

Optimal Movement of Mobile Sensors for Barrier Coverage of a Planar Region*

Binay Bhattacharya^{*¶} Mike Burmester[†] Yuzhuang Hu^{*}
Evangelos Kranakis^{‡¶} Qiaosheng Shi^{*} Andreas Wiese[§]

Abstract

Intrusion detection, area coverage and border surveillance are important applications of wireless sensor networks today. They can be (and are being) used to monitor large unprotected areas so as to detect intruders as they cross a border or as they penetrate a protected area. We consider the problem of how to optimally move mobile sensors to the fence (perimeter) of a region delimited by a simple polygon in order to detect intruders from either entering its interior or exiting from it. We discuss several related issues and problems, propose two models, provide algorithms and analyze their optimal mobility behavior.

1 Introduction

Monitoring and surveillance are two of the main applications of wireless sensor networks today. Typically, one is interested in monitoring a given geographic region either for measuring and surveying purposes or for reporting various types of activities and events. Another important application concerns critical security and safety monitoring systems. One is interested in detecting intruders (or movements thereof) around *critical* infrastructure facilities and geographic delimiters (chemical plants, forests, etc). As a matter of fact, since the information security level of the monitoring system might change rapidly because of hostile attacks targeted at it, research efforts are currently underway to extend the scalability of wireless sensor networks so that they can be used to monitor international borders as well. For example, [11] reports the possibility of using wireless sensor networks for replacing traditional barriers (more than a kilometer long) at both the building and estate level. Also, “Project 28” concerns the construction of a virtual fence as a way to complement a physical fence that will include 370 miles of pedestrian fencing and 300 miles of vehicle barrier (see [8] which reports delays in its deployment along the U.S.-Mexico border).

To begin, we say that a point is *covered* by a sensor if it is within its range. In this paper we will use the concept of *barrier coverage* as used in [11] and which differs from the

^{*}School of Computing Science, Simon Fraser University, Vancouver, BC, Canada.

[†]Department of Computer Science, Florida State University, Tallahassee, Florida, USA.

[‡]School of Computer Science, Carleton University, Ottawa, Ontario, Canada.

[§]Institut für Mathematik, Technische Universität Berlin, Berlin, Germany.

[¶]Research supported in part by NSERC and MITACS.

*A first version of this manuscript was presented at the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA 2008), St John’s, Newfoundland.

more traditional concept of *full coverage*. In the latter case one is interested in covering the entire region by the deployment of sensors, while in the former all crossing paths through the region are covered by sensors. Thus, one is not interested in covering the entire deployment region but rather to detect potential intruders by guaranteeing that there is no path through this region that can be traversed undetected by an intruder that crosses the border. Clearly, barrier coverage is an appropriate model of movement detection that is more efficient than full coverage since it requires less sensors for detecting intruders (this is the case, for example, when the width of the deployment region is three times the range of the sensors).

In the article of [3] the authors consider the problem of how individual sensors can determine barrier coverage *locally*. In particular, they prove that it is possible for individual sensors to locally determine the existence of barrier coverage, even when the region of deployment is arbitrarily curved. Although local barrier coverage does not always guarantee global barrier coverage, they show that for thin belt regions, local barrier coverage almost always provides global barrier coverage. They also consider the concept of *L-local barrier coverage* whereby if the bounding box that contains the entire trajectory of a crossing path has length at most L then this crossing path is guaranteed to be detected by at least one sensor.

1.1 Motivation, model and problem statement

Motivated from the works of [3] and [11], in this paper we go beyond by asking a different question not examined by any of these papers. More precisely, given that the mobile sensors have detected the existence of a crossing path (e.g., using any of the above algorithms) how do they reposition themselves *most efficiently* within a specified region so as to repair the existing *security hole* and thereby prevent intruders.

Further, we stipulate the existence of a geometric planar region (the critical region to be protected) delimited by a simple polygon, the *barrier*, and mobile sensors (or robots) that are lying in the interior of this polygon. The sensors can move autonomously in the plane. Each sensor has knowledge of the region to be barrier-covered, of its geographic location and can move from its starting position A to a new position A' on the perimeter of this polygon (see Figure 1). For each sensor A , we consider the distance $d(A, A')$ between the initial and

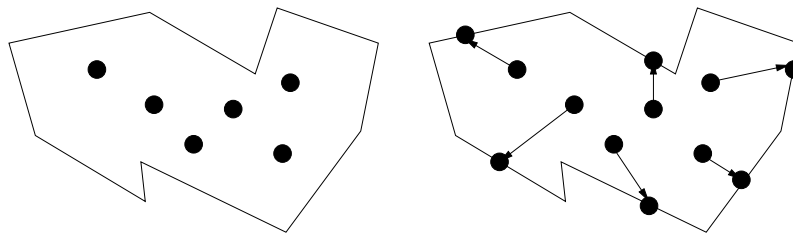


Figure 1: Sensors move from their initial position to positions on the perimeter of a simple polygon.

final positions of the sensors, respectively, and investigate how to move the sensors within this region so as to optimize either the *minimum sum* or the *minimum of the maximum* of the distances covered by the respective sensors. We call this, the *barrier coverage problem*. In the sequel we investigate the complexity of this problem for various types of regions and types of movement of the mobile sensors.

1.2 Related work

An interesting research article is by [1] which surveys the different kinds of holes that can form in geographically correlated problem areas of wireless sensor networks. The authors discuss relative strengths and short-comings of existing solutions for combating different kinds of holes such as coverage holes, routing holes, jamming holes, sink/black holes, wormholes, etc. [2] looks at critical density estimates for coverage and connectivity of thin strips (or annuli) of sensors. In addition, [5] and [6] design a distributed self deployment algorithm for coverage calculations in mobile sensor networks and consider various performance metrics, like coverage, uniformity, time and distance traveled till the algorithm converges. Related is also the research on art gallery theorems (see [14]) which is concerned with finding the minimal number of positions for guards or cameras so that every point in a gallery is observed by at least one guard or camera.

In addition to the research on barrier coverage already mentioned there is extensive literature on detection and tracking in sensor networks. [12] considers the problem of event tracking and sensor resource management in sensor networks and transforms the detection problem into finding and tracking the cell that contains the point in an arrangement of lines. [9] addresses the problem of tracking multiple targets using a network of communicating robots and stationary sensors by introducing a region-based approach for controlling robot deployment. [16] considers the problem of accurate mobile robot localization and mapping with uncertainty using visual landmarks. Finally, related to the problem of detecting a path through a region that can be traversed undetected by an intruder is the paper [15] which gives necessary and sufficient conditions for the existence of vertex disjoint simple curves homotopic to certain closed curves in a graph embedded on a compact surface.

1.3 Outline and results of the paper

Section 2 gives the formal model on a disk and defines the min-max (minimizing the maximum) and min-sum (minimizing the sum) problems for a set of sensors within a disk or a simple polygon. Section 3 looks at the simpler one dimensional case and derives simple optimal algorithms for the case the sensors either all lie on a line or on the perimeter of the disk. Section 4 and Section 5 are the core of the paper and provide algorithms for solving the min-sum and min-max problems, respectively. That is, in Section 4, an $O(n^{3.5} \log n)$ -time algorithm for the min-max problem on a disk and an $O(mn^{3.5} \log n)$ -time algorithm for the min-max problem on a simple polygon are proposed (m is the number of edges of the simple polygon). Our approximation algorithms for min-sum problems on a disk or a simple polygon are presented in Section 5, where we give a PTAS with approximation $1 + \epsilon$ having running time $O(\frac{1}{\epsilon} mn^5)$ for a given constant ϵ . We also present experimental results on the min-sum problem and Section 6 gives the conclusion.

2 Preliminaries and Formal Model

First we describe the formal model on a disk and provide the basic definitions and preliminary concepts.

2.1 Optimization on the unit disk

The simpler scenario we envision concerns n mobile sensors which are located in the interior of a unit-radius circular region (see Figure 2). We assume that the sensors are location aware

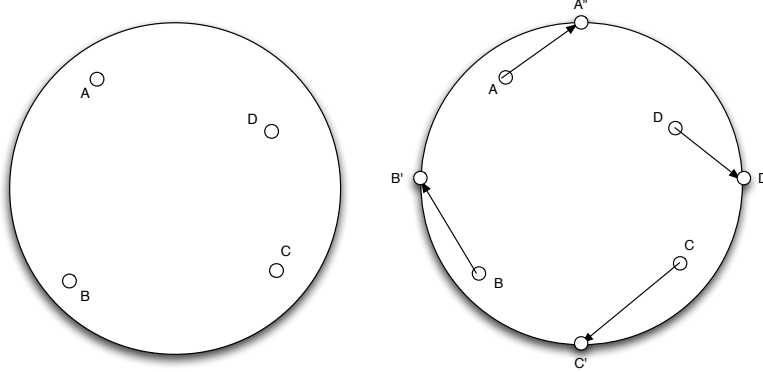


Figure 2: Four mobile sensors A, B, C, D located in the interior of a disk move to new positions A', B', C', D' on the perimeter of the disk so that $A'B'C'D'$ forms a regular 4-gon.

(i.e., they know their geometric coordinates) and also know the location of the center of the disk. We would like to move all the sensors from their initial positions to the perimeter of the disk so as to:

1. Form a regular n -gon, and
2. Minimize the total/maximum distance covered (see Figure 2).

The motivation for placing the sensors on the perimeter is because it provides the most efficient way to protect the disk from intruders. Observe that when all n sensors lie on the perimeter and form a regular n -gon, then each sensor need only cover a circular arc of size $2\pi/n$ so as to be able to monitor the entire perimeter. Using elementary trigonometry, it follows easily that the transmission range of each sensor must be equal to $r = \sin(\pi/n)$.

More formally, for n given sensors in positions A_1, A_2, \dots, A_n , respectively, that move to new positions A'_1, A'_2, \dots, A'_n on the perimeter forming a regular n -gon, the total distance covered is:

$$\sum_{i=1}^n d(A_i, A'_i). \quad (1)$$

It is clear that the sum is minimized when each sensor A_i moves to its destination A'_i in a straight line.

The reason for having the sensors form a regular n -gon is because this is evidently the optimal final arrangement that will enable the sensors to detect intruders (i.e., by being equidistant on the perimeter). Since the final positions A'_1, A'_2, \dots, A'_n of the sensors form a regular n -gon it is clear that the final configuration can be parametrized by using a single angle $0 \leq \theta \leq 2\pi$. However, a difficulty arises in view of the fact that we must also specify a permutation $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ of the sensors such that the i -th sensor moves from position $A_{\sigma(i)}$ to the new position A'_i .

Let the n sensors have coordinates (a_i, b_i) , for $i = 1, 2, \dots, n$. Let us parametrize the regular polygon with respect to the angle of rotation say θ . The n vertices of the regular n -gon that lie on the perimeter of the disk can be described by

$$(a_i(\theta), b_i(\theta)) = \left(\cos \left(\theta + \frac{(i-1)2\pi}{n} \right), \sin \left(\theta + \frac{(i-1)2\pi}{n} \right) \right), \text{ for } i = 1, 2, \dots, n, \quad (2)$$

respectively, where $(a_i(\theta), b_i(\theta))$ are the vertices of the regular n -gon when the angle of rotation is θ .

2.1.1 Minimizing the sum

We are interested in minimizing the sum

$$S_n(\theta) := \sum_{i=1}^n \sqrt{(a_i - a_i(\theta))^2 + (b_i - b_i(\theta))^2}, \quad (3)$$

as a function of the angle θ . The optimization problem is:

$$\min_{\theta} S_n(\theta). \quad (4)$$

This of course assumes that the i -th sensor is assigned to position $(\cos(\theta + (i-1)2\pi/n), \sin(\theta + (i-1)2\pi/n))$ on the perimeter. In general, we have to determine the minimum over all possible permutations $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ of the sensors. If for a given permutation $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ we define the following sum

$$S_n(\sigma, \theta) := \sum_{i=1}^n \sqrt{(a_{\sigma(i)} - a_i(\theta))^2 + (b_{\sigma(i)} - b_i(\theta))^2} \quad (5)$$

then the general optimization problem is:

$$\min_{\sigma, \theta} S_n(\sigma, \theta). \quad (6)$$

2.1.2 Minimizing the maximum

The previous problem was concerned with minimizing the sum of the distance traveled by the robots. In view of the fact that the robots are moving simultaneously it makes sense to consider the problem of minimizing the maximum of the distances traveled by each robot. More formally, given n sensors in positions A_1, A_2, \dots, A_n , respectively, that move to new positions A'_1, A'_2, \dots, A'_n forming a regular n -gon on the barrier, the maximum distance covered is:

$$\max_{1 \leq i \leq n} d(A_i, A'_i). \quad (7)$$

It is clear that the maximum is minimized when each sensor A_i moves in a straight line to its destination A'_i . The min-max problem involves minimizing the maximum

$$M_n(\theta) := \max_{1 \leq i \leq n} \sqrt{(a_i - a_i(\theta))^2 + (b_i - b_i(\theta))^2}, \quad (8)$$

as a function of the angle θ . The optimization problem is therefore to compute

$$\min_{\theta} M_n(\theta). \quad (9)$$

This of course assumes that the i -th sensor is assigned to position $(\cos(\theta + (i-1)2\pi/n), \sin(\theta + (i-1)2\pi/n))$ on the perimeter. In general, we have to determine the minimum over all possible permutations $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ of the sensors. If for a given permutation $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ we define the following maximum

$$M_n(\sigma, \theta) := \max_{1 \leq i \leq n} \sqrt{(a_{\sigma(i)} - a_i(\theta))^2 + (b_{\sigma(i)} - b_i(\theta))^2}, \quad (10)$$

then the general optimization problem is:

$$\min_{\sigma, \theta} M_n(\sigma, \theta). \quad (11)$$

2.2 Optimization on a simple polygon

We define the problem of minimizing the sum and minimizing the maximum on a simple polygon in a similar way, as follows.*

Let P be a simple polygon. (From now on, a polygon is always assumed to be simple.) We denote the boundary of P by ∂P . We assume that ∂P is oriented in the clockwise (also called positive) direction. For any two points $A, C \in \partial P$, $\hat{\pi}_P(A, C)$ denotes the set of all points $B \in \partial P$ such that when starting after A in positive direction along ∂P , B is reached before C . Let P_1, P_2, \dots, P_m denote the vertices of P ordered in the positive direction, and let the edges of P be e_1, e_2, \dots, e_m , where edge e_i has endpoints P_i and P_{i+1} , $1 \leq i \leq m$ (the indices are computed modulo m : so $P_0 = P_m$). Denote by $\ell(e_i)$ the length of edge e_i , $1 \leq i \leq m$, and by $\hat{d}_P(A, B)$ the length of $\hat{\pi}_P(A, B)$ for any two points A and B on ∂P (called *polygonal distance* between A and B). Let $L(P) = \sum_{i=1}^m \ell(e_i)$.

We are given n mobile sensors which are located in the interior, or on the boundary of P . Each sensor has the knowledge of its geometric coordinates and the simple polygon (i.e., the coordinates of all vertices P_i , $1 \leq i \leq m$ and the clockwise ordering of these vertices). The objective is to move all the sensors from their initial positions to ∂P such that:

1. The polygonal distance between any two consecutive sensors on the polygon is $L(P)/n$, and
2. The total/maximum distance covered is minimized.

More formally, we are given n sensors located at positions A_1, A_2, \dots, A_n , respectively. Let A'_i be the destination position of A_i on ∂P , $i = 1, 2, \dots, n$. Without loss of any generality, assume that for any i , $1 \leq i < n$, A'_{i+1} is the first position after A'_i in the positive direction along ∂P . The new positions A'_1, A'_2, \dots, A'_n should satisfy $\hat{d}_P(A'_i, A'_{i+1}) = L(P)/n$, $1 \leq i \leq n$ (taken modulo n), and we consider the following two objectives:

1. **Minimizing the sum:** $\min \sum_{i=1}^n d(A_i, A'_i)$, and
2. **Minimizing the maximum:** $\min \max_{i=1}^n d(A_i, A'_i)$,

where $d(\cdot, \cdot)$ could be the straight-line distance metric or any other distance metric that can be computed in $O(m^2)$ time.

*Although the approach proposed later (parametric search) will also work for arbitrary simple curves, we refrain from such a generalization so as to avoid unnecessary complications.

3 Mobile Sensors in One Dimension

In this section we look at the one dimensional problem and provide efficient algorithmic solutions. In particular, since optimization for the minimum maximum is similar (and simpler than the two dimensional analogue) we provide algorithms only for the minimum sum.

3.1 Sensors on the unit line segment

In this model we suppose that the sensors can move on a line segment. Further, instead of protecting a circular range the sensor can now protect an interval of a given size centered at the sensor. Consider the minimum sum optimization problem for the case of n sensors on a line. Without loss of generality assume the segment has length 1 and let the n sensors be at the initial locations $A_1 \leq A_2 \leq \dots \leq A_n$, respectively. Then the destination locations must be at the positions $\frac{2i-1}{2n}$, for $i = 1, 2, \dots, n$.

Theorem 1 *The optimal arrangement is obtained by moving point A_i to position $\frac{2i-1}{2n}$, for $i = 1, 2, \dots, n$, respectively.*

Proof. It is clear that the destinations must be equidistant and cover the endpoints 0 and 1. The optimal configuration is therefore: $1/2n, 3/2n, \dots, (2n-1)/2n$. For any point X let $d(X)$ be its destination. Recall that our goal is to determine the destinations of each point X so that the sum

$$\sum_X |X - d(X)| \tag{12}$$

is minimized. We call the route from X to $d(X)$ the directed path on the line from X to $d(X)$ which is traced by the sensor located at X . The length of this path is equal to $|X - d(X)|$. For any two points $X < Y$ we say that their paths *inversely overlap* if they are in opposite directions and they also overlap. As depicted in Figure 3, it is clear that for two points $X < Y$

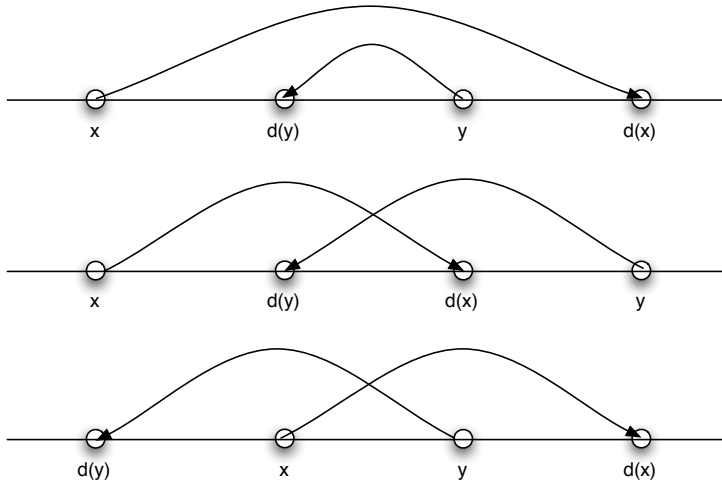


Figure 3: Three possible cases for overlapping paths between the two points $X < Y$.

there are three possible cases of overlapping paths. It is easy to show that in each case we

can improve on the optimization sum (Equation 12) by merely switching the destinations of X and Y , respectively. We call such a configuration between two points *inversion*.

Now consider the configuration resulting by moving point A_i to position $\frac{2i-1}{2n}$, for $i = 1, 2, \dots, n$, respectively. Observe that for this arrangement there exist no two pairs $X < Y$ of

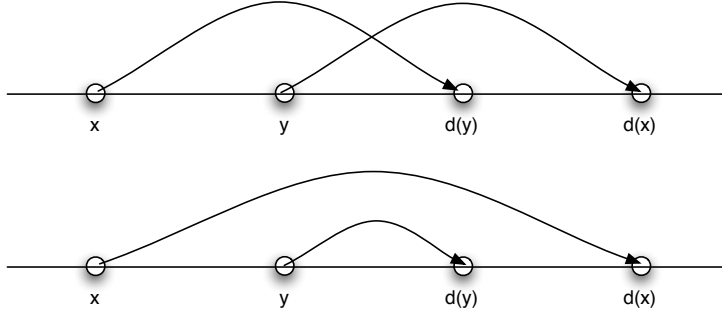


Figure 4: A non-overlapping transformation leaves the sum unchanged.

sensors which inversely overlap. Moreover, we claim that any two possible assignments that do not have pairs $X < Y$ that inversely overlap must have identical sum (Equation 12). Indeed, the sum of the top configuration depicted in Figure 4 is $(d(Y) - X) + (d(X) - Y)$ and of the bottom it is $(d(X) - X) + (d(Y) - Y)$. Since $(d(Y) - X) + (d(X) - Y) = (d(X) - X) + (d(Y) - Y)$ the result of the claim follows. This completes the proof of the theorem. ■

3.2 Sensors on the perimeter of the unit disk

In this model we suppose that the sensors can move on the perimeter of a disk. Further, instead of protecting a circular range the sensor can now protect an arc on the perimeter of a given size centered at the sensor. The same idea as for a line segment should work for the case of the unit disk when the sensors lie on its perimeter. The main difficulty here is that we no longer have a unique destination. Instead, we can parametrize all possible destinations of the n points by $\phi + \frac{2j\pi}{n}$, for $j = 0, 1, \dots, n - 1$, using a fixed angle $0 \leq \phi < \frac{2\pi}{n}$. First of all we prove the following lemma.

Lemma 2 *The minimal cost assignment of the destinations for the given points, must be among the n assignments*

$$(A_1, \dots, A_j, \dots, A_n) \rightarrow \left(A_i + (i - 1) \frac{2\pi}{n}, \dots, A_i + (i - j) \frac{2\pi}{n}, \dots, A_i + (i - n) \frac{2\pi}{n} \right),$$

for $i = 1, 2, \dots, n$. Note that each such assignment has a fixed point.

Proof. (Lemma 2) Consider the minimal cost assignment based on the angle A . Suppose that in the corresponding n -gon none of the initial points A_1, A_2, \dots, A_n is a vertex. Now rotate this n -gon and observe that either its clockwise or its counter clockwise rotation decreases the cost or remains the same. ■

Now we prove the main result. Suppose we are given n points A_1, A_2, \dots, A_n in this cyclical order along the perimeter of the disk. The main steps of the algorithm are the following.

1. For each point $A_i \in \{A_1, A_2, \dots, A_n\}$, map all points A_j to destinations $A_i + (i - j)\frac{2\pi}{n}$, for $j = 1, 2, \dots, n$ (this implies that A_i is mapped to itself).
2. Select the point $A_i \in \{A_1, A_2, \dots, A_n\}$ that optimizes the sum

$$\sum_{j=1}^n \left| A_j - A_i - (i - j)\frac{2\pi}{n} \right| \quad (13)$$

It is easy to show that in this mapping each point A is mapped to a destination $d(A)$ such that $|A - d(A)| \leq \pi$. Now just like the case of a line (imagine Figure 3 on a disk) we can show that the optimal assignment cannot have paths that inversely overlap. Moreover, according to Figure 4 (imagine it on a disk) the assignments will be optimal each for the given starting position $A_i + (i - 1)\frac{2\pi}{n}$. Therefore we have proved the following theorem.

Theorem 3 *There is an algorithm that computes an optimal cost arrangement of the sensors.*

3.2.1 When the sensors' movement is in the interior of the disk

In this model we suppose that the sensors and their destination positions are located on the perimeter of a disk and the sensors can move to their destination along straight lines. Here we have the same result as Lemma 2. Its proof is based on the fact that there is an optimal solution in which one sensor does not move at all.

Assume to the contrary that all sensors move in an optimal solution. For each $i, 1 \leq i \leq n$, let α_i be the angle of rotation of the sensor A_i to its new position A'_i : a positive angle denotes clockwise rotation while a negative angle denotes counter-clockwise rotation. For sensors which move to the opposite side of the disk we take $\alpha_i = -\pi$. From the law of cosines we derive that $d(A_i, A'_i) = \sqrt{2 - 2\cos\alpha_i}$. Let $f(\alpha) = \sqrt{2 - 2\cos\alpha}$. Then $f'(\alpha) = \sin\alpha(2 - 2\cos\alpha)^{-1/2}$ and $f''(\alpha) = -(\cos\alpha - 1)^2(2 - 2\cos\alpha)^{-3/2}$. It follows that f is concave over $[-\pi, \pi)$. Define $\alpha' := \max_{\alpha_i \leq 0} \alpha_i$ and $\alpha'' := \min_{\alpha_i > 0} \alpha_i$. Then $\alpha' < 0$ since all $\alpha_i \neq 0$ and $\alpha'' > 0$. Next define the function $g : [a, b] \rightarrow \mathbb{R}$:

$$g(\beta) := \sum_{i=1}^n f(\alpha_i + \beta).$$

The value of $g(\beta)$ represents the total cost of the solution which we get by taking the original solution $\{\alpha_i\}$ and changing the positions of the sensors on the perimeter by the angle β . Note that g is concave for all $\beta \in (a, b)$, since it is the sum of concave functions. We want to show that there is a $\beta \neq 0$ such that $g(\beta) < g(0)$. If $g'(0) \neq 0$ then there is clearly a β with $g(\beta) < g(0)$. This is a contradiction. If $g'(0) = 0$ then $\beta = 0$ is a local maximum since $g''(0) < 0$. So there is a β with $g(\beta) < g(0)$. This is a contradiction.

Since Lemma 2 is still valid in the model using the straight-line distance metric, the same idea as for the arc distance metric can be applied here. Therefore, we have the following theorem.

Theorem 4 *There is an algorithm that computes an optimal cost arrangement of the sensors.*

3.3 Sensors in the unit disk

In this model we suppose that some of the sensors are in the interior of the disk, and that they move to their destinations along straight lines. We use a similar argument to that in Section 3.2.1 to get an approximation for the min-sum assignment.

Let A_1, A_2, \dots, A_n be the sensors, and let α_i be the angle of rotation of A_i to its destination A'_i , B_i the nearest point to A_i on the disk and γ_i the distance $d(O, A_i)$, where O is the center of the disk, for $i = 1, 2, \dots, n$. If A_i is on the circle then $B_i = A_i$ and $\gamma_i = 1$. If A_i is at the center of the disk then we take B_i to be any point on the circle and $\gamma_i = 0$.

By using the law of cosines we get $d(A_i, A'_i) = (1 + \gamma_i^2 - 2\gamma_i \cos \alpha_i)^{1/2}$. For $f_\gamma(\alpha) = (1 + \gamma^2 - 2\gamma \cos \alpha)^{1/2}$, we get $f'_\gamma(\alpha) = \sin \alpha (1 + \gamma^2 - 2\gamma \cos \alpha)^{-1/2}$, and $f''_\gamma(\alpha) = -(\cos \alpha - \gamma)(\cos \alpha - 1/\gamma)(1 + \gamma^2 - 2\gamma \cos \alpha)^{-3/2}$. It follows that $f''_\gamma(\alpha)$ is concave when $\cos \alpha > \gamma$. As in Section 3.2.1 we define $g(\beta) := \sum_{i=1}^n f_{\gamma_i}(\alpha_i + \beta)$ for $\beta \in [a, b]$, to represent the total cost of the solution obtained by taking the original solution and changing the positions of the sensors on the perimeter by angle β . In this case we want to show that $\cos \alpha_i \geq \gamma_i$ for some $i = 1, 2, \dots, n$, which implies that A'_i lies in the arc about B_i of length $\pm \cos^{-1} \gamma_i$. The proof is by contradiction. Suppose that $\cos \alpha_i < \gamma_i$ for all $i = 1, 2, \dots, n$. Then $f''_{\gamma_i}(\alpha_i) < 0$ for all i , and therefore $g''(0) < 0$. Again, if $g'(0) \neq 0$ there is a β with $g(\beta) < g(0)$, and if $g'(0) = 0$ then $\beta = 0$ is a local maximum so there is a β with $g(\beta) < g(0)$. In both cases we get a contradiction. We therefore have:

Lemma 5 *The minimal cost assignment of the destinations for the given points, must be among the n assignments*

$$(A_1, \dots, A_j, \dots, A_n) \rightarrow \left(\hat{A}_i + (i-1) \frac{2\pi}{n}, \dots, \hat{A}_i + (i-j) \frac{2\pi}{n}, \dots, \hat{A}_i + (i-n) \frac{2\pi}{n} \right),$$

for $i = 1, 2, \dots, n$, where \hat{A}_i is some point on the arc about B_i of length $\pm \cos^{-1} \gamma_i$.

4 Min-max problem

In this section we study the problem of minimizing the maximum (min-max problem) on a unit disk and a simple polygon, and provide efficient algorithmic solutions.

4.1 On the disk

Let $\lambda_{m,C}^*$ be the optimal value of the min-max problem on a disk \mathcal{C} , i.e.,

$$\lambda_{m,C}^* = \min_{\sigma, \theta} M_n(\sigma, \theta).$$

It is easy to see that $\lambda_{m,C}^*$ is no more than the diameter of the disk \mathcal{C} , i.e., $\lambda_{m,C}^* \leq 2$. In this section we propose a parametric-searching approach [13] to compute $\lambda_{m,C}^*$.

A non-negative value λ is *feasible* for the min-max problem if all the sensors can move from their initial positions to the perimeter of the disk such that the new positions form a regular n -gon and the maximum covered distance is no more than λ , otherwise λ is *infeasible*. Clearly, the min-max problem is to compute the minimum feasible value, which is equal to $\lambda_{m,C}^*$.

The remaining part of this section is organized as follows. We first show that a feasibility test of a given value λ ($0 \leq \lambda \leq 2$) can be performed in time $O(n^{3.5})$. Then, a parametric-searching approach for the min-max problem is presented, which runs in $O(n^{3.5} \log n)$ time.

4.1.1 Algorithm to check the feasibility test of λ

For each $i, 1 \leq i \leq n$, we construct a circle of radius λ centered at position A_i , denoted by \mathcal{C}_i . If \mathcal{C}_i is contained in \mathcal{C} for some i , then λ is infeasible since sensor A_i cannot move to the perimeter of \mathcal{C} within distance λ . We therefore assume that for each $i, 1 \leq i \leq n$, either \mathcal{C}_i contains \mathcal{C} or \mathcal{C}_i intersects \mathcal{C} . Refer to Figure 5. For each $i, 1 \leq i \leq n$, we denote by Q_i the

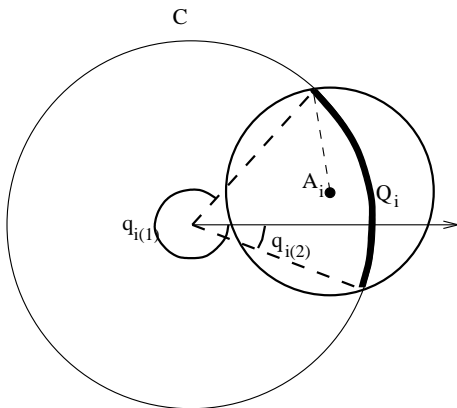


Figure 5: The arc Q_i , and the angles $q_{i(1)}, q_{i(2)}$ of its endpoints.

arc of \mathcal{C} that lies in \mathcal{C}_i . Let $q_{i(1)}, q_{i(2)}$ be the angles of the two endpoints of arc Q_i in clockwise order, $i = 1, 2, \dots, n$. We let $q_{i(1)} = 0$ and $q_{i(2)} = 2\pi$ if \mathcal{C}_i contains \mathcal{C} .

The following property is important to our algorithm for the feasibility test of λ .

Lemma 6 *If $\lambda > 0$ is feasible for the min-sum problem on the disk \mathcal{C} then there exists an optimal arrangement such that one of its vertices A_i' ($1 \leq i \leq n$) is an endpoint of arc Q_i .*

Proof. Each of the destination points A_i' lies on the corresponding arc Q_i of the disk, cut by the λ -radius circle centered at A_i . Move the configuration of the destination points clockwise (or counter-clockwise) until one of the destination points reaches an endpoint of one of the arcs Q_i . ■

The following algorithm is used to check the feasibility of λ .

Algorithm Check.

Step 1 Sort the angles of the endpoints of the arcs $Q_i, 1 \leq i \leq n$. Let $q'_1, q'_2, \dots, q'_{2n}$ be the angles in increasing order. These angles partition the interval $[0, 2\pi]$ into at most $2n$ pairwise disjoint intervals, denoted by I_j ($1 \leq j \leq 2n$).

Step 2 For each interval $I_j, 1 \leq j \leq 2n$, determine the set of sensors, denoted by S_j , that lie within distance λ of the corresponding arc on \mathcal{C} .

Step 3 For each $1 \leq j \leq 2n$, do the following:

- a Compute the vertices of the regular n -gon $B_1^j, B_2^j, \dots, B_n^j$, defined by the angle $B_1^j := q'_j$. It is easy to see that the vertices of this n -gon are: $q'_j, (q'_j + \frac{2\pi}{n}) \bmod 2\pi, \dots, (q'_j + (n-1)\frac{2\pi}{n}) \bmod 2\pi$.
- b Construct a bipartite graph between the sensors A_1, A_2, \dots, A_n and the vertices $B_1^j, B_2^j, \dots, B_n^j$: sensor A_i is linked to vertex B_k if $d(A_i, B_k) \leq \lambda$ ($1 \leq i, k \leq n$).
- c Check if there exists a perfect matching. If it is so, terminate the process and return “Feasible”.

Return “Infeasible” .

It is easy to see that the sorting in the first step can be done in $O(n \log n)$ time and the computation of sets of sensors S_j ($1 \leq j \leq 2n$) can be done in $O(n^2)$ time. In the third step, the process might try all $O(n)$ regular n -gons. For each regular n -gon, it takes $O(n^{2.5})$ time (see [7]). Therefore, we have the following lemma.

Lemma 7 *One can determine whether a given positive value λ is feasible in the min-max problem in $O(n^{3.5})$ time.*

4.1.2 A parametric-searching approach

Our approach for the solution to the min-max problem is to run Algorithm *Check* parametrically, which has a single parameter λ , without specifying the value of $\lambda_{m,C}^*$ a priori. Note that for a fixed value of the parameter, the algorithm is executed in $O(n^{3.5})$ steps. Imagine that we start the algorithm without specifying a value of the parameter λ . The parameter is restricted to some interval which is known to contain the optimal value $\lambda_{m,C}^*$. (Initially, we may start with the interval $[0, 2]$.) As we go along, at each step of the algorithm we update and shrink the size of the interval, ensuring that it includes the optimal value $\lambda_{m,C}^*$. The final interval contains $\lambda_{m,C}^*$ and any value in it is feasible. Therefore, the minimum value of the final interval is the optimal value $\lambda_{m,C}^*$.

The whole approach for the min-max problem is described as follows. The sorting step (the first step) of Algorithm *Check* with unknown $\lambda_{m,C}^*$ can be performed by solving $O(\log^2 n)$ feasibility tests if a parallel sorting network that runs $O(\log n)$ steps with n processors is used [13]. Actually, it can be reduced to $O(\log n)$ feasibility tests if Cole’s result is applied [4]. The second step does not need to solve feasibility tests since the order of the endpoints of the arcs $Q_i, 1 \leq i \leq n$, provides enough information to compute sets $S_j, 1 \leq j \leq 2n + 1$.

We now show how to run the third step with unknown $\lambda_{m,C}^*$. We can see that no feasibility test is needed for the Steps 3(b) and 3(c). However, to find the vertices of the regular n -gons $B_1^j, B_2^j, \dots, B_n^j, j = 1, \dots, 2n$, we solve some feasibility tests since these vertices and the endpoints of intervals are low-degree polynomials of unknown value $\lambda_{m,C}^*$.

To speed up Step 3(c), we can compute the angles in parallel. For each angle, we locate the interval containing it using a binary-search. Clearly, this can be done in $O(\log n)$ steps, and each step is a comparison between two low-degree polynomials that can be achieved by solving a constant number of feasibility tests.

Therefore, the computation in Step 3(a) for all possible $j = 1, 2, \dots, n$, can be done in $O(\log n)$ parallel steps by using $O(n^2)$ processors (there are $2n^2$ queries in total), where each step is a comparison between two low-degree polynomials with unknown $\lambda_{m,C}^*$. By applying the idea of Cole [4], the computation in Step 3(a) for all possible $j = 1, 2, \dots, n$ can be done by solving $O(\log n)$ feasibility tests. Summing up, we have the following theorem.

Theorem 8 *The min-max problem on the disk can be solved in $O(n^{3.5} \log n)$ time.*

Note that our algorithm can be easily extended to the model in which all sensors are arbitrarily located on the plane (not restricted to the interior of the disk \mathcal{C}).

4.2 On a simple polygon

In this model we discuss the min-max problem on a simple polygon as defined in Section 2.2. The parametric-searching approach for a disk (described in Section 4.1) should work for the case of a polygon where the destination positions of all sensors lie on the perimeter of the polygon. The main difficulty here is that to check the feasibility of a positive value λ , there might be $O(m)$ isolated polygonal chains of ∂P within the disk \mathcal{C}_i (of radius λ centered at position A_i) for each sensor A_i . In other words, for a given positive value of λ each sensor will contribute $O(m)$ candidate sets of n destination positions on P instead of at most two candidate sets on a circle. Hence, one can determine whether a given positive value λ is feasible in the min-max problem on a simple polygon by solving $O(mn)$ matching problems of size n . Therefore, the feasibility test of the min-max problem on a simple polygon can be solved in $O(mn^{3.5})$ time.

Theorem 9 *The min-max problem on a simple polygon can be solved in $O(mn^{3.5} \log n)$ time where m is the size of the simple polygon.*

5 Approximation algorithms for the min-sum problem

In this section we discuss the problem of minimizing the sum (min-sum problem) on a disk and a simple polygon, and provide approximation solutions.

5.1 On the disk

Let $\lambda_{s,C}^*$ be the optimal value of the min-sum problem on the disk, i.e., $\lambda_{s,C}^* = \min_{\sigma,\theta} S_n(\sigma, \theta)$.

We present two approximation algorithms for the min-sum problem. One algorithm (labeled as the *first approach*) has an approximation ratio $\pi + 1$ (Section 5.2). The other (labeled as the *second approach*) uses the first approach as a subroutine to obtain lower and upper bounds for $\lambda_{s,C}^*$ and has an approximation ratio $1 + \epsilon$, where ϵ is an arbitrary constant (Section 5.3).

More notations are introduced as follows. Let $\hat{d}_C(X, Y)$ denote the arc distance between two points X and Y on the boundary of the disc \mathcal{C} and let $\hat{\pi}_C(X, Y)$ denote the arc of length $\hat{d}_C(X, Y)$ between X and Y . For a point X on \mathcal{C} , we denote by $\hat{Q}_X(r)$ the arc consisting of all points Y on \mathcal{C} such that $\hat{d}_C(X, Y) \leq r$.

For each $i = 1, 2, \dots, n$, let ω_i be the smallest distance between A_i and the disk \mathcal{C} , and denote by B_i the point on \mathcal{C} for which the distance $d(A_i, B_i) = \omega_i$. We note that for each $i = 1, 2, \dots, n$, B_i is unique if A_i is not located at the center of \mathcal{C} . In the case when A_i is located at the center of \mathcal{C} , an arbitrary point on \mathcal{C} is selected to be B_i . Let $\Omega = \sum_{i=1}^n \omega_i$. Obviously, we have the following lemma.

Lemma 10 $\Omega \leq \lambda_{s,C}^*$.

5.2 The first approach

The first approach, called Algorithm 1, consists of three steps.

Algorithm 1

Step 1 For each sensor A_i , $i = 1, 2, \dots, n$, compute B_i .

Step 2 Compute the optimal min-sum assignment for the set of n points B_1, B_2, \dots, B_n , by using the algorithm for sensors on the perimeter of the disk described in Section 3.2. Let B'_i be the destination of B_i , $1 \leq i \leq n$.

Step 3 Move A_i to B'_i , $1 \leq i \leq n$, and compute $S_n^1 = \sum_{i=1}^n d(A_i, B'_i)$.

In Section 3.2 we showed that Step 2 of Algorithm 1 can be implemented in $O(n^2)$ time. Thus the above algorithm can be solved in $O(n^2)$ time.

5.2.1 Approximation bound of Algorithm 1

In this section, we show that S_n^1 computed by the first approach is bounded by $(\pi + 1) \times \lambda_{s,C}^*$.

Suppose that A'_i is the destination of sensor A_i , $i = 1, 2, \dots, n$, in an optimal solution. Clearly, A'_1, A'_2, \dots, A'_n lie on \mathcal{C} and form a regular n -gon. Also, $\sum_{i=1}^n \hat{d}_C(B_i, B'_i) \leq \sum_{i=1}^n \hat{d}_C(B_i, A'_i)$ since $\{B'_1, B'_2, \dots, B'_n\}$ is an optimal solution for the one dimensional min-sum problem with the input $\{B_1, B_2, \dots, B_n\}$. The following Lemma is easy to show.

Lemma 11 For any two points x, y on \mathcal{C} , $\hat{d}_C(x, y) \leq \frac{\pi}{2} \times d(x, y)$.

Therefore,

$$\begin{aligned}
S_n^1 &= \sum_{i=1}^n d(A_i, B'_i) \\
&\leq \sum_{i=1}^n [d(A_i, B_i) + \hat{d}_C(B_i, B'_i)] \\
&\leq \sum_{i=1}^n d(A_i, B_i) + \sum_{i=1}^n \hat{d}_C(B_i, A'_i) \\
&\leq \sum_{i=1}^n d(A_i, B_i) + \frac{\pi}{2} \times \sum_{i=1}^n d(B_i, A'_i) \text{ (Lemma 11)} \\
&\leq \sum_{i=1}^n d(A_i, B_i) + \frac{\pi}{2} \times \sum_{i=1}^n [d(B_i, A_i) + d(A_i, A'_i)] \\
&\leq (\pi + 1) \times \lambda_{s,C}^* \text{ (Lemma 10)}.
\end{aligned}$$

From the above discussion it follows that

Theorem 12 Algorithm 1 can be implemented in $O(n^2)$ time and its approximation ratio is no more than $\pi + 1$.

5.3 The second approach

The following Lemma is crucial for the second approach (Algorithm 2).

Lemma 13 *In an optimal solution, there exists at least one sensor A_i ($1 \leq i \leq n$), such that its destination A'_i on \mathcal{C} is on the arc $\hat{Q}_{B_i}(\frac{\pi}{2} \times \frac{S_n^1}{n})$.*

Proof. It is clear that $S_n^1 \geq \lambda_{s,C}^*$. Let A'_i be the destination of sensor A_i in an optimal solution, $i = 1, 2, \dots, n$. Then there is at least one sensor, say A_k ($1 \leq k \leq n$), such that the distance $d(A_k, A'_k)$ is no more than $\frac{S_n^1}{n}$.

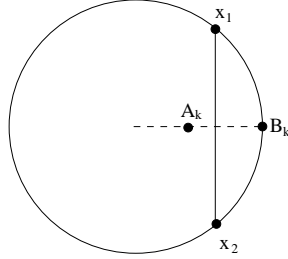


Figure 6: Lemma 13: $\hat{d}_C(x_1, B_k) = \hat{d}_C(x_2, B_k) = \frac{\pi}{2} \times \frac{S_n^1}{n} \Rightarrow d(x_1, x_2) \geq \frac{2S_n^1}{n}$ (Lemma 11).

By Lemma 11, all points on \mathcal{C} at distance no more than $\frac{S_n^1}{n}$ from A_k lie on the arc $\hat{Q}_{B_k}(\frac{\pi}{2} \times \frac{S_n^1}{n})$ (recall that B_k is the point on \mathcal{C} closest to A_k), which completes the proof of Lemma 13. \blacksquare

We now describe the second algorithm.

Algorithm 2

Step 1 Use Algorithm 1 to compute S_n^1 , as defined above.

Step 2 For each $i = 1, 2, \dots, n$, find the arc $\hat{Q}_{B_i}(\frac{\pi}{2} \times \frac{S_n^1}{n})$ and compute the set of points that partition the arc into $\lceil \frac{1}{\epsilon'} \rceil$ pieces of equal length where $\epsilon' = \frac{2\epsilon}{\pi(\pi+1)}$. Clearly, there are $n \times (\lceil \frac{1}{\epsilon'} \rceil + 1)$ such points in total.

Step 3 For each point X , construct a regular n -gon P_X with one vertex located at X , and find the optimal assignment of the n sensors A_1, A_2, \dots, A_n , to the vertices of P_X by solving a weighted bipartite matching problem. (The Hungarian method to solve the weighted matching problem in a complete bipartite graph of size n takes $O(n^3)$ time (see [10])).

Step 4 Among all $n \times (\lceil \frac{1}{\epsilon'} \rceil + 1)$ regular n -gons thus constructed, find the one with the minimum cost (denoted by S_n^2) and output the optimal arrangement of the n sensors to the vertices of the n -gon.

Remark. Step 1 and Step 2 of Algorithm 2 use Algorithm 1 as a subroutine to locate a destination point that lies within the arc $\hat{Q}_{B_i}(\frac{\pi}{2} \times \frac{S_n^1}{n})$, which is then partitioned into pieces. An alternative approach could use Lemma 5.

5.3.1 Analysis of the second approach

First, it is evident that the running time of the second approach is determined by the time needed to solve $n \times (\lceil \frac{1}{\epsilon} \rceil + 1) \in O(\frac{n}{\epsilon})$ bipartite matching problems.

According to Lemma 13, there exists an optimal solution in which one of the vertices of the corresponding regular n -gon is located at a point on the arc $\hat{Q}_{B_k}(\frac{\pi}{2} \times \frac{S_n^1}{n})$ for some k , $1 \leq k \leq n$. In Step 2, the arc $\hat{Q}_{B_k}(\frac{\pi}{2} \times \frac{S_n^1}{n})$ is partitioned into $\lceil \frac{1}{\epsilon} \rceil$ pieces, and therefore, the length of each piece is no more than $\frac{\pi S_n^1 \epsilon'}{n}$ (note that the length of $\hat{Q}_{B_k}(\frac{\pi}{2} \times \frac{S_n^1}{n})$ is $\frac{\pi S_n^1}{n}$). Since all possible values of k are considered, the difference between S_n^2 (computed by the second approach) and $\lambda_{s,C}^*$ (the optimal cost) is no more than

$$n \times \frac{1}{2} \times \frac{\pi S_n^1 \epsilon'}{n} = \frac{\pi S_n^1 \epsilon'}{2} = \frac{S_n^1 \epsilon}{\pi + 1} \leq \epsilon \lambda_{s,C}^* \text{ (Theorem 12).}$$

Therefore, we have the following theorem.

Theorem 14 *The approximation ratio of Algorithm 2 is no more than $1 + \epsilon$ for a given constant ϵ , and the running time is $O(\frac{1}{\epsilon} n^4)$.*

5.4 On a simple polygon

Let $\lambda_{s,P}^*$ be the optimal value of the min-sum problem on a polygon P . In this subsection we present an approximation algorithm for the min-sum problem on P , which has an approximation ratio $1 + \epsilon$ (ϵ is an arbitrary constant).

Our algorithm for a simple polygon is very similar to the second approach for the disk. In that approach, we used Algorithm 1 as a subroutine to obtain lower and upper bounds for $\lambda_{s,C}^*$. However, our approximation algorithm for a simple polygon will use the solution for the min-max problem on the polygon to obtain lower and upper bounds for $\lambda_{s,P}^*$. Let $\lambda_{m,P}^*$ be the optimal value of the min-max problem on P . It is easy to see that $\lambda_{m,P}^* \leq \lambda_{s,P}^* \leq n \times \lambda_{m,P}^*$.

Our algorithm for a simple polygon P is described below.

Step 1 Use the approach for the min-max problem on P to compute $\lambda_{m,P}^*$ as described above.

Step 2 For each i, j , where $1 \leq i \leq n$ and $0 \leq j < n$, find the sub-edge $e'_{i,j}$ of edge e_j that is within the circle of radius $\lambda_{m,P}^*$ centered at position A_i , and compute a set of points that partitions the sub-edge into $\lceil \frac{n}{\epsilon} \rceil$ pieces of equal length. Clearly, there are $mn \times (\lceil \frac{n}{\epsilon} \rceil + 1) \in O(\frac{mn^2}{\epsilon})$ such points in total.

Step 3 For each point X , construct a set of n positions on P such that one of them is located at X and the polygonal distance between any two consecutive positions is $L(P)/n$, and find the optimal assignment of the n sensors A_1, A_2, \dots, A_n to the set of n positions by using the algorithm [10].

Step 4 Among all $O(\frac{mn^2}{\epsilon})$ candidate sets of n positions thus constructed, find the one with the minimum cost.

It is evident that the running time of the above approach is determined by the time needed for solving $O(\frac{mn^2}{\epsilon})$ weighted bipartite matching problems.

The reason why the approximation ratio of the above approach is bounded by $1 + \epsilon$, is as follows. Since $\lambda_{s,P}^* \leq n \times \lambda_{m,P}^*$, there is at least one sensor whose moving distance to its destination is no more than $\lambda_{m,P}^*$ in an optimal solution. Let A_i be one such sensor: its destination position lies on edge e_j in the optimal solution. In Step 2, the sub-edge $e'_{i,j}$ is partitioned into $\lceil \frac{n}{\epsilon} \rceil$ pieces, and therefore, the length of each piece is no more than $\frac{2\epsilon\lambda_{m,P}^*}{n}$. Since all possible values of i and j are considered, the difference between the value computed by the above approach and $\lambda_{s,P}^*$ (the optimal cost) is no more than

$$n \times \frac{1}{2} \times \frac{2\lambda_{m,P}^*}{n} = \epsilon\lambda_{m,P}^* \leq \epsilon\lambda_{s,P}^*.$$

Therefore, we have the following theorem.

Theorem 15 *The approximation ratio of the approach for a simple polygon is no more than $1 + \epsilon$ for a given constant ϵ , and the running time is $O(\frac{1}{\epsilon}mn^5)$.*

5.5 Experimental Results on the complexity of the min-sum problem

It is not known whether the min-sum problem can be solved optimally. Two related problems which could help clarify the issue are the following.

1. Given a counter-clockwise ordering of n sensors on the perimeter of the disk \mathcal{C} , solve the min-sum problem.
2. Find the number of different counter-clockwise orderings of n sensors on the perimeter of \mathcal{C} when we sweep a regular n -gon along the perimeter of \mathcal{C} .

Table 1 shows our experimental results to resolve the second problem described above. The

Table 1: Experimental result for the number of different orderings (over 20 test sets)

# of sensors (n)	# of regular n -gons (t)	average # of orderings
10	1,000	5.40
20	2,000	8.40
30	3,000	11.75
40	4,000	15.45
50	5,000	18.10
60	6,000	19.80
70	7,000	24.50
80	8,000	26.60
90	9,000	30.85
100	10,000	35.55

first column in the table represents the number n of sensors contained in \mathcal{C} (the n sensors are randomly generated), the second column represents the number t of regular n -gons being tested (i.e., the arc distance between two consecutive regular n -gons is $\frac{2\pi}{t*n}$), and the third column represents the average number of different counter-clockwise orderings of the n sensors

to the vertices of the t n -gons (for each value of n , 20 different sets of n sensors are generated). From this table, we can see that there are no more than n different counter-clockwise orderings in the experiment.

6 Conclusion and Open Problems

In this paper we gave an algorithm for solving the min-max problem and a PTAS (Polynomial Time Approximation Scheme) for the min-sum problem in both one and two dimensions. Although it is unknown whether the min-sum problem is NP -hard, we conjecture that it can be solved in polynomial time. In addition, several other variants of the problem on simple polygons and regions are of interest for further investigation, including k -barrier coverage, regions with holes, and various types of sensor placements and motions. Thus, in Subsection 2.2, in order to minimize the number of sensors used when scanning the perimeter one should take into account sections already scanned. For example, this is the case if the polygon is a narrow rectangle of height less than the range of a sensor; this in itself is an interesting optimization problem which is worth of further investigation. Also of interest is to refine the sensor motion model, the network model, and the communication model in order to enable effective intrusion detection and barrier coverage. For example, the communication model becomes crucial when assuming the sensors either do not have knowledge of the region or do not know their coordinates.

References

- [1] N. Ahmed, S.S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):4–18, 2005.
- [2] P. Balister, B. Bollobas, A. Sarkar, and S. Kumar. Reliable density estimates for coverage and connectivity in thin strips of finite length. *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 75–86, 2007.
- [3] A. Chen, S. Kumar, and T.H. Lai. Designing localized algorithms for barrier coverage. *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 63–74, 2007.
- [4] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *Journal of the ACM (JACM)*, 34(1):200–208, 1987.
- [5] N. Heo and PK Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3, 2003.
- [6] N. Heo and PK Varshney. Energy-efficient deployment of Intelligent Mobile sensor networks. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 35(1):78–92, 2005.
- [7] J.E. Hopcroft and R.M. Karp. An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

- [8] S. S. Hu. 'Virtual Fence' along border to be delayed. *Washington Post*, Thursday, February 28, 2008.
- [9] B. Jung and G.S. Sukhatme. Tracking Targets Using Multiple Robots: The Effect of Environment Occlusion. *Autonomous Robots*, 13(3):191–205, 2002.
- [10] H. W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
- [11] S. Kumar, T.H. Lai, and A. Arora. Barrier coverage with wireless sensors. *Wireless Networks*, 13(6):817–834, 2007.
- [12] J. Liu, P. Cheung, F. Zhao, and L. Guibas. A dual-space approach to tracking and sensor management in wireless sensor networks. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 131–139, 2002.
- [13] N. Megiddo. Applying Parallel Computation Algorithms in the Design of Serial Algorithms. *Journal of the ACM (JACM)*, 30(4):852–865, 1983.
- [14] J. O'Rourke. *Art gallery theorems and algorithms*. Oxford University Press, Inc. New York, NY, USA, 1987.
- [15] A. Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *Journal of Combinatorial Theory Series B*, 51(1):127–159, 1991.
- [16] S. Se, D. Lowe, and J. Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *The International Journal of Robotics Research*, 21(8):735, 2002.