Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multilevel Coverage

You-Chiun Wang and Yu-Chee Tseng

Abstract—One of the key research issues in *wireless sensor networks* (*WSNs*) is how to efficiently deploy sensors to cover an area. In this paper, we solve the *k-coverage sensor deployment problem* to achieve multilevel (k) coverage of the area of interest $\mathcal{I}$. We consider two subproblems: *k-coverage placement problem* and *distributed dispatch problem*. The placement problem asks how to determine the minimum number of sensors required and their locations in $\mathcal{I}$ to guarantee that $\mathcal{I}$ is k-covered and the network is connected, while the dispatch problem asks how to schedule mobile sensors to move to the designated locations according to the result computed by the placement strategy, if they are not in the current positions, such that the energy consumption due to movement is minimized. Our solutions to the placement problem consider both the binary and probabilistic sensing models, and allow an arbitrary relationship between the communication distance and sensing distance of sensors, thereby relaxing the limitations of existing results. For the dispatch problem, we propose a competition-based scheme and a pattern-based scheme. The competition-based scheme allows mobile sensors to bid for their closest locations, while the pattern-based scheme allows sensors to derive the target locations on their own. Our proposed schemes are efficient in terms of the number of sensors required and are distributed in nature. Simulation results are presented to verify their effectiveness.

Index Terms—mobile sensors, network planning, pervasive computing, sensor coverage problem, topology control, wireless sensor networks.

1 Introduction

In recent years, with the rapid progress in embedded micro-sensing MEMS and wireless communication technologies, *wireless sensor networks* (*WSNs*) have been studied intensively for various applications such as environment monitoring, smart home, and surveillance. A WSN usually consists of numerous wireless devices deployed in a region of interest, each able to collect and process environmental information and communicate with neighboring devices.

Sensor deployment is an essential issue in WSN because it not only determines the cost to construct the network but also affects how well a region is monitored by sensors. In this paper, we consider the sensor deployment problem for a WSN with multilevel coverage. In particular, given a region of interest, we say that the region is *k-covered* if every location in that region can be monitored by at least k sensors, where k is a given parameter. A large amount of applications may impose the requirement of $k > 1$. For instance, military or surveillance applications with a stronger monitoring requirement may impose that $k \geq 2$ to avoid leaving uncovered holes when some sensors are broken. Positioning protocols using triangulation [1] require at least three sensors (i.e., $k \geq 3$) to detect each location where an object may appear. Moreover, several strategies are based on the assumption of $k \geq 3$ to conduct data fusion [2] and to minimize the impact of sensor failure [3]. In addition, to extend a WSN's lifetime, sensors are separated into k sets, each capable of covering the whole area, to work in shifts [4]–[6].

In this paper, we address the sensor deployment problem with the following requirements:

- Multiple-level coverage of the area of interest is required.
- Connectivity between sensors (in terms of their communications) should be maintained.
- The area of interest may change over time.
- Sensors are autonomous and mobile and thus can be dispatched to desired locations when being instructed so.

We call this the *k-coverage sensor deployment problem*, where the k-level coverage of a given area of interest $\mathcal{I}$ is needed. We consider two subproblems: the *k-coverage sensor placement problem* and the *distributed sensor dispatch problem*. The placement problem asks how to decide the minimum number of sensors required and their locations in $\mathcal{I}$ to ensure that $\mathcal{I}$ is k-covered and that the network is connected. Note that coverage is affected by sensors' sensing distance, while connectivity is determined by their communication distance. Considering that sensors are mobile and the area $\mathcal{I}$ may change over time, the objective of the dispatch problem is to schedule sensors to move to the designated locations (according to the result computed by the placement strategy) such that the total energy consumption of sensors due to movement can be minimized.

In the literature, one related area is the *art gallery problem* [7] in computational geometry. It intends to use the minimum number of observers to monitor a polygon area. The problem assumes that an observer can watch any point as long as line of sight exists and it does not address the (wireless) communication issue between observers. Another relevant issue is the *base station (BS) placement problem*. This problem discusses how to determine the optimal number and locations of BSs within an environment so as to satisfy the coverage and throughput requirements [8]. To solve this problem, many studies propose their discrete optimization models by multi-objective genetic

The authors are with the Department of Computer Science, National Chiao-Tung University, Hsin-Chu, 30010, Taiwan.
E-mail: {wangyc, yctseng}@cs.nctu.edu.tw

algorithms [8], [9], parallel evolutionary algorithms [10], and simulated annealing [11] to determine the optimal placement of BSs. However, these results cannot be directly applied to our sensor placement problem.

Sensor placements for 1-coverage have been studied in several works. For example, the works in [12], [13] consider placing sensors in a gridlike fashion to satisfy some coverage requirements, while [14] suggests placing sensors strip by strip to achieve both coverage and connectivity. In [15], a 1-coverage sensor placement method for the sensing field with obstacles is proposed. Several studies have also considered the sensor placement problem of multilevel coverage. In [3], a hexagon-like placement is proposed to guarantee the sensing field to be k-covered, under the assumption that the communication distance of sensors r_c is no smaller than twice their sensing distance r_s. The work in [16] models the sensing field by grids and considers two kinds of sensors with different costs and sensing capabilities to be deployed in the sensing field. The objective is to make every grid point k-covered and the total cost is minimum. However, both [3] and [16] do not address the relationship between r_c and r_s. How to compute the coverage level of a given placement is addressed in [17].

Some works address the coverage and connectivity issue by assuming that there is redundancy in the initial deployment and the goal is to select a minimal set of active sensors to achieve energy conservation and maintain complete coverage of the sensing field and connectivity of the network. References [4], [18] address how to arrange some sensors to go to sleeping modes to extend the network lifetime while maintaining 1-coverage of the sensing field. On the other hand, the works [19]–[23] consider how to select these active sensors to maintain k-coverage of the sensing field and the connectivity of the network.

The use of mobile sensors has also been discussed in several works. The work [24] considers moving nodes to make the network biconnected. When events occur, [25] discusses how to move some sensors to the event locations while still maintaining complete 1-coverage of the sensing field. The works [26]–[29] study how to move sensors to enhance coverage of the sensing field by using the Voronoi diagram or attractive/repulsive forces between sensors. In [30], the sensing field is partitioned into grids, and sensors are moved from high-density grids to low-density ones to achieve more uniform coverage. The work [31] considers adding several mobile sensors into a stationary sensor network to improve the coverage and connectivity of the original network. As can be seen, the attention of prior works was mainly paid to the use of mobile sensors to improve the topology of an existing network, which is different from the sensor dispatch problem discussed in this paper. Actually, several studies [32]–[34] have proposed their design and implementation of mobile sensors. Such mobile platforms are controlled by embedded computers and mounted with sensors. These studies motivate us to investigate the sensor dispatch problem.

In this paper, we consider more complete solutions to the k-coverage sensor deployment problem, by addressing both the placement and dispatch subproblems. In particular, for the sensor placement problem, we allow an arbitrary relationship between sensors' communication distance r_c and their sensing distance r_s. We consider two types of sensing models: *binary* and *probabilistic*. Under the binary sensing model [14], [15], [28], a location can be either monitored or not monitored by a sensor, depending on whether the location is within the sensor's r_s range. Under the probabilistic sensing model [12], [23], [35], a location will be monitored by a sensor according to some probability function. We first consider the binary sensing model of sensors and propose two solutions to the placement problem. The first one is based on an intuitive duplication idea, while the second one is based on a more complicated interpolating scheme and thus can save more sensors. Then, we adapt these solutions to the probabilistic sensing model by properly adjusting the sensing distances of sensors. For the sensor dispatch problem, we propose two distributed schemes to let sensors move to the designated locations (computed by the placement result) on their own. The first scheme assumes that sensors have full knowledge of all target locations in the area of interest. Sensors will compete with each other to move toward their closest locations. The second scheme relaxes the above assumption in a way that sensors can derive other target locations based on several known locations, according to the patterns in our placement strategies. Therefore, we can give several locations as seeds in the beginning, and sensors will then extend their range based on the placement pattern in a distributed manner.

In this paper, we consider that the area of interest $\mathcal{I}$ may change over time (based on users' application requirements). So sensors may be dispatched in multiple rounds. Specifically, in each round, when a new $\mathcal{I}$ is generated, the sink first calculates the locations to be placed with sensors in $\mathcal{I}$ by the proposed placement solutions and announces the complete or partial locations to sensors. Sensors then can automatically move to these designated locations by the proposed dispatch solutions to ensure k-coverage of $\mathcal{I}$. Because the sink does not know the current statuses and positions of mobile sensors, it cannot determine which sensor should move to which location in a centralized manner. Therefore, distributed dispatch solutions are more desirable.

The major contributions of this paper are twofold. First, our schemes allow change of the monitoring region and coverage level of the WSN in an autonomous and distributed manner. This is quite important for those applications where the region of interest may change over time. For example, one can image that a wide area is contaminated by some hazardous material such as the leakage of nuclear or poisonous chemicals. By quickly providing multilevel coverage of these movable regions of pollution, the whole situation can be assessed immediately, and such information can be conveniently used by the rescue team. Second, our deployment solutions are helpful in conditions where the precise initial deployment (e.g., by humans) is almost impossible, because the region of interest is very dangerous or even inaccessible to people. By introducing the concept of sensor dispatch, mobile sensors can automatically move to designated locations in an efficient way and thus the region of interest can be self-deployed by these sensors.

The rest of this paper is organized as follows: Section 2 formally defines the sensor placement and dispatch problems. Sections 3 and 4 propose our solutions to these problems. Section 5 presents simulation results to evaluate the proposed schemes. Section 6 concludes this paper.

2 Problem Statement

We are given a field $\mathcal{A}$, an area of interest $\mathcal{I}$ inside $\mathcal{A}$, and a set of mobile sensors $\mathcal{S}$ resident in $\mathcal{A}$. For convenience, we assume that $\mathcal{I}$ is a rectangular region. Each sensor has a communication distance r_c and a sensing distance r_s. Sensors are homogenous, but the relationship of r_c and r_s can be arbitrary. For connectivity, we assume that two sensors can communicate with each other if their distance is not larger than r_c. For coverage, we consider both the binary and probabilistic sensing models of sensors. Under the binary sensing model, a location can be monitored by a sensor if it is within the sensor's sensing region. Thus, a location in $\mathcal{A}$ is defined as k*-covered* if it is within k sensors' sensing regions, where k is a given parameter. Under the probabilistic sensing model, the detection probability of a sensor will decay with the distance from the sensor to the monitored location. In particular, the detection probability of a location u by a sensor s_i can be evaluated as [12], [35]:

$$p(u, s_i) = \begin{cases} e^{-\varepsilon d(u,s_i)} & \text{if } d(u, s_i) \le r_s \\ 0 & \text{otherwise,} \end{cases} \qquad (1)$$

where ε is a parameter indicating the physical characteristics of the sensor, and $d(u, s_i)$ is the distance between u and s_i. Thus, a location in $\mathcal{A}$ is considered as k-covered if the probability that there are at least k sensors which can detect this location is not smaller than a predefined threshold p_{th}, where $0 < p_{\text{th}} < 1$. With the above definitions, an area in $\mathcal{A}$ is considered as k-covered if every location inside that area is k-covered. We assume that sensors can be aware of their own positions, which can be obtained by the *Global Positioning System (GPS)* [36] or other localization techniques [37], [38].

Given an integer k, the k*-coverage sensor deployment problem* can be divided into two subproblems: the k*-coverage sensor placement problem* and the *distributed sensor dispatch problem*. The objective of the placement problem is to determine the minimum number of sensors required and their locations in the area of interest $\mathcal{I}$ to guarantee that $\mathcal{I}$ is k-covered and that the network is connected. Considering that mobile sensors are arbitrarily placed inside $\mathcal{A}$ and that there are sufficient sensors, the dispatch problem asks how to move sensors to designated locations (according to the result computed by the placement strategy) in a distributed manner such that the total energy consumption of sensors due to movement is minimized, i.e., $\min \sum_{i \in \mathcal{S}} e_i^{\text{move}} \times d_i$, where e_i^{move} is the energy cost for sensor i to move in one unit distance, and d_i is the total distance that sensor i has traveled. Note that here we assume that a sensor will move at a constant speed and will incur a constant rate of energy drain during its motion [30], [39]. However, the energy model may be defined in a different way from this one.

3 k-Coverage Sensor Placement Schemes

In this section, we deal with the k-coverage sensor placement problem. We first consider the binary sensing model of sensors and propose two placement solutions. The first solution is based on a naive duplication idea, while the second solution is inspired by a more complicated interpolating concept. Then, we discuss how to adapt these placement schemes to the probabilistic sensing model.

3.1 The Naive Duplicate Placement Scheme

One intuitive idea to achieve a k-coverage placement is to use a good sensor placement method to determine the locations of sensors to ensure 1-coverage and connectivity in $\mathcal{I}$, and then duplicate k sensors on each designated location. For the 1-coverage placement, we adopt the method proposed in [15], which has been proved to be able to use the minimum number of sensors to achieve 1-coverage and connectivity [40]. In this 1-coverage placement, sensors are placed row by row, where each row of sensors will guarantee continuous coverage and connectivity, while adjacent rows will guarantee continuous coverage of the whole area. According to the relationship of r_c and r_s, we separate the discussion into two cases, as shown in Fig. 1. When $r_c < \sqrt{3}r_s$, sensors on each row are separated by a distance of r_c, so the connectivity of sensors in each row can be guaranteed. Since $r_c < \sqrt{3}r_s$, each row of sensors can cover a beltlike area of width 2δ, where $\delta = \sqrt{r_s^2 - \frac{1}{4}r_c^2}$. Adjacent rows will be separated vertically by a distance of $r_s + \delta$ and will be shifted horizontally by a distance of $\frac{r_c}{2}$. This guarantees the coverage of the whole area. When $r_c \ge \sqrt{3}r_s$, the aforementioned placement will use too many sensors, so a common regular placement of triangular lattice [41] should be adopted, where adjacent sensors will be regularly separated by a distance of $\sqrt{3}r_s$.

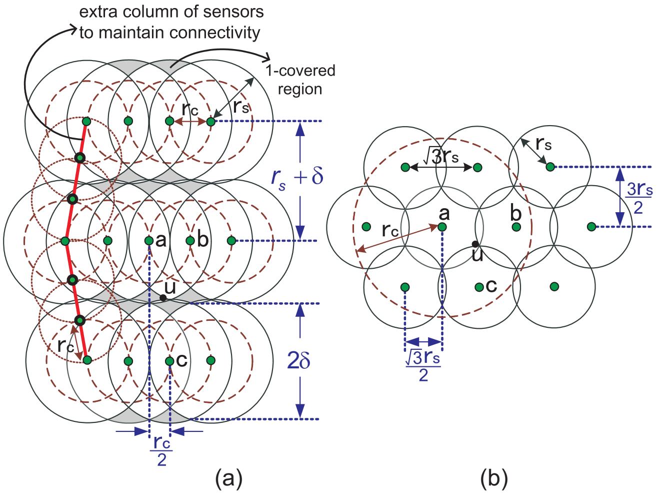


Fig. 1: The 1-coverage sensor placement method proposed in [15]: (a) the case of $r_c < \sqrt{3}r_s$ and (b) the case of $r_c \ge \sqrt{3}r_s$.

After determining the 1-coverage placement, we can duplicate k sensors on each location to ensure k-coverage of the whole area. Note that in the case of $r_c < \sqrt{3}r_s$, since the distance between sensors on adjacent rows is larger than r_c, it is necessary to add some extra columns of sensors, where sensors on each column are separated by a distance not larger than r_c, to connect adjacent rows.

3.2 The Interpolating Placement Scheme

The previous duplicate scheme may result in some subregions in $\mathcal{I}$ that have coverage levels much higher than k. Consequently, the following interpolating placement scheme will try to balance the coverage levels of subregions. Observe that in Fig. 1(a), a large amount of subregions in a row are actually more than 1-covered. Thus, we can reuse these subregions when generating a multilevel coverage placement. Based on this observation, the interpolating placement scheme will first find out those insufficiently covered subregions and then place the least number of sensors to cover these regions. Note that these newly added sensors should remain connected with the formerly placed sensors. According to the relationship of r_c and r_s, we separate the discussion into three cases.

The case of $r_c \le \frac{\sqrt{3}}{2}r_s$: In Fig. 1(a), we can observe that the insufficiently covered subregions (i.e., only 1-covered regions)

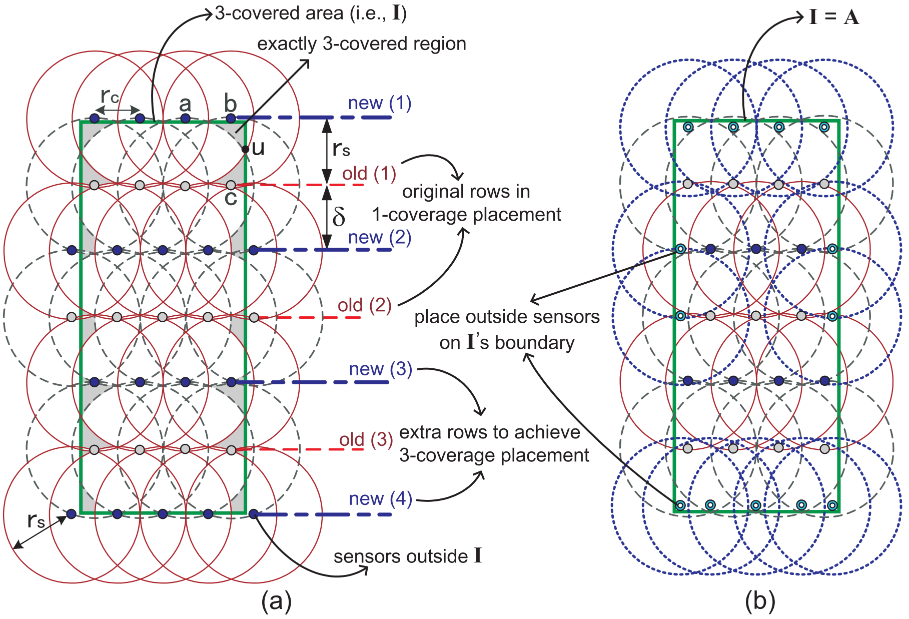


Fig. 2: The interpolating placement scheme in the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$: (a) the solution when $\mathcal{I} \subset \mathcal{A}$ and (b) the modified solution when $\mathcal{I} = \mathcal{A}$.

are located between adjacent rows (marked by gray). If we add an extra row of sensors between each pair of adjacent rows in Fig. 1(a), as Fig. 2(a) shows, the coverage level of the sensing field will directly become three. Here each extra row is placed above the previous row by a distance of r_s, and neighboring sensors in each extra row are still separated by a distance of r_c. Note that in Fig. 2(a), some sensors may be placed outside the area of interest $\mathcal{I}$. This may lead to the failure of the interpolating scheme to calculate a feasible solution when $\mathcal{I} = \mathcal{A}$. To solve this problem, we can place these outside sensors on the boundary of $\mathcal{I}$, as shown in Fig. 2(b). In this case, 3-coverage of $\mathcal{I}$ can still be achieved because sensors are placed more compactly.

In the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$, since the distance between sensors on adjacent rows is r_s (which is larger than r_c), we have to add at least one column of sensors, each separated by a distance not larger than r_c, to connect adjacent rows.

To summarize, the previous duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a beltlike area of width $(x-1)r_s + (x+1)\delta$, while this interpolating scheme uses only $2x+1$ rows of sensors to ensure 3-coverage of the same region. In general, for $k > 3$, we can apply $\lfloor \frac{k}{3} \rfloor$ times of the above 3-coverage placement and apply (k mod 3) times of the 1-coverage placement to achieve k-coverage in $\mathcal{I}$. Therefore, while the duplicate placement requires kx rows of sensors to cover a region, this interpolating placement requires only $\left(\lfloor \frac{k}{3} \rfloor (2x+1) + (k \bmod 3) \cdot x\right)$ rows of sensors.

The case of $\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$: In this case, if the desired coverage level k is two, we can directly apply the same placement in the previous case. The result is shown in Fig. 3(a). However, because the sensing distance r_s is relatively smaller (as opposed to the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$), there are some subregions that are only 2-covered but not 3-covered (marked by gray in Fig. 3(a)). Therefore, if the desired k is three, we need to add one extra row of sensors (marked as new′ i) between each new row i and old row i, as Fig. 3(b) shows. Note that these extra rows are shifted horizontally by a distance of $\frac{r_c}{2}$ from the previous rows and neighboring sensors are separated regularly by a distance of $2r_c$. Also note that each new′ row i can connect with its adjacent new row i and old row i, as shown in Fig. 3(c). In particular, because

$$|\overline{s_n s_a}| = |\overline{s_n s_b}| = |\overline{s_n s_c}| = |\overline{s_n s_d}|$$
$$= \sqrt{\left(\tfrac{1}{2} r_s\right)^2 + \left(\tfrac{1}{2} r_c\right)^2} < \tfrac{1}{2}\sqrt{\left(\tfrac{2}{\sqrt{3}} r_c\right)^2 + r_c^2} < r_c,$$

the sensor s_n in a new′ row i can communicate with its four neighbors s_a, s_b, s_c, and s_d in the adjacent new and old rows.

In the case of $\frac{\sqrt{3}}{2} r_s < r_c < r_s$, because the distance between sensors on adjacent rows may be larger than r_c, we have to add extra sensors between them to maintain the network connectivity. There are two cases to be discussed. When $k = 2$, we need to add at least one column of sensors between every two adjacent rows to connect them. When $k \geq 3$, because a new′ row has already connected with its adjacent new and old rows, we only have to add these extra columns of sensors between each old row i and new row $i+1$ to maintain the network connectivity.

To summarize, the previous duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a beltlike area of width $(x-1)r_s + (x+1)\delta$, while this interpolating scheme can use only $2.5x + 1$ rows of sensors to ensure 3-coverage of the same region (the third addition of rows only needs about $0.5x$ extra sensors). In general, for $k > 3$, we can also apply $\lfloor \frac{k}{3} \rfloor$ times of the above 3-coverage placement and apply (k mod 3) times of the 1-coverage placement to achieve k-coverage in $\mathcal{I}$. Therefore, while the duplicate placement requires kx rows of sensors to cover a region, this interpolating placement only requires $\left(\lfloor \frac{k}{3} \rfloor (2.5x+1) + (k \bmod 3) \cdot x\right)$ rows of sensors.

The case of $r_c > \frac{2+\sqrt{3}}{3} r_s$: In the previous case, when r_c increases, the areas of these only 2-covered regions in Fig. 3(a) also increase. To achieve the 3-coverage placement using fewer sensors, each sensor s_n in a new′ row should completely cover two only 2-covered regions (marked by gray), as shown in Fig. 3(d). In this case, we should make $|\overline{x s_n}| \leq r_s$, so we can

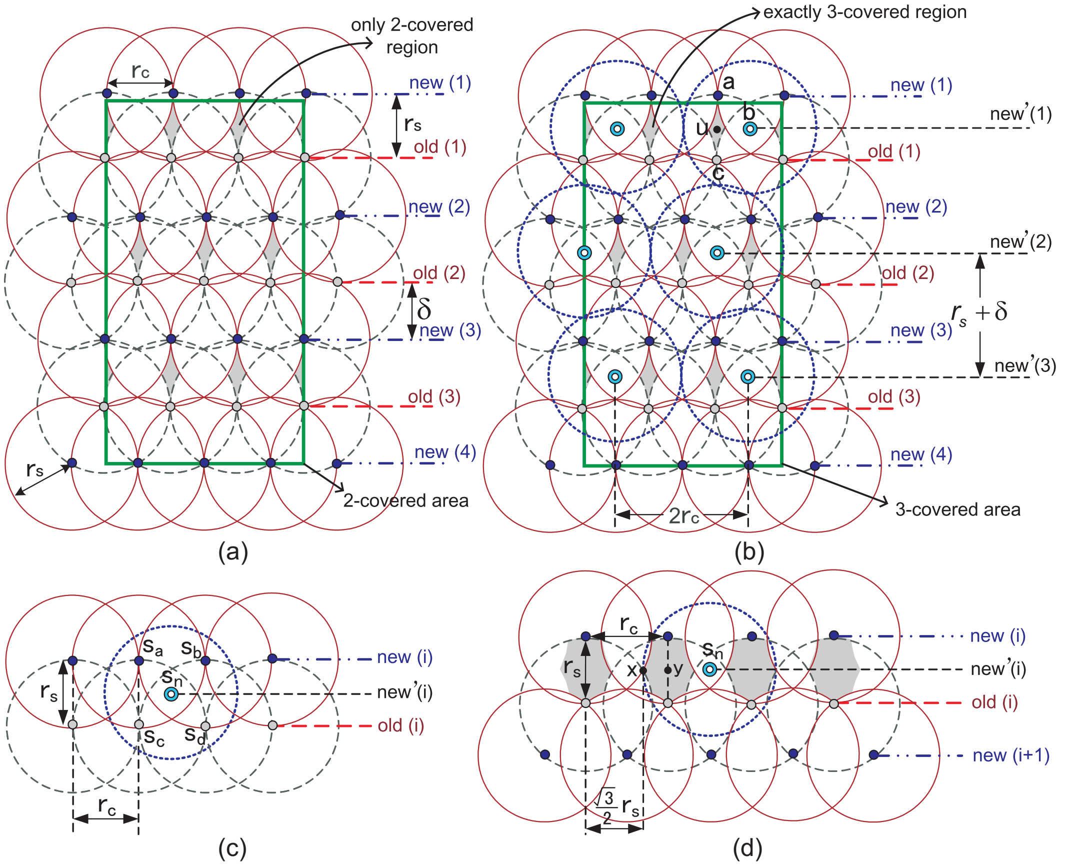


Fig. 3: The interpolating placement scheme in the case of $\frac{\sqrt{3}}{2}r_s < r_c \le \frac{2+\sqrt{3}}{3}r_s$: (a) the placement for $k = 2$, (b) the placement for $k = 3$, (c) an example to show that the connectivity between a new′ row and its adjacent rows is guaranteed, and (d) the boundary case.

obtain

$$|\overline{xs_n}| = |\overline{xy}| + |\overline{ys_n}| = \left(r_c - \frac{\sqrt{3}}{2}r_s\right) + \frac{1}{2}r_c \le r_s$$

$$\Rightarrow \; r_c \le \frac{2+\sqrt{3}}{3}r_s.$$

Clearly, when $r_c > \frac{2+\sqrt{3}}{3}r_s$, sensor s_n can no longer cover the nearest two 2-covered regions. Thus, we need to add one extra sensor in the new′ row to cover every 2-covered region. In this case, if the duplicate scheme uses $3x$ rows of sensors to ensure 3-coverage of a beltlike area of width $(x-1)r_s + (x+1)\delta$, this interpolating scheme should use $3x+1$ rows to achieve the same goal. Since the interpolating placement will not save sensors compared to the duplicate placement, we adopt the duplicate scheme in the case of $r_c > \frac{2+\sqrt{3}}{3}r_s$.

3.3 Adapting to the Probabilistic Sensing Model

In this section, we discuss how to adapt the previous two placement schemes to the probabilistic sensing model, where the detection probability of a sensor to any location follows that specified in Eq. (1). To simplify the presentation, we call the probability that a location u can be detected by at least k sensors as the *k-covered probability* of location u. To adapt our placement schemes, we first find the minimum k-covered probability $p_{\min}$ in our placement. Then, we calculate a pseudo sensing distance r_s^p according to $p_{\min}$ and p_{th}, and replace the original sensing distance r_s by r_s^p in the placement to guarantee that every location inside $\mathcal{I}$ is still k-covered under the probabilistic sensing model. In this section, we assume that $\mathcal{I} \subset \mathcal{A}$ and the desired coverage level $k \ge 3$.

3.3.1 Adaptation of the Duplicate Placement Scheme

Observing in Fig. 1, there must be a location u covered by only one sensor with a distance approximate to r_s. Such a location u is very close to the sensing boundary of the sensor placed at location a, but not inside the sensing ranges of sensors placed at locations b and c. Thus, we can derive the detection probability of location u by the sensor s_a located at a as $p(u, s_a) = e^{-\varepsilon d(u,s_a)} \approx e^{-\varepsilon r_s}$. Because the duplicate scheme places k sensors on each location specified in Fig. 1, location u will have the minimum k-covered probability $p_{\min}$. In particular, location u will be detected by a set $\mathcal{S}_a$ of k sensors placed at location a with the probability

$$p_{\min} = p(u, \mathcal{S}_a) = \prod_{s_i \in \mathcal{S}_a} p(u, s_i) \approx e^{-k\varepsilon r_s}.$$

Therefore, the duplicate scheme can guarantee a k-covered probability of at least $e^{-k\varepsilon r_s}$ in any location of the area of interest $\mathcal{I}$. On the other hand, if we want to guarantee that every location inside $\mathcal{I}$ has a k-covered probability not smaller than the given threshold p_{th}, we can calculate the pseudo sensing distance r_s^p by

$$e^{-k\varepsilon r_s^p} \ge p_{\text{th}} \;\Rightarrow\; r_s^p \le \frac{-\ln p_{\text{th}}}{k\varepsilon}.$$

With the above argument, if we replace r_s by r_s^p when executing the duplicate scheme, we can guarantee that $\mathcal{I}$ is still k-covered under the probabilistic sensing model.

3.3.2 Adaptation of the Interpolating Placement Scheme

According to the relationship of r_c and r_s, we separate the discussion into three cases.

The case of $r_c \leq \frac{\sqrt{3}}{2}r_s$: We first consider the case of $k = 3$. Observing in Fig. 2(a), there are some subregions covered by exactly three sensors (marked by gray). Among these regions, there will be a location u with a minimum 3-covered probability. In particular, such a location u will be covered by sensors s_a, s_b, and s_c located at a, b, and c, respectively. Since $d(u, s_a) = r_s$, we have $p(u, s_a) = e^{-\varepsilon r_s}$. In addition, because

$$\begin{aligned} d(u, s_b) = d(u, s_c) &< \sqrt{\left(\tfrac{1}{2}r_c\right)^2 + \left(\tfrac{1}{2}r_s\right)^2} \\ &\leq \tfrac{1}{2}\sqrt{\left(\tfrac{\sqrt{3}}{2}r_s\right)^2 + r_s^2} = \tfrac{\sqrt{7}}{4}r_s, \end{aligned}$$

we can obtain $p(u, s_b) = p(u, s_c) > e^{-\frac{\sqrt{7}}{4}\varepsilon r_s}$. Thus, the 3-covered probability of location u is

$$p_{abc} = p(u, s_a) \cdot p(u, s_b) \cdot p(u, s_c) > e^{-\frac{2+\sqrt{7}}{2}\varepsilon r_s}.$$

In the interpolating scheme, when $k \geq 3$ is a multiple of three, we will place $\frac{k}{3}$ sensors on each location specified in Fig. 2(a). Therefore, we can obtain

$$p_{\min} = (p_{abc})^{\frac{k}{3}} > e^{-\frac{2+\sqrt{7}}{6}k\varepsilon r_s}. \tag{2}$$

When k is not a multiple of three, we will add extra (k mod 3) sensors on each location in the old rows in Fig. 2(a). Thus, we have

$$p_{\min} = (p_{abc})^{\frac{k}{3}} \cdot (p(u, s_c))^{(k \bmod 3)}. \tag{3}$$

By combining Eqs. (2) and (3), we can derive that

$$p_{\min} > e^{-(\frac{2+\sqrt{7}}{6}k + \frac{\sqrt{7}}{4}(k \bmod 3))\varepsilon r_s}. \tag{4}$$

To calculate the pseudo sensing distance r_s^p, we can make

$$\begin{aligned} &e^{-(\frac{2+\sqrt{7}}{6}k + \frac{\sqrt{7}}{4}(k \bmod 3))\varepsilon r_s^p} \geq p_{\text{th}} \\ \Rightarrow\ & r_s^p \leq \frac{-\ln p_{\text{th}}}{\left(\frac{2+\sqrt{7}}{6}k + \frac{\sqrt{7}}{4}(k \bmod 3)\right)\varepsilon}. \end{aligned}$$

The case of $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$: Again, we first consider the case of $k = 3$. Observing in Fig. 3(b), there are some subregions covered by exactly three sensors (marked by gray). Among these regions, there will be a location u that has the minimum 3-covered probability. In particular, such a location u is covered by sensors s_a, s_b, and s_c at locations a, b, and c, respectively. Because $d(u, s_a) = d(u, s_c) = \frac{1}{2}r_s$, we have $p(u, s_a) = p(u, s_c) = e^{-\frac{1}{2}\varepsilon r_s}$. Moreover, since $d(u, s_b) = \frac{1}{2}r_c \leq \frac{2+\sqrt{3}}{6}r_s$, we can obtain $p(u, s_b) \geq e^{-\frac{2+\sqrt{3}}{6}\varepsilon r_s}$. Thus, the 3-covered probability of location u will be

$$p_{abc} = p(u, s_a) \cdot p(u, s_b) \cdot p(u, s_c) \geq e^{-\frac{8+\sqrt{3}}{6}\varepsilon r_s}.$$

Similar to Eq. (4), when $k \geq 3$, we can derive the minimum k-covered probability $p_{\min}$ as

$$(p_{abc})^{\frac{k}{3}} \cdot (p(u, s_c))^{(k \bmod 3)} \geq e^{-(\frac{8+\sqrt{3}}{18}k + \frac{1}{2}(k \bmod 3))\varepsilon r_s}.$$

Again, the pseudo sensing distance r_s^p can be derived as

$$\begin{aligned} &e^{-(\frac{8+\sqrt{3}}{18}k + \frac{1}{2}(k \bmod 3))\varepsilon r_s^p} \geq p_{\text{th}} \\ \Rightarrow\ & r_s^p \leq \frac{-\ln p_{\text{th}}}{\left(\frac{8+\sqrt{3}}{18}k + \frac{1}{2}(k \bmod 3)\right)\varepsilon}. \end{aligned}$$

The case of $r_c > \frac{2+\sqrt{3}}{3}r_s$: In this case, since the duplicate scheme is adopted, we can obtain $p_{\min} \geq e^{-k\varepsilon r_s}$ and $r_s^p \leq \frac{-\ln p_{\text{th}}}{k\varepsilon}$.

Table 1 summarizes the approximate threshold values of the minimum k-covered probability $p_{\min}$ and the pseudo sensing distance r_s^p in the interpolating scheme.

4 Distributed Sensor Dispatch Schemes

After determining the locations to be placed with sensors, the next issue is how to move existing sensors in the field $\mathcal{A}$ to the designated locations in $\mathcal{I}$ such that the energy consumption of sensors due to movement can be minimized. Since we cannot obtain the current statuses and positions of sensors, it is impossible to compute an optimal solution to dispatch sensors in a centralized manner. Thus, we propose two distributed dispatch schemes in this section.

4.1 The Competition-Based Dispatch Scheme

In this scheme, when an area of interest $\mathcal{I}$ is determined, the sink will first calculate a set of locations $\mathcal{L} = \{(x_1, y_1, n_1), (x_2, y_2, n_2), \cdots, (x_m, y_m, n_m)\}$ to be placed with sensors in $\mathcal{I}$, according to our placement schemes in Section 3. Here each element (x_j, y_j, n_j), $j = 1..m$, indicates that n_j sensors need to be placed on location (x_j, y_j). The sink then broadcasts $\mathcal{L}$ to all sensors.

On receiving $\mathcal{L}$ from the sink, sensors will compete with each other to move toward these locations. In particular, each sensor s_i will construct a table $OCC[1..m]$ such that every entry $OCC[j] = \{(s_{j_1}, d_{j_1}), (s_{j_2}, d_{j_2}), \ldots, (s_{j_\alpha}, d_{j_\alpha})\}$, $\alpha \leq n_j$, contains the set of sensors that have already moved into, or are still on their ways moving toward, location (x_j, y_j) and their corresponding distances to (x_j, y_j). Specifically, each record $(s_{j_\beta}, d_{j_\beta})$, $\beta = 1..\alpha$, indicates that sensor s_{j_β} has chosen to cover location (x_j, y_j) and its current estimated distance to (x_j, y_j) is d_{j_β}. When $d_{j_\beta} = 0$, it means that sensor s_{j_β} has already arrived at (x_j, y_j). Initially, $OCC[j] = \emptyset$ for all $j = 1..m$. To simplify the presentation, we say that a location (x_j, y_j) is *covered* if a sufficient number n_j of sensors have committed to move toward (x_j, y_j) (i.e., $|OCC[j]| = n_j$); otherwise, (x_j, y_j) is *uncovered*. A sensor s_i is *engaged* if it has chosen to move to, or has already moved into, any location in $\mathcal{L}$; otherwise, it is *free* or *terminated*. The initial state of each sensor is free. A free sensor will try to become engaged and move toward a destination. When the free sensor finds that there is no location that it can cover, it will enter the terminated state. Fig. 4 illustrates the state transition diagram of a sensor.

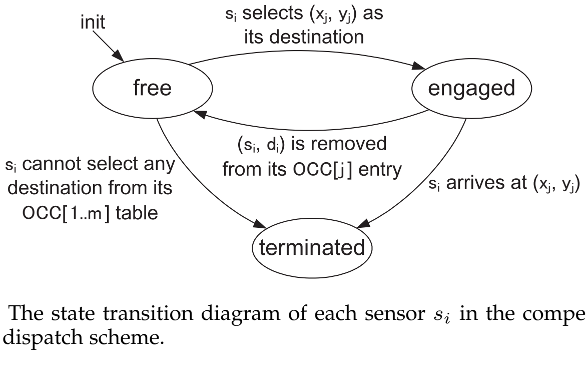

init
s$_i$ selects (x$_j$, y$_j$) as its destination
free
engaged
(s$_i$, d$_i$) is removed from its OCC[j] entry
s$_i$ cannot select any destination from its OCC[1..m] table
s$_i$ arrives at (x$_j$, y$_j$)
terminated

Fig. 4: The state transition diagram of each sensor s_i in the competition-based dispatch scheme.

When the state of a sensor s_i is free, it will check its $OCC[1..m]$ table to select a location in $\mathcal{L}$ as its destination. The selection is as follows:

case	$p_{\min}$	r_s^p
$r_c \leq \frac{\sqrt{3}}{2} r_s$	$e^{-(0.77k+0.66(k \bmod 3))\varepsilon r_s)}$	$\frac{-\ln p_{\text{th}}}{(0.77k+0.66(k \bmod 3))\varepsilon}$
$\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$	$e^{-(0.54k+0.5(k \bmod 3))\varepsilon r_s)}$	$\frac{-\ln p_{\text{th}}}{(0.54k+0.5(k \bmod 3))\varepsilon}$
$r_c > \frac{2+\sqrt{3}}{3} r_s$	$e^{-k\varepsilon r_s}$	$\frac{-\ln p_{\text{th}}}{k\varepsilon}$

TABLE 1: Approximate threshold values of $p_{\min}$ and r_s^p in the interpolating placement scheme.

- The first priority is to consider uncovered locations. Specifically, if there is a location (x_j, y_j) such that $|OCC[j]| < n_j$, (x_j, y_j) will be considered first. If multiple locations are qualified, the location (x_j, y_j) such that $d(s_i, (x_j, y_j))$ is minimized will be selected, where $d(s_i, (x_j, y_j))$ is the distance between s_i's current position to (x_j, y_j). In this case, s_i will add a record $(s_i, d(s_i, (x_j, y_j)))$ in its $OCC[j]$ entry and enter the engaged state.
- If all locations in $\mathcal{L}$ are already covered (i.e., $|OCC[j]| = n_j$, $\forall j = 1..m$), s_i selects a location (x_j, y_j) such that there is a record $(s_k, d_k) \in OCC[j]$ and $e_i^{\text{move}} \times d(s_i, (x_j, y_j)) < e_k^{\text{move}} \times d_k$. If multiple locations are qualified, the location (x_j, y_j) such that $e_k^{\text{move}} \times d_k - e_i^{\text{move}} \times d(s_i, (x_j, y_j))$ is maximized will be selected. In this case, s_i will replace the original record $(s_k, d_k) \in OCC[j]$ by the new record $(s_i, d(s_i, (x_j, y_j)))$ in its $OCC[j]$ entry and enter the engaged state. Here both sensors s_i and s_k are competing for the same location (x_j, y_j). Because s_i can consume less energy to move to (x_j, y_j), we should replace s_k's mission by s_i to reduce the total moving energy. Note that sensor s_k will realize that it loses the competition when it receives an update message originated from s_i later.

When sensor s_i becomes engaged, it begins moving toward its destination. Otherwise, s_i will enter the terminated state because it does not need to cover any location.

For the maintenance purpose, each sensor s_i will periodically perform the following two actions:

- Update the content of its $OCC[1..m]$ table. Specifically, for each $(s_{j_\beta}, d_{j_\beta}) \in OCC[j]$, $j = 1..m$, we decrease d_{j_β} by the expected moving distance of s_{j_β} during the last period of time, unless $d_{j_\beta} = 0$.
- Broadcast s_i's current status to its one-hop neighbors, including its ID, its moving energy cost e_i^{move}, its $OCC[1..m]$ table, and its current position and state.

The above actions can be controlled by setting two timers $T_{\text{update\_OCC}}$ and $T_{\text{broadcast}}$. Note that the update of the $OCC[1..m]$ table is based on the assumption that sensors all move in the same constant speed. If this assumption is not valid, d_{j_β} is only an estimated distance for sensor s_{j_β} to location (x_j, y_j). In this case, we can make an extension by including each sensor's moving speed in its broadcast message.

When a sensor s_i receives an update message from another sensor s_k, two actions will be taken:

- First, s_i has to update its $OCC[1..m]$ table as follows: Let us denote by $OCC_i[1..m]$ and $OCC_k[1..m]$ the tables of s_i and s_k, respectively. For each $j = 1..m$, we calculate the union $U_j = OCC_i[j] \cup OCC_k[j]$. If $|U_j| \leq n_j$, we will replace $OCC_i[j]$ by U_j. Otherwise, it means that there are too many sensors scheduled to cover (x_j, y_j), in which case we will truncate those records (s_k, d_k) in U_j that have more moving energy (i.e., large value of $e_k^{\text{move}} \times d_k$), until the size $|U_j| = n_j$. Then we replace $OCC_i[j]$ by the truncated U_j. Note that the above merge of two sets may lead to a special case that s_i was in the original $OCC_i[j]$ entry, but is not in the new $OCC_i[j]$ entry. In this case, it means that s_i has been replaced by some other sensors with a lower moving energy to (x_j, y_j). If so, sensor s_i should change its state from engaged to free and then reselect another destination.
- After the above merge, if s_i remains engaged, say, with (x_j, y_j) being its destination, we will conduct the following optimization. We will check if

$$e_i^{\text{move}} \times d(s_i, (x_l, y_l)) + e_k^{\text{move}} \times d(s_k, (x_j, y_j)) < e_i^{\text{move}} \times d(s_i, (x_j, y_j)) + e_k^{\text{move}} \times d(s_k, (x_l, y_l)),$$

  where (x_l, y_l) is the current destination of s_k. If so, it means that the total moving energy of s_i and s_k can be reduced if we exchange their destinations. In this case, s_i will communicate with s_k for this trade. Once the trade is confirmed, s_i will replace the records (s_i, d_i) and (s_k, d_k) in $OCC_i[j]$ and $OCC_i[l]$ by the new records $(s_k, d(s_k, (x_j, y_j)))$ and $(s_i, d(s_i, (x_l, y_l)))$, respectively. Note that s_k will also update its $OCC_k[j]$ and $OCC_k[l]$ entries with the same records.

In the above steps, if any entry in the $OCC_i[1..m]$ table has been changed, s_i will broadcast the modified content to its direct neighbors.

When a sensor s_i is in the engaged state, it will keep moving toward its destination (x_j, y_j). When s_i arrives at (x_j, y_j), it will change its state to terminated and begin its monitoring job at the designated location. Meanwhile, it still executes the maintenance actions until the sink commands it to stop. Since the sink will eventually observe that $\mathcal{I}$ is k-covered (by receiving the sensing reports from sensors), it can notify all sensors to exit from the dispatch algorithm. Fig. 5 summarizes the main steps of the competition-based scheme. Theorem 1 shows that the competition-based scheme can guarantee $\mathcal{I}$ to be k-covered if there are sufficient sensors.

Step 1: Sink broadcasts a set of location $\mathcal{L}$ to all sensors.

Step 2: Each sensor constructs an OCC table from $\mathcal{L}$ and sets its state as free.

Step 3: A free sensor s_i selects an uncovered (or replaceable) location (x_j, y_j) as its destination. If s_i cannot find such (x_j, y_j), it enters the terminated state and stops. Otherwise, s_i becomes engaged and moves to (x_j, y_j).

Step 4: Each sensor periodically broadcasts and updates its OCC table. If the record (s_i, d_i) is discarded during the updating operation, sensor s_i changes its state to free and goes back to step 3.

Step 5: When sensor s_i arrives at (x_j, y_j), it changes to terminated and begins the monitoring job. s_i still conducts step 4 until the sink commands to stop.

Fig. 5: The main steps of the competition-based dispatch scheme.

Theorem 1. Given an area $\mathcal{I} \subseteq \mathcal{A}$, the competition-based dispatch scheme guarantees that $\mathcal{I}$ will be eventually k-covered if there are sufficient mobile sensors inside $\mathcal{A}$.

Proof: Since the proposed placement schemes in Section 3 can compute a set of locations $\mathcal{L}$ inside $\mathcal{I}$ to be placed with sensors to ensure that $\mathcal{I}$ is k-covered, we only have to show that every location $(x_j, y_j) \in \mathcal{L}$ will eventually be covered by n_j sensors. Observe that in the competition-based scheme, it is guaranteed that an engaged sensor s_i will eventually arrive at location (x_j, y_j) if the record (s_i, d_i) remains in s_i's $OCC[j]$ entry. However, if the record (s_i, d_i) is removed during s_i's movement toward (x_j, y_j), it means that either another sensor s_k trades its current destination (x_l, y_l) with s_i or s_i loses the competition. In the former case, the locations (x_j, y_j) and (x_l, y_l) will be covered by s_k and s_i, respectively. In the latter case, it means that (x_j, y_j) has already been committed by more than n_j sensors, so it is safe for sensor s_i to give up the location (x_j, y_j). In this case, s_i has to reselect another destination. If s_i finds that $|OCC[j]| = n_j$ for all $j = 1..m$, then every location in $\mathcal{L}$ has been committed by sufficient sensors. So all locations will be eventually covered by n_j sensors. Therefore, the competition-based dispatch scheme guarantees that $\mathcal{I}$ will be eventually k-covered if there are sufficient mobile sensors. □

Remark 1. Theorem 1 also shows that the competition-based scheme can converge when there are sufficient sensors. However, when the number of sensors is not sufficient to cover $\mathcal{I}$, the competition-based scheme still guarantees that each sensor can eventually find a location to cover. In this case, if the sink knows in advance the total number of mobile sensors, it can also notify all sensors to exit from the dispatch algorithm earlier. If this assumption is not valid, a time-out mechanism should be applied to guarantee the convergence of this dispatch scheme. In this case, the sink can maintain a timer to decide when to terminate the dispatch algorithm.

Remark 2. There is a hidden assumption that the initial deployment of the network is connected, so sensors can receive the target locations $\mathcal{L}$ from the sink safely. For those sensors isolated from the initial network, they can only receive $\mathcal{L}$ when other sensors with $\mathcal{L}$ move close to them (by step 4). However, to alleviate the worst situation that some sensors may be always isolated from other sensors, we can enforce sensors to roam around randomly from time to time to increase the probability of information exchange.

Remark 3. Most message exchanges in the competition-based scheme rely on broadcast mechanism (steps 1 and 4). Since sensors will periodically broadcast their statuses and OCC tables, this scheme can tolerate the slight loss of messages. Thus, no extra acknowledgement mechanism is required to ensure proper operations of the competition-based scheme. In addition, to ensure that sensors correctly update their OCC tables, a timestamp or a sequence number is needed in each message to distinguish new from old messages.

Remark 4. In the competition-based scheme, sensors will find out and move to their destinations on their own, without any interaction with the sink. The sink only announces available target locations in the beginning. Thus, the competition-based scheme is essentially distributed.

neighbor	$r_c < \sqrt{3}r_s$	$r_c \geq \sqrt{3}r_s$
n_1	$(x + r_c, y)$	$(x + \sqrt{3}r_s, y)$
n_2	$(x + \frac{1}{2}r_c, y - r_s - \delta)$	$(x + \frac{\sqrt{3}}{2}r_s, y - \frac{3}{2}r_s)$
n_3	$(x - \frac{1}{2}r_c, y - r_s - \delta)$	$(x - \frac{\sqrt{3}}{2}r_s, y - \frac{3}{2}r_s)$
n_4	$(x - r_c, y)$	$(x - \sqrt{3}r_s, y)$
n_5	$(x - \frac{1}{2}r_c, y + r_s + \delta)$	$(x - \frac{\sqrt{3}}{2}r_s, y + \frac{3}{2}r_s)$
n_6	$(x + \frac{1}{2}r_c, y + r_s + \delta)$	$(x + \frac{\sqrt{3}}{2}r_s, y + \frac{3}{2}r_s)$

TABLE 2: Coordinates of the six neighbors of a sensor located at (x, y) in the duplicate placement scheme.

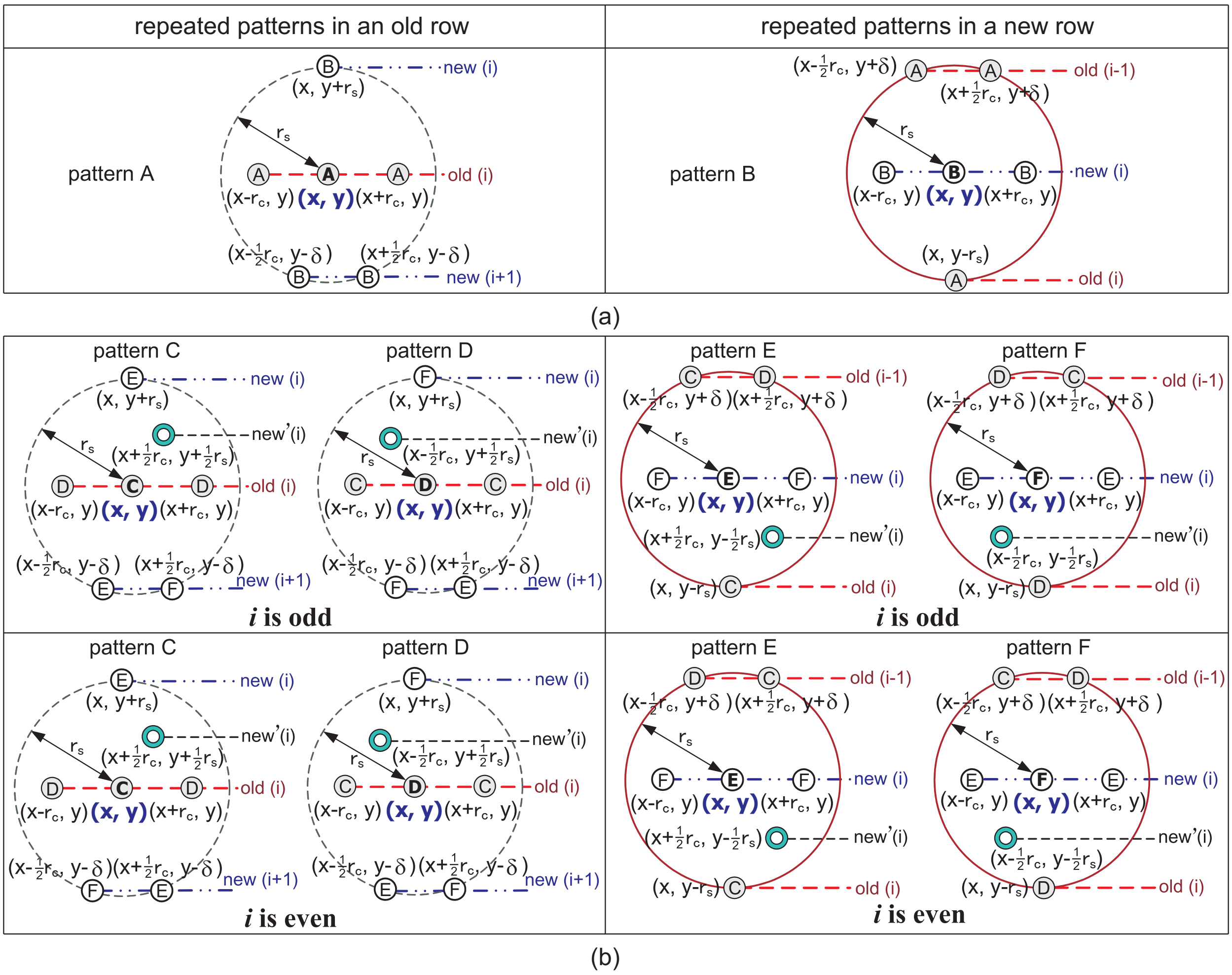
4.2 The Pattern-Based Dispatch Scheme

The previous competition-based scheme assumes that every sensor has full knowledge of all target locations inside $\mathcal{I}$. This requires the sink to execute the placement scheme for $\mathcal{I}$ and then to broadcast all target locations to every sensor. Consequently, in this section, we propose a pattern-based dispatch scheme, which allows sensors to derive the target locations on their own, thus relaxing the above limitation.

Observe that our placement schemes in Section 3 actually place sensors with some regular patterns. Specifically, in the duplicate placement scheme, sensors will be placed in a hexagonlike fashion. Thus, each sensor at the location (x, y) can derive its potential six neighbors' positions according to Table 2. When the interpolating placement scheme is adopted, the pattern will be changed according to the relationship of r_c and r_s:

- $r_c \leq \frac{\sqrt{3}}{2}r_s$. Recall the placement in Fig. 2(a). There are two patterns A and B, which will be repeated in each old row and new row, as shown in Fig. 6(a). Therefore, a sensor s_i located at (x, y) can derive its five neighbors' positions according to its pattern. Moreover, s_i can also derive the patterns of its neighbors, depending on its own pattern (indicated by the letters inside circles in Fig. 6(a)).
- $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$. In this case, if the desired coverage level k is two, we can directly apply patterns A and B in the previous case. However, when $k \geq 3$, there is an extra row (marked as new') between each new and old rows in Fig. 3(b). This will result in four placement patterns C, D, E, and F, as Fig. 6(b) shows, depending on a sensor's position and its row number. Thus, a sensor s_i located at (x, y) can derive its six neighbors' positions based on its pattern. In addition, s_i can also derive the patterns of its neighbors according to its own pattern (indicated by the letters inside circles in Fig. 6(b)). Note that we do not derive the patterns for sensors at the extra new' rows (although this is feasible, deriving these patterns will complicate the problem a lot). That is why sensors marked by double circles are not assigned with any pattern letter.
- $r_c > \frac{2+\sqrt{3}}{3}r_s$. In this case, since the duplicate placement scheme is adopted, a sensor can compute its neighbors' positions according to Table 2.

To summarize, the above observations allow a sensor to derive its direct neighbors (within the r_s range) as well as the patterns to be used by them. This property allows us to expand from a partial deployment to a full deployment of sensors in $\mathcal{I}$. Note that since the values of r_c and r_s are known, each sensor can maintain a small table to record the related positions of its



(a)

(b)

Fig. 6: The repeated patterns in the interpolating placement scheme: (a) the case of $r_c \leq \frac{\sqrt{3}}{2} r_s$ and (b) the case of $\frac{\sqrt{3}}{2} r_s < r_c \leq \frac{2+\sqrt{3}}{3} r_s$ when the desired coverage level $k \geq 3$.

neighbors in each pattern. Thus, the calculation of neighbors' positions can be translated to a simple table lookup procedure.

With the above property, the pattern-based dispatch scheme works as follows. Each sensor initially keeps a set of seed locations $\mathcal{L}' = \{(x_1, y_1, n_1, \rho_1), (x_2, y_2, n_2, \rho_2), \cdots, (x_\alpha, y_\alpha, n_\alpha, \rho_\alpha)\}$, which is a partial list of locations to be placed with sensors in $\mathcal{I}$, where ρ_j is the pattern used by the sensor at location (x_j, y_j). Clearly, $\mathcal{L}'$ can be considered as a subset of $\mathcal{L}$. Note that these seed locations should be sparsely distributed over $\mathcal{I}$, so that sensors may not crowd into only few locations in the beginning. Each sensor then executes the competition-based scheme to contend for their closest locations in $\mathcal{L}'$. However, the original steps 3 and 5 in the competition-based scheme should be modified as follows:

- **Step 3':** A free sensor s_i will try to select a location in $\mathcal{L}'$ as its destination. If s_i cannot find any available location from its current $OCC[\cdot]$ table, it will calculate some new locations based on the known locations and their patterns in the $OCC[\cdot]$ table. Then, s_i will try to select a destination among these newly derived locations. However, if s_i cannot calculate any new location from its current $\mathcal{L}'$ (which means that $\mathcal{L}' = \mathcal{L}$), s_i will enter the terminated state since it does not need to cover any location.
- **Step 5':** When an engaged sensor s_i arrives at its destination, it will derive some new locations from its current $\mathcal{L}'$ and add the corresponding new entries in its $OCC[\cdot]$ table.

***Corollary* 1.** Given an area $\mathcal{I} \subseteq \mathcal{A}$, the pattern-based dispatch scheme guarantees that $\mathcal{I}$ can be k-covered if there are sufficient mobile sensors inside $\mathcal{A}$.

Proof: From Theorem 1, we know that the competition-based dispatch scheme can ensure that $\mathcal{I}$ is k-covered if there are sufficient sensors. Since the pattern-based scheme works similarly to the competition-based one, we only need to show that the complete information of $\mathcal{L}$ can be eventually known by all sensors. Observing in the pattern-based scheme, since a sensor can either derive new locations by itself (according to steps 3' and 5') or learn new locations from other sensors (by step 4 in the competition-based scheme), the complete information of $\mathcal{L}$ can be propagated throughout the whole network. Therefore, the pattern-based scheme also guarantees $\mathcal{I}$ to be k-covered when there are sufficient mobile sensors. □

5 Experimental Results

In this section, we present some experimental results to evaluate the performances of the proposed schemes. The evaluation includes three parts. First, we measure the numbers of sensors required by different placement schemes discussed in Section 3. Second, we verify the effectiveness of our sensor dispatch schemes proposed in Section 4. Finally, we study the effect of seed locations on the pattern-based dispatch scheme.

5.1 Evaluations of the Sensor Placement Schemes

The first experiment measures the numbers of sensors required by different placement schemes. We design an area of interest $\mathcal{I}$ as a 1000 m (meter) × 1000 m square region to be placed with sensors. The communicate distance r_c is set to 10 m, which is approximate to that specified in the IEEE 802.15.4 standard [42] in an indoor environment. To reflect the relationships of $r_c < \frac{\sqrt{3}}{2}r_s$, $r_c \approx \frac{\sqrt{3}}{2}r_s$ (boundary case), $\frac{\sqrt{3}}{2}r_s < r_c < \frac{2+\sqrt{3}}{3}r_s$, $r_c \approx \frac{2+\sqrt{3}}{3}r_s$ (boundary case), and $r_c > \frac{2+\sqrt{3}}{3}r_s$, we set the sensing distance r_s to 15 m, 11.55 m, 10 m, 8.04 m, and 6 m, respectively. We mainly compare the results of the duplicate and interpolating placement schemes discussed in Section 3. For baseline reference, we also calculate the theoretical lower bound of the number of sensors required by $\left\lceil \frac{|\mathcal{I}|}{\pi r_s^2} \right\rceil \times k$, where $|\mathcal{I}|$ is the area of $\mathcal{I}$ (i.e., $10^6\,\mathrm{m}^2$ in this experiment). Note that the above lower bound can never be achieved, because it does not consider the connectivity and coverage overlapping between sensors.

Fig. 7 illustrates the numbers of sensors required when the desired coverage level k increases from two to seven. When $k = 2$, the interpolating scheme requires slightly more sensors compared with the duplicate scheme, because the former needs an extra row of sensors to ensure 2-coverage of $\mathcal{I}$'s boundary. However, when $k \geq 3$, the interpolating scheme can save approximately 19.4% ∼ 32.5% and 10.1% ∼ 16.8% sensors as opposed to the duplicate scheme in the case of $r_c \leq \frac{\sqrt{3}}{2}r_s$ and $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$, respectively. When $r_c > \frac{2+\sqrt{3}}{3}r_s$, the interpolating scheme works the same as the duplicate scheme, so they require the same number of sensors. Note that when r_c becomes larger, our placement schemes will be dominated by the value of r_s. So the numbers of sensors required by the duplicate and interpolating schemes are closer to the theoretical lower bound as r_c increases.

5.2 Performances of the Sensor Dispatch Schemes

In the second experiment, we estimate the total moving energy and average moving distance of sensors when different dispatch schemes are adopted. We design a field $\mathcal{A}$ as a 600 m × 600 m square region. The area of interest $\mathcal{I}$ is a 300 m × 300 m square region located at the center of $\mathcal{A}$. Three scenarios, namely *hollow*, *right*, and *central*, are considered. In the hollow scenario, sensors are randomly distributed inside the region of $\mathcal{A}-\mathcal{I}$. In the right scenario, sensors are arbitrarily placed inside a 150 m × 600 m rectangle region located at the right side of $\mathcal{A}-\mathcal{I}$. In the central scenario, sensors are initially concentrated inside a 100 m × 100 m square region located at the center of $\mathcal{I}$. With the setting of (r_c, r_s) = (34.7 m, 20.0 m), (24.1 m, 13.9 m), (19.3 m, 11.1 m), (16.7 m, 9.62 m), (14.9 m, 8.6 m), (13.4 m, 7.71 m), and (12.5 m, 7.16 m), we can obtain 100, 200, 300, 400, 500, 600, and 700 locations to be placed with sensors inside $\mathcal{I}$, respectively, according to the interpolating placement scheme (in the case of $r_c > \frac{2+\sqrt{3}}{3}r_s$). We set the desired coverage level $k = 3$, so that there will be 300, 600, 900, 1200, 1500, 1800, and 2100 sensors needed to be dispatched to $\mathcal{I}$. The moving speed of each sensor is set to 1 m/s. The moving energy cost e_i^{move} of a sensor i is randomly selected as [0.8 J (joule), 1.2 J] per meter. For our sensor dispatch schemes, the two timers $T_{\mathrm{update\_OCC}}$ and $T_{\mathrm{broadcast}}$ are set to five seconds. In the pattern-based dispatch scheme, we randomly select 10%, 20%, and 30% target locations inside $\mathcal{I}$ as the seed locations. For comparison purpose, we design a *greedy* dispatch scheme, where sensors are assumed to know all target locations inside $\mathcal{I}$ and they will simply move toward their closest locations without exchanging any information with other sensors. In this case, a sensor can realize that its destination has been occupied by sufficient sensors only when the sensor moves close to its destination (i.e., not larger than the communication distance r_c). For baseline reference, we also design a *centralized* dispatch scheme. In this scheme, we assume that the sink knows the positions and statuses of all sensors and thus can calculate an optimal dispatch.

Fig. 8 shows the total moving energy and average moving distance of sensors under the greedy, competition-based, and centralized dispatch schemes. As can be seen, when the number of sensors increases, the average moving distances of the greedy and competition-based schemes also increase. This is because each sensor has to compete with more other sensors and thus increases its moving distance. Nevertheless, the greedy scheme will lead sensors to move much longer distances (and thus consumes more energy) compared with the competition-based scheme. This is because sensors just blindly move toward their nearest locations without exchanging necessary information to avoid moving to the same locations. In the hollow and right scenarios, the situation becomes worse as the number of sensors increases since the number of unnecessary contests also increases in the greedy scheme. In the central scenario, the average moving distance of the greedy scheme always keeps very high (as compared with the other two dispatch schemes), because sensors are initially concentrated in a small region. Thus, from Fig. 8, we can observe that simply taking a greedy strategy to dispatch sensors will make them exhaust much energy, thereby greatly shortening the network lifetime. On the other hand, by properly exchanging and maintaining necessary information of sensors, our competition-based scheme can consume slightly more energy compared with the centralized scheme (especially in the hollow and right scenarios). Note that in the central scenario, since sensors have similar initial positions, there will be more sensors that compete for the same destinations. Thus, the competition-based scheme will cause sensors to move longer distances compared with the centralized scheme.

Fig. 9 illustrates the total moving energy and average moving distance of sensors under the pattern-based and competition-based dispatch schemes. The competition-based scheme outperforms the pattern-based scheme, because sensors have full knowledge of target locations. In the pattern-based scheme, the average moving distance will arise as the number of sensors increases. This is because sensors have to compete for those few known locations in the beginning, thus increasing their moving distances. However, the average moving distance (and the total moving energy) of the pattern-based scheme can decrease when there are more target locations selected as seeds.

5.3 Effect of Seed Locations on the Pattern-Based Dispatch Scheme

The third experiment evaluates the effect of seed locations on the average moving energy of sensors in the pattern-based dispatch scheme. In this experiment, we set the number of

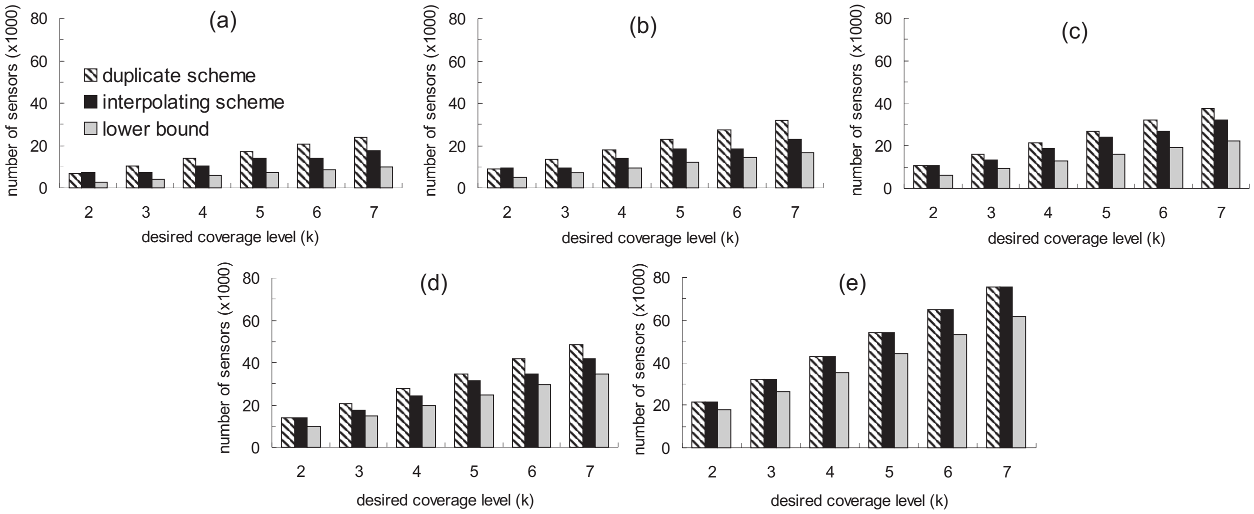


Fig. 7: Comparison on the numbers of sensors required under different coverage level k. (a) $r_c < \frac{\sqrt{3}}{2}r_s$. (b) $r_c \approx \frac{\sqrt{3}}{2}r_s$. (c) $\frac{\sqrt{3}}{2}r_s < r_c < \frac{2+\sqrt{3}}{3}r_s$. (d) $r_c \approx \frac{2+\sqrt{3}}{3}r_s$. (e) $r_c > \frac{2+\sqrt{3}}{3}r_s$

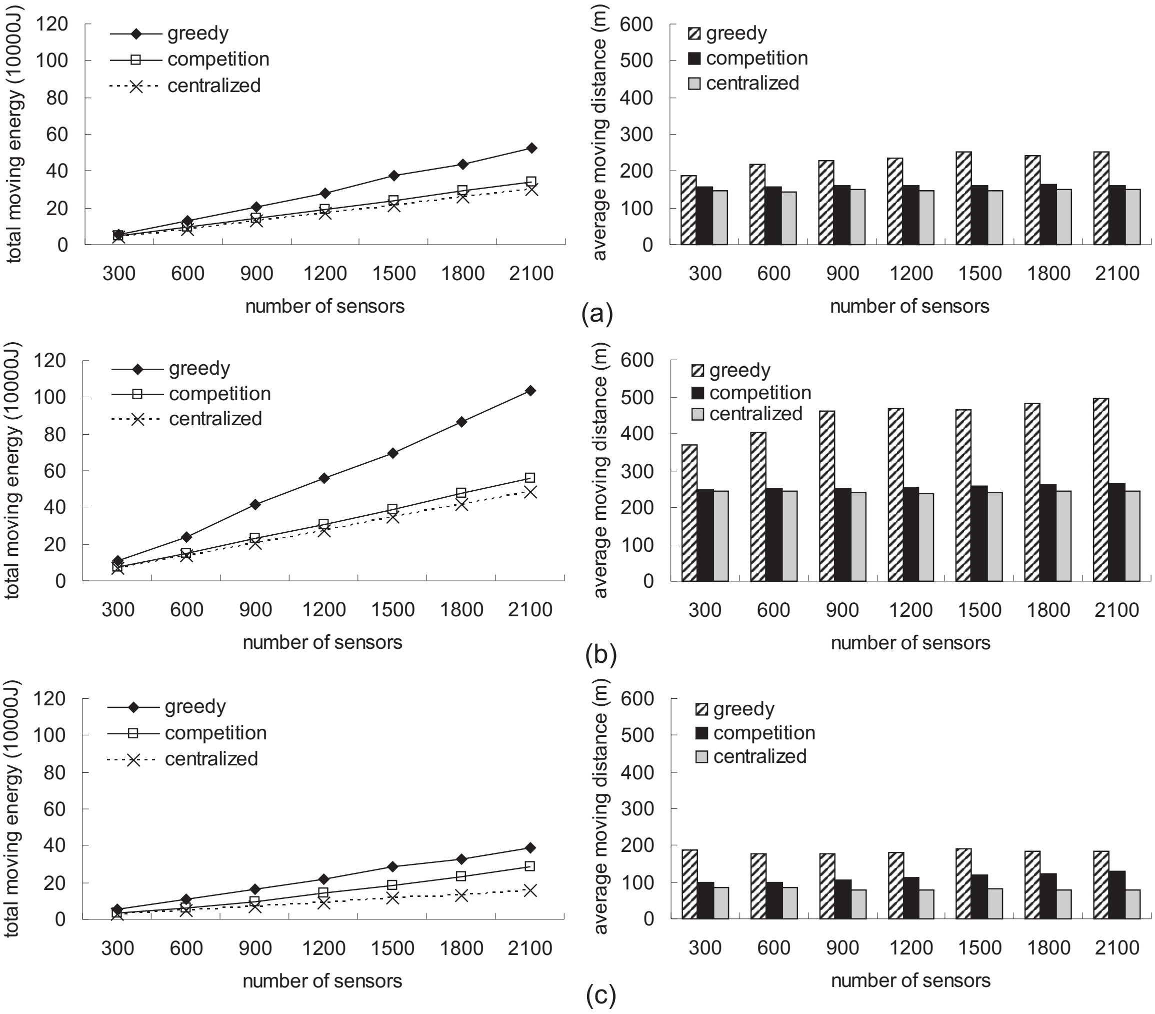


Fig. 8: Comparison on the total moving energy and average moving distance of sensors under the greedy, competition-based, and centralized dispatch schemes. (a) Hollow scenario. (b) Right scenario. (c) Central scenario.

sensors as 600 and 1500, and randomly select 5% to 70% target locations inside $\mathcal{I}$ as the seed locations.

Fig. 10 shows the effect of seed locations. As can be seen, the average moving energy of sensors can be reduced when the number of seed locations increases. When the percentage of seed locations arrives at 100%, the pattern-based scheme will work the same as the competition-based scheme. From Fig. 10(a), we can observe that in the hollow scenario, the difference between the average moving energies of the pattern-based scheme and competition-based scheme can be smaller

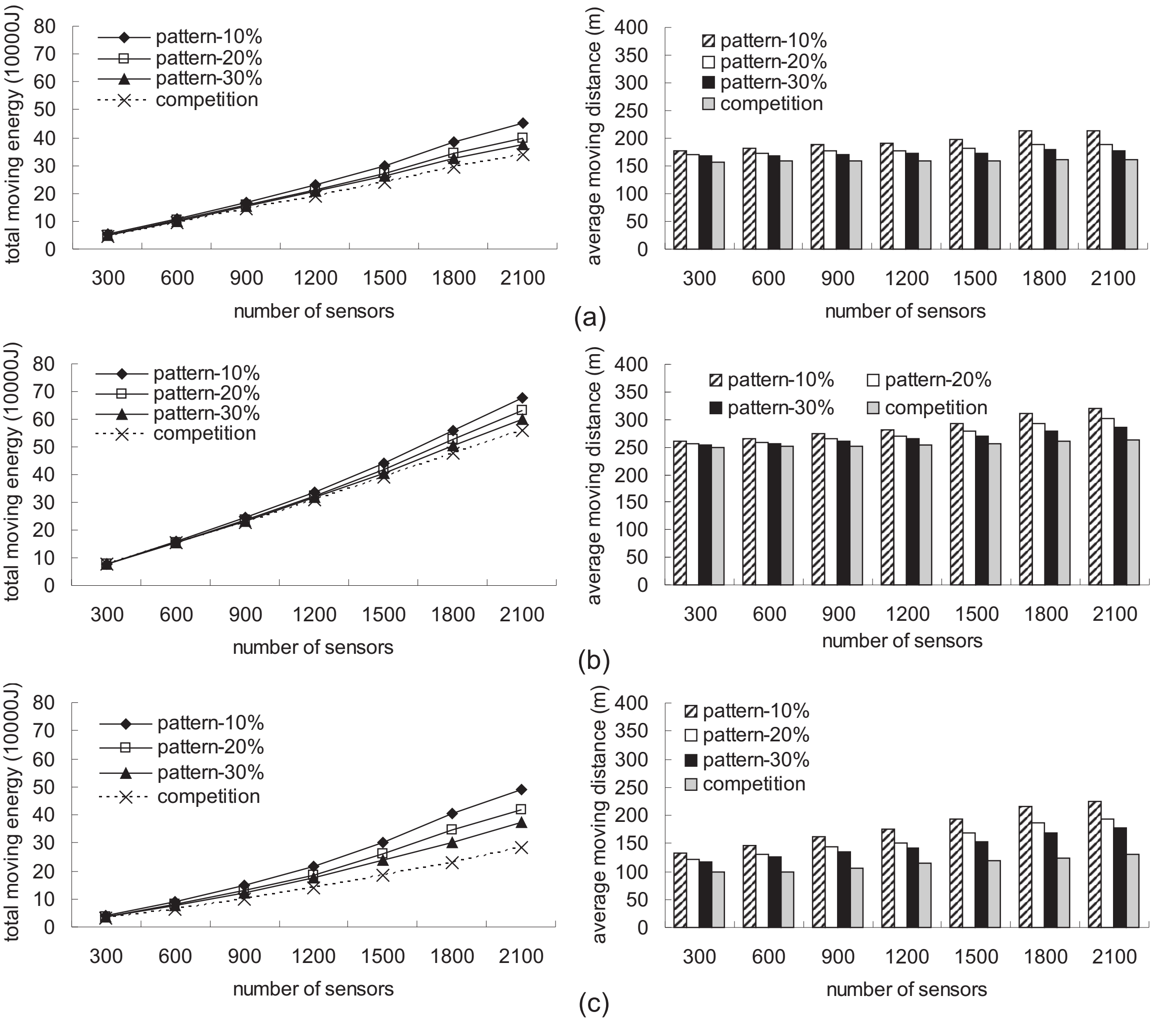


Fig. 9: Comparison on the total moving energy and average moving distance of sensors under the pattern-based and competition-based dispatch schemes. (a) Hollow scenario. (b) Right scenario. (c) Central scenario.

than 10 J when there are more than 40% ∼ 45% target locations selected as seeds. In the right scenario (i.e., Fig. 10(b)), when the number of sensors is 600 (respectively, 1500), such a difference can be achieved if there are more than 15% ∼ 20% (respectively, 35% ∼ 40%) target locations selected as seeds. On the other hand, in the central scenario (i.e., Fig. 10(c)), such a difference can be achieved if we select at least 45% ∼ 50% seed locations. To summarize, from Fig. 10, we can observe that the performance of the pattern-based scheme can be significantly improved by selecting 40% ∼ 50% target locations as seeds.

6 Conclusions and Future Work

In this paper, we have proposed systematical solutions to the k-coverage sensor placement problem and distributed sensor dispatch problem. Our placement solutions allow an arbitrary relationship of sensors' communication distance and their sensing distance, and can work properly under both binary and probabilistic sensing models. It is verified that the interpolating placement scheme requires fewer sensors to ensure k-coverage of the sensing field and connectivity of the network as compared with the duplicate placement scheme. Our dispatch solutions are based on the competitive nature of a distributed network. Simulation results have shown that the competition-based dispatch scheme performs better than the greedy and pattern-based dispatch schemes. However, by selecting sufficient seed locations, the pattern-based scheme can work as efficiently as the competition-based scheme.

As to future work, sensor deployment in arbitrary-shaped regions for multilevel coverage deserves further study. When the area of interest $\mathcal{I}$ is of an arbitrary shape, one potential approach is to form a rectangle region that can fully cover $\mathcal{I}$. Then, we can apply our solution to this rectangle and then remove those sensors that are outside the area of interest. Another way is to approximate $\mathcal{I}$ by multiple smaller rectangles. In our model, sensors' energy drain is at a constant speed when moving around. More sophisticated energy consumption models of mobile sensors can be defined, and this deserves further investigation. For example, a start-up energy cost may be incurred when first moving a sensor, and a cost may be incurred to enforce a sensor to turn around. In addition, variable moving speeds can be considered too.

References

[1] D. Nicules and B. Nath, "Ad-hoc positioning system (APS) using AoA," *Proc. IEEE INFOCOM*, pp. 1734–1743, 2003.

[2] L.A. Klein, "A boolean algebra approach to multiple sensor voting fusion," *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 317–327, 1993.

[3] T. Sun, L.J. Chen, C.C. Han, and M. Gerla, "Reliable sensor networks for planet exploration," *Proc. IEEE Int'l Conf. Networking, Sensing and Control*, pp. 816–821, 2005.

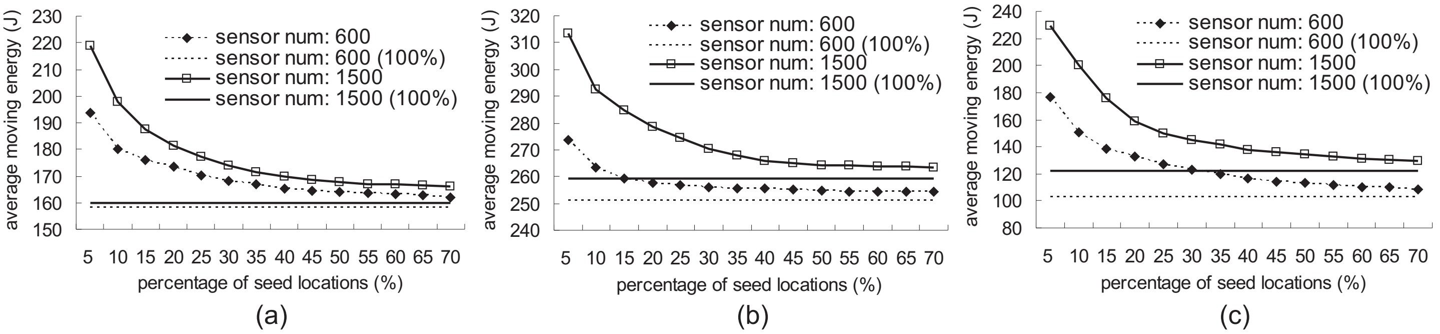


Fig. 10: Effect of seed locations on the average moving energy of sensors in the pattern-based dispatch scheme. (a) Hollow scenario. (b) Right scenario. (c) Central scenario.

[4] D. Tian and N.D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," *Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications*, pp. 32–41, 2002.

[5] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," *Proc. IEEE Int'l Symp. Information Processing in Sensor Networks*, pp. 424–432, 2004.

[6] M. Cardei, M.T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," *Proc. IEEE INFOCOM*, pp. 1976–1984, 2005.

[7] T.C. Shermer, "Recent results in art galleries," *Proc. IEEE*, vol. 80, no. 9, pp. 1384–1399, 1992.

[8] M.D. Adickes, R.E. Billo, B.A. Norman, S. Banerjee, B.O. Nnaji, and J. Rajgopal, "Optimization of indoor wireless communication network layouts," *IIE Trans.*, vol. 34, no. 9, pp. 823–836, 2002.

[9] K.S. Tang, K.F. Man, and K.T. Ko, "Wireless LAN design using hierarchical genetic algorithm," *Proc. Int'l Conf. Genetic Algorithm*, pp. 629–635, 1997.

[10] L. Raisanen, R.M. Whitaker, and S. Hurley, "A comparison of randomized and evolutionary approaches for optimizing base station site selection," *Proc. ACM Symp. Applied Computing*, pp. 1159–1165, 2004.

[11] H.R. Anderson and J.P. McGeehan, "Optimizing microcell base station locations using simulated annealing techniques," *Proc. IEEE Vehicular Technology Conf.*, pp. 858–862, 1994.

[12] S.S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," *Proc. IEEE Wireless Comm. and Networking Conf.*, pp. 1609–1614, 2003.

[13] F.Y.S. Lin and P.L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," *IEEE Comm. Letters*, vol. 9, no. 1, pp. 43–45, 2005.

[14] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," *Proc. Int'l Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[15] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," *Proc. IEEE Wireless Internet Conf.*, pp. 114–121, 2005.

[16] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.

[17] C.F. Huang and Y.C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[18] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, pp. 28–37, 2003.

[19] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems*, pp. 28–39, 2003.

[20] Z. Zhou, S. Das, and H. Gupta, "Connected k-coverage problem in sensor networks," *Proc. IEEE Int'l Conf. Computer Comm. and Networks*, pp. 373–378, 2004.

[21] S. Yang, F. Dai, M. Cardei, and J. Wu, "On multiple point coverage in wireless sensor networks," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems*, 2005.

[22] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Ensuring area k-coverage in wireless sensor networks with realistic physical layers," *Proc. IEEE Conf. Sensors*, pp. 880–883, 2006.

[23] J.P. Sheu and H.F. Lin, "Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks," *Proc. IEEE Wireless Comm. and Networking Conf.*, pp. 2631–2636, 2007.

[24] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *IEEE Network*, vol. 18, no. 4, pp. 36–44, 2004.

[25] Z. Butler and D. Rus, "Event-based motion control for mobile-sensor networks," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 34–42, 2003.

[26] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," *Proc. IEEE INFOCOM*, pp. 1293–1303, 2003.

[27] G. Wang, G. Cao, and T.L. Porta, "Movement-assisted sensor deployment," *Proc. IEEE INFOCOM*, pp. 2469–2479, 2004.

[28] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Trans. Systems, Man and Cybernetics–Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005.

[29] K. Akkaya and M. Younis, "C^2AP: coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks," *Proc. IEEE Int'l Performance, Computing, and Comm. Conf.*, pp. 281–288, 2007.

[30] G. Wang, G. Cao, T.L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," *Proc. IEEE INFOCOM*, pp. 2302–2312, 2005.

[31] X. Du and F. Lin, "Improving sensor network performance by deploying mobile sensors," *Proc. IEEE Int'l Performance, Computing, and Comm. Conf.*, pp. 67–71, 2005.

[32] T.A. Dahlberg, A. Nasipuri, and C. Taylor, "Explorebots: a mobile network experimentation testbed," *Proc. ACM SIGCOMM Workshop Experimental Approaches to Wireless Network Design and Analysis*, pp. 76–81, 2005.

[33] D. Johnson, T. Stack, R. Fish, D.M. Flickinger, L. Stoller, R. Ricci, and J.Lepreau, "Mobile Emulab: a robotic wireless and sensor network testbed," *Proc. IEEE INFOCOM*, 2006.

[34] Y.C. Tseng, Y.C. Wang, K.Y. Cheng, and Y.Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[35] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Trans. Computers*, vol. 54, pp. 978–991, 2005.

[36] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice (fourth edition)*, Springer Verlag, 1997.

[37] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Comm.*, vol. 7, no. 5, pp. 28–34, 2000.

[38] L. Hu and D. Evans, "Localization for mobile sensor networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking*, pp. 45–57, 2004.

[39] R. Rao and G. Kesidis, "Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 3, pp. 225–231, 2004.

[40] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T.H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," *Proc. ACM Int'l Symp. Mobile Ad Hoc Networking and Computing*, pp. 131–142, 2006.

[41] R. Kershner, "The number of circles covering a set," *American J. Mathematics*, vol. 61, pp. 665–671, 1939.

[42] *IEEE Standard 802.15.4-2003, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, 2003, IEEE.