International Journal of Computational Geometry & Applications

EFFICIENT CONSTRUCTION OF LOW WEIGHTED BOUNDED DEGREE PLANAR SPANNER

XIANG-YANG LI

Department of Computer Science, Illinois Institute of Technology
10 W. 31st Street, Chicago, Illinois 60616, USA
xli@cs.iit.edu

YU WANG

Department of Computer Science, Illinois Institute of Technology
10 W. 31st Street, Chicago, Illinois 60616, USA
wangyu1@iit.edu

Received (received date)
Revised (revised date)
Communicated by (Name)

Given a set V of n points in a two-dimensional plane, we give an $O(n \log n)$-time centralized algorithm that constructs a planar t-spanner for V, for $t = \rho(\alpha) = \max\{\frac{\pi}{2}, \pi \sin \frac{\alpha}{2} + 1\} \cdot C_{del}$, such that the degree of each node is bounded from above by $19 + \lceil \frac{2\pi}{\alpha} \rceil$, where $0 < \alpha < \pi/2$ is an adjustable parameter. Here C_{del} is the spanning ratio of the Delaunay triangulation, which is at most $\frac{4\sqrt{3}}{9}\pi$. We also show, by applying the greedy method in [14], how to construct a low weighted bounded degree planar spanner with spanning ratio $\rho(\alpha)^2(1+\epsilon)$ and the same degree bound, where ϵ is any positive real constant. Here, a structure is called *low weighted* if its total edge length is proportional to the total edge length of the Euclidean minimum spanning tree of V. Moreover, we show that our method can be extended to construct a planar bounded degree spanner for unit disk graphs with the adjustable parameter α satisfying $0 < \alpha < \pi/3$. Previously, only centralized method [6] of constructing bounded degree planar spanner is known, with degree bound 27 and spanning ratio $t \simeq 10.02$. The distributed implementation of this centralized method takes $O(n^2)$ communications in the worst case. Our method can be converted to a localized algorithm where the total number of messages sent by all nodes is at most $O(n)$.

Keywords: Spanner; bounded degree; low weight; planar; localized algorithm.

1. Introduction

Let $d_G(u, v)$ be the total edge weight of the shortest path in an edge weighted graph G connecting two vertices u and v. Given a set of points V in a two-dimensional plane, a subgraph $G = (V, E)$ of a graph H is a t-spanner of H if for any two nodes u and v, we have $d_G(u, v) \leq t \cdot d_H(u, v)$. In this paper, we consider the case when the weight of an edge is the Euclidean distance between its two endpoints. When H is the complete graph, we simply say that G is a t-spanner. A graph G is *sparse* if it has only $O(n)$ edges. If the total edge length of G is within a constant factor of the Euclidean minimum spanning tree of V, then G is *low weighted*. Many algorithms are known that compute sparse t-spanners with some additional properties such as bounded node degree, small spanner diameter (i.e., any two points are connected by a t-spanner path consisting of only a small number of edges), low weight, and fault-tolerance, see, e.g., [1,2,3,8,10,14,19,21]. All these algorithms compute t-spanners for any given constant $t > 1$.

We consider how to construct planar spanners for a set of two-dimensional points or a unit disk graph. Here a unit disk graph is a graph which has an edge uv if and only if the Euclidean distance $\|uv\|$ between u and v is less than one unit. It is known that the relative neighborhood graph [4,15] and Gabriel graph [4,12,13] are not spanners, while the Delaunay triangulation [11,16,17] is a t-spanner where t is a constant upper bounded by $\leq \frac{4\sqrt{3}}{9}\pi$. For the convenience of our notations, we use C_{del} to denote the spanning ratio of the Delaunay triangulation. Das and Joseph [9] showed that the minimum weight triangulation and the greedy triangulation are t-spanners for some constant t. Levcopoulos and Lingas [18] showed, for any real number $r > 0$, how to construct a planar t-spanner from the Delaunay triangulation, whose total edge length is at most $2r + 1$ times the weight of a minimum spanning tree of V, where $t = (1 + 1/r)C_{del}$. Notice that all these structures could have unbounded node degree.

Recently Bose *et al.* [6] proposed a centralized $O(n \log n)$-time algorithm that constructs a planar t-spanner for a given node set V, for $t = (1 + \pi) \cdot C_{del} \simeq 10.02$, such that the node degree is bounded from above by 27. As far as we know, this algorithm is the first method to compute a planar spanner of bounded degree.

In this paper, we give a simpler method to construct a bounded degree planar t-spanner. The bounds achieved here are better than those by Bose *et al.* [6]. In addition, we show, by applying the greedy method in [14], how to construct a low weighted bounded degree planar spanner. The main results of this paper are summarized in the following theorems.

Theorem 1. *There is an $O(n \log n)$-time centralized algorithm that, given a set V of n points in a two-dimensional plane and a real constant $0 < \alpha \leq \pi/2$, constructs a graph*

(1) that is planar,
(2) that is a t-spanner, for $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$,
(3) in which each point of V has degree at most $19 + \lceil\frac{2\pi}{\alpha}\rceil$.

For our convenience of notations, hereafter, let $\rho(\alpha) = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$.

Theorem 2. *There is an $O(n \log n)$-time centralized algorithm that, given a set V of n points in a two-dimensional plane and two real constants $0 < \alpha \leq \pi/2$ and $\epsilon > 0$, constructs a graph*

(1) that is planar,
(2) that is a t-spanner, for $t = \rho(\alpha)^2 \cdot (1 + \epsilon)$,
(3) in which each point of V has degree at most $19 + \lceil\frac{2\pi}{\alpha}\rceil$,
(4) and whose total edge length is bounded from above by a constant factor times the total edge length of the Euclidean minimum spanning tree of V. Here the constant factor depends on ϵ.

We also extend our method to unit disk graph and get the following result.

Theorem 3. *There is an $O(n \log n)$-time centralized algorithm that, given a connected unit disk graph $UDG(V)$ of a set V of n points in a two-dimensional plane and a real constant $0 < \alpha \leq \pi/3$, constructs a graph*

(1) that is planar,
(2) that is a t-spanner of $UDG(V)$, for $t = \rho(\alpha)$,
(3) in which each point of V has degree at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.

The rest of the paper is organized as follows. In Section 2, we propose a method constructing bounded degree planar t-spanner with low weight for a two-dimensional point set. In Section 3, we extend our method to construct a bounded degree planar t-spanner for a unit disk graph defined over a two-dimensional point set. The degree bound is larger than that achieved by the method for point set, but the spanning ratio is smaller. Moreover, this centralized method can be converted to a localized algorithm, which can be used for wireless networks. We conclude our paper in Section 4.

2. Bounded Degree and Planar Spanner on Point Set

2.1. *Priori Arts*

Our algorithms borrow some idea from the algorithm by Bose *et al.* [6]. For the completeness of presentation, we briefly review the basic steps of their algorithm as follows.

(1) Compute the Delaunay triangulation of V, $Del(V)$, and a degree-3 spanning subgraph $BDS(V)$ of $Del(V)$ that includes the convex hull $CH(V)$ of V. This graph $BDS(V)$ partitions $CH(V)$ into (possibly degenerate) simple polygons, such that each node of V is on the boundary of at most three polygons.
Notice that, the Euclidean minimum spanning tree is a degree-5 spanning subgraph of $Del(V)$. They use degree-3 spanning subgraph $BDS(V)$ to improve the degree bound achieved by their structure.
(2) For each polygon P in the above partition, their algorithm first orders the nodes of the polygon according to a geometry based breadth-first search (BFS), and processes the nodes of P in increasing order. It prunes this part of the Delaunay triangulation (edges inside P or on the boundary of P) such that each node of P has low degree. The resulting graph is a planar spanner for the nodes of P in the sense that any two nodes u and v of P are connected by a path whose length is at most a constant times the length of a shortest path between u and v that is *completely contained* in P. By combining all the spanners for each of the polygons, they obtain a planar spanner of bounded degree.
(3) Run a greedy algorithm by Gudmundsson *et al.* [14] on the planar spanner with bounded degree to bound the total weight of the graph.

They show that the length stretch factor of the final graph is $(1+\epsilon)(\pi+1)C_{del}$ [a] and node degree is at most 27. The running time of their algorithm is $O(n \log n)$. However, their method is not suitable to be converted into a localized one even efficient distributed version, since they used BFS and other operations on polygons (such as degree-3 partitions). Here, a distributed algorithm constructing a graph G is a *localized algorithm* if every node u can exactly decide all edges of G incident on u based only on the information of all nodes within a constant hops of u. Notice that the breadth-first-search may use $O(n^2)$ communications in distributed or localized algorithms. Here, the total communication cost of a distributed or localized algorithm is the total number of messages sent by all nodes under broadcasting communication model. In this section, we will give a new method for constructing a planar spanner with bounded node degree for a point set V. The basic idea of our methods is to combine the Delaunay triangulation and the ordered Yao structure [5].

Given any edge-weighted graph $G = (V, E, w)$, greedy method has long been known to construct a t-spanner H for G. It works as follows: (1) sort all edges in G incrementally according to its weight; (2) initialize H to be an empty graph; (3) process all edges in order of their weights, and an edge e is added to H if there is no path in H with weight at most $t \cdot w(e)$. A simple implementation of this greedy method is expensive due to the query cost for each edge. For complete geometry graph, Gudmundsson *et al.* [14,10] showed how this greedy method can be implemented in time $O(n \log n)$ under the indirect addressing model.

Notice that, given a point set V, the greedy algorithm by Gudmundsson *et al.* [14] has a preprocessing step that helps to eliminate all but a linear number of edges from further consideration. This step generally will compute a sparse spanner G with spanning ratio $t/\sqrt{t \cdot t'}$, which will then be used by the greedy method. Given a sparse spanner $G = (V, E)$ for the complete graph, their main contribution is an $O(n \log n)$ time method to construct a spanner H of G using greedy approach. They use $\sqrt{t \cdot t'}$ as the parameter in the greedy method. To guarantee that the total edge length of H is within a constant factor of the minimum spanning tree of the point set V, they need the follow condition: there exists a constant ϕ such that $t \geq t' \geq \phi t + 1 - \phi > 1$. They proved that the edges of the graph constructed by their method satisfy the (t', t)-leapfrog property, which in turn implies that the resulted graph is low-weighted when $t \geq t' \geq \phi t + 1 - \phi > 1$.

2.2. *Our Algorithm*

We then present our method of constructing low weighted bounded degree planar spanner for a two-dimensional point set V. We assume that every node u has a unique ID denoted by $ID(u)$.

Algorithm: Constructing Bounded Degree Planar Spanner with Low Weight

(1) Compute the Delaunay triangulation of a set V of n nodes, $Del(V)$. Let $N_{Del}(u)$ be

[a] We suspect the correctness of this bound when low-weighted bounded degree planar spanner is needed. As will see later, to use the method by Gudmundsson *et al.* [14] to bound the weight of the graph, the spanning ratio achieved should be modified to $((\pi+1)C_{del})^2 \cdot (1+\epsilon)$.

the neighbors of a node u in the Delaunay triangulation, and let d_u be the degree of node u in $Del(V)$. $N_{del}(u)$ and d_u can easily be computed in linear time since the Delaunay triangulation only has a linear number of edges.

(2) Find an order π of V as follows. Let $G_1 = Del(V)$ and $d_{G,u}$ be the degree of u in a graph G. Remove the node u with the smallest value of $(d_{G_i,u}, ID(u))$ and all its incident edges from G_i. Assign $\pi_u = n - i + 1$, and call the remaining graph G_{i+1}. Repeat the above procedure for $1 \leq i \leq n$.
Let P_v denote the predecessors of v in π, i.e., $P_v = \{u \in V : \pi_u < \pi_v\}$. Notice that since G_i is a planar graph, we know that the smallest value of $d_{G_i,u}$ is at most 5. Then, for ordering π, node u has at most 5 edges to its predecessors P_u in $Del(V)$.

(3) Let E be the edge set of $Del(V)$, and let E' be the edge set of the desired spanner. Initialize E' to be the empty set and all nodes in V are marked as *unprocessed*. Then, for each node u in V, following the increasing order of π, run the following steps to add some edges from E to E' (we only consider Delaunay neighbors $N_{Del}(u)$ of u):

   (a) We use $v_1, v_2, \cdots, v_k$ to denote the predecessors of node u (see Figure 1). Notice that u can have at most 5 edges to its predecessors (already processed Delaunay neighbors), i.e., $k \leq 5$. Then there are $k \leq 5$ *open* sectors at node u whose boundaries are rays emanated from u to the processed neighbors v_i of u in $Del(V)$. For each such sector apexed at u, we divide it into a minimum number of *open* cones of degree at most α, where $\alpha \leq \pi/2$ is a parameter.

   (b) For each such cone, let $s_1, s_2, \cdots, s_m$ be the geometrically ordered[b] neighbors $N_{Del}(u)$ of u in this cone. That is, $s_1, s_2, \cdots, s_m$ are all *unprocessed* nodes that are connected by some edges of E to u in this cone. For this cone, we first add the shortest edge in E that is connected to u to the edge set E', then add to E' all the edges $s_j s_{j+1}$, $1 \leq j < m$. Notice that from the Delaunay triangulation definition, $s_j s_{j+1} \in E$.

   (c) Mark node u *processed*.

   Repeat this procedure in the increasing order of π, until all nodes are marked as *processed*. The final graph formed by edges E' is denoted by $BPS(V)$.

(4) If a low weighted structure is also needed, run the greedy spanner algorithm by Gudmundsson *et al.* [14] to bound the weight of the graph $BPS(V)$. Let $LBPS(V)$ be the resulted low-weighted structure. We will set $\sqrt{tt'} = \rho(\alpha)(1+\epsilon)$ as the input parameter of the greedy method.

Notice that in the algorithm we use *open* sectors, which means that we do not consider adding the edges on the boundaries (any edge involved previously processed neighbors). For example, in Figure 1, the cones do not include any edges uv_i. This guarantees that the algorithm does not add any edges to node v_i after v_i has been processed. This approach, as we will show it later, bounds the node degree.

[b]Here geometrically ordered means links $us_1, us_2, \cdots, us_m$ are clockwise or anticlockwise distributed around node u. In Figure 1, s_1, s_2, s_3 are anticlockwise distributed in the cone defined by v_4 and v_5.

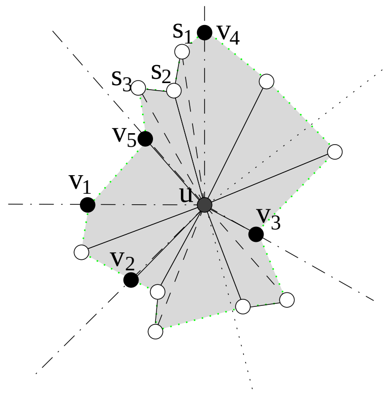

Fig. 1. Constructing planar spanner with bounded degree for a point set: process node u.

2.3. *Analysis of Algorithm*

A simple proof by induction can show that the final graph $BPS(V)$ is connected. This is also implied by the later theorem of spanner property, thus, we ignore the proof here.

We then show that the degree of $BPS(V)$ is bounded by a constant.

Theorem 4. *The maximum node degree of the graph $BPS(V)$ is at most $19 + \lceil \frac{2\pi}{\alpha} \rceil$.*

PROOF. Notice that for a node u there are 2 cases that an edge uv is added to $BPS(V)$.

Case 1: When we process node u, some edges uv have already been added by some processed nodes w before. There are two subcases for this case.

Subcase 1.1: The edge uv has been added by the processed node v (i.e., here $w = v$). For example, in Figure 1, node u has edges from v_2, v_3 and v_5 before it is processed. For each predecessor v, it only adds one such edge to node u.

Subcase 1.2: The edge uv has been added by a processed node w (here w is not v). In this case, node v must be an unprocessed node when processing w. For example, in Figure 1, node s_2 has edges from s_1 and s_3 added by processing node u before node s_2 is processed. Notice that both v and u are neighbors of this processed node w. For each predecessor w, it at most adds two such edges to node u.

Remember that node u can have at most 5 predecessor neighbors (processed neighbors), and each of predecessor neighbors can add at most 3 edges to it (either Subcase 1.1 or Subcase 1.2, or both). Thus, the number of this kind of edges (edges added by its predecessors before node u is processed) is bounded by 15.

Case 2: When node u is processed, we can add one edge uv for each of the partitioned cones. Since we have at most 5 sectors emanated from u and each cone must have angle at most α, it is easy to show that we have at most $4 + \lceil \frac{2\pi}{\alpha} \rceil$ cones at u. So the number of edges uv added when processing u is bounded by $4 + \lceil \frac{2\pi}{\alpha} \rceil$.

Notice that after node u is processed, no edges will be added to it. Consequently, the degree of each node u is bounded by $19 + \lceil \frac{2\pi}{\alpha} \rceil$ in the final structure. $\square$

For example, when $\alpha = \pi/2$, then the maximum node degree is at most 23; when $\alpha = \pi/3$, then the maximum node degree is at most 25. Either case improves the previous bound 27 on the maximum node degree by Bose *et al.* [6].

It is trivial that $BPS(V)$ is a planar graph, since $Del(V)$ is a planar graph and the algorithm only adds the Delaunay edges to $BPS(V)$. Notice that all edges $s_i s_{i+1}$ are also in $Del(V)$ since s_i and s_{i+1} are consecutive Delaunay neighbors of node u.

Finally, we prove that the graph $BPS(V)$ is a spanner. Notice that the following theorem also implies that the graph $BPS(V)$ is connected.

Theorem 5. *The graph $BPS(V)$ is a t-spanner, where $t = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}$.*

PROOF. First, remember that $Del(V)$ is a spanner with a constant length stretch factor $C_{del} = \frac{4\sqrt{3}}{9}\pi \approx 2.42$. Keil and Gutwin [17] proved it using induction on the order of the lengths of all pair of nodes (from the shortest to the longest). We can show that the path connecting nodes u and v constructed by the method given by Keil and Gutwin [17] also satisfies that all edges of that path are shorter than $\|uv\|$. So if we can prove this claim: *for any edge $uv \in Del(V)$, there exists a path in $BPS(V)$ connecting u and v whose length is at most a constant ℓ times* $\|uv\|$, then we know $BPS(V)$ is a $\ell \cdot C_{del}$-spanner.

We then prove the above claim. Consider an edge uv in $Del(V)$. If $uv \in BPS(V)$, the claim holds. So assume that $uv \notin BPS(V)$.

Assume w.l.o.g. that $\pi_u < \pi_v$. It follows from the algorithm that, when we process node u, there must exist a node v' in the same cone with v such that $\|uv\| > \|uv'\|$, $uv' \in BPS(V)$, and $\angle v'uv < \alpha \leq \pi/2$. Let $v' = s_1, s_2, \cdots, s_k = v$ be the sequence of nodes in the ordered unprocessed neighborhood of u from v' to v.

Same with the proof by Bose *et al.* [6], consider the polygon P, consisting of nodes $u, s_1, \cdots, s_k$. We will show that the path $s_1 s_2 \cdots s_k$ has length that is at most a small constant factor of the length $\|uv\|$. Let us consider the shortest path from s_1 to s_k that is *totally inside* the polygon P. Let $S(s_1, s_k)$ denote such path. This path consists of diagonals of P and is contained inside $\triangle u s_1 s_k$. For example, in Figure 2, $S(s_1, s_k) = s_1 s_7 s_9$.

Assume that $\|uv'\| = x$. Let w be the point on segment uv such that $\|uw\| = \|uv'\|$. Assume that $\|uv\| = y$, then $\|wv\| = y - x$. Notice that node v' is the closest Delaunay neighbors in such cone. Obviously, all Delaunay neighbors s_i in this cone is outside of the triangle defined by segments uw, wv' and uv'. We will show that such path $S(s_1, s_k)$ is contained inside the triangle $\triangle w s_1 s_k$. First, if no Delaunay neighbor is inside $\triangle w s_1 s_k$, then $S(s_1, s_k) = s_1 s_k$. Thus, the claim trivially holds. If there are some Delaunay neighbors inside $\triangle w s_1 s_k$, then s_1 will connect to the one s_i forming the smallest angle $\angle u s_1 s_i$. Similarly, node s_k will connect to the one s_j forming the smallest angle $\angle u s_k s_j$. Obviously s_i and s_j are inside $\triangle w s_1 s_k$, thus, the shortest path connecting them is also inside $\triangle w s_1 s_k$ (proved by induction). Since path $S(s_1, s_k)$ is the shortest path inside the polygon P to connect s_1 and s_k, by convexity, the length of $S(s_1, s_k)$ is at most $\|v'w\| + \|wv\| = 2x \sin\frac{\theta}{2} + y - x$. Here $\theta = \angle v'uv < \alpha$.

An edge $s_i s_j$ of $S(s_1, s_k)$ has endpoints s_i and s_j in the neighborhood of u. Let $D(s_i, s_j)$ be the sequence of edges of the polygon P between s_i and s_j in the ordered neighborhood of u, which are added by processing u. For example, in Figure 2,

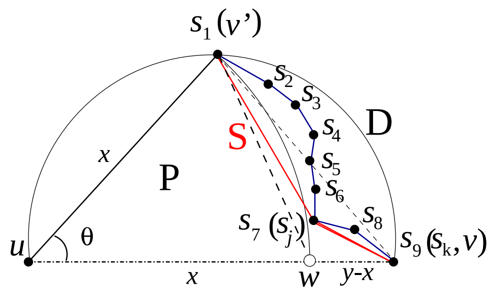

Fig. 2. The shortest path in polygon P.

$D(s_1, s_7) = s_1s_2s_3s_4s_5s_6s_7$. This path is in $BPS(V)$. We can bound the length of $D(s_i, s_j)$ by $\pi/2\|s_is_j\|$ by the argument in Ref. [6,7]. In Ref. [7], it is shown that the length of $D(s_i, s_j)$ is at most $\pi/2$ times $\|s_is_j\|$, provided that (1) the straight-line segment between s_i and s_j lies outside the Voronoi region induced by u, and (2) that the path $D(s_i, s_j)$ lies on one side of the line through s_i and s_j. In other words, we need $D(s_i, s_j)$ to be *one-sided Direct Delaunay path* [c]; See Ref. [11]. In Ref.[6], they showed [d] that both these two conditions hold when $\angle s_ius_j < \pi/2$. This is trivially satisfied since $\angle s_ius_j < \alpha \le \pi/2$.

Thus, we have a path $us_1s_2\cdots s_k$ to connect u and v with length at most

$$\begin{aligned}
&\|uv'\| + \sum_{i=1}^{k-1}\|s_is_{i+1}\| \\
\le& x + \frac{\pi}{2}\cdot S(s_1, s_k) \\
\le& x + (2x\sin\frac{\theta}{2} + y - x)\cdot\pi/2 \\
=& y\cdot(\frac{\pi}{2} + \frac{x}{y}\cdot(\pi\sin\frac{\alpha}{2} - \frac{\pi}{2} + 1)) \\
\le& y\cdot\max\{\frac{\pi}{2}, \pi\sin\frac{\alpha}{2} + 1\}
\end{aligned}$$

The last inequality comes from $0 < \frac{x}{y} \le 1$. Putting it all together, we know $BPS(V)$ is a spanner with length stretch factor at most $\max\{\frac{\pi}{2}, \pi\sin\frac{\alpha}{2} + 1\}\cdot C_{del}$. □

[c] For any pair of nodes u and v, let $u = w_1, w_2, \cdots, w_k = v$ be the sequence of nodes whose Voronoi region intersect segment uv and the Voronoi regions at w_i and w_j share a common boundary segment. Then the Direct Delaunay path $DT(u, v)$ is $w_1w_2\cdots w_k$.

[d] Firstly, the Voronoi region centered at u will not intersect the segment s_is_j. This can be proved by showing that $\|up\| > \max\{\|s_ip\|, \|s_jp\|\}$ for any point p on segment s_is_j, which is due to $\angle us_ip + \angle us_jp > \angle s_iup + \angle s_jup = \angle s_ius_j$. Notice that $\angle s_ius_j < \alpha \le \pi/2$. Secondly, the path $D(s_i, s_j)$ is on one-side of s_is_j because it is part of the shortest path connecting s_1 and s_k. Thirdly, the path $D(s_i, s_j)$ is Direct Delaunay path $DT(s_i, s_j)$. This can be proved by showing that $Vor(s_q)$ intersects the segment s_is_j for any $i \le q \le j$. This is obvious since the circumcenter (belonging to $Vor(s_q)$) of any triangle $us_{q-1}s_q$ is on the same side of s_is_j as u.

For simplicity, we denote the spanning ratio of $BPS(V)$ as

$$\rho(\alpha) = \max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot C_{del}.$$

For example, when $\alpha = \pi/2$, then the spanning ratio is at most $(\frac{\sqrt{2}\pi}{2} + 1) \cdot C_{del}$; when $\alpha = \pi/3$, then the spanning ratio is at most $(\frac{\pi}{2} + 1) \cdot C_{del}$; when $\alpha = 2\arcsin(\frac{1}{2} - \frac{1}{\pi}) \simeq 20.9^o$, then the spanning ratio is at most $\frac{\pi}{2} \cdot C_{del}$. Notice that, the method by Bose *et al.* [6] actually can achieve the same spanning ratio as this one, although they did not prove this. However, the node degree of the graph generated by our method is smaller than that by Ref. 6.

Notice that the time complexity of our centralized algorithm is $O(n \log n)$ too. We can build Delaunay triangulation in $O(n \log n)$, and do ordering in time $O(n \log n)$ (using heap for the ordering based on degrees), and Yao structure in $O(n)$ (each edge is processed at most a constant times and there are $O(n)$ edges to be processed). When using heap for the ordering, initially building a heap needs $O(n \log n)$, then we remove one node and it has at most 5 adjacent edges, it needs at most 5 times of updating the heap based on degree (each of which can be done in time $O(\log n)$). So the ordering can be done in $O(n \log n)$. Consequently, the time complexity is $O(n \log n)$, which is the same as the method by Bose *et al.* [6]. However, our algorithm has smaller node degree bound, is easier to implement, and (more importantly) has potential to become a localized version for wireless ad hoc networks application as we will describe later.

We then show that the structure $LBPS(V)$ is indeed a low-weighted bounded degree planar spanner. Notice that since we apply the fast greedy method by Gudmundsson *et al.* [14] on top of the structure $BPS(V)$ instead of their preprocessing step, the structure $LBPS(V)$ clearly is planar, and has bounded degree. Remember that, in the preprocessing step, they construct a $\frac{t}{\sqrt{tt'}}$-spanner. Since we use $BPS(V)$ to substitute their preprocessing step, we have $\frac{t}{\sqrt{tt'}} = \rho(\alpha)$. From the transition property of the spanning ratio, the spanning ratio of $LBPS(V)$ is at most $\rho(\alpha) \cdot \sqrt{tt'} = \rho(\alpha)^2(1 + \epsilon)$. In other words, in applying the method by Gudmundsson *et al.* [14], we choose $t = \rho(\alpha)^2(1 + \epsilon)$, and $t' = 1 + \epsilon$. It is easy to show that the constant $\phi = \frac{\epsilon}{\rho(\alpha)^2(1+\epsilon)-1}$ does satisfy that $1 > \phi > 0$, and $t \geq t' \geq \phi t + 1 - \phi > 1$. Consequently, the edges of $LBPS(V)$ do satisfy the (t', t)-leapfrog property and thus the total edge length of $LBPS(V)$ is within a constant factor of that of $MST(V)$. Here the constant depends on t, and t', i.e., depending on ϵ.

Notice that there is a penalty to be paid to achieve the additional low-weight property: a larger spanning ratio proved for the low-weighted bounded degree planar spanner. In addition, the $O(n \log n)$ time complexity of constructing $LBPS(V)$ only works for the indirect addressing model [14].

3. Bounded Degree and Planar Spanner on Unit Disk Graph

We consider a wireless ad hoc network (or sensor network) with all nodes distributed in a two-dimensional plane. Assume that all wireless nodes have distinctive identities and each static wireless node knows its position information either through a low-power Global Position System (GPS) receiver or through some other way. For simplicity, we also assume that all wireless nodes have the same maximum transmission range and we normalize it to one unit. By one-hop broadcasting, each node u can gather the location information of all nodes within the transmission range of u. Consequently, all wireless nodes V together define the original communication graph (a unit-disk graph $UDG(V)$), which has an edge uv if and only if the Euclidean distance $\|uv\|$ between u and v is less than one unit. Notice, throughout this paper, a *broadcast* by a node u means u sends the message to all nodes within its transmission range. In wireless ad hoc networks, the radio signal sent out by a node u can be received by all nodes within the transmission range of u. The main communication cost is to send out the signal while the receiving cost of a message is neglected here. In this section we give a centralized algorithm to construct a planar spanner with bounded degree for a connected $UDG(V)$, which can be used for wireless ad hoc networks.

3.1. *Our Algorithm*

Our method of constructing a bounded degree planar spanner for a connected unit disk graph is similar to our algorithm for a two-dimensional point set, but with the following difference: the parameter α is at most $\pi/3$ here since we have to ensure that an edge $s_i s_{i+1}$ does belong to UDG for any two nodes s_i and s_{i+1} from a cone. The proof of the planar property and the proof of the spanner property are different since we may add some edges that are not in Delaunay triangulation.

Algorithm: Constructing Planar Spanner with Bounded Degree for $UDG(V)$

(1) Same with the algorithm for point set, first, compute Delaunay triangulation $Del(V)$.
(2) Removing the edges whose length are longer than one unit in $Del(V)$. Call the remaining graph unit Delaunay triangulation, denoted by $UDel(V)$. For every node u, we find its unit Delaunay neighbors $N_{UDel}(u)$ and its node degree d_u in $UDel(V)$.
(3) Then, same with the algorithm for point set, find an order π of V as follows: Let $G_1 = UDel(V)$ and $d_{G,u}$ be the node degree of u in graph G. Remove the node u with the smallest value of $(d_{G_i,u}, ID(u))$ from G_i, let $\pi_u = n - i + 1$, and call the remaining graph G_{i+1}. Repeat this procedure for $1 \le i \le n$. Obviously, in ordering π, node u at most have 5 edges to its predecessors P_u in $UDel(V)$.
(4) Let E and E' be the edge sets of $UDel(V)$ and the desired spanner. Initialize $E' = \emptyset$ and all nodes in V are unprocessed. Then, same with the algorithm for point set, for each node u in V, following the increasing order π, run the following steps to add some edges to E':

    (a) Node u uses its predecessors (processed Unit Delaunay neighbors) in E to define at most 5 *open* sectors at node u (see Figure 3). For each sector, we divide it into a minimum number of *open* cones of degree α, where $\alpha \le \pi/3$.
    (b) For each cone, let $s_1, s_2, \cdots, s_m$ be the ordered neighbors $N_{UDel}(u)$ of u in this cone. That is, $s_1, s_2, \cdots, s_m$ are all unprocessed nodes that are connected by an edge of the unit Delaunay triangulation to u. For each cone, first add the shortest edge in E that is adjacent to u to the edge set E', then add to E' all the edges

$s_j s_{j+1}$ between its geometrically ordered unprocessed neighbors in this cone, $1 \leq j < m$. Notice that, here such edges $s_j s_{j+1}$ are *not* necessarily in $UDel(V)$. For example, when node u has a Delaunay neighbor x such that ux intersects edge $s_i s_{i+1}$ and $\|ux\| > 1$.

(c) Mark node u *processed*.

Repeat this procedure in order of π, until all nodes are marked *processed*. Let $BPS(UDG(V))$ denote the final graph formed by edge set E'.

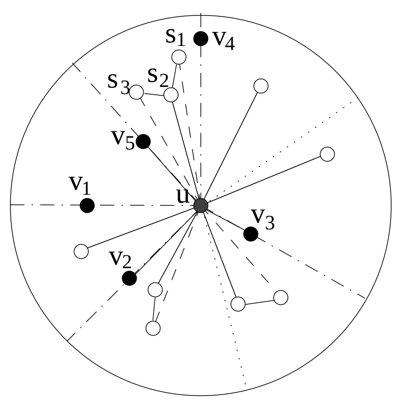

Fig. 3. Constructing planar spanner with bounded degree for $UDG(V)$: process node u. Here $v_1, \cdots, v_5$ are the processed neighbors of node u in $UDel(V)$.

Notice that, for UDG we need all edges be less than one unit, while for point set we do not worry about whether an edge is in the original graph. In the algorithms for $UDG(V)$, we change the cone angle bound from $\pi/2$ to $\pi/3$. The reason is that in the proof of spanner property we need to guarantee the edge $s_i s_j$ and vv' must be in $UDG(V)$, i.e., $\|s_i s_j\| \leq 1$ and $\|vv'\| \leq 1$.

3.2. *Analysis of Algorithm*

The bounded node degree property of the final structure $BPS(UDG(V))$ is trivial. The proof is similar to the one for point set. Only difference is that the angle of open cone is $\alpha \leq \pi/3$ instead of $\alpha \leq \pi/2$. Notice that node degree is bounded by 25 if $\alpha = \pi/3$.

Remember that it is straightforward that $BPS(V)$ is planar since it is a subgraph of the Delaunay triangulation. This is not true anymore for $BPS(UDG(V))$: the graph $BPS(UDG(V))$ may contain edges that are not in Delaunay triangulation. We will prove that the structure $BPS(UDG(V))$ is still planar.

Theorem 6. *$BPS(UDG(V))$ is a planar graph.*

PROOF. Observe that $UDel(V)$ is a planar graph. When each node u is being processed, we add two kinds of edges: (1) edge us_i, where s_i is the nearest unprocessed node in some cone divided by u; (2) some edges $s_i s_{i+1}$, when s_i and s_{i+1} are consecutive unprocessed neighbors of u in graph $UDel(V)$. See Figure 3 for illustration. These edges us_i belong to $UDel(V)$, so they will not intersect each other. If edge $s_i s_{i+1}$ is in $UDel(V)$, then it will not break the planar property of the graph also. Otherwise, the only possible reason which makes $s_i s_{i+1} \notin UDel(V)$ is that there are some edges (such as uw in Figure 4) in $Del(V)$ between us_i and us_{i+1} with length longer than 1. Then all such endpoints w of these long

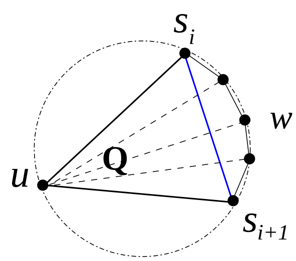

Fig. 4. No new edges can be added by other nodes to intersect $s_i s_{i+1}$, where $s_i s_{i+1}$ is added by u and not in $UDel(V)$.

edges and s_i, s_j, u will form a polygon, denoted by Q, in $UDel(V)$. We will show that after $s_i s_{i+1}$ is added no intersecting edges can be added in $BPS(UDG(V))$. Notice that all the edges which are possible to add in $BPS(UDG(V))$ must be diagonals of some polygons in $UDel(V)$. However, all the diagonals of polygon Q intersecting $s_i s_{i+1}$ are longer than 1, as uw is, i.e., they will never be considered by our algorithm. Consequently, adding edge $s_i s_{i+1}$ will not break the planar property. This finishes our proof. □

Next, we prove that $BPS(UDG(V))$ is a spanner of $UDG(V)$. Notice that we cannot directly apply for proof that $BPS(V)$ is a spanner here since the edges added are different.

Theorem 7. *$BPS(UDG(V))$ is a $\ell \cdot C_{del}$-spanner of $UDG(V)$, where $\ell = \max\{\frac{\pi}{2}, \pi \sin \frac{\alpha}{2} + 1\}$.*

PROOF. Keil and Gutwin [17] showed that the Delaunay triangulation is a t-spanner for a constant $C_{del} = \frac{4\sqrt{3}}{9}\pi$ using induction on the increasing order of the lengths of all pair of nodes. We can show that the path connecting nodes u and v constructed in Ref. [17] also satisfies that all edges of that path is shorter than $\|uv\|$. Consequently, for any edge $uv \in UDG(V)$ we can find a path in $UDel(V)$ with length at most a $t = \frac{4\sqrt{3}}{9}\pi$ times $\|uv\|$, and all edges of the path is shorter than $\|uv\|$. So we only need to show that for any edge $uv \in UDel(V)$, there exists a path in $BPS(UDG(V))$ between u and v whose length is at most a constant ℓ times $\|uv\|$. Then $BPS(UDG(V))$ is a $\ell \cdot C_{del}$-spanner.

Consider an edge uv in $UDel(V)$. If edge uv is in $BPS(UDG(V))$, the claim trivially holds.

Then consider the case $uv \notin BPS(UDG(V))$. The rest of the proof is similar to the proof of Theorem 5. There must exist a node v' in the same cone with v such that $\|uv\| >$

$\|uv'\|$, $uv' \in BPS(UDG(V))$, and $\angle v'uv < \alpha \leq \pi/3$. Let $v' = s_1, s_2, \cdots, s_k = v$ be the sequence of nodes in the ordered unprocessed neighborhood of u in $UDel(V)$ from v' to v. Let $v' = w_1, w_2, \cdots, w_k = v$ be the sequence of nodes in the ordered unprocessed neighborhood of u in $Del(V)$ from v' to v. Obviously, the set $\{s_1, s_2, \cdots, s_k\}$ is a subset of $\{w_1, w_2, \cdots, w_k\}$. Similar to Theorem 5, we know that the length of the path $uw_1w_2 \cdots w_k$ to connect u and v with length at most $\max\{\frac{\pi}{2}, \pi \sin\frac{\alpha}{2} + 1\} \cdot \|uv\|$, where $w_1 = s_1$ is the nearest neighbor of u in the cone, and $w_k = v$. Since any such node w_i is not inside the polygon Q (defined in the Figure 4 of proof for Theorem 6), the path $us_1s_2 \cdots s_k$ is not longer than the length of path $uw_1w_2 \cdots w_k$. This finishes the proof. □

The spanner theorem also implies the connectivity of the final topology. In addition, the computation cost of the algorithm is $O(n \log n)$.

It is unclear whether we can run the greedy spanner algorithm by Gudmundsson *et al.* [14], using the above method as the preprocessing step, to obtain a low-weighted bounded degree planar spanner for $UDG(V)$. Notice that if we run the greedy method with the naive implementation on $BPS(UDG(V))$, we will obtain a graph whose edges satisfying the t-leapfrog property, i.e., the final structure is low-weighted. We leave it as an open problem whether we can construct low-weighted planar bounded degree spanner for UDG in time $O(n \log n)$.

Notice that here our algorithm for UDG removes long edges from Delaunay triangulation before we process them. We can also first process the Delaunay edges and then remove the long edges from the resulted graph. In other words, we can first run the algorithm for point set to build $BPS(V)$ with parameter $\alpha \leq \pi/3$, then remove the edges with length longer than 1 in $BPS(V)$. The final graph, denoted as $BPS'(UDG(V))$, could be different from $BPS(UDG(V))$ since (1) the ordering of nodes could be different; (2) $BPS(UDG(V))$ could add some edges (some s_is_{i+1} type edges) that do not belong to the unit Delaunay triangulation $UDel(V) = Del(V) \cap UDG(V)$, while $BPS'(UDG(V))$ always uses the edges from $UDel(V)$. It is not difficult to prove $BPS'(UDG(V))$ also has same properties as $BPS(UDG(V))$, such as planar, bounded spanning ratio and bounded node degree.

3.3. *Localized Algorithm*

Due to the limited power and resources of wireless nodes, wireless ad hoc networks prefer that the underlying network topology can be constructed and maintained in a localized manner. Therefore, in Ref. [22], we extended our centralized algorithm for $UDG(V)$ to a localized algorithm with $O(n)$ total messages so that it can be used in wireless ad hoc networks. Surprisingly enough, the proof of the localized algorithm is much more complicated. For the completeness of presentation, we briefly review that method to support our claim that the centralized algorithm can be extended to a localized one. The algorithm has three steps as following.

Firstly, since we cannot build Delaunay triangulation locally, we construct a planar spanner, localized Delaunay triangulation for UDG. In Ref. [20], Li *et. al* presented the localized method to build the LDel for UDG and proved it is a planar spanner with the same spanning ratio $t = C_{Del}$ as the Delaunay triangulation. Let $LDel(V)$ be the localized Delaunay triangulation of $UDG(V)$, and let $N_{LDel}(u)$ be the neighbors of node u in $LDel(V)$.

Secondly, instead of building a global order, we build a local order π of V using the following method. Every node u initializes $\pi_u = 0$, i.e., unordered. For node u with $\pi_u = 0$, if its degree $d_u \leq 5$ then node u queries each node v, from its unordered neighbors, the current degree d_v. If node u has the smallest ID among all unordered neighbors v with $d_v \leq 5$, node u sets $\pi_u = \max\{\pi_v \mid v \in N_{LDel}(u)\}+1$, and broadcasts π_u to its neighbors $N_{LDel}(u)$ in $LDel(V)$. Notice that if all unordered neighbors with $d_v \leq 5$ has larger ID, we call such query round a *failed round*. Node u performs a new round of queries only if it finds that the unordered neighbors have been reduced from previous failed round. If node u receives a message from its neighbor v saying that $\pi_v = k$, it updates its $d_u = d_u - 1$ and also updates the order π_v stored locally. So d_u represents how many neighbors are not ordered so far. If node u finds that $d_u \leq 5$ and $\pi_u = 0$, it goes to do querying. When node u finds that $d_u = 0$ and $\pi_u > 0$, it can go to next step to bound node degree. Notice that different nodes may have the same order and go to next step in the same time. However, most importantly, in the local ordering, each node u has different order with its neighbors.

Finally, we apply the same technique in the previous algorithm to bound the node degree following the local order π. Initialize all nodes unprocessed. If an unprocessed node u has the highest local order among its unprocessed neighbors $N_{LDel}(u)$, it applies the same procedure (steps (4a) and (4b) in the centralized algorithm for UDG) to bound degree, and then marks itself processed. When all nodes are processed, the algorithm terminates.

4. Conclusion

In this paper, we first proposed a new structure which is a planar spanner with bounded node degree for any point set V. We can further bound the total weight of the structure by applying the method by Gudmundsson *et al.* [14]. Then we gave one centralized algorithm to construct bounded degree planar spanner for $UDG(V)$. The centralized algorithms can be implemented in time $O(n \log n)$. An advantage of the centralized methods presented here, compared with the previous methods, is that these methods can be extended to localized methods, although the extension is not straightforward.

5. Acknowledgment

The authors would like to thank Dr. P. Bose and Dr. P.-J. Wan for valuable discussions on Ref. [6]. And we also greatly appreciate the anonymous reviewers and the editor for their constructive comments for improving the paper.

References

1. ARYA, S., DAS, G., MOUNT, D., SALOWE, J., AND SMID, M. Euclidean spanners: short, thin, and lanky. In *Proc. 27th ACM STOC* (1995), pp. 489–498.

2. ARYA, S., MOUNT, D. M., AND SMID, M. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. 35th Annual Symp. on Foundations of Computer Science* (1994), pp. 703–712.
3. ARYA, S., AND SMID, M. Efficient construction of a bounded degree spanner with low weight. *Algorithmica 17* (1997), 33–54.
4. BOSE, P., DEVROYE, L., EVANS, W., AND KIRKPATRICK, D. On the spanning ratio of gabriel graphs and beta-skeletons. In *Proceedings of the Latin American Theoretical Infocomatics (LATIN)* (2002).
5. BOSE, P., GUDMUNDSSON, J., AND MORIN, P. Ordered theta graphs. In *Proceedings of the Canadian Conference on Computational Geometry (CCCG)* (2002).
6. BOSE, P., GUDMUNDSSON, J., AND SMID, M. Constructing plane spanners of bounded degree and low weight. In *Proceedings of the European Symposium on Algorithms (ESA)* (2002).
7. BOSE, P., AND MORIN, P. Online routing in triangulations. In *Proc. of the 10 th Annual Int. Symp. on Algorithms and Computation ISAAC* (1999).
8. CHANDRA, B., DAS, G., NARASIMHAN, G., AND SOARES, J. New sparseness results on graph spanners. *International Journal on Computational Geometry and Applications*, 5 (1995), 125–144.
9. DAS, G., AND JOSEPH, D. Which triangulations approximate the complete graph? In *Proceedings of International Symposium on Optimal Algorithms (LNCS 401)* (1989), pp. 168–192.
10. DAS, G., AND NARASIMHAN, G. A fast algorithm for constructing sparse euclidean spanners. *International Journal on Computational Geometry and Applications 7*, 4 (1997), 297–315.
11. DOBKIN, D., FRIEDMAN, S., AND SUPOWIT, K. Delaunay graphs are almost as good as complete graphs. *Discrete Computational Geometry* (1990).
12. EPPSTEIN, D. Beta-skeletons have unbounded dilation. *International Journal on Computational Geometry and Applications 23*, 1 (2002), 43–52.
13. GABRIEL, K., AND SOKAL, R. A new statistical approach to geographic variation analysis. *Systematic Zoology 18* (1969), 259–278.
14. GUDMUNDSSON, J., LEVCOPOULOS, C., AND NARASIMHAN, G. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal of Computing 31*, 5 (2002), 1479–1500.
15. JAROMCZYK, J., AND TOUSSAINT, G. Relative neighborhood graphs and their relatives. *Proceedings of IEEE 80*, 9 (1992), 1502–1517.
16. KEIL, J., AND GUTWIN, C. The delaunay triangulation closely approximates the complete euclidean graph. In *Proc. 1st Workshop Algorithms Data Structure (LNCS 382)* (1989).
17. KEIL, J. M., AND GUTWIN, C. A. Classes of graphs which approximate the complete euclidean graph. *Discrete Computational Geometry 7* (1992).
18. LEVCOPOULOS, C., AND LINGAS, A. There are planar graphs almost as good as the complete graphs and almost as cheap as minimum spanning trees. *Algorithmica 8* (1992), 251–256.
19. LEVCOPOULOS, C., NARASIMHAN, G., AND SMID, M. Improved algorithms for constructing fault tolerant geometric spanners. *Algorithmica* (2000).
20. LI, X.-Y., CALINESCU, G., AND WAN, P.-J. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)* (2002), vol. 3.
21. LUKOVSZKI, T. New results on fault tolerant geometric spanners. *Proceedings of the 6th Workshop on Algorithms an Data Structures (WADS'99), Lecture Notes in Computer Science* (1999), 193–204.
22. WANG, Y., AND LI, X.-Y. Localized construction of bounded degree planar spanner for wireless networks. In *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing* (2003).