcal{P}^\dagger$  maximize the amount of flows in the flow network, we have

$$\sum_{g_i \in \mathbb{G}} f^\dagger(g_i, t) = \sum_{g_i \in \mathbb{G}} f'(g_i, t). \quad (22)$$

From (20)–(22), we have  $C(\mathcal{P}^\dagger) > C(\mathcal{P}')$ , which contradicts (19). Therefore, it can be concluded that the  $\text{BMrD}_{k, \infty}$  algorithm minimizes the  $L_\infty$ -norm of the gap vector, i.e., minimizes the maximum grid gap, and the maximum grid gap is  $k - \alpha^*$ .

To prove that the  $\text{BMrD}_{k, \infty}$  algorithm minimizes the total movement cost:

We use  $\mathcal{P}^*$  to denote the movement plan determined by the  $\text{BMrD}_{k, \infty}$  algorithm. From the procedure of the  $\text{BMrD}_{k, \infty}$  algorithm,  $\mathcal{P}^*$  makes any grid have at least  $\alpha^*$  sensors. Recall that  $\mathbb{F}_2$  is the set of all optimal movement plans in step 2 of the three-step optimization problem, as shown in Section II. Then, we need to prove that  $\mathcal{P}^*$  minimizes the movement cost, among all the movement plans in  $\mathbb{F}_2$ . From the above proof, it can be concluded that each movement plan in  $\mathbb{F}_2$  makes each grid at least  $\alpha^*$ -covered. Recall that among all those feasible movement plans in  $\mathbb{F}_2$ ,  $\mathcal{P}^*$  is an optimal matching of the flow network. This means  $\mathcal{P}^*$  minimizes the cost function given by

$$\begin{aligned} &\sum_{m_j \in \mathbb{M}, g_i \in \mathbb{G}} f(m_j, g_i) \cdot c_m(m_j, g_i) \\ &\quad + \sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^*}(u(g_i, t) - f(u_i, t)). \end{aligned} \quad (23)$$

Among all the movement plans in  $\mathbb{F}_2$ , the second term in (23) is

$$\begin{aligned} &\sum_{g_i \in \mathbb{G}} D \cdot h_{\alpha^*}(u(g_i, t) - f(u_i, t)) \\ &= \sum_{g_i \in \mathbb{G}} D \cdot A \cdot [u(g_i, t) - f(u_i, t)] \\ &= D \cdot A \sum_{g_i \in \mathbb{G}} u(g_i, t) - D \cdot A \sum_{g_i \in \mathbb{G}} f(u_i, t) \end{aligned} \quad (24)$$

which has a fixed value since all the movement plans in  $\mathbb{F}_2$  maximize the amount of flows in the flow network. Therefore,  $\mathcal{P}^*$  minimizes the first term in (23), which is exactly the movement cost of a movement plan.

This completes the proof.  $\blacksquare$

#### ACKNOWLEDGMENT

The authors would like to thank Dr. S. Wan at Northwestern Polytechnical University, Xi'an, China, and the anonymous reviewers for their constructive comments and suggestions that helped to improve the quality of the manuscript.

#### REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
- [2] K. T. Phan, R. Fan, H. Jiang, S. A. Vorobyov, and C. Tellambura, "Network lifetime maximization with node admission in wireless multimedia sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3640–3646, Sep. 2009.

- [3] P. Wang, J. Zheng, F. Yang, and C. Li, "Joint data aggregation and encryption using Slepian-Wolf coding for clustered wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 10, no. 4, pp. 573–583, Apr. 2010.
- [4] R. Fan, H. Jiang, S. Wu, and N. Zhang, "Ranging error-tolerable localization in wireless sensor networks with inaccurately positioned anchor nodes," *Wireless Commun. Mobile Comput.*, vol. 9, no. 5, pp. 705–717, May 2009.
- [5] H. Zhang and J. Hou, "Is deterministic deployment worse than random deployment for wireless sensor networks?" in *Proc. IEEE INFOCOM*, Apr. 2006.
- [6] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *Proc. 7th ACM Int. Symp. MobiHoc*, May 2006, pp. 131–142.
- [7] B. Wang, H. B. Lim, and D. Ma, "A survey of movement strategies for improving network coverage in wireless sensor networks," *Comput. Commun.*, vol. 32, no. 13/14, pp. 1427–1436, Aug. 2009.
- [8] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *Proc. 6th Int. Symp. DARS*, Jun. 2002, pp. 299–308.
- [9] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. IEEE INFOCOM*, Mar./Apr. 2003, pp. 1293–1303.
- [10] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.
- [11] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Trans. Comput.*, vol. 56, no. 7, pp. 946–958, Jul. 2007.
- [12] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Comput.*, vol. 7, no. 2, pp. 262–274, Feb. 2008.
- [13] C.-Y. Chang, C.-T. Chang, Y.-C. Chen, and H.-R. Chang, "Obstacle-resistant deployment algorithms for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 6, pp. 2925–2941, Jul. 2009.
- [14] J. Wu and S. Yang, "SMART: A scan-based movement-assisted sensor deployment method in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2313–2324.
- [15] J. Wu and S. Yang, "Optimal movement-assisted sensor deployment and its extensions in wireless sensor networks," in *Proc. 12th ICPADS*, Jul. 2006, pp. 261–270.
- [16] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 199–211, Feb. 2007.
- [17] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Trans. Mobile Comput.*, vol. 6, no. 10, pp. 1142–1157, Oct. 2007.
- [18] G. Wang, G. Cao, T. F. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2302–2312.
- [19] Z. Jiang, J. Wu, R. Kline, and J. Krantz, "Mobility control for complete coverage in wireless sensor networks," in *Proc. 28th ICDCS Workshops*, Jun. 2008, pp. 291–296.
- [20] S. Yang, J. Wu, and F. Dai, "Localized movement-assisted sensor deployment in wireless sensor networks," in *Proc. IEEE Int. Conf. MASS*, Oct. 2006, pp. 753–758.
- [21] R. Jain, A. Dursesi, and G. Babic, "Throughput Fairness Index: An Explanation," Feb. 1999, ATM Forum Doc. No. ATM\_Forum/99-0045.
- [22] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.



**Zhong Shen** received the B.E. degree from the University of Shanghai for Science and Technology, Shanghai, China, in 1992 and the M.E. degree in computer science and the Ph.D. degree in information and communication engineering from Xidian University, Xi'an, China, in 2002 and 2006, respectively.

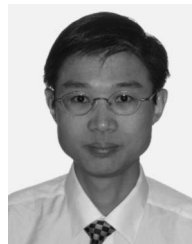
Since April 2006, he has been a Faculty Member with the School of Telecommunications Engineering, Xidian University. His research interests include network management, wireless *ad hoc* networks, and

wireless sensor networks.



**Yilin Chang** received the B.Sc. degree in physics from Xi'an Jiaotong University, Xi'an, China, in 1970 and the M.Sc. degrees in communication and information systems from Xidian University, Xi'an, in 1982.

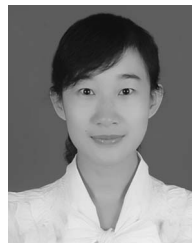
He is currently a Professor with the School of Telecommunications Engineering, Xidian University. His major research interests are in the fields of network management, multimedia communications, and wireless video transmission.



**Hai Jiang** (M'07) received the B.Sc. and M.Sc. degrees in electronics engineering from Peking University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree (with an Outstanding Achievement in Graduate Studies Award) in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 2006.

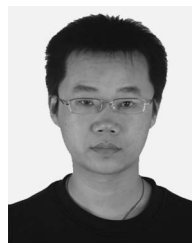
Since July 2007, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include radio resource management, cognitive radio networking, and cross-layer design for wireless multimedia communications.

Dr. Jiang is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He served as a Cochair for the General Symposium at the International Wireless Communications and Mobile Computing Conference in 2007, the Communications and Networking Symposium at the Canadian Conference on Electrical and Computer Engineering in 2009, and the Wireless and Mobile Networking Symposium at the IEEE International Conference on Communications in 2010. He was a recipient of an Alberta Ingenuity New Faculty Award in 2008 and a Best Paper Award from the IEEE Global Communications Conference in 2008.



**Yanling Wang** received the B.E. and M.E. degrees in communication and information systems from Xidian University, Xi'an, China, in 2007 and 2010, respectively.

She is currently with China Mobile as a Network Maintenance Engineer. Her research interests include management and deployment of wireless *ad hoc* and sensor networks.



**Zhongjiang Yan** received the B.E. degree in communication and information systems in 2006 from Xidian University, Xi'an, China, where he is currently working toward the Ph.D. degree with the School of Telecommunications Engineering.

His major research interests include wireless network management, topology control, and mobility control.