An Efficient Relay Sensors Placing Algorithm for Connectivity in Wireless Sensor Networks*

Jyh-Huei Chang ** and Rong-Hong Jan

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, 30050, Taiwan
jhchang@cis.nctu.edu.tw

Abstract. Randomly deployed sensor networks often make initial communication gaps inside the deployed area even in an extremely high-density network. How to add relay sensors such that the underlying graph is connected and the number of relay sensors added is minimized is an important problem in wireless sensor networks. This paper presents an Efficient Relay Sensors Placing Algorithm (ERSPA) for solving such a problem. Compared with minimum spanning tree algorithm and greedy algorithm, ERSPA achieves a better performance in terms of number of relay sensors added. Simulation results show that the average number of relay sensors added by minimal spanning tree algorithm is approximately up to two times than ERSPA algorithm.

1 Introduction

Randomly deployed networks often make initial communication gaps inside the deployed area even in an extremely high-density networks. In the random sensor network topology, the sensors may be sparsely located and the connectivity is no guaranteed. Therefore, finding an efficient algorithm for improving connectivity in wireless sensor networks and minimizing the number of relay sensors added is an important topic of researches.

In a finite domain, the connectivity of random network depends only on the probability distribution of critical transmission range. Many studies try to find efficient algorithms for determining the critical transmission range for connectivity [1-3]. The asymptotic distribution of the critical transmission radius for k-connectivity is derived in [1]. This study proved the critical transmission range in the unit-area square is $r_n = \sqrt{\frac{\log n + (2k-1)\log\log n + \xi}{\pi n}}$ where n is the number of network nodes and ξ is a constant. The critical transmitting range for connectivity in mobile ad hoc networks is proved in [2]. The author showed the critical transmission range (CTR) for a mobility model M is $r_M = c\sqrt{\frac{\ln n}{\pi n}}$ for some constant $c \geq 1$ where n is the number of nodes in the network. The mobility model

* This research was supported in part by the National Science Council, Taiwan, ROC, under grant NSC 94-2219-E-009-005 and NSC94-2752-E-009-005-PAE and in part by the Intel.

** Corresponding Author. Fax: 886-3-5721490

M is assumed to be obstacle free and nodes are allowed to move only within a certain bounded area. In addition, many researches focus on maintaining sensing coverage and connectivity in wireless sensor networks [4-7]. The transmission range (R_t) must be at least twice of the sensing range (R_s) is the sufficient condition to ensure that complete coverage preservation implies connectivity among active nodes [4]. Another study [5] enhanced the work in [4] to prove that the sufficient condition for complete coverage implies connectivity is $R_t = 2R_s$. In [7], a coverage configuration protocol is proposed to achieve guaranteed degrees of coverage and connectivity. This work provided different degrees of coverage requested by applications. To measure the coverage, the work divides the sensing area into $1m \times 1m$ patches. The coverage degree of a patch is approximately by measuring the number of active nodes that cover the center of the patch.

Note that the above studies [1-7] do not discuss how to place relay sensors to improve connectivity for a disconnected ad hoc network. Thus, there are many papers proposed for finding the optimal location to place the additional nodes to achieve network connectivity [8-11]. This problem can be reduced to a minimal Steiner tree problem. In [8], a relay sensor placement algorithm to maintain connectivity is proposed. They formulated this problem into a network optimization problem, named Steiner Minimum Tree with Minimum Number of Steiner Points (SMT-MSP). This study restricts transmission power of each sensor to a small value and adds relay sensor to guarantee connectivity. Simulation results show that their method can achieve better performance in terms of total consumed power and maximum degree, especially for sparse network topology. However, their algorithm runs a time complexity in $O(N^3)$. Some heuristic algorithms for the bounded edge-length Steiner tree problem with a good approximate ratio are proposed in [9-11]. Nevertheless, these heuristic algorithms do not consider the heterogeneous transmission ranges of terminal nodes and relay nodes.

Many researches focus on finding efficient heuristic algorithms to solve the minimal additional nodes placing problems and prolong the network lifetime [12-14]. A heuristic algorithm for energy preserving problem is proposed in [12]. This algorithm transforms the mixed-integer nonlinear problem into a linear programming problem. This study provides additional energy on the existing nodes and deploys relay nodes into the network to prolong network lifetime. In [13], three heuristic algorithms are proposed for achieving connectivity of a randomly deployed ad hoc wireless networks. This work connects the network with a minimum number of additional nodes and maximize utility from a given number of additional nodes for the disconnected network. The time complexity of the greedy algorithms is $O(N^2)$ in a two dimension space where N is the number of terminal nodes.

Our motivation is to find an efficient relay sensors placing algorithm to construct a connected communication graph for connectivity and minimize total relay sensors. Assume that all terminal and relay sensors have the same transmission range and all sensors are location aware. Simulation results show that ERSPA algorithm gives better performance in terms of the average number of relay sensors with respect to minimal spanning tree algorithm and greedy algorithm [13]. The average number of relay sensors in minimum spanning tree algorithm is approximately up to two times than ERSPA algorithm. This is because the ERSPA places relay sensors in optimal location to connect the maximal number of initial connected sub-graphs.

The remainder of this paper is organized as follows. In section 2, we describe problem formulation and network model. Section 3 illustrates the details of ERSPA algorithm. The simulation results and performance analysis are shown in section 4. Finally, the conclusions are given in section 5.

2 Problem Formulation and Network Model

Consider a wireless sensor network. We assume that the sensing area is in a two dimension space which is a bounded convex subset R of the Euclidean space. In this sensing area, the initially deploy sensors, called as terminal nodes, have been placed and a set of relay sensors are available to be added for connectivity. All terminal nodes and relay nodes are location aware such that the location information can be collected. The set of the terminal nodes is denoted as $N_t = \{N_{t1}, N_{t2}, ..., N_{tn}\}$. The transmission range of each terminal node is adjustable. Initially, the terminal nodes can adjust their transmission range to convey their location information to base station, then limit their transmission range in a bounded value R_t for energy efficient. The set of the locations of n terminal nodes denoted as $L_t = \{p_i \in R \mid i = 1, .., n\}$. The set of the initial network topology is in the form of undirected graph denoted as $G(N_t, E(L_t, R_t))$. Where $E(L_t, R_t) = \{(l_i, l_j) \mid l_i, l_j \in L_t, i \neq j, \| l_i - l_j \| \leq R_t\}$. In order to construct the connected communication graph, we can add the relay sensors to connect the initial separated sub-graphs. A solution is a set of locations to place relay sensors, $L_r = \{q_i \in R \mid i = 1, .., r\}$. The set of the relay sensors denoted as $N_r = \{N_{r1}, N_{r2}, ..., N_{rm}\}$. We formulate our problem as follows: A randomly deployed sensor network with $nR_t \times nR_t$ sensing area in the two dimension space. Given N_t and R_t, find the L_r for minimum relay sensor set N_r to make the graph $G(N_t \bigcup N_r, E(L_t \cup L_r, R_t))$ connected.

3 The Details of ERSPA Algorithm

The ERSPA algorithm includes the following three phases: 1) Find the initial graph $G(N_t, E(L_t, R_t))$; 2) Construct Delaunay; and 3) Add relay nodes. The details of the algorithm are given as follows.

Phase 1: Find the initial graph $G(N_t, E(L_t, R_t))$

Initially, divide the sensing area into $n \times n$ grids. The grid width is equal to the transmission range R_t. Connect all sensor nodes within the transmission range to construct initial connected sub-graphs by grid range searching method (see Figure 1). The details are described as follows.

Step 1: Search each grid using $R_t \times R_t$ searching range (see Figure 1-(a)). The area of $R_t \times R_t$ is equal to the area of each grid. If there exists any pair of nodes within this range, then connect them. This operation formed the sub-graphs G_{ij}

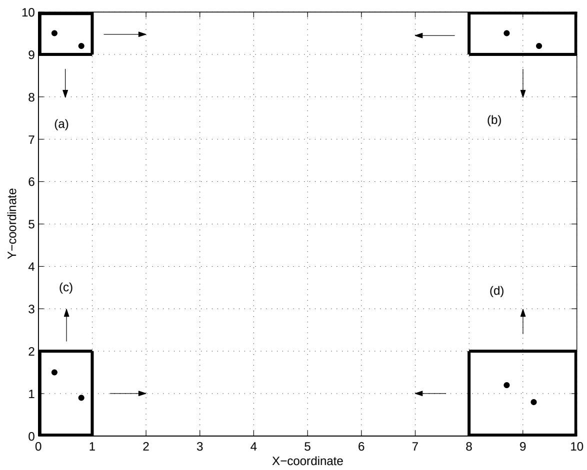

Fig. 1. An example of grid range searching method. (a) Searching range is $R_t \times R_t$. (b) Searching range is $R_t \times 2R_t$. (c) Searching range is $2R_t \times R_t$. (d) Searching range is $2R_t \times 2R_t$.

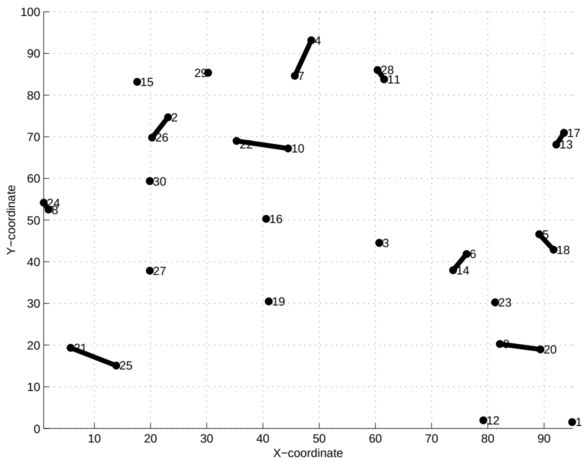

Fig. 2. The initial connected sub-graphs for a randomly deployed network with 30 sensors. The transmission range is 10 percentage of the side of the square sensing field. The initial disconnected terminal nodes indicated by the '•'-sign. The initial connected sub-graphs indicated by the coarse solid line.

where $i = 1, ..., n, j = 1, ..., n$. G_{ij} are the connected graphs in each grid.

Step 2: Search the sensing area from left to right and from top to down using $R_t \times 2R_t$ searching range. If the distance between any disconnected pair of nodes is equal or less than R_t, then connect them (see Figure 1-(b)). This operation formed the sub-graphs $G_{ij} \cup G_{i,j+1}$. The '$\cup$'-sign indicated connecting the connected graph in adjacent grids.

Step 3: Repeat step 2 and replace the searching range by using $2R_t \times R_t$ (see Figure 1-(c)). This operation constructed the sub-graphs $G_{ij} \cup G_{i+1,j}$.

Step 4: Repeat step 2 and replace the searching range by using $2R_t \times 2R_t$ (see Figure 1-(d)). This operation constructed the sub-graphs $(G_{ij} \cup G_{i+1,j+1}) \cup (G_{i,j+1} \cup G_{i+1,j})$.

After above operations, the initial resulting graphs are constructed (see Figure 2). The resulting graphs are illustrated as equation (1).

$$G = (G_{ij} \cup G_{i,j+1}) \cup (G_{ij} \cup G_{i+1,j}) \cup (G_{ij} \cup G_{i+1,j+1}) \cup (G_{i,j+1} \cup G_{i+1,j}) \quad (1)$$

where $i = 1, ..., n, j = 1, ..., n$. G_{ij} are the connected graphs in each grid. The '$\cup$'-sign indicated connecting the connected graphs in adjacent grids.

Phase 2: Construct Delaunay

Construct the Delaunay by using terminal nodes [15]. The construction of Delaunay is illustrated as follows. Let S be a set of points in a two dimension space. The Voronoi diagram of S, denoted as $Vol(S)$ which is decomposed into Voronoi cells $\{V_a : a \in S\}$ defined as equation (2).

$$V_a = \{x \in R^2 : |x - a| \leq |x - b| \forall b \in S\} \quad (2)$$

The dual of the Voronoi diagram is the Delaunay triangulation $Del(S)$. $Del(S)$ is geometrically realized as a triangulation of the convex hull of S (see Figure 3). As shown in Figure 3, the convex hull of initial connected sub-graphs in phase 1 is indicated by the coarse solid line. The purpose of constructing delaunay is used to find the nearest neighbor node for given node to connect by adding relay node(s). For example, nodes 3, 22, 10, 30, 27, and 19 are the neighbor nodes of node 16 (see Figure 3). Node 10 is the nearest neighbor node of node 16. Thus, for node 16, we can choose node 10 to connect in the next phase (phase 3).

Phase 3: Add relay nodes

After constructing Delaunay, we add the relay nodes to connect the disconnected sub-graphs.

Step 1: We only search the triangle including three points on its apexes and the three points are belong to three different sub-graphs. The triangles inside the convex hull of initial connected sub-graphs are not required to search. Then add a node on the circumcenter of the triangle and check whether the node can connect three sub-graphs or not. The circumcenter is the intersection of the perpendicular bisectors of the sides of the triangle. For example, triangle $(30, 26, 22)$ includes three points on its apexes that are belong to three different sub-graphs (see Figure 4). R_1 is the circumcenter of the triangle (30, 26, 22). If the distance

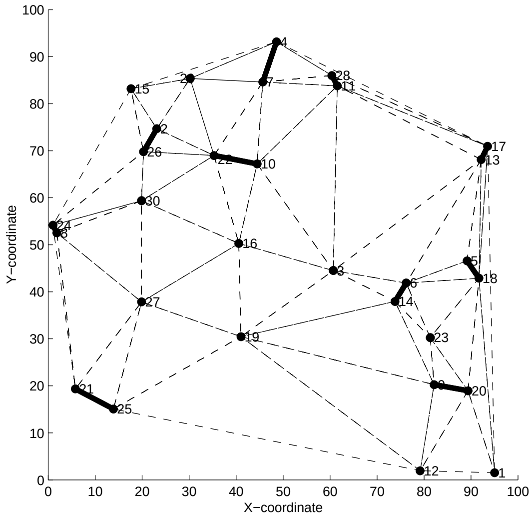

Fig. 3. An example of constructing Delaunay using 30 terminal nodes. The dash lines represented the edges of Delaunay triangulation that are not connected. The initial disconnected terminal nodes indicated by the '•'-sign. The initial connected sub-graphs indicated by the coarse solid line.

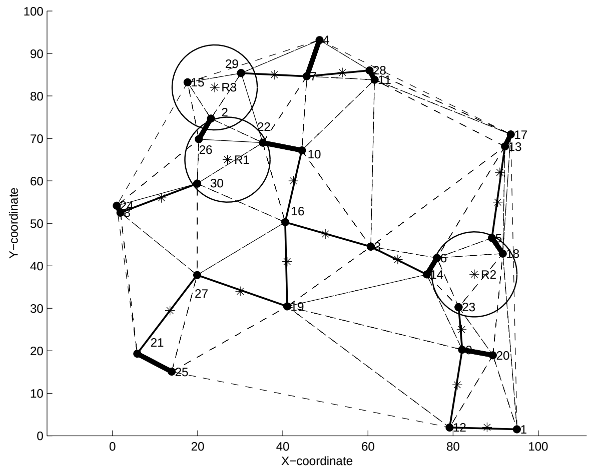

Fig. 4. An example of adding relay sensors with 30 terminal nodes. Place one relay node to connect three nodes is indicated by the circle centered at R_1. The relay sensors indicated by the '$*$'-sign. The initial disconnected terminal nodes indicated by the '•'-sign. The coarse solid line represented initial connected sub-graphs. The dash line represented the edge of Delaunay triangulation that are not connected. After adding relay node, the edge becomes connected that is indicated by slight solid line.

Table 1. The pseudo code of ERSPA algorithm.

```
begin Connect
  Randomly deploy N nodes
    for (every N_t) //N_t is the set of terminal nodes
      while (d(u,v) < R_t ) // u,v are any two terminal nodes
          Connect (u,v)
      end while
    end for
end Connect
begin Construct Delaunay
    for ( every N_t )
          Construct Delaunay
    end for
end Construct Delaunay
begin Add relay nodes
    for ( every nearest pair of nodes (G_i, G_j) )
      while (K × R_t < d(u,v) ≤ (K+1) × R_t, K = 0,1,...,n)
           Add K relay nodes //K is the number of relay nodes
      end while
    end for
end Add relay nodes
```

from $R1$ to each apex of the triangle is less than transmission range R_t, then add this relay node. The relay sensors indicated by the '$*$'-sign. The '$\bullet$'-sign represented initial disconnected terminal nodes and the initial connected sub-graphs represented by the coarse solid line.

Step 2: The disconnected terminal node only require to connect to its nearest node along the edge of the triangle. For example, the disconnected terminal node 16 has six neighbors. The distance between node 16 and node 10 is the smallest. Add a relay node to connect node 16 and node 10. This new connected edge is indicated by slight solid line. The number of required relay sensors to add into the edge of the nearest pair of nodes is illustrated as equation (3).

$$K \times R_t < d(u,v) \leq (K+1) \times R_t, K = 0,1,...,n \tag{3}$$

Where K is the number of relay nodes. $d(u,v)$ is the distance between node u and node v. Repeat phase 3 until the complete communication graph is connected.

In phase 1, the grid search takes $O(N^2)$ time. In phase 2, construct Delaunay takes $O(NlogN)$ times. Phase 3 requires $O(N)$ times to add relay sensors. Where N is the number of terminal nodes. The total time complexity of ERSPA algorithms is $O(N^2)$. It is feasible in a two dimension space. The pseudo code of ERSPA is illustrated as in table 1.

4 Simulation Results

4.1 Performance Metrics and Environment Setup

This section presents the performance analysis of the ERSPA algorithm. The metrics for performance are given as follows.

1) Average number of relay sensors: The average number of relay sensors is defined as the required minimal average number of additional sensors to make the network connected.

2) Time complexity: Time complexity is defined as the time to run the algorithm.

The environment setup of simulation is described as follows. There are different number of terminal sensors that randomly deployed in a 100×100 two-dimensional sensing area. The maximum transmission range is 10 percentage of the side of the square sensing field. The network can convey location information of terminal nodes to base station, then limit the transmission range to a bounded value R_t for energy efficient.

4.2 Numerical Results

Comparisons of the two performance metrics were made for three schemes: ERSPA algorithm, Minimum Spanning Tree (MST) algorithm and greedy algorithm [13]. The MST algorithm has two steps. First, generates a minimum spanning tree to connect the terminal nodes. Then, place the relay nodes on the edges of the minimal spanning tree that are longer than the transmission range R_t. The MST algorithm takes $O(NlogN)$ times.

The performance metrics includes the total number of relay sensors and the time complexity. The details will illustrate as follows.

1) Average number of relay sensors: Figure 5 shows that the average number of relay nodes of ERSPA is smaller than minimum spanning tree algorithm and greedy algorithm when the the number of terminal nodes are 50 and 90. Figure 6 shows that the average number of relay nodes of ERSPA is smaller than minimum spanning tree algorithm and greedy algorithm under different terminal nodes. The average number of relay sensors in MST is approximately up to two times than ERSPA algorithm. This is because ERSPA find the optimal location to place relay sensors and connected the maximal number of disconnected subgraphs.

2) Time complexity: The time complexity of ERSPA algorithm is $O(N^2)$. The minimum spanning tree takes $O(NlogN)$ times. Greedy algorithm takes $O(N^2)$ times. N is the number of terminal nodes. The time complexity of ERSPA is feasible in a two dimension space.

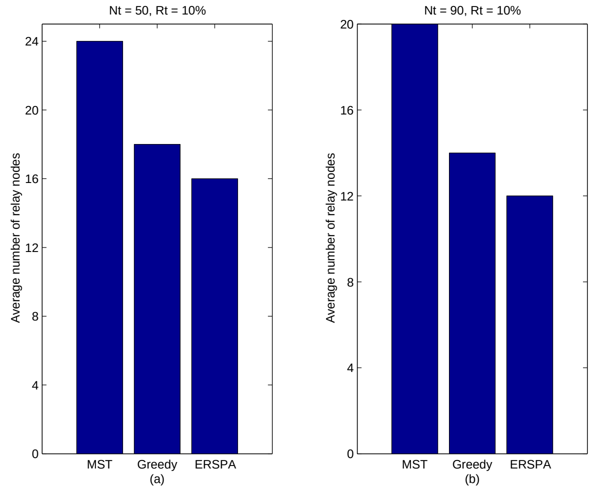

Fig. 5. (a)The average number of relay nodes for connectivity with 50 terminal nodes. (b)The average number of relay nodes for connectivity with 90 terminal nodes. The transmission range is 10 percentage of the side of the square sensing field.

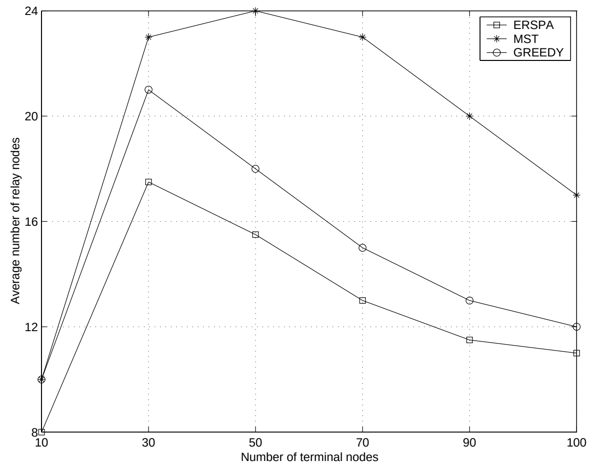

Fig. 6. The average number of relay nodes for connectivity under different terminal nodes. The transmission range is 10 percentage of the side of the square sensing field.

5 Conclusions

This paper presents an efficient relay sensors placing algorithm for connectivity in wireless sensor networks. Compared with minimal spanning tree algorithm and greedy algorithm, our ERSPA algorithm gives better performance in terms of the average number of relay sensors. This is because ERSPA places the relay sensors in optimal place to connect the maximum number of initial connected sub-graphs such that the average number of relay sensors can be minimized. We are confident that ERSPA is an efficient and useful algorithm for further wireless ad hoc sensor networks.

References

[1] R.-J Wan and C.-W. Yi: Asymptotic Critical Transmission Radius and Critical Neighbor Number for K-connectivity in Wireless Ad Hoc Networks. Proceedings of the 5th ACM international symposium on mobile ad hoc networking and computing. (2004) 1-8

[2] P. Santi: The Critical Transmitting Range for Connectivity in Mobile Ad Hoc Networks. IEEE Transaction on Mobile Computing. (2005) 310-317

[3] H. Koskinen: A Simulation-based Method for Predicting Connectivity in Wireless Multihop Networks. Telecommunicatin Systems. (2004) 321-338

[4] H. Zhang and J. C. Hou: Maintain Sensing Coverage and Connectivity in Largen Sensor Networks. Ad Hoc and Sensor Networks. (2005) 89-124

[5] D. Tian and N. D. Georganas: Connectivity Maintence and Coverage Preservation in Wireless Sensor Networks. Ad Hoc Networks. (2005) 744-761

[6] H. Gupta, S. R. Das, and Q. Gu: Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. Prcoeedings of MobiHoc'03.

[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill: Integrated Coverage and Connectivity Confuration in Wireless Sensor Networks. ACM SENSYS'03.

[8] X. Cheng, D.-Z. Du, L. Wang, and B. Xu: Relay Sensor Placement in Wireless Sensor Networks. IEEE Transaction on Computers. (2001)

[9] H. Koskinen, J. Karvo, and O. Apilo: On Improving Connectivity of Static Ad-Hoc Networks by Adding Nodes. Proceedings of the 6th ACM International Symposium on Low Power Electronics and Design (ISLPED'03). (2003) 251-254

[10] I. Mandoiu and A. Zelikovsky: A Note on the MST Heuristic for Bounded Edge-length Steiner Trees with Minimum Number of Steiner Points. Information Processing Letters. (2000)

[11] D. Chen, D.-Z. Du, X. Xu, G.-H. Lin, L. Wan, and G. Xue: Approximations for Steiner Trees with Minimum Number of Steiner Points. Journal of Global Optimization. (2000)

[12] Y. T. Hou, Y. Shi, H. D. Sherali, and S.F. Midkiff: On Energy Provisioning and Relay Node Placement for Wireless Sensor Networks. IEEE Transactions on Wireless Communications. (2005) 2579-2590

[13] H. Koskinen, J. Karvo, and O. Apilo: On Improving Connectivity of Static Ad-Hoc Networks by Adding Nodes. Med-Hoc-Net. (2005)

[14] D. Ganesan, R. Christescu, and B. B. Lozano: Power-Efficient Sensor Placement and Transmission Structure for Data Gathering under Distortion Constrain. ACM Transcactin on Sensor Networks. (2005)

[15] K. Mulmuley: Computational Geometry, Prentice Hall (1994)