

2-Sensor Problem

Michael Segal *

Communication Systems Engineering Department, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, Tel: 972-8-6477234, Fax: 972-8-6472883

* Author to whom correspondence should be addressed. Email: segal@cse.bgu.ac.il

Received: 20 July 2004 / Accepted: 28. November 2004 / Published: 30 November 2004

Abstract: Ad-hoc networks of sensor nodes are in general semi-permanently deployed. However, the topology of such networks continuously changes over time, due to the power of some sensors wearing out to new sensors being inserted into the network, or even due to designers moving sensors around during a network re-design phase (for example, in response to a change in the requirements of the network). In this paper, we address the problem of covering a given path by a limited number of sensors — in our case to two, and show its relation to the well-studied matrix multiplication problem.

Keywords: Sensors, matrix multiplication, covering.

Introduction

Ad-hoc sensor networks are emerging as a new sensing paradigm and have thus, received massive research interest recently. Usually sensor nodes are semi-permanently deployed, since the sensors themselves barely have any moving capacity. However, the topology of such networks continuously changes over time due to a variety of reasons: For example, a sensor node may wear out due to its very limited battery power, a new sensor node may be inserted into the network, or the layout of a sensor network may need to be changed in order to improve the quality of the network coverage in response to a change in the network requirements, which is accomplished by changing the placement of current (or inserting, deleting) sensors in network.

In this paper, we address a problem of covering a path in the network using a limited number of sensors. In a sensor network, each sensor bears the ability to detect objects around it. The coverage of a sensor is limited by its energy level. Assume the sensor's detecting ability is omnidirectional, then we can model the coverage of a sensor as a disk (under 2-norm on the Euclidean plane¹) centered at the sensor. The radii of such disks are determined by the energy level of the sensors. The coverage area (or simply coverage) of the sensor network is the union of all such disks.

A sensor network is often used to detect intruders. An intruder may start at a point S on the plane, may follow an arbitrary trajectory (path), and may stop at some other point T on the plane. In some applications, a sensor network may need to keep track of the intruder at all times, as it follows its trajectory; in some other applications, the network's function may be simply to detect the presence of an intruder, in which case the network only needs to cover some part of the trajectory. Thus, given two points S and T , two relevant types of trajectories are proposed [1]: The *maximum breach path* and the *maximum support path* (In [1], these paths are called *maximal* breach path and *maximal* support path, respectively).

The maximum breach path measures the vulnerability of a sensor network by, as the name suggests, completely avoiding the coverage area of the sensor network: It is a trajectory between the start point S and the stop point T that stays "as far away" from the sensors as possible. On the other hand, the maximum support path measures the efficiency of the network coverage: This path is a trajectory between S and T , which stays "as close to the sensors" as possible. The *distance of a point P to the sensor network* is defined as the smallest Euclidean distance from P to one of the sensor nodes. A *maximum breach path* from S to T is a path from S to T such that the minimum distance from a point P in the path to the sensor network is maximized: This distance is called the *worst-case coverage distance* of the network. Similarly, a *maximum support path* from S to T is a path such that the maximum distance of a point P in the path to the sensor network is minimized: This distance is called the *best-case coverage distance* of the network.

We consider this problem of covering an entire trajectory using two sensors only, where each of the sensors should be located on opposite sides of the trajectory. This problem is an extension of the well-studied two-center problem, which we describe in the next section "Two-sensor-problem". We

¹A *disk* of radius r centered at (x, y) under 2-norm in R^2 is the set of points (p, q) such that $\sqrt{(p-x)^2 + (q-y)^2} \leq r$.

present an $O(n^{2.688})$ time algorithm for this problem, improving on the exhaustive search bound of $O(n^3)$. The solution proposed is of interest on its own, since it establishes an interesting relation to the matrix multiplication problem. In the section "Two-sensor-problem" we define our problem formally and give an efficient solution for it.

Two-sensor problem

We consider the following problem: There are two groups of sensors A and B of total size n divided by a polygonal xy -monotone path P containing n edges, such that each group is located on opposite sides of P . A path $P = \langle v_1, v_2, \dots, v_{n+1} \rangle$ is called xy -monotone if $x(v_1) \leq x(v_2) \leq \dots \leq x(v_{n+1})$ and $y(v_1) \leq y(v_2) \leq \dots \leq y(v_{n+1})$. Each sensor s_i has its covering area radius R_i , so it can detect any object at a distance at most R_i from it. We are interested in determining whether there exists a sensor from group A and a sensor from group B that cover the entire path P . The path P is playing role of a possible way for intruder penetration inside the sensors area and each group of sensors wants to catch the intruder while it is important to keep a visual guard of him all the time. We will refer to our problem as the *two-sensor problem*. Some similar problem has been considered by Huang and Tseng [2], who tried to cover a target area by k sensors from a set consisting of n sensors.

Clearly, our task can be performed in $O(n^3)$ time by taking all possible pairs of sensors from A and B and checking for each edge of P whether it is covered or not. We present an $O(n^{2.688})$ runtime algorithm based on matrix multiplication and Matoušek algorithm [3] for computing dominance in n -dimensional space.

The two-sensor problem naturally extends a list of optimization problems related to the famous two-center problem. In the two-center problem, we want to cover a set S of n points on the plane by two disks of smallest possible common radius. Agarwal and Sharir give an $O(n^2 \log n)$ -time algorithm in [4] for determining whether S can be covered by two disks of radius r . Plugging this algorithm into the parametric search machinery, one obtains an $O(n^2 \log^3 n)$ -time algorithm for the two-center problem. The runtime of the decision algorithm of Agarwal and Sharir was improved by Hershberger [5] to $O(n^2)$. Jaromczyk and Kowaluk [6] use the algorithm by Hershberger [5] to obtain an $O(n^2 \log n)$ running time algorithm for the two-center problem. A major progress on this problem was recently made by Sharir [7], who presented an $O(n \log^9 n)$ -time algorithm, by combining the parametric search technique with several additional techniques, including a variant of the matrix search algorithm of Frederickson and Johnson [8]. Eppstein [9] has simplified Sharir's algorithm, using randomization and different data structures, obtaining an improved solution with expected runtime of $O(n \log^2 n)$.

Recently, Agarwal et al. [10] have developed an $O(n^{\frac{4}{3}} \log^5 n)$ -time algorithm for the discrete two-center problem, where the disks should be centered at the input points. Shin et al. [11] considered the problem of covering convex polygons using two disks of smallest possible radius by giving two algorithms: The first algorithm runs in time $O(n \log^3 n)$ and assumes that the disks are constrained to be centered at the input points; the second algorithm runs in $O(n^2 \log^3 n)$ time and assumes arbitrary disks location. Huang et al. [12] considered a variation of the two-center problem, where the distance between two solution disks should be below some given constant, providing a $O(n^2 \log^2 n)$ runtime algorithm for this problem.

The two-sensor problem considered in this paper is harder than all of the related variations of the two-center problem in the literature. The hardness comes from two sources: First, we need to cover the entire path and not just a set of discrete points; second, in the solution obtained, an edge may be partially covered by each of the sensors, implying that the sets of edges covered by each sensor may overlap significantly and, therefore, a simple binary search can no longer be used. We follow the notation used by Matoušek [3]. Let X be an $n \times n$ matrix of real numbers; we will use the notation $x_{i,*}$ for the i -th row of X ; we say that $x_{j,*}$ dominates $x_{i,*}$, $1 \leq i, j \leq n, i \neq j$, if $x_{i,k} \leq x_{j,k}$, for $k = 1, \dots, n$. Matoušek [3] presents an algorithm for computing an $n \times n$ matrix C , whose element $c_{i,j}$ is:

$$c_{i,j} = |\{k : x_{i,k} \leq x_{j,k}\}| \quad (1)$$

In other words, $c_{i,j} = n$ if and only if $x_{i,*}$ is dominated by $x_{j,*}$. The runtime of Matoušek's algorithm is $O(n^{\frac{3}{2}}M(n)^{\frac{1}{2}})$, where $M(n)$ denotes the time needed to multiply two $n \times n$ matrices. With the best known results on matrix multiplication we have $M(n) = O(n^{2.376})$ [13], and the running time of Matoušek's algorithm is $O(n^{2.688})$.

In what follows, we show how to apply Matoušek's algorithm in order to get a solution to our problem. We start with the following observation, which is crucial for the solution of the 2-sensor problem.

Observation 1. Given an $n \times n$ matrix of real numbers X and $h \in R^n$, we can determine whether there exists two rows in X , $x_{i,*}$ and $x_{j,*}$ such that $x_{i,k} + x_{j,k} \geq h_k$, $k = 1, \dots, n$ using Matoušek's algorithm.

1. We define a new, $2n \times 2n$ matrix X' as follows. The first n rows of X' are the rows of X followed by zeros in places $n+1, \dots, 2n$. The next n rows of X' are defined as follows. The element $x_{i,k}$ is defined as $h_k - x_{i-n,k}$, if $k = 1, \dots, n$, $i = n+1, \dots, 2n$ and $x_{i,k} = 0$, if $k = n+1, \dots, 2n$, $i = n+1, \dots, 2n$. Now, we apply Matoušek's algorithm that computes an $2n \times 2n$ matrix C such that $c_{i,j} = n$ if and only if $x_{i,*}$ is dominated by $x_{j,*}$.

We are interested in the quarter corresponding to C , i.e. in the submatrix $C[n+1..2n, 1..n]$. It is obvious that $c_{i,j} = n$, $i = n+1, \dots, 2n$, $j = 1, \dots, n$, if and only if $h_k - x_{i,k} \leq x_{j,k}$, $k = 1, \dots, n$, or in other words, if and only if $x_{i,k} + x_{j,k} \geq h_k$, $k = 1, \dots, n$.

The runtime remains as the runtime of Matoušek's algorithm.

We will use the above observation in order to solve the *two-sensor problem*. For each sensor $s_i \in A \cup B$, we draw a disk of radius R_i . Denote by $|v_i v_{i+1}|$, $i = 1, \dots, n$ the length of an edge (v_i, v_{i+1}) of P , we define an $n \times n$ matrix X in the following way:

$$x_{i,j} = \frac{|R_i \cap (v_j, v_{j+1})|}{|v_j v_{j+1}|} \quad (2)$$

After we build the matrix X , we apply an algorithm for observation 1 with $h = (1, 1, \dots, 1)$. We consider the submatrix $C[n+1..2n, 1..n]$. It is clear that if two sensors $s_l \in A$ and $s_m \in B$ with radii R_l and R_m correspondingly cover the entire P then $x_{l,j} + x_{m,j} \geq 1$, for $j = 1, \dots, n$. On the other hand, if the

sum of row l and row m dominates $(1, \dots, 1)$, i.e. either $C(n+m, l) = n$ or $C(n+l, m) = n$ then, we first check whether sensors s_l and s_m belong to different groups. If the answer is negative, we reject this pair of sensors as a potential solution. If the answer is positive, it can still happen that P is uncovered.

Lemma 1. The only possible case, when P is uncovered and row l and row m dominates $(1, \dots, 1)$ (s_l and s_m belong to different groups), is when the first or the last edge of P are uncovered.

2. Consider some internal edge $(v_i, v_{i+1}), 1 < i < n$ of P . We need to show that it is covered by s_l and s_m . If sensor, say s_m , does not cover neither v_i nor v_{i+1} , then it follows that s_l should cover the entire (v_i, v_{i+1}) , since it is located on other side than s_m and the path P is monotone. Otherwise, assume that s_m covers, wlog v_i , but not v_{i+1} . Then, if v_{i+1} is also uncovered by s_l , the following edge (which always exists because our choice of i) is completely uncovered by two sensors, which violates condition of either $C(n+m, l) = n$ or $C(n+l, m) = n$. Notice that in this case, since we know that either $C(n+m, l) = n$ or $C(n+l, m) = n$, the edge (v_i, v_{i+1}) is covered. We can check in $O(1)$ time whether the first and the last edge of P are covered by $R_l \cup R_m$.

Thus, we proved the following Theorem.

Theorem 1. The 2-sensor problem can be solved in $O(n^{3/2}M(n)^{1/2})$ time.

Future work

In this paper, we present an efficient algorithm for covering a given path by two sensors. A challenging vein for future work is to consider approximations for the k -sensor problem, where $k > 2$. The k -sensor problem, for general k , is NP-complete [14].

References

1. Meguerdichian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M.B. Coverage problems in wireless ad-hoc sensor networks, *INFOCOM*, **2001**, 1380.
2. Huang, C.-F.; Tseng, Y.-C. The coverage problem in a wireless sensor networks, *ACM International Workshop on Wireless Sensor Networks and Applications*, **2003**.
3. Matoušek, J. Computing dominances in E^n , *Information Processing Letters*, **1991**, 38, 277.
4. Agarwal, P.; Sharir, M. Planar geometric location problems, *Algorithmica*, **1994**, 11, 185.
5. Hershberger, J. A faster algorithm for the two-center decision problem, *Inf. Process. Lett.*, **1993**, 47, 23.
6. Jaromczyk, J.; Kowaluk, M. An efficient algorithm for the Euclidian two-center problem, *Proc. 10th ACM Sympos. Comput. Geom.*, **1994**, 303.
7. Sharir, M. A near-linear algorithm for the planar 2-center problem, *Discrete Comput. Geom.*, **1997**, 18, 125.

8. Frederickson, G.; Johnson, D. Generalized selection and ranking: sorted matrices, *SIAM J. Comput.* **1984**, *13*, 14.
9. Eppstein, D. Faster construction of planar two-centers, *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms*, **1997**, 131.
10. Agarwal, P.; Sharir, M.; Welzl, E. The discrete 2-center problem, *Proc. 13th ACM Symp. on Computational Geometry*, **1997**, 147.
11. Shin, C.; Kim, J.; Kim, S.K. ; Chwa, K. Two-center problems for a convex polygon, *Proc. of ESA '98*, **1998**.
12. Huang, P.H.; Tsai, Y.T.; Tang, C.Y. A Near-quadratic algorithm for the alpha-connected two-center decision problem, *Proc. of 14th Canadian Conference on Computational Geometry*, **2002**, 133.
13. Coppersmith, D.; Winograd, S. Matrix multiplication via arithmetic progression, *Journal of Symbolic Computation*, **1990**, *9*(3), 251.
14. Bereg, S. Private communication, **2003**.