Coverage in Wireless Ad-hoc Sensor Networks

Xiang-Yang Li Peng-Jun Wan Ophir Frieder

Dept. of Comp. Sci., Illinois Institute of Technology, Chicago, IL, 60616. Email: {xli,wan,ophir}@cs.iit.edu

Abstract

Sensor networks pose a number of challenging conceptual and optimization problems such as location, deployment, and tracking [1]. One of the fundamental problems in sensor networks is the calculation of the coverage. In [1], it is assumed that the sensor has the uniform sensing ability. We provide efficient distributed algorithms to optimally solve the *best-coverage problem* raised in [1]. In addition, we consider a more general sensing model: the sensing ability diminishes as the distance increases. As energy conservation is a major concern in wireless (or sensor) networks, we also consider how to find an optimum best-coverage-path with the least energy consumption and how to find an optimum best-coverage-path that travels a small distance. In addition, we justify the correctness of the method proposed in [1] that uses the Delaunay triangulation to solve the best coverage problem and show that the search space of the best coverage problem can be confined to the relative neighborhood graph, which can be constructed locally.

I. Introduction

Sensor networks pose a number of challenging conceptual and optimization problems such as location, deployment, and tracking [1]. In Meguerdichian, *et al.*[1], they addressed one of the fundamental problems, namely coverage, which in general answers the questions about the quality of service that can be provided by a particular sensor network. They gave polynomial time centralized algorithms to solve the questions optimally. However, their algorithms rely heavily on some geometrical structures such as the Delaunay triangulation and the Voronoi diagram which can not be constructed locally or even efficiently in a distributed manner. Typically, we say that a distributed algorithm is communication efficient if its total communication cost is linear in the number of nodes. In addition, the correctness of using these two geometry structures is not presented in their paper.

In a wireless ad hoc network (or sensor network), each wireless node has a maximum transmission power so that it can send signals to all nodes within its transmission range. If a node v is not within the transmission range of the sender u, nodes u and v communicate through multi-hop wireless links by using intermediate nodes to relay the message. Each node in the wireless network also acts as a router, forwarding data packets for other nodes. We assume that each static wireless node knows its position information, either through a low-power Global Position System (GPS) receiver. If GPS is not available, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths. Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [2]. For simplicity, we also assume that all wireless nodes have the same maximum transmission range, and we normalize it to one unit. We assume that all wireless nodes have distinctive identities. Consequently, all wireless nodes S together define a unit disk graph $UDG(S)$, which has an edge uv if and only if the Euclidean distance between u and v is less than one unit.

We call all nodes within a constant k hops of a node u in the unit disk graph $UDG(S)$ as the *k-local nodes* of u. Usually, here the constant k is 1 or 2, denoted by $N_k(u)$, which will be omitted if it is clear from the context. By broadcasting, each node u can gather the location information of all nodes within the transmission range of u. The total communication cost to do so is obviously $O(n)$ when omni-directional antennas are used.

In wireless networks, distributed algorithms often have advantages over centralized algorithms in reducing communication cost. Notice, energy conservation is one of the critical issues in designing wireless networks. Thus, efficient distributed algorithms are always demanded in solving many challenging questions in wireless networks. The geometrical nature of the multi-hop wireless networks allows a promising idea: localized algorithms.

A distributed algorithm is a *localized algorithm* if it uses only the information of all local nodes plus the information of a constant number of additional nodes. A graph G can be constructed locally in the ad hoc wireless environment if each wireless node u can compute the edges of G incident on u by using only the location information of all local nodes.

Given a wireless sensor networks, we are interested in designing a localized algorithm that finds a path connecting a point s and a point t, which maximizes the smallest observability of all points on the path. It is called the *best coverage problem*[1]. Meguerdichian, *et al.* [1] presented a centralized method using the Delaunay triangulation to solve the best coverage problem. Several related problems were also studied recently. The *minimum exposure problem* [3] is to find a path connecting two points in the domain that minimizes the integral observability over the time traveled from the source point to the destination point. While the *worst coverage problem*[1] is to find the path that maximizes the distance of the path to all sensor nodes. Meguerdichian, *et al.*[1] presented a centralized method using the Voronoi diagram to solve the worst coverage problem. The minimum exposure problem was studied in [3]. We concentrate on the best coverage problem. We provide efficient distributed algorithms to solve it. In addition, we justify the correctness of using the Delaunay triangulation to solve the best coverage problem. Moreover, we show that the search space of the best coverage problem can be confined to the relative neighborhood graph, which can be constructed locally. It automatically gives an efficient distributed algorithm for the best coverage problem. Some extensions of the best coverage are discussed.

The rest of the paper is organized as follows. In Section II, we define terms and notations used in presenting our algorithms. We also briefly review the algorithms by Meguerdichian, *et al.*[1] and outline some discrepancies in their algorithms. In Section III, we present the first localized algorithm that solves the best coverage problem efficiently. We also discuss several extensions of the best-coverage problem. Specifically, we consider how to find an optimum best-coverage-path that conserves energy, and how to find an optimum best-coverage-path with small traveling distance. Both the correctness of our algorithm and the correctness of the algorithm by Meguerdichian *et al.* are justified in Section IV. We conclude our paper and discuss possible future research directions in Section V.

II. Preliminaries

A. Problem formulation

We assume that the wireless sensor nodes are given as a set of n points S distributed inside a two-dimensional domain Ω. For simplicity, we assume that the domain Ω is given as a *planar-straight-line graph* (PSLG), which is a collection of line segments and points in the plane, closed under intersection. Let B be the set of points that define the domain boundary. For simplicity, we assume that the convex hull $CH(S)$ of the set of sensors S is contained inside the domain Ω. We also assume that every wireless node has the same maximum transmission range. Then the set of wireless sensors S defines a unit disk graph $UDG(S)$. We always assume that the graph $UDG(S)$ is connected.

We first give some geometry notations that will be used in the remainder of this Section to mathematically formulate the problems considered. Let $\|xy\|$ denote the Euclidean distance of two points x and y.

Definition 1: The distance of a point x to a set of points V, denoted by $dist(x, V)$, is the smallest distance of x to all points of V. In other words,

$$dist(x, V) = \min_{y \in V} \|xy\|.$$

Notice that the point set V may be infinite. For example, V could be all points lying on a segment uv. We use $dist(x, uv)$ to denote the smallest distance from x to all points on the segment uv.

Given two point sets U and V, the breach distance $dist(U, V)$ is defined as $\min_{x \in U} dist(x, V)$. In other words, $dist(U, V) = \min_{x \in U, y \in V} \|xy\|$. Usually, the breach distance is called just *distance* in the literature.

Definition 2: The *coverage-distance* of a point set U by another point set V, denoted by $cover(U, V)$, is the maximum distance of every point $x \in U$ to V. That is,

$$cover(U, V) = \max_{x \in U} dist(x, V).$$

Notice that, the breach distance $dist(U, V)$ is symmetric, i.e., $dist(U, V) = dist(V, U)$, while the coverage distance $cover(U, V)$ is *not* symmetric. Here both point sets U and V can be infinite. For example, U can be a path connecting two points s and t and V all sensor nodes. Given a path $\Pi(s, t)$ inside Ω connecting s and t, the *coverage-distance* $\max_{x \in \Pi(s,t)} dist(x, S)$ of the path $\Pi(s, t)$ specifies how well the path is protected by the sensors, while, on the reverse side, the breach distance $\min_{x \in \Pi(s,t)} dist(x, S)$ specifies how far the path is from all sensors. Thus, for wireless sensor networks, the coverage problem has two folds: the best coverage and the worst coverage, which are defined as follows.

Definition 3: A path $\Pi(s, t)$ that achieves the minimum coverage-distance $cover(\Pi(s, t), S)$ is called a *best-coverage-path*. The minimum coverage-distance $cover(\Pi(s, t), S)$ of all paths connecting s and t is called the *best-coverage-distance* or the *support-distance*.

Thus, given a set of sensors S inside a two-dimensional domain Ω, a starting point $s \in \Omega$, and an ending point $t \in \Omega$, we find a path $\Pi(s,t)$ inside Ω to connect s and t such that the coverage distance $cover(\Pi(s,t), S) = \max_{x \in \Pi(s,t)} dist(x, S)$ is minimized. In other words, we try to find a path connecting s and t such that every point x of the path is covered by some sensor nodes with small distance.

This problem has several interesting applications. For example, consider a war-field denoted by a two-dimensional domain Ω. Assume that a postman soldier wants to travel from position s to position t in Ω. There are some randomly distributed protection soldiers, denoted by a set of points S, which will protect the postman soldier. Then it is always desirable to find a path in Ω such that the maximum distance of the postman soldier from the protection soldiers is minimized when the soldier travels from s to t.

There are several variations for the best coverage problem. Notice that, as shown later, there are many paths that achieve the *best-coverage-distance*. As the energy conservation is a critical issue in the wireless networks, we wish to find a path that consumes the least energy possible while still achieving the best-coverage-distance. The other variation is to find a path with the minimum total traveling distance among all optimum paths with the best-coverage-distance. This is justified by the above postman soldier example.

Definition 4: A path $\Pi(s,t)$ that achieves the maximum breach-distance $dist(\Pi(s,t), S)$ is called a *worst-coverage-path*. The maximum breach-distance $dist(\Pi(s,t), S)$ of all paths that connecting s and t is called the *worst-coverage-distance* or the *breach-distance*.

This problem also has several interesting applications. Again consider the same postman soldier problem. Here we assume that there is a set of mines, denoted by a two-dimensional point set S, randomly distributed in the domain Ω. Then the postman soldier want to travel from a position s to a position t in Ω such that the path walked is far from any mine to minimize the risk.

Meguerdichian, *et al.*[1] presented a centralized method using the Voronoi diagram to solve the worst coverage problem. Notice, for the best coverage problem, we will show that the search space could be refined to the relative neighborhood graph. Thus, an efficient localized algorithm is almost straightforward. However, for the worst coverage problem, currently, no efficient distributed algorithm is known except the adaption of the Voronoi Diagram. We leave it as a possible future work to design such efficient algorithm.

Sensing devices generally have widely different theoretical and physical characteristics. Interestingly, in most sensing device models, the sensing ability diminishes as distance increases. Let $\mathcal{D}(\int, \surd)$ be the sensing ability of sensor s at point p. When point p is out of the sensing range of the sensor s, i.e., $\|sp\| > 1$, then $\mathcal{D}(\int, \surd) = \imath$. Notice that the sensing range is normalized to one unit here. In [3], they assumed that $\mathcal{D}(\int, \surd) = \frac{\lambda}{\|\int \surd\|^{\kappa}}$ for sensor-technology dependent parameters λ and κ. We adopt the following sensing model.

1. The sensing ability of every sensor device is *uniform*, i.e., $\mathcal{D}(\int_\rangle, \sqcap) = \mathcal{D}(\int_|, \sqsubseteq)$ if $\|s_i u\| = \|s_j v\|$.

2. The sensing ability satisfies a *monotone property*: $\mathcal{D}(\int, \sqcap) > \mathcal{D}(\int, \sqsubseteq)$ if $\|su\| < \|sv\|$.

Given a point p, its closest-sensor observability $I_c(p)$ is defined as $\mathcal{D}(\int_\surd, \surd)$, where s_p is the closest sensor to point p. In other words, we define

$$I_c(p) = \max_{s_p \in S} \mathcal{D}(\int_\surd, \surd).$$

Then given a path $\Pi(s,t)$ connecting points s and t, its closest-sensor observability is defined as

$$I_c(\Pi(s,t)) = \min_{p \in \Pi(s,t)} I_c(p).$$

In [3], Meguerdichian *et al.* also studied the so call all field observability. Define the all-sensor observability of point p, denoted by $I_a(p)$, as follows

$$I_a(p) = \sum_{s_i \in S} \mathcal{D}(\int_\rangle, \surd).$$

Similarly, we define the all-sensor observability as

$$I_a(\Pi(s,t)) = \min_{p \in \Pi(s,t)} I_a(p).$$

However, we are not aware of applications that specifically needs the all-sensor observability.

We concentrate on the closest-sensor observability. In other words, we try to find a path connecting s and t such that all point on the path is well-observed by some sensors. It is easy to show the correctness of the following lemma from the definitions of the best-coverage-path and the closest-sensor observability.

Lemma 1: The best-coverage-path also achieves the maximum closest-sensor observability.

Consequently, in the rest of the paper, we must only study how to find the best-coverage-path, which also achieves the maximum closest-sensor observability.

B. Geometry Notations

Delaunay triangulation and Voronoi diagram [4], [5], [6] are widely used in many areas. We begin with definitions of the Voronoi diagram and the Delaunay triangulation. We assume that all wireless nodes are given as a set S of n vertices in a two dimensional space. Each node has some computational power. We also assume that there are no four vertices of S that are co-circular. A triangulation of S is a *Delaunay triangulation*, denoted by $Del(S)$, if the circumcircle of each of its triangles does not contain any other vertices of S in its interior. A triangle is called the *Delaunay triangle* if its circumcircle is empty of vertices of S. The *Voronoi region*, denoted by $Vor(p)$, of a vertex p in S is a collection of two dimensional points such that every point is closer to p than to any other vertex of S. The *Voronoi diagram* for S is the union of all Voronoi regions $Vor(p)$, where $p \in S$. The Delaunay triangulation $Del(S)$ is also the dual of the Voronoi diagram: two vertices p and q are connected in $Del(S)$ if and only if $Vor(p)$ and $Vor(q)$ share a common boundary. The shared boundary of two Voronoi regions $Vor(p)$ and $Vor(q)$ is on the perpendicular bisector line of segment pq. The boundary segment of a Voronoi region is called the *Voronoi edge*. The intersection point of two Voronoi edge is called the *Voronoi vertex*. When there is no four points of S that are co-circular, then every Voronoi vertex has only exactly three Voronoi edges incident on it. The Voronoi vertex is the circumcenter of some Delaunay triangle. Figure 1 gives an example of the Voronoi Diagram and the Delaunay triangulation of a set of two dimensional points.

Notice that, generally, we cannot construct the Delaunay triangulation or the Voronoi diagram efficiently in a distributed way. For example, the radius of circumcircle of three nodes u, v and w could be as large as ∞. To decide whether $\triangle uvw$ is a triangle in

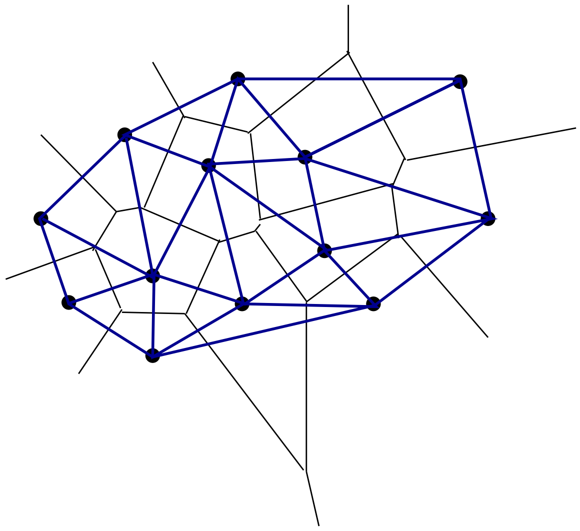

Fig. 1. The Voronoi Diagram and the Delaunay triangulation of a set of two dimensional points. The Delaunay triangulation is represented by thicker lines.

the Delaunay triangulation, we have to check if its circumcircle is empty of other nodes inside, which implies that we almost for sure have to check if any given node is inside the circumcircle of $\triangle uvw$. It is not appropriate, thus, to require the construction of the Delaunay triangulation in the wireless communication environment because of the possible massive communications it requires. Therefore, Li [7] studied a subset of the Delaunay triangulation. Let $UDel(S)$ be the graph by removing all edges of $Del(S)$ that are longer than one unit, i.e., $UDel(S) = Del(S) \cap UDG(S)$. Call $UDel(S)$ the *unit Delaunay triangulation.* Li, *et al.* [7], [8] provided an efficient localized algorithm that constructs a planar graph, called localized Delaunay triangulation $PLDel(S)$, which contains $UDel(S)$ as a subgraph. Thus, the constructed graph can be used by almost all algorithms that require to use the structure $UDel(S)$ or even $Del(S)$. We review how to construct the localized Delaunay triangulation $PLDel(S)$ later.

Various proximity subgraphs of the unit disk graph were studied [9], [10]. For convenience, let $disk(u, v)$ be the closed disk with diameter uv; let $disk(u, v, w)$ be the circumcir-

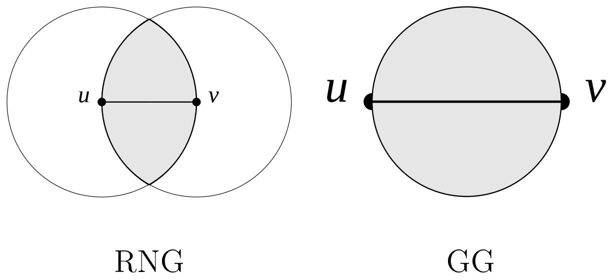

Fig. 2. The definitions of RNG and GG on point set. Left: The shaded area is the lune defined by uv, which is empty for RNG. Right: The diametric circle using uv is empty of nodes inside for GG.

cle defined by the triangle $\triangle uvw$; let $B(u,r)$ be the circle centered at u with radius r. Call the interior of the intersection $B(u,\|uv\|)\cap B(v,\|uv\|)$ the *lune*, denoted by $lune(u,v)$, defined by two points u and v. The constrained *relative neighborhood graph* $RNG(V)$ over a point set V has an edge (u,v) if $\|uv\|\leq 1$ and the $lune(u,v)$ does not contain any point from V in the interior. Toussaint and Jaromczyk [11], [12] gave the first definition of the relative neighborhood graph to study the pattern recognition. The *constrained Gabriel graph* of a point set V, denoted by $GG(V)$, consists of all edges uv such that $\|uv\|\leq 1$ and the $disk(u,v)$ does not contain any node from V. Gabriel and Sokal [13] defined the Gabriel graph for the geographic variation analysis. Obviously, the relative neighborhood graph is always a subgraph of the Gabriel graph. See Figure 2 for an illustration of their definitions.

It was well known that Delaunay triangulation, Voronoi Diagram, Gabriel graph and the relative neighborhood graph in two dimensions can be constructed in $O(n\log n)$ time. See [6] for efficient construction of the Delaunay triangulation and the Voronoi Diagram. Since relative neighborhood graph is a subgraph of the Delaunay triangulation, a simple method with $O(n^2)$ time complexity is to check whether the lune of each edge in the Delaunay triangulation is empty. If it is empty, then we add that edge to RNG. Supowit [14] gave the first $O(n \log n)$ time-complexity algorithm to construct the relative neighborhood graph in two dimensions using the l_2 metric, which eliminates edges from the Delaunay triangulation in $O(n \log n)$ time. Several methods [15], [16] were proposed then for constructing RNG efficiently. The construction of the Gabriel graph is a tad easier; see [17]. The algorithm by Matula and Sokal [17] is based on the observation that the Gabriel graph only contains those edges in the Delaunay triangulation that do not intersect their Voronoi edges.

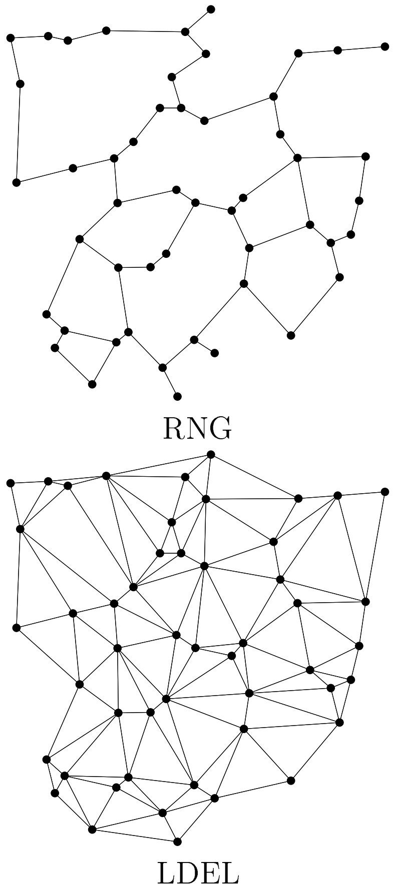

Fig. 3. Various topologies for best-coverage. Except Delaunay triangulation, all topologies can be constructed locally.

Figure 3 illustrates four different topologies that could be used to solve the best coverage problem. All graphs are planar. The relative neighborhood graph, the Gabriel graph and the localized Delaunay triangulation can be constructed efficiently in a localized manner with communication cost $O(n \log n)$ bits. However, the Delaunay triangulation can only be constructed efficiently in a centralized manner.

C. Prior Arts

Wireless sensor networks are used practically in our life for many years. Although many researchers have mentioned the coverage notion in wireless sensor networks, it seems that Meguerdichian, *et al.* [1] were among the first several researchers to identify the importance of using Delaunay triangulation and Voronoi diagram in sensor network coverage. A related problem is the so called *art gallery problem* [18], in which one must determine the number of observers necessary to cover an art gallery room such that every point of the room is watched by at least one observer. A sensor network was also used to detect the global ocean color by assembling and merging data from satellites at different orbits [19].

Notice that, the coverage problem we study here is different from the coverage problem studied in cellular network. Traditionally in cellular networks, coverage studies the maintainence of connectivity and thus continuouity of network service. In those scenarios, we often have to find the optimum number of base stations required to achieve some system objectives; see [20] for more detail. In [21], Hall studied how many wireless nodes with fixed coverage radius r are needed so that every point of a unit square region is covered by some wireless node with high probability. If the connectivity coverage is of concern, then we need to find how many wireless nodes are needed in a unit area square such that the resulting UDG is connected with high probability. Gupta and Kumar [22] studied the dual of this problem: if the number of nodes is fixed, then what is the smallest r such that the resulting UDG is connected with high probability.

In [1], Meguerdichian, *et al.* developed centralized algorithms to solve the best coverage problem using the Delaunay triangulation. Notice that the Delaunay triangulation can be constructed in $O(n \log n)$ time in a centralized manner if the geometrical information of all sensors are available. Thus, their algorithm has the best possible time complexity among centralized algorithms. However, no justification as to why the search space can be confined to the Delaunay triangulation for the best coverage problem was provided. Later, we provide a formal proof of the correctness of using Delaunay triangulation in Section IV. Additionally, they connect the starting point s to its closest sensor node u_s and connect the ending point t to its closest sensor node u_t. This is based on intuition [1]. We formally prove that there is an optimum best-coverage-path with this property if the unit disk graph $UDG(S)$ is connected.

To find an optimum best-coverage-path, they assign each Delaunay edge a weight equal to its Euclidean distance and then apply some graph algorithms on it to find the path with the min max weight. Remember that we want to find a path such that the maximum distance of all points of this path to its closest sensor node is minimized. Thus, we assign the weight of an edge uv as $\max_{x \in uv} dist(x, S)$ instead, which is *at most* $\frac{1}{2}\|uv\|$. Surprisingly, their weight assignment method also leads to a correct solution. Notice, intuitively, it is insufficient to just consider the midpoint of a Delaunay edge uv to compute its weight $\max_{x \in uv} dist(x, S)$. This is because it is possible that there is some other sensor node w that lies inside $dist(u, v)$. See Figure 4. The weight of the Delaunay edge uv is less than $\frac{1}{2}\|uv\|$. We later show that the search space can be confined to a much smaller graph, namely the relative neighborhood graph. Moreover, the weight of each edge uv in that graph is guaranteed to be *exactly* $\frac{1}{2}\|uv\|$.

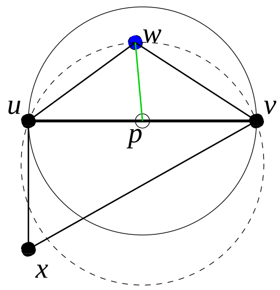

Fig. 4. The weight of a Delaunay edge uv is less than $\frac{1}{2}\|uv\|$.

Our later analysis shows that there is always an optimum best-coverage-path that only uses the edges in the relative neighborhood graph. In other words, we can still find an optimum best-coverage-path without using the edges of the Delaunay triangulation that are not in the relative neighborhood graph. We denote such edges by Del-RNG. Consequently, although the weight assignment of the edges of Del-RNG in [1] is not exactly correct, the computed min max path is still an optimum path. Notice, by definition, an edge uv of RNG has coverage-distance exactly equal to $\frac{1}{2}\|uv\|$.

D. Growing Disks

Assume that every sensor node originally has a disk centered at it with radius 0 and every disk starts growing with the same speed. See Figure 5. Let $D(S, r)$ be the region covered by all disks centered at points of S with radius r. Let $\overline{D(S, r)}$ be the complementary region of $D(S, r)$ in domain Ω. Then the best coverage problem asks what is the smallest radius value r such that there is a path, inside the region $D(S, r)$, connecting points s and t. On the other hand, the worst coverage problem asks what is the largest radius value r such that there is a path, inside the region $\overline{D(S, r)}$, connecting points s and t.

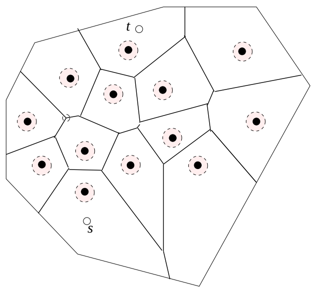

Fig. 5. Each sensor node has a disk specifying the region covered by it. All disks have the same radius and grow with the same speed.

III. The Best Coverage Problem

A. Algorithm

We first give an efficient distributed algorithm that solves the best coverage problem efficiently. Here assume that we are given a set of sensors S, a starting point s, and an ending point t in a two-dimensional domain Ω such that the unit disk graph $UDG(S)$ is connected and the convex hull $CH(S)$ of S is contained inside Ω.

Algorithm 1: FindBestCoverage(S, Ω, s, t)

1. Find the closest sensor node of the starting point s if itself is not a sensor node. Assume u_s is the closest sensor node. Similarly, find the closest sensor node u_t of the ending point t.

2. Each sensor node u locally constructs all edges uv of the relative neighborhood graph $RNG(S)$, where v is also a sensor node. This can be constructed as follows. Each node u broadcasts its location information and listens to the broadcasting by its neighbors. Thus, after this step, we assume that each node u has the location information of $N_1(u)$. Node u adds an edge uv if and only if the $lune(u, v)$ does not contain any nodes from $N_1(u)$ inside.

3. Assign each constructed edge uv with weight $\frac{1}{2}\|uv\|$.

4. Run a distributed *shortest path* algorithm to compute the *shortest path* connecting u_s and u_t. Here, the *weight* of a path is the maximum weight of all of its edges. A path is the shortest path if it has the

minimum weight among all paths connecting u_s and u_t. The Bellman-Ford algorithm [23] can be modified to solve this shortest path problem.

5. Let $\Pi(u_s, u_t)$ be a computed path and $\|\Pi(u_s, u_t)\|$ be the weight of the path. Then the path concatenating the edge su_s, path $\Pi(u_s, u_t)$, and the edge $u_t t$ is an optimum best-coverage-path. The best-coverage-distance is $\max(\|su_s\|, \|\Pi(u_s, u_t)\|, \|u_t t\|)$. Here $\|su_s\|$ and $\|u_t t\|$ are the Euclidean distance between points.

B. The time and communication complexity

We can also implement the above algorithm in a centralized manner: constructing the relative neighborhood graph $RNG(S)$, and then apply the Bellman-Ford algorithm [23] to find the *shortest path* between nodes u_s and u_t. The time complexity of this centralized algorithm is $O(n \log n)$: The first step costs $O(n)$ time; we can construct the relative neighborhood graph in $O(n \log n)$ time; and we can compute the shortest path connecting two vertices in a planar graph in time $O(n \log n)$. This centralized approach has the same complexity as that given in [1], but it has an advantage of being run efficiently in a distributed manner.

For wireless sensor networks, however, it is impractical to collect the location information of all sensors due to the massive communication it requires. Thus, a distributed algorithm is a must. Notice that the relative neighborhood graph of all sensors S can be constructed efficiently by using a localized approach. It can be constructed in $O(N_1(u) \log N_1(u))$ time using the approach in [14] by Supowit or in time $O(N_1(u)^2)$ if simply checking each edge incident on u. The communication cost is also small as compared to collecting the locations of all sensor nodes. The communication cost of constructing the graph $RNG(S)$ using a distributed manner is $O(n \log n)$ bits. We assume that the identity of each wireless node can be represented by $O(\log n)$ bits and the geometry location information can be represented by $O(1)$ bits.

C. Extensions

In addition, we consider some extensions of the best coverage problem and present efficient distributed algorithms to solve them. Notice that, the coverage-distance of two points s and t depends on their distances to the closest sensors. If we want to improve the coverage-distance of all pairs of points in the domain by adding more sensors, these new sensors should be placed at the circumcenters of Delaunay triangles that have the largest circumradius.

C.1 Energy Conservation

As energy conservation is critical, the first extension is to find a path with the best-coverage-distance while the total energy consumed by this path is minimized among all optimum best-coverage-paths. We assume that the energy needed to support a link uv is proportional to $\|uv\|^{\alpha}$, where α is a real constant between 2 and 5. In the best-coverage problem, finding areas of high observability from sensors and identifying the best support and guidance regions are of primary concern [1]. For example, in a sensor network for detecting fire, it is not only required that the sensor network observe a given region, it is also necessary that the sensor that detects the fire can report the fire to a center station efficiently. We need find a reporting path that consumes less energy.

Algorithm 2: EnergyConsrvngBestCoverage(S, Ω, s, t)

1. Run a distributed *shortest path* algorithm to compute the *coverage distance* of the best-coverage-path connecting u_s and u_t. Let β be the best coverage distance.

2. Construct the Gabriel graph $GG(S)$ and prune out all edges of the Gabriel graph $GG(S)$ with weight larger than β, and call the remaining graph the residue graph G.

3. Assign each edge uv of the residue graph G the weight equal to $\|uv\|^{\alpha}$, where α is the propagation constant depending on the transmission environment.

4. Run a distributed *shortest path* algorithm to compute the *shortest path* connecting u_s and u_t. Here, the *weight* of a path is the total weight of all of its edges. A path is the shortest path if it has the minimum weight among all paths connecting u_s and u_t.

5. Let $\Pi(u_s, u_t)$ be a computed path and $\|\Pi(u_s, u_t)\|$ be the weight of the path. The path concatenating the edge su_s, path $\Pi(u_s, u_t)$, and the edge $u_t t$ is an optimum best-coverage-path with the minimum energy consumption. The best-coverage-distance is max($\|su_s\|$,β,$\|u_t t\|$). Here $\|su_s\|$ and $\|u_t t\|$ are the Euclidean distance between points.

The correctness of the algorithm is based on the following observation. Consider an edge uv of the best-coverage-path that consumes the minimum energy among all best-coverage-paths. If there is a sensor node w inside $disk(u, v)$, then $\|wu\| \leq \|uv\|$ and $\|wv\| \leq \|uv\|$. It is obvious that the path uwv is in the residue graph G. Thus, the path by substituting edge uv with edges uw and wv is still a best-coverage-path and consumes less energy, which is a contradiction. Consequently, edge uv must be a Gabriel edge.

The time complexity of the above algorithm is $O(n \log n)$ if it is implemented using a centralized manner. The total communication cost by all wireless nodes of the above algorithm is $O(n \log n)$ bits if it is implemented in a distributed manner. Here, we assume that we use a synchronized distributed algorithm to construct the shortest path between two given wireless nodes.

C.2 Travel Distance

The second extension is to find a path with the best-coverage-distance with the total length of the edges of this path is not more than 5/2 times the shortest path among all optimum best-coverage-paths. It is well-known that the relative neighborhood graph and the Gabriel graph are not spanners [24], [10]. Although Delaunay triangulation is a spanner [25], [26], but we know that it cannot be always constructed efficiently in a distributed manner. Thus, we have to use some other geometry structure that is a spanner and can be constructed efficiently. Recently, Li *et al.* [8] proposed such a geometry structure, namely *local Delaunay triangulation*, and give efficient distributed algorithm to construct it.

Triangle $\triangle uvw$ is called a *k-localized Delaunay triangle* if the interior of the circumcircle of $\triangle uvw$, denoted by $disk(u, v, w)$ hereafter, does not contain any vertex of V that is a k-neighbor of u, v, or w; and all edges of the triangle $\triangle uvw$ have length no more than one unit. The *k-localized Delaunay graph* over a vertex set V, denoted by $LDel^{(k)}(V)$, has exactly all Gabriel edges and edges of all k-localized Delaunay triangles. We then review the algorithm proposed in [8].

Algorithm 3: Localized Unit Delaunay Triangulation

1. Each wireless node u broadcasts its identity and location and listens to the messages from other nodes.
2. Assume that every wireless node u gathers the location information of $N_1(u)$. Node u computes the Delaunay triangulation $Del(N_1(u))$ of its 1-neighbors $N_1(u)$, including u itself.
3. Node u finds all Gabriel edges uv and marks them as *Gabriel edges*. Notice that here $\|uv\| \leq 1$.
4. Node u finds all triangles $\triangle uvw$ such that all three edges of $\triangle uvw$ have length at most one unit. If angle $\angle wuv \geq \frac{\pi}{3}$, node u broadcasts a proposal to form a 1-localized Delaunay triangle $\triangle uvw$ in $LDel^{(1)}(S)$ and listens to the messages from other nodes.

5. Node v accepts the proposal of constructing $\triangle uvw$ if $\triangle uvw$ belongs to the Delaunay triangulation $Del(N_1(v))$; otherwise, it rejects the proposal. Similarly does node w.

6. Node u accepts the triangle $\triangle uvw$ if both nodes v and w accept the proposal. Similarly do node v and w.

It is shown in [8] that the graph constructed by the above approach has linear number of links but not necessarily a planar graph. They also gave efficient method to make this graph planar and denoted the final planarized graph by $PLDel(S)$. They proved that $PLDel(S)$ contains $UDel(S)$ as a subgraph. Thus, $PLDel(S)$ is a planar t-spanner of $UDG(S)$.

Algorithm 4: Planarize $LDel^{(1)}(S)$

1. Each wireless node u broadcasts the Gabriel edges incident on u and the triangles $\triangle uvw$ of $LDel^{(1)}(S)$ and listens to the messages from other nodes.

2. Assume node u gathered the Gabriel edge and 1-local Delaunay triangles information of all nodes from $N_1(u)$. For two intersected triangles $\triangle uvw$ and $\triangle xyz$ known by u, node u removes the triangle $\triangle uvw$ if its circumcircle contains a node from $\{x, y, z\}$.

3. Each wireless node u broadcasts all remaining triangles incident on u and listens to the broadcasting by other nodes.

4. Node u keeps triangle $\triangle uvw$ if both v and w have triangle $\triangle uvw$ remained.

The above method constructs the local Delaunay triangulation $LDel(S)$ using optimum $O(n)$ communications. Here the communication unit is $O(\log n)$ bits, which is the number of bits representing a node ID. They also show that the number of edges in $PLDel(S)$ is $O(n)$ by proving that $PLDel(S)$ is indeed a planar graph. We then propose our algorithm that finds a path with the best-coverage-distance and the length of this path is not more than 5/2 times the shortest path among all optimum best-coverage-paths.

Algorithm 5: SmallTravellingBestCoverage(S, Ω, s, t)

1. Run a distributed *shortest path* algorithm to compute the *coverage distance* of the best-coverage-path connecting u_s and u_t. Let β be the best coverage distance.

2. Construct the local Delaunay triangulation. Prune out all edges of the local Delaunay triangulation $PLDel(S)$ with weight larger than β, and call the remaining graph the residue graph G.

3. Assign each edge uv of the residue graph G the weight equal to $\|uv\|$.

4. Run a distributed *shortest path* algorithm to compute the *shortest path* connecting u_s and u_t. Here, the *weight* of a path is the total weight of all of its edges. A path is the shortest path if it has the minimum weight among all paths connecting u_s and u_t.

5. Let $\Pi(u_s, u_t)$ be a computed path and $\|\Pi(u_s, u_t)\|$ be the length of the path. The path concatenating the edge su_s, path $\Pi(u_s, u_t)$, and the edge $u_t t$ is an optimum best-coverage-path with small traveling distance. The best-coverage-distance is $\max(\|su_s\|, \beta, \|u_t t\|)$. Here $\|su_s\|$ and $\|u_t t\|$ are the Euclidean distance between points.

Hence the communication complexity of the above algorithm is $O(n \log n)$. The correctness of the algorithm is based on the following observation. Consider an edge uv of the best-coverage-path that has the minimum total edge lengths among all best-coverage-paths. It is proved in [7] that if edge uv is not in the localized Delaunay triangulation $PLDel(S)$, then there exists a path $\Pi(u, v)$ in $PLDel(S)$ such that all edges of $\Pi(u, v)$ are shorter than uv and the total length of all edges of $\Pi(u, v)$ is no more than $\frac{4\sqrt{3}\pi}{9}\|uv\|$. It is obvious that the path $\Pi(u, v)$ is in the residue graph G. Consequently, the shortest path in the residue graph G has length no more than $\frac{4\sqrt{3}\pi}{9}$ (which is less than 5/2) factor of the length of the shortest best-coverage-path in the unit disk graph $UDG(S)$.

Here we have a trade-off between the quality of performance and the time-complexity. If the graph $UDG(S)$ is used, we get the shortest best-coverage-path but the communication complexity of the algorithm is $O(m \log n)$, where m is the number of edges in $UDG(S)$ which could be as large as $O(n^2)$. On the other hand, our algorithm approximates the shortest best-coverage-path with total communication cost $O(n \log n)$.

IV. Algorithm Correctness

This section is devoted to study the correctness of Algorithm 1. Given two points s and t, let $b_{s,t}$ be the smallest radius r such that points s and t are connected inside the region $D(S, r)$. Let $D(s_{i_1}, r)$, $D(s_{i_2}, r)$, $\cdots$, $D(s_{i_k}, r)$ be the sequence of disks centered at sensor nodes traveled by a path connecting s and t. Then, obviously, the following path starting from s, then using the path $s_{i_1} s_{i_2} \cdots s_{i_k}$, and finally ending at t has the same optimum best-coverage-distance (i.e., the radius r) as any optimum best-coverage-path. See Figure 6 for an illustration. This implies the following lemma.

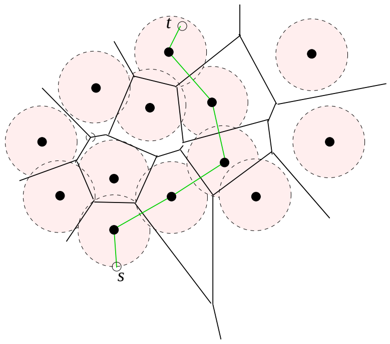

Fig. 6. There is a path inside the region covered by all disks that connects the source point s and the destination point t.

Lemma 2: There is an optimum best-coverage-path that uses only the following edges: the edges of the unit disk graph $UDG(S)$, the edges by connecting s to every sensor node, and the edges by connecting t to every sensor node.

We show that it is sufficient to consider only the edges su_s and $u_t t$, where u_s and u_t are the closest sensor nodes to s and t, respectively. Notice that Meguerdichian, *et al.* [1] had already applied this approach. We just give a formal proof here.

Lemma 3: There is an optimum best-coverage-path that connects s to its closest sensor node u_s and connects t to its closest sensor node u_t.

Proof: Consider an optimum path that does not connect s to its closest sensor node u_s. Assume that s is connected to a node v. We concentrate on the edge sv. We construct an alternative subpath connecting s and v using the edge su_s. Without loss of generality, let $u_0 = u_s$, u_1, u_2, $\cdots$, u_{m-1}, $u_m = v$ be the vertices corresponding to the sequence of Voronoi regions traversed by walking from s to v along the segment sv. See Figure 7. If a Voronoi edge or a Voronoi vertex happens to lie on the segment sv, then choose the Voronoi region lying above sv. Assume that line sv is the x-axis. The sequence of vertices u_i, $0 \leq i \leq m$, defines a path from u_s to v. In general, we refer to the path constructed this way between some nodes u_s and v as the *direct DT path* from u_s to v, denoted by $DT(u_s, v)$, which is also used by [25].

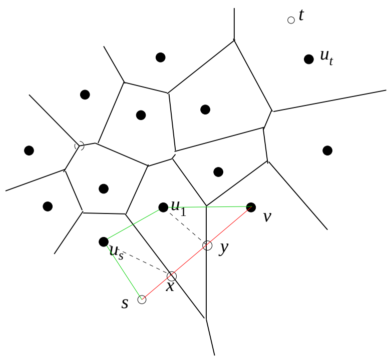

Fig. 7. Connect the starting point s to its closest sensor node u_s.

Then we show that the path, denoted by $DT(s, v)$, consisting of edge su_s and the direct DT path $DT(u_s, v)$ from u_s to v is not worse than the edge sv in terms the coverage-distance. Figure 8 illustrates the proof that follows. Let x_i denote the point on the x-axis

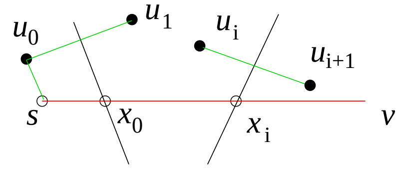

Fig. 8. Proof of that point s is connected to its closest sensor node u_s.

that also lies on the boundary between the Voronoi regions $Vor(u_i)$ and $Vor(u_{i+1})$ for $i = 0, 1, \cdots, m-1$. The definition of the Voronoi diagram immediately gives that the circle centered at x_i passing through the vertices u_i and u_{i+1} contains no points of S in its interior. We denote such circle as C_i, i.e., $C_i = disk(x_i, \|x_i u_i\|)$. For each point x_i on the subpath $\Pi(s, v)$, its coverage-distance is exactly $\|x_i u_i\|$. Consequently, the coverage-distance of the edge sv is at least (by considering only the point s and all points x_i, $0 \leq i < m$)

$$\max(\|su_0\|, \max_{0 \leq i < m}(\|x_i u_i\|)).$$

Notice that the coverage-distance of any edge $u_i u_{i+1}$, $0 \leq i < m$, is at most $\frac{1}{2}\|u_i u_{i+1}\|$; the coverage distance of the edge su_0 is exactly $\|su_0\|$. Consequently, the coverage-distance of the subpath $DT(s, v)$ is at most

$$\max(\|su_0\|, \max_{0 \leq i < m}(\frac{1}{2}\|u_i u_{i+1}\|)).$$

The definition of the Voronoi region immediately implies that $\|x_i u_i\| \geq \frac{1}{2}\|u_i u_{i+1}\|$. Consequently, the coverage-distance of the subpath $DT(s, v)$ is at most as large as the coverage-distance of the edge sv. Substituting the subpath $\Pi(s, v)$ by the subpath $DT(s, v)$ gives an optimum best-coverage-path that connects s to its closest sensor node $u_s = u_0$. Then, the lemma follows. ■

For simplicity, from now on, we will not consider the starting point s and the ending point t. Instead, we must only determine the best-coverage-path connecting a pair of sensor nodes. As shown by Lemma 2, the search can be confined to the paths in the unit disk graph $UDG(S)$. However, the unit disk graph $UDG(S)$ may have too many edges, which in the worst case could be as large as $O(n^2)$. We then show that the search space of the best covering problem can be further confined to the Delaunay triangulation $Del(S)$ of the set S of sensors. Notice that the algorithm given in [1] uses this approach without the justification of its correctness. We prove this by the following lemma.

Lemma 4: There is an optimum best-coverage-path that uses only the edges of the Delaunay triangulation $Del(S)$.

Proof: Consider any optimum best-coverage-path connecting two sensor nodes. We show that there is another optimum best-coverage-path such that every edge uv in the path is a Delaunay edge. Remember that an edge uv is a Delaunay edge if and only if the Voronoi regions $Vor(u)$ and $Vor(v)$ share some common Voronoi edge. Consider any edge uv of an optimum best-coverage-path. We show that the direct DT path $DT(u, v)$ has a coverage-distance at most of that of uv. The proof is similar to the proof of Lemma 3. Without loss of generality, let $b_0 = u$, b_1, b_2, $\cdots$, b_{m-1}, $b_m = v$ be the vertices corresponding to the sequence of Voronoi regions traversed by walking from u to v along the segment uv. See Figure 9. Let x_i denote the point on the x-axis that also lies on the boundary between the Voronoi regions $Vor(b_{i-1})$ and $Vor(b_i)$ for $i = 1, 2, \cdots, m$. Then $\|x_i b_i\| \geq \frac{1}{2}\|b_i b_{i+1}\|$. Consequently, the coverage-distance of the subpath $DT(u, v)$, which is at most $\max_{1 \leq i \leq m} \frac{1}{2}\|b_{i-1} b_i\|$, is at most as large as the coverage-distance of the

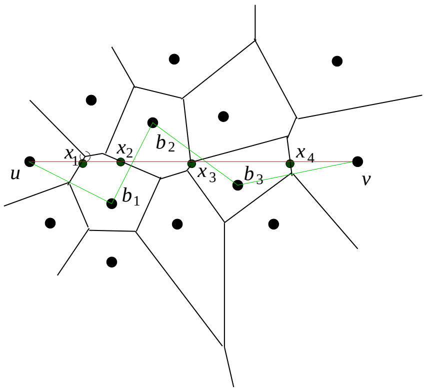

Fig. 9. There is an optimum best-path that uses only the edges of the Delaunay triangulation.

edge uv, which is at least $\max_{1\leq i\leq m} \|x_i b_i\|$. Notice that every edge $b_{i-1}b_i$ of the direct DT path $DT(u,v)$ is a Delaunay edge because the Voronoi regions $Vor(u_i)$ and $Vor(u_{i+1})$ are adjacent. Thus, the lemma follows. ■

The following lemma shows that we can confine our search space to a much smaller graph $GG(S)$, which can be constructed in a distributed manner efficiently.

Lemma 5: There is an optimum best-coverage-path that uses only the edges of the Gabriel graph $GG(S)$.

Proof: The following Figure 10 illustrates the proof that follows. Similar to the proof

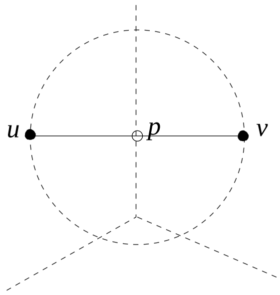

Fig. 10. There is an optimum best-coverage-path that uses only the edges of the Gabriel graph.

of Lemma 4, we know that there is an optimum best-coverage-path connecting any two sensor nodes such that every edge uv of the path only intersects the Voronoi edge shared by $Vor(u)$ and $Vor(v)$. Let p be the midpoint of the segment uv. Then the circle centered at p with radius $\frac{1}{2}\|uv\|$ is empty. It implies that the edge uv is an Gabriel edge. The lemma then follows. ■

Actually we can further confine our search space based on the following lemma.

Lemma 6: There is an optimum best-coverage-path that uses only the edges of the relative neighborhood graph $RNG(S)$.

Proof: The following Figure 11 illustrates the proof that follows. Consider an opti-

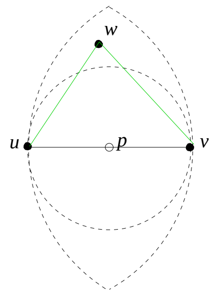

Fig. 11. There is an optimum best-coverage-path that uses only the edges of the relative neighborhood graph.

mum best-coverage-path using edges of the Gabriel graph $GG(S)$. Consider any edge uv of this path. Assume that the $lune(u, v)$ contains a sensor w from S in the interior. Then node w can not be inside the circle $disk(u, v)$ because uv is a Gabriel edge. Thus, the coverage-distance of the midpoint p of edge uv is exactly $\frac{1}{2}\|uv\|$. Notice that the coverage-distance of edge uw is at most[1] $\frac{1}{2}\|uw\|$, and the coverage-distance of edge wv is at most $\frac{1}{2}\|wv\|$. Thus, the coverage-distance of the subpath uwv is at most $\max(\frac{1}{2}\|uw\|, \frac{1}{2}\|wv\|) \leq \frac{1}{2}\|uv\|$. Consequently, substituting the edge uv by the subpath uwv will not increase the coverage-distance of the optimum best-coverage-path. It then follows that there is an optimum best-coverage-path such that it only uses the edges of the relative neighborhood graph $RNG(S)$. ■

[1]It is possible that there is some other sensor node inside the circle $disk(u, w)$. Thus, we can only claim that it is at most $\frac{1}{2}\|uw\|$.

Since there is no sensor node inside the disk using a Gabriel edge uv as diameter, the coverage-distance of the Gabriel edge uv is exactly $\frac{1}{2}\|uv\|$, which is achieved at the midpoint of the edge uv. For an edge uv of the relative neighborhood graph, same reasoning holds.

V. Conclusion

We discussed efficient algorithms to find a path with maximum observability under a general assumption of the sensing model. We proved that it is the same as the best coverage problem, which can be solved by an efficient distributed algorithm using the relative neighborhood graph. In addition, we considered some extensions of the best coverage problem: to find a path with the best-coverage-distance while the total energy consumed by this path is minimized among all optimum best-coverage-paths; to find a path with the best-coverage-distance with the total length of edges of this path is no more than 2.5 times the shortest best-coverage-path. We gave efficient distributed algorithms for both extended problems. We also justified the correctness of the algorithm proposed by Meguerdichian, *et al.* [1] using the Delaunay triangulation to solve the best coverage problem. Moreover, we showed that the search space of the best coverage problem can be confined to the relative neighborhood graph, which can be constructed locally.

References

[1] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *IEEE INFOCOM'01*, 2001, pp. 1380–1387.

[2] S. Capkun, M. Hamdi, and J.P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proc. Hawaii Int. Conf. on System Sciences*, 2001.

[3] Seapahn Meguerdichian, Farinaz Koushanfar, Gang Qu, and Miodrag Potkonjak, "Exposure in wireless ad-hoc sensor network," in *IEEE MOBICOM'01*, 2001, pp. 139–150.

[4] S. Fortune, "Voronoi diagrams and Delaunay triangulations," in *F. K. Hwang and D. -Z. Du, editors, Computing in Euclidean Geometry.* 1992, pp. 193–233, World Scientific, Singapore.

[5] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[6] Franco P. Preparata and Michael I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, 1985.

[7] Xiang-Yang Li, "Localized Delaunay triangulation is as good as unit disk graph," *Journal of Computer Networks*, 2002, Submitted for publication.

[8] Xiang-Yang Li, G. Calinescu, and Peng-Jun Wan, "Distributed construction of planar spanner and routing for ad hoc wireless networks," in *The 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002, vol. 3.

[9] Xiang-Yang Li, Peng-Jun Wan, and Yu Wang, "Power efficient and sparse spanner for wireless ad hoc networks," in *IEEE International Conference on Computer Communications and Networks (ICCCN01)*, 2001, pp. 564–567.

[10] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Ophir Frieder, "Sparse power efficient topology for wireless networks," in *IEEE Hawaii Int. Conf. on System Sciences (HICSS)*, 2002.

[11] Godfried T. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

[12] J.W. Jaromczyk and G.T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of IEEE*, vol. 80, no. 9, pp. 1502–1517, 1992.

[13] K.R. Gabriel and R.R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, pp. 259–278, 1969.

[14] K. J. Supowit, "The relative neighborhood graph, with an application to minimum spanning trees," *Journal of Associate Computing Machine*, , no. 30, 1983.

[15] J. W. Jaromczyk and M. Kowaluk, "Constructing the relative neighborhood graph in three-dimensional Euclidean space," *Discrete Applied Mathematics*, , no. 31, pp. 181–192, 1991.

[16] J. W. Jaromczyk, M. Kowaluk, and France Yao, "An optimal algorithm for constructing β-skeletons in l_p metric," *SIAM Journal of Computing*, 1991.

[17] D. W. Matula and R. R. Sokal, "Properties of Gabriel graphs relevant to geographical variation research and the clustering of points in the plane," *Geographical Analysis*, , no. 12, pp. 205–222, 1984.

[18] M. Marengoni, B. A. Draper, A. Hanson, and R. A. Sitaraman, "System to place observers on a polyhedral terrain in polynomial time," *Image and vision computing*, vol. 18, pp. 773–780, 1996.

[19] W.W. Gregg, W.E. Esaias, G.C. Feldman, Frouin R, S.B. Hooker, C.R. McClain, and R.H. Woodward,

"Coverage opportunities for global ocean color in a multimission era," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, pp. 1620–1627, 1998.

[20] A. Molina, G.E. Athanasiadou, and A.R. Nix, "The automatic location of base stations for optimized cellular coverage: a new combinatorial approach," in *IEEE 49th Vehicular Technology Conference*, 1999, vol. 1, pp. 606–610.

[21] P. Hall, *Introduction to the Theory of Coverage Processes*, Wiley, New York, 1998.

[22] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W. M. McEneaney, G. Yin, and Q. Zhang (Eds.)*, 1998.

[23] T. J. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press and McGraw-Hill, 1990.

[24] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick, "On the spanning ratio of Gabriel graphs and beta-skeletons," in *Proceedings of the Latin American Theoretical Infocomatics (LATIN)*, 2002.

[25] D.P. Dobkin, S.J. Friedman, and K.J. Supowit, "Delaunay graphs are almost as good as complete graphs," *Discrete Computational Geometry*, 1990.

[26] J. M. Keil and C. A. Gutwin, "Classes of graphs which approximate the complete Euclidean graph," *Discrete Computational Geometry*, vol. 7, 1992.