Optimized Barrier Location for Barrier Coverage in Mobile Sensor Networks

Xiaoyun Zhang, Mathew L. Wymore, and Daji Qiao
Iowa State University, Ames, IA 50011
{zxydut, mlwymore, daji}@iastate.edu

***Abstract*—Barrier coverage is an important application of sensor networks. This paper studies how to build a strong barrier with mobile sensors in which the maximum moving distance of sensors is minimized. Our work differs from others in the way the y-coordinate of the barrier is determined. We optimize the y-coordinate of the barrier instead of fixing it *a priori*. An efficient algorithm is proposed, in which the search space of the y-coordinate of the barrier is first discretized and then searched over iteratively. In the theoretical worst case, $O(N^4)$ iterations may be needed to find the optimal barrier location, where N is the number of sensors, but in practice, our algorithm requires less than $O(N^2)$ iterations, as confirmed in simulation.**

I. Introduction

Intruder detection and border surveillance are intuitive applications of sensor networks. In these applications, sensors are deployed along the perimeter of a protected area such that no intruder can cross the perimeter without being detected. The arrangement of sensors for this purpose is referred to as the *barrier coverage* problem in sensor networks.

Early research in barrier coverage focused on static sensors [1]–[4]. Different kinds of barrier coverage were defined and the corresponding critical densities of sensors were calculated. Also, various schemes were proposed to identify the barriers formed by static sensors. More recent research has examined barrier coverage with mobile sensors [5]–[8]. The primary focus of these papers has been minimizing the moving distances of sensors when building the barrier. These works either restricted the sensors' deployment and movement regions to a 1-dimensional (1D) line, or fixed the y-coordinate of the barrier if considering a 2-dimensional (2D) rectangular deployment region with the minimum number of sensors. In practice, it is difficult to deploy sensors along a 1D line, and in the 2D case, fixing the y-coordinate of the barrier *a priori* results in a suboptimal solution, since the y-coordinate of the barrier affects the moving distance of sensors.

In this paper, we study the *strong barrier coverage* problem in a 2D rectangular region, where the goal is to detect intruders that may follow any path to cross the deployment region. This is in contrast to *weak barrier coverage*, which deals with intruders that only perpendicularly cross the deployment region. We assume that a strong barrier is formed with the minimum number of sensors, along a horizontal line that spans the length of the deployment region. We present an algorithm that decides the optimal y-coordinate of the barrier, and identifies a subset of sensors to move to their corresponding destinations, so that the maximum moving distance of sensors is minimized. To achieve this, our algorithm first discretizes the search space for the y-coordinate of the barrier, then quickly iterates over this discretized search space. Simulation results show that our algorithm is efficient and scalable.

The rest of this paper is organized as follows. In Section II, we discuss related work in detail. In Section III, the system model is presented and the problem is formally defined. We present our algorithm in Section IV and evaluate it in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

Various 1D mobile barrier coverage problems have been studied in [5, 9, 10]. In these papers, mobile sensors are initially deployed on the line where the barrier will be built, then the sensors move along this line to form the barrier. The sensors move according to various objectives, such as minimizing the maximum (min-max) or the sum (min-sum) of moving distances, maximizing the coverage, or minimizing the number of moving sensors. Regardless of the objective, the y-coordinate of the barrier is not considered because the sensors are assumed to already be on the line.

Barrier coverage has also been studied for mobile sensors deployed in a 2D region. Bhattacharya et al. [5] solved the min-max and min-sum problems for sensors moving from a circle's interior to its circumference to form a barrier. Their algorithm optimizes an angle variable, analogous to our y-coordinate of the barrier. However, their solution cannot be applied to our problem due to different search spaces of the angle variable and the y-coordinate, and the difference in how these factors decide sensors' final positions. Saipulla et al. [6] studied the min-max problem of forming a barrier with a minimum number of mobile sensors in a rectangular region. However, they fixed the y-coordinate of the barrier *a priori*. Shen et al. [7] did not fix the y-coordinate of the barrier in their investigation into the min-sum problem for a rectangular region, but they instead fixed the order of sensors along the barrier. In contrast, our algorithm does not fix either the y-coordinate or the order of the sensors along the barrier, thus allowing for optimal min-max results with no assumptions about how the sensors are deployed.

Distributed algorithms have also been proposed for the barrier coverage problem [7, 8, 11, 12], but the goal of these algorithms is usually to reduce the communication between sensors. They are suboptimal when considering objectives such as min-max or min-sum of sensors' moving distances.

III. System Model and Problem Statement

A. System Model

1) Sensor Network: We study a network of N mobile sensors deployed in a long rectangular region of size $L \times W$, where $L \gg W$, as shown in Fig. 1. Sensors are named s_1 to s_N from left to right of the deployment region, and the initial position of sensor s_i is denoted by (x_i, y_i). The set of all the sensors is denoted by S. We adopt the widely-used disk coverage model and denote the coverage radius as R. An intruder can be detected by a sensor if and only if it is within R of the sensor. In addition, we assume sensors can acquire their positions from GPS or another localization scheme.

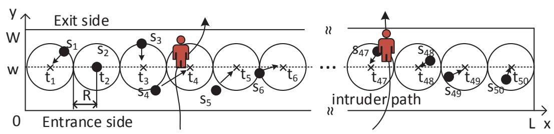


Fig. 1. An example sensor network of 50 mobile sensors.

2) Intruder and Barrier: We assume that intruders may take any path to cross the deployment region from bottom to top, as shown in Fig. 1. In order to detect such intruders, *strong barrier coverage* is required. This is in contrast to *weak barrier coverage*, which assumes intruders only take paths perpendicular to the x-axis of the deployment region.

The minimum number of sensors needed to form a strong barrier is $N_{\text{min}} = \lceil \frac{L}{2R} \rceil$. For simplicity, and without loss of generality, we assume that L is a multiple of $2R$; hence, $N_{\text{min}} = \frac{L}{2R}$. To form a strong barrier with N_{min} sensors, which is the focus of our study in this paper, these sensors must be aligned along a horizontal line parallel to the x-axis of the deployment region. In other words, the destination positions of these sensors (denoted by t_j, $1 \le j \le N_{\text{min}}$) must have the coordinates of (α_j, w_j), where $\alpha_j = (2j-1)R$, $1 \le j \le N_{\text{min}}$, and $w_1 = w_2 = \cdots = w_{N_{\text{min}}}$. We use T to denote the set of destination positions, and use w to denote their common y-coordinate, called the *barrier location*.

3) System: We assume that the sensor network remains connected during sensor movement, and that there is a central processing unit which collects information from sensors, executes the proposed algorithm, and disseminates the movement strategy to sensors.

B. Problem Statement

Our goal is to minimize the maximum moving distance of mobile sensors that form a strong barrier, with selected N_{min} sensors from a network of N ($N \ge N_{\text{min}}$) sensors with known initial positions. More specifically, our algorithm must decide (1) where the barrier shall be formed, i.e., the optimal barrier location, denoted by w^{opt}; and (2) how to form the barrier at w^{opt}, i.e., the optimal sensor movement strategy, which can be described as a mapping function M^{opt} from S to T. In general, a mapping function $M : S \to T$ is defined as follows: $M(i) = j \neq 0$ means that sensor s_i moves to destination t_j, while $M(i) = 0$ means that sensor s_i remains stationary. Sensors not used in the initial barrier may participate in forming future barriers after the operational lifetime of the current barrier has expired. Formally, the problem we try to address in this paper is described as follows:

Given:
- rectangular deployment region: $L \times W$
- coverage radius: R
- total number of deployed sensors: N
- initial sensor positions: (x_i, y_i) for each $s_i \in S$
- x-coordinate of the destinations: α_j for each $t_j \in T$

Output:
- optimal barrier location w^{opt} and sensor movement strategy M^{opt}: $\{w^{\text{opt}}, M^{\text{opt}}\} = \arg\min_{\{w,M\}} D(M, w)$ where

$$D(M, w) = \max_{s_i \in S, M(i) \neq 0} \sqrt{\left(x_i - \alpha_{M(i)}\right)^2 + (y_i - w)^2}.$$

Constraint: $0 \le w^{\text{opt}} \le W$

Essentially, we solve a min-max problem. Our motivation is the assumption that the sensor with the maximum moving distance runs out of energy first, which would break the barrier. Therefore, our scheme produces a barrier with the longest operational lifetime. We use $D(M, w)$ to denote the maximum moving distance of the sensors when they follow strategy M to form a barrier at location w. We restrict the barrier location between 0 and W.

IV. Proposed Scheme

A. Overview of the Proposed Scheme

To determine the optimal barrier location w^{opt}, we first reduce its search space from a continuous range $[0, W]$ to a discrete set with less than $N^2 N_{\text{min}}^2$ points, which are called *candidate barrier locations*. Afterwards, we propose an efficient iterative algorithm to search over this discrete set. The worst-case number of iterations is $O(N^2 N_{\text{min}}^2)$, but in practice, our algorithm uses approximately $O(N N_{\text{min}})$ iterations.

B. Identification of Candidate Barrier Locations

Candidate barrier locations are derived from the minimum and intersection points of the group of functions defined below.

Definition 1 (Function of Moving Distance)**.** *Suppose a sensor s_i moves to a destination t_j. The moving distance of s_i can be represented as a function of the barrier location w, i.e.,*

$$f_{i,j}(w) = \sqrt{[x_i - (2j-1)R]^2 + (y_i - w)^2}, \qquad (1)$$

where $0 \le w \le W$.

We use F to denote the set of all f associated with a sensor network of N sensors, i.e.,

$$F = \{f_{i,j}(w) | 1 \le i \le N, 1 \le j \le N_{\text{min}}, 0 \le w \le W\}. \qquad (2)$$

For simplicity, we also define $f_{i,0}(w) = 0, \forall i, \forall w$. With these functions defined, we can now define the candidate barrier locations.

Definition 2 (Candidate Barrier Locations)**.** *The set of candidate barrier locations is $\Phi = \Phi^{mins} \cup \Phi^{ints}$, where*
- $\Phi^{mins} = \bigcup_{\forall f_{i,j} \in F} \{w | \arg\min_{w} f_{i,j}(w)\}$ *includes all the w values that yield a minimum value for at least one of the $f_{i,j}$ functions in F;*
- $\Phi^{ints} = \bigcup_{\forall f_{i,j} \in F, f_{m,n} \in F} \{w | f_{i,j}(w) = f_{m,n}(w)\}$ *includes all the w values at which two functions in F intersect.*

Let $|\Phi|$ denote the total number of candidates in Φ. We sort these candidates in ascending order and name them $\{K_i | i = 1, \cdots, |\Phi|\}$. Let $|\Phi^{\text{mins}}|$ and $|\Phi^{\text{ints}}|$ denote the number of candidates in Φ^{mins} and Φ^{ints}, respectively. Then, we have $O(N^4)$ candidates, as follows:

$$|\Phi| = |\Phi^{\text{mins}}| + |\Phi^{\text{ints}}| \le NN_{\text{min}} + \binom{NN_{\text{min}}}{2} = O(N^2N_{\text{min}}^2) = O(N^4). \tag{3}$$

Fig. 3 plots all the functions in F corresponding to the sensor network shown in Fig. 2. In this example, $|\Phi^{\text{mins}}| = 3$, $|\Phi^{\text{ints}}| = 10$, and $|\Phi| = 13$.

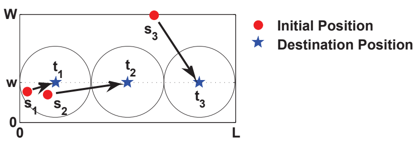


Fig. 2. An example sensor network of 3 mobile sensors.

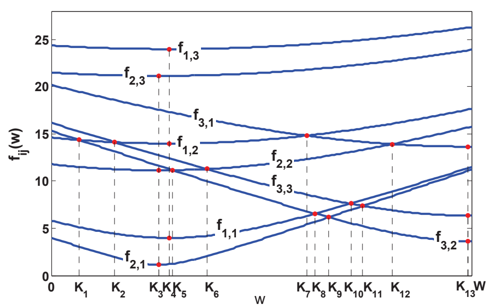


Fig. 3. For the example network in Fig. 2, there are 9 functions in F and 13 candidate barrier locations. $\Phi^{\text{mins}} = \{K_3, K_4, K_{13}\}$ and $\Phi^{\text{ints}} = \{K_1, K_2, K_5, K_6, K_7, K_8, K_9, K_{10}, K_{11}, K_{12}\}$.

The optimal barrier location w^{opt} is guaranteed to be at one of the candidate barrier locations, as Theorem 1 below proves, any movement strategy M for a barrier at any w between two adjacent candidate locations can always yield a smaller maximum moving distance at one of these two candidates.

Theorem 1. *$\forall M$, $\forall w \in (K_j, K_{j+1})$, where $1 \le j \le |\Phi| - 1$, $D(M, w) > \min\{D(M, K_j), D(M, K_{j+1})\}$.*

Proof: Let s_{i^*} denote the sensor that has the maximum moving distance in M at w. This means

$$\forall i,\ f_{i,M(i)}(w) \le f_{i^*,M(i^*)}(w). \tag{4}$$

Since K_j and K_{j+1} are adjacent candidate locations, no other functions in F intersect between them. Therefore, (4) implies that

$$\begin{cases} f_{i,M(i)}(K_j) \le f_{i^*,M(i^*)}(K_j),\ \forall i, \\ f_{i,M(i)}(K_{j+1}) \le f_{i^*,M(i^*)}(K_{j+1}),\ \forall i. \end{cases} \tag{5}$$

In other words,

$$\begin{cases} f_{i^*,M(i^*)}(K_j) = \max\limits_{1\le i\le N} f_{i,M(i)}(K_j), \\ f_{i^*,M(i^*)}(K_{j+1}) = \max\limits_{1\le i\le N} f_{i,M(i)}(K_{j+1}). \end{cases} \tag{6}$$

Moreover, according to the definition of candidate barrier locations, all the functions in F are monotone between two adjacent candidates such as K_j and K_{j+1}. Therefore, we have:

$$\begin{aligned} D(M, w) &= \max_{1\le i\le N} f_{i,M(i)}(w) = f_{i^*,M(i^*)}(w) && \text{(Definition)} \\ &> \min\{f_{i^*,M(i^*)}(K_j), f_{i^*,M(i^*)}(K_{j+1})\} && \text{(Monotone)} \\ &= \min\left\{\max_{1\le i\le N} f_{i,M(i)}(K_j), \max_{1\le i\le N} f_{i,M(i)}(K_{j+1})\right\} \\ &= \min\{D(M, K_j), D(M, K_{j+1})\}. \end{aligned} \tag{7}$$

∎

C. Iterative Algorithm

With the candidate barrier locations being identified, we further reduce the search complexity by proposing an iterative algorithm in which only a small portion of candidates are checked. Fig. 4 shows the flowchart of our iterative algorithm, which is explained in detail next.

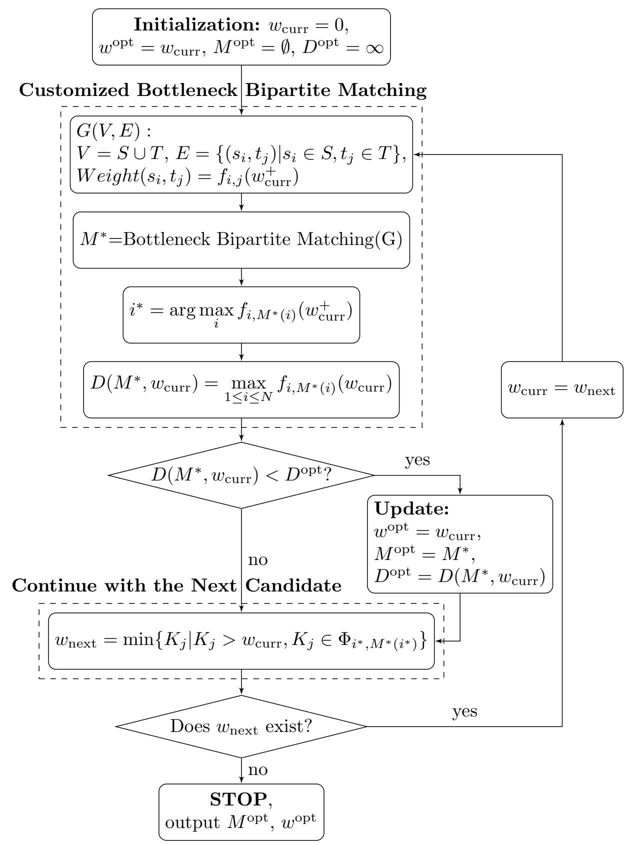


Fig. 4. Flowchart of the iterative algorithm (note that $w^+_{\text{curr}} = w_{\text{curr}} + \delta w$).

The algorithm starts with $w_{\text{curr}} = 0$. After initialization with the optimal min-max moving distance D^{opt} set to infinity, the Bottleneck Bipartite Matching (BBM) algorithm [13] is applied to determine the best movement strategy for the current barrier location as follows. The input to BBM is a weighted bipartite graph $G(V, E)$:

- $V = S \cup T$ where S is the set of sensors, T is the set of destinations along w_{curr};
- $E = \{(s_i, t_j)|\ s_i \in S, t_j \in T\}$;
- Weight$(s_i, t_j) = f_{i,j}(w^+_{\text{curr}})$ where $w^+_{\text{curr}} = w_{\text{curr}} + \delta w$ and δw is a positive offset which is sufficiently small so that w^+_{curr} does not reach the next candidate in Φ. Recall that $f_{i,j}$ is the function of moving distance for sensor s_i to reach destination t_j.

BBM returns a max-cardinality matching whose maximum edge weight is minimized. In other words, it produces a movement strategy which minimizes the maximum moving distance of sensors from their initial positions to the destination positions along w_{curr}.

In this customized BBM algorithm, the offset δw is used as a tie breaker in case there are multiple movement strategies that all yield the same min-max moving distance at w_{curr}. For example, in Fig. 5, at $w_{\text{curr}} = K_2$, there exist two such movement strategies: $M_1 = \{1{\to}2, 2{\to}1, 3{\to}3\}$ and $M_2 = \{1{\to}1, 2{\to}2, 3{\to}3\}$. By adding a small offset, the tie is broken and M_2 is chosen which yields a smaller maximum moving distance than M_1 at K_2^+. This serves as an important base for the next round of iteration, as shown in the proof of Theorem 2.

As shown in the flowchart, we record the output of the customized BBM algorithm as follows:

- M^*: the min-max matching at w_{curr}^+;
- i^*: the index of the sensor that has the maximum moving distance in M^* at w_{curr}^+;
- $D(M^*, w_{\text{curr}})$: the min-max moving distance at w_{curr}.

The next candidate location to check is

$$w_{\text{next}} = \min\{K_j | K_j > w_{\text{curr}}, K_j \in \Phi_{i^*,M^*(i^*)}\}, \tag{8}$$

where $\Phi_{i^*,M^*(i^*)}$ is the set of candidates along $f_{i^*,M^*(i^*)}$ which is formally defined below.

Definition 3 (Candidate Barrier Locations along $f_{i,j}$)**.** *The set of candidate barrier locations along the $f_{i,j}$ function is $\Phi_{i,j} = \Phi_{i,j}^{mins} \cup \Phi_{i,j}^{ints}$, where*

- $\Phi_{i,j}^{mins} = \{w | \arg\min_w f_{i,j}(w)\}$ *includes the w value at which $f_{i,j}$ achieves its minimum;*
- $\Phi_{i,j}^{ints} = \bigcup_{\forall f_{m,n} \in F} \{w | f_{i,j}(w) = f_{m,n}(w)\}$ *includes all the w values at which $f_{i,j}$ intersects with another function in F.*

Note that w_{next} is the first candidate location along the $f_{i^*,M^*(i^*)}$ function after w_{curr}. There might exist other candidate locations between w_{curr} and w_{next} but belong to a different f function, which are skipped in our iterative algorithm to reduce the search complexity. For example, in Fig. 5, the next candidate to check after K_2 is K_6. It is determined with the following steps: (1) the min-max matching at K_2^+ is $M^* = \{1{\to}1, 2{\to}2, 3{\to}3\}$; (2) the sensor that has the maximum moving distance in M^* at K_2^+ is s_3; (3) the first candidate along $f_{3,3}$ after K_2 is K_6. Comparing Fig. 5 with Fig. 3, we can see that candidate locations K_3, K_4, and K_5 are skipped. The reason why these candidate locations can be skipped is that, as we will show in Theorem 2, at any w between w_{curr} and w_{next}, the maximum moving distance of any movement strategy is always larger than or equal to that of M^* at w_{curr} or w_{next}.

Theorem 2. *$\forall M'$, $\forall w \in [w_{curr}, w_{next}]$, where w_{curr} is the current candidate location in the iterative algorithm, and w_{next} is the next candidate location to check as defined in (8), we always have $D(M', w) \geq \min\{D(M^*, w_{curr}), D(M^*, w_{next})\}$, where M^* is the min-max matching at w_{curr}^+.*

Proof: Recall that i^* is the index of the sensor that has the maximum moving distance in M^* at w_{curr}^+, i.e.,

$$i^* = \arg\max_i f_{i,M^*(i)}(w_{\text{curr}}^+). \tag{9}$$

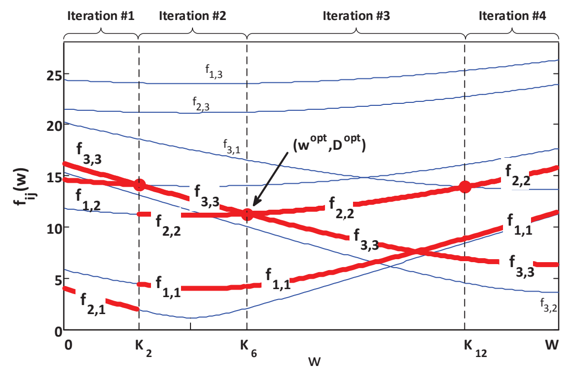


Fig. 5. The iterative algorithm starts with $w_{\text{curr}} = 0$. Iteration #1: The min-max matching at w_{curr}^+ is $\{1{\to}2, 2{\to}1, 3{\to}3\}$, in which s_3 has the maximum moving distance at w_{curr}^+. The next candidate to check is the first candidate location along $f_{3,3}$ after $w_{\text{curr}} = 0$, which is K_2. Iteration #2: The min-max matching at K_2^+ is $\{1{\to}1, 2{\to}2, 3{\to}3\}$, in which s_3 has the maximum moving distance at K_2^+. The next candidate to check is the first candidate along $f_{3,3}$ after K_2, which is K_6. The same process repeats for Iteration #3 and Iteration #4 till the algorithm terminates.

Therefore, the following inequality holds for all i:

$$f_{i^*,M^*(i^*)}(w_{\text{curr}}^+) \geq f_{i,M^*(i)}(w_{\text{curr}}^+). \tag{10}$$

As w_{next} is the first candidate location along the $f_{i^*,M^*(i^*)}$ function after w_{curr}, no other functions in F intersect with $f_{i^*,M^*(i^*)}$ between w_{curr} and w_{next}. Consequently, we have:

$$\forall w \in [w_{\text{curr}}, w_{\text{next}}], \forall i, f_{i^*,M^*(i^*)}(w) \geq f_{i,M^*(i)}(w). \tag{11}$$

Hence, we have:

$$\forall w \in [w_{\text{curr}}, w_{\text{next}}], f_{i^*,M^*(i^*)}(w) = D(M^*, w). \tag{12}$$

On the other hand, let i' denote the index of the sensor that has the maximum moving distance in M' at w_{curr}^+, i.e.,

$$i' = \arg\max_i f_{i,M'(i)}(w_{\text{curr}}^+). \tag{13}$$

As M^* is the min-max matching at w_{curr}^+, we have:

$$f_{i',M'(i')}(w_{\text{curr}}^+) \geq f_{i^*,M^*(i^*)}(w_{\text{curr}}^+). \tag{14}$$

Similarly, due to the fact that no other functions in F intersect with $f_{i^*,M^*(i^*)}$ between w_{curr} and w_{next}, we have:

$$\forall w \in [w_{\text{curr}}, w_{\text{next}}], f_{i',M'(i')}(w) \geq f_{i^*,M^*(i^*)}(w). \tag{15}$$

As the definition of $D(M', w)$ implies:

$$D(M', w) = \max_{1 \leq i \leq N} f_{i,M'(i)}(w) \geq f_{i',M'(i')}(w), \tag{16}$$

we have:

$$\forall w \in [w_{\text{curr}}, w_{\text{next}}], D(M', w) \geq f_{i^*,M^*(i^*)}(w). \tag{17}$$

Furthermore, as the $f_{i^*,M^*(i^*)}$ function is monotone between w_{curr} and w_{next}, (17) implies:

$$\begin{aligned} &\forall w \in [w_{\text{curr}}, w_{\text{next}}], \\ &D(M', w) \geq \min\{f_{i^*,M^*(i^*)}(w_{\text{curr}}), f_{i^*,M^*(i^*)}(w_{\text{next}})\}. \end{aligned} \tag{18}$$

Finally, by combining (12) with (18), we have:

$$\begin{aligned} &\forall w \in [w_{\text{curr}}, w_{\text{next}}], \\ &D(M', w) \geq \min\{D(M^*, w_{\text{curr}}), D(M^*, w_{\text{next}})\}. \end{aligned} \tag{19}$$

■

The algorithm terminates when w_{next} cannot be found, meaning that we have completed the search over the entire range $[0, W]$. Fig. 5 illustrates the iterative algorithm with an example and the iteration process is explained in its caption.

D. Complexity Analysis

In theory, there is a total of $O(N^2 N_{\text{min}}^2)$ candidates in Φ, which means that, in the worst case, we may need $O(N^2 N_{\text{min}}^2)$ iterations to complete the search. However, in practice, the number of iterations is more comparable to $O(N N_{\text{min}})$. This is because our algorithm checks the candidate barrier locations iteratively along a continuous function that is composed of multiple sections from different f functions. In other words, in each iteration, our algorithm only checks the candidate locations along a particular section of a single f function. Thus, the total number of checked locations is on the same order as the number of candidate locations along a single f function, which is $O(N N_{\text{min}})$. In Section V, we validate this complexity analysis by simulation.

V. Evaluation Results

We evaluate our algorithm by simulating two types of sensor deployment: random uniform deployment, and line-based deployment. In both scenarios, we compare the optimal min-max moving distance found by our algorithm to that of a naive algorithm that fixes the barrier location to $\frac{W}{2}$, which is the middle point of the deployment region. We refer to our algorithm as "Opt" and the naive algorithm as "Mid". The time complexity of our algorithm is also evaluated in terms of the number of checked candidate locations.

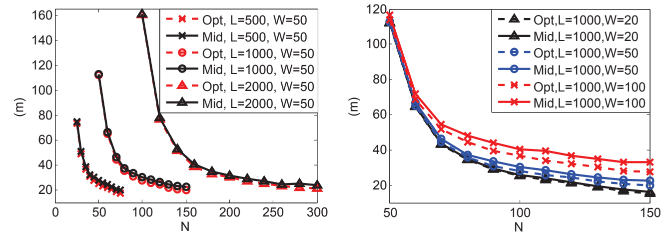
A. Uniform Deployment

We first evaluate our algorithm with a random uniform deployment of N sensors in an $L \times W$ region.

1) Min-max Moving Distance: Fig. 6 shows the min-max moving distances of the two algorithms varying with N in the uniform case. We test different sizes of deployment region. For each size, the effect of redundant sensors on the min-max moving distance is also tested.

We first note that, regardless of the size of the deployment region, the min-max moving distance of both algorithms decreases as the number of sensors increases. This is an intuitive result, as more sensors deployed in the area means a higher chance that one or more sensors will already be close to each final destination. We also note that, regardless of the size of the deployment region, Opt outperforms Mid more when N is larger. This is because the horizontal moving distance tends to dominate the overall 2D moving distance as $L \gg W$, but as the number of sensors increases, the horizontal moving distance decreases. Therefore, the benefit of optimizing vertical moving distance becomes relatively larger. Finally, we note from Fig. 6(b) that, given the same L and N, the difference between Opt and Mid is larger when W is larger. This is again due to the increased proportion of vertical distance in the total moving distance.

Fig. 6 shows the average min-max moving distances of Opt and Mid. To further illustrate the improvement of Opt over Mid, Fig. 7 presents the cumulative distribution function (CDF) of the absolute and relative improvement of Opt over Mid in terms of min-max moving distance, which is a result of 1000 trials for each of three different setups shown in the figure. In the scenario where $N = 150$, Opt has a min-max moving distance which is on average 2.5 m or 11.2% less than that of Mid, but in around 40% of the trials, the improvement is greater. In the most extreme case, Opt reduces the min-max moving distance by 9.4 m or 38%. Similar observations can be made for the other two setups.



(a) Fixed W and varying L. (b) Fixed L and varying W.

Fig. 6. Min-max moving distance when N sensors are uniformly deployed in an $L \times W$ region. The coverage radius of each sensor is $R = 10$ m.

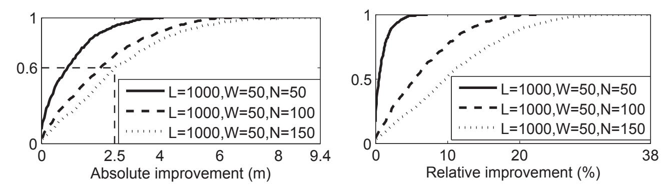


Fig. 7. CDF of the absolute and relative improvement of Opt over Mid in terms of min-max moving distance.

2) Number of Checked Candidate Locations: Table I shows the number of total and checked candidate locations of selected experiments from the uniform scenario. "Total" is the number of candidates in Φ, which is up to $N^2 N_{\text{min}}^2$, while "check" is the number of candidates checked in our iterative algorithm. When N, L, or W increases, the number of total and checked candidates increases accordingly. However, in any scenario, only a small portion of candidates are checked, with a number even less than $N N_{\text{min}}$, which indicates that our algorithm is efficient and scalable. For example, when $L = 2000$ m, $W = 100$ m, $N_{\text{min}} = 100$, given $N = 300$ sensors, $N N_{\text{min}} = 30000$, but only 1783 candidates are checked.

TABLE I

Total and checked candidates for the uniform deployment case. $R = 10$ m and $N_{\text{min}} = L/2R$.

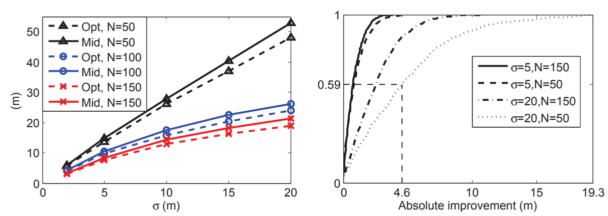
L=1000,W=50 $N_{\text{min}} = 50$			L=2000,W=50 $N_{\text{min}} = 100$			L=2000,W=100 $N_{\text{min}} = 100$		
N	total	check	N	total	check	N	total	check
50	31656	38	100	156114	54	100	512099	208
100	128167	306	200	629182	591	200	2062666	1286
150	288983	469	300	1422913	959	300	4663225	1783

B. Line-based Deployment

The second deployment scenario is the line-based sensor deployment strategy proposed in [14], where N sensors are deployed in an $L \times W$ region along the line of $y = \frac{W}{2}$. Each sensor s_i is deployed at its final position $[(2j-1)R, \frac{W}{2}]$ with a random x-axis error δ_i^x and y-axis error δ_i^y, where

$\delta_i^x, \delta_i^y \sim N(0, \sigma)$. In practice, this error could be the result of wind or other environmental conditions during an air drop. When redundant sensors are deployed, sensors are assumed to be dropped in groups, e.g., for $N = 2N_{\min}$, two sensors are dropped at each position.

1) Min-max Moving Distance: Fig. 8(a) shows the min-max moving distances of Opt and Mid with varying σ. As σ increases, the min-max moving distances of both Opt and Mid increase, because the sensors tend to be initially deployed farther from their final positions. Also, as σ increases, Opt outperforms Mid more. This is because when σ is larger, the sensors will be scattered in a wider region. It is more necessary to optimize the vertical moving distance in this case. Interestingly, when σ is fixed, the difference between Opt and Mid is larger when N is smaller, which is the opposite of the observations in the uniform deployment case. This is due to the following reasons. Firstly, when N increases and multiple sensors are dropped at each point, we tend to have a combination of sensors that yields an optimal barrier location close to $\frac{W}{2}$; therefore, the difference between the vertical moving distance of Opt and Mid is smaller. Secondly, the vertical and horizontal moving distances are comparable in the line-based deployment case; hence, the reduction of the vertical moving distance can be reflected in the overall 2D moving distance.



(a) Min-max moving distance vs. σ given three different N values.

(b) CDF of the absolute improvement of Opt over Mid in terms of min-max moving distance.

Fig. 8. The min-max moving distance for the line-based deployment strategy, with $L = 1000$ m, $W = 50$ m, $R = 10$ m, and $N_{\min} = 50$.

Similar to before, we plot in Fig. 8(b) the CDF of the improvement of Opt over Mid in four different setups. We can see that when $\sigma = 20$ m and $N = 50$, on average, the min-max moving distance of Opt is 4.6 m or 8.7% less than Mid. In the most extreme case, the min-max moving distance of Opt may be up to 19.3 m or 36% less than Mid. Similar observations can be made for the other setups.

2) Number of Checked Candidate Locations: Table II shows the number of total and checked candidate locations of selected experiments from the line-based deployment scenario. Similar to before, when N or σ increases, the number of total and checked candidates increases, but only a small proportion of the total candidates are checked in any scenario.

C. Discussions

As shown above, our proposed scheme outperforms the scheme that fixes the barrier location, significantly in terms of the min-max moving distance, thus saving much energy on sensor movement. As a tradeoff, it may increase the computational complexity at the central processing unit, which usually is much more powerful than sensor nodes and has a sustainable power supply. On the other hand, these two schemes incur the same communication overhead, as both algorithms are executed by the central processing unit thus all sensors need to report their initial positions to there.

TABLE II
TOTAL AND CHECKED CANDIDATES FOR THE LINE-BASED DEPLOYMENT CASE. $L = 1000$ M, $W = 50$ M, $R = 10$ M, AND $N_{\text{MIN}} = 50$.

N	$\sigma = 5$		$\sigma = 20$	
	total	check	total	check
50	6997	83	39561	197
100	30833	104	161154	338
150	72210	159	363082	495

VI. Conclusions and Future Work

In conclusion, we have shown that our proposed algorithm reduces the min-max moving distance for mobile sensors in a barrier coverage network by determining the optimal barrier location, instead of fixing it *a priori*. Depending on the deployment scheme, this reduction can be up to 38% of the min-max moving distance obtained by an algorithm that uses an intuitive fixed barrier location. Additionally, simulation results show that our proposed algorithm for finding the optimal barrier location is both efficient and scalable. Future work includes extending the proposed scheme to K-barrier coverage and to use the probabilistic coverage model, in which sensors may collaborate with each other to further reduce the moving distance and the number of sensors used.

Acknowledgment

Xiaoyun Zhang would like to thank China Scholarship Council for the financial support.

References

[1] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proc. of ACM MobiCom*, Aug. 2005.
[2] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," *in Proc. of ACM MobiHoc*, May 2008.
[3] A. Chen, S. Kumar, and T. H. Lai, "Designing localized algorithms for barrier coverage," *in Proc. of ACM MobiCom*, Sept. 2007.
[4] G. Yang and D. Qiao, "Barrier information coverage with wireless sensors," in *Proc. of IEEE INFOCOM*, Apr. 2009.
[5] B. Bhattacharya, B. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese, "Optimal movement of mobile sensors for barrier coverage of a planar region," in *Proc. of COCOA*, Aug. 2008.
[6] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang, "Barrier coverage with sensors of limited mobility," in *Proc. of ACM MobiHoc*, Sep. 2010.
[7] C. Shen, W. Cheng, X. Liao, and S. Peng, "Barrier coverage with mobile sensors," in *Proc. of IEEE I-SPAN*, May 2008.
[8] M. Eftekhari, E. Kranakis, D. Krizanc, O. Morales-Ponce, L. Narayanan, J. Opatrny, and S. Shende, "Distributed algorithms for barrier coverage using relocatable sensors," in *Proc. of ACM PODC*, July 2013.
[9] J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. Yazdani, "On minimizing the sum of sensor movements for barrier coverage of a line segment," in *Proc. of ADHOC-NOW*, Aug. 2010.
[10] M. Mehrandish, L. Narayanan, and J. Opatrny, "Minimizing the number of sensors moved on line barriers," in *Proc. of IEEE WCNC*, 2011.
[11] T. M. Cheng and A. V. Savkin, "A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 11, pp. 877–879, 2009.
[12] L. Kong, X. Liu, Z. Li, and M.-Y. Wu, "Automatic barrier coverage formation with mobile sensor networks," in *Proc. of IEEE ICC*, 2010.
[13] A. P. Punnen and K. Nair, "Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem," *Discrete Applied Mathematics*, vol. 55, no. 1, pp. 91–93, 1994.
[14] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks," in *Proc. of IEEE INFOCOM*, April 2009.