On Spanners and Lightweight Spanners of Geometric Graphs*

Iyad A. Kanj[†] Ljubomir Perković[‡] Ge Xia[§]

Abstract

We consider the problem of computing spanners of Euclidean and unit disk graphs embedded in the 2-dimensional Euclidean plane. We are particularly interested in spanners that possess useful properties such as planarity, bounded degree, and/or light weight. Such spanners have been extensively studied in the area of computational geometry and have been used as the building block for constructing efficient and reliable wireless network communication topologies.

We study the above problem under two computational models: the centralized and the distributed model. In the distributed model we focus on algorithms that are *local*. Such algorithms are suitable for the relevant applications (e.g., wireless computing).

Under the centralized model, we present an $O(n \lg n)$ time algorithm that computes a bounded-degree plane spanner of a complete Euclidean graph, where n is the number of points in the graph. Both upper bounds on the degree and the stretch factor significantly improve the previous bounds. We extend this algorithm to compute a bounded-degree plane *lightweight* spanner of a complete Euclidean graph.

Under the distributed model, we give the first *local* algorithm for computing a spanner of a unit disk graph that is of bounded degree and plane. The upper bounds on the degree, stretch factor, and the locality of the algorithm dramatically improve the previous results, as shown in the paper. This algorithm can also be extended to compute a bounded-degree plane *lightweight* spanner of a unit disk graph.

Our algorithms rely on structural and geometric results that we develop in this paper.

1 Introduction

A *spanner* of a weighted graph is a spanning subgraph in which the weight of a shortest path between any pair of points is at most a constant times the weight of a shortest path in the original graph. This constant is called the *stretch factor* of the spanner. A spanner of a graph is *lightweight* if its weight is at most a constant times the weight of a minimum spanning tree of the graph.

In this paper we consider the problem of computing spanners and lightweight spanners of a complete Euclidean graph or a (connected) unit disk graph on n points in the plane. We are interested in spanners that possess the following useful properties: bounded degree, planarity, and light weight. The weight of an edge in the graph in this case is its Euclidean distance, and a

*Preliminary versions of the results in this paper appeared at STACS'08 and DISC'08.

[†]School of Computing, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604. Email: `ikanj@cs.depaul.edu`.

[‡]School of Computing, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604. Email: `lperkovic@cs.depaul.edu`.

[§]Department of Computer Science, Lafayette College, Easton, PA 18042. Email: `gexia@cs.lafayette.edu`.

minimum spanning tree of the graph is a Euclidean Minimum Spanning Tree (abbreviated EMST henceforth) on the point-set of the graph.

The problem of constructing a bounded degree or lightweight plane geometric spanner has been extensively studied within computational geometry, and much of the early work on spanners was done from that perspective under the centralized model of computation (for example, see [1, 4, 11, 12, 13, 16, 20, 22, 30], and the following book on spanners [26]). More recently, wireless network researchers have approached the problem as well. Emerging wireless distributed system technologies, such as wireless ad-hoc and sensor networks, are often modeled as a *unit disk graph* (UDG) in the Euclidean plane: the points of the UDG correspond to the mobile wireless devices, and its edges connect pairs of points whose corresponding devices are in each other's transmission range equal to one unit. Spanners and lightweight spanners of UDGs are fundamental to wireless *distributed* systems because they represent topologies that can be used for efficient unicasting, multicasting, *and/or* broadcasting (see [4, 6, 16, 17, 21, 24, 28], to name a few). For these applications, spanners are typically required to be planar and have bounded degree: the planarity requirement is for efficient routing, while the bounded degree requirement is motivated by interference issues and the physical limitations of wireless devices [4, 6, 16, 17, 21, 28].

When the problem is considered from the perspective of distributed wireless computing, fault tolerance, scalability, and robustness are all major concerns. In this case the *local* distributed computational model, in which the computation of any point in the system (graph) depends only on the information available within its "vicinity" (to be defined precisely later), is a suitable working model. Efficient local distributed algorithms are naturally fault-tolerant and robust because faults and changes can be handled locally by such algorithms. These algorithms are also scalable because the computation performed by a device does not depend on the size of the network.

In this paper we study the problem of computing spanners and lightweight spanners of Euclidean and unit disk graphs under both the centralized and the local distributed models of computation. We present state-of-the-art results on this problem that improve the previous work in several aspects. Our work reveals interesting structural results that are of independent interest.

We summarize below the main results of the paper and how they compare to the relevant work in the literature.

1.1 Spanners

We start with the problem of constructing geometric spanners of complete Euclidean graphs, a well studied problem (see, for example, the recent book [26] for a survey on geometric spanners and their applications in networks). Dobkin et al. [15] showed that the Delaunay graph is a plane geometric spanner of the complete Euclidean graph with stretch factor $(1+\sqrt{5})\pi/2 \approx 5.08$. This ratio was improved by Keil et al. [20] to $C_{del} = 2\pi/(3\cos(\pi/6)) < 2.42$, which currently stands as the best upper bound on the stretch factor of the Delaunay graph. While Delaunay graphs are good plane geometric spanners of Euclidean graphs, they may have unbounded degree. Other geometric (sparse) spanners were also proposed in the literature including the Yao graphs [30], the Θ-graphs [20], and many others (see [26]); however, most of these proposed spanners either do not guarantee planarity, or do not guarantee bounded degree.

Bose et al. [3, 4] were the first to show how to extract a subgraph of the Delaunay graph that is a bounded-degree, plane geometric spanner of the complete Euclidean graph (with stretch factor bounded by 10.02 and degree bounded by 27). In the context of UDGs, Li et al. [23, 24] gave a distributed algorithm, which is not local, that constructs a plane geometric spanner of a unit disk graph with stretch factor C_{del}; however, the spanner constructed can have unbounded degree. Wang and Li [28] then showed how to construct a bounded-degree plane spanner of a unit disk graph with stretch factor $\max\{\pi/2, 1+\pi\sin(\alpha/2)\}\cdot C_{del}$ and degree bounded by $19+2\pi/\alpha$, where $0 < \alpha < 2\pi/3$ is a parameter. Very recently, Bose et. al [7] improved the earlier result in [3, 4] and showed how to construct a subgraph of the Delaunay graph that is a geometric spanner of the complete Euclidean graph with stretch factor: $\max\{\pi/2, 1+\pi\sin(\alpha/2)\}\cdot C_{del}$ when $\alpha < \pi/2$, and $(1 + 2\sqrt{3} + 3\pi/2 + \pi\sin(\pi/12))\cdot C_{del}$ when $\pi/2 \le \alpha \le 2\pi/3$, and whose degree is bounded by $14 + 2\pi/\alpha$. Bose et al. then applied their construction to obtain a plane geometric spanner of a unit disk graph with stretch factor $\max\{\pi/2, 1 + \pi\sin(\alpha/2)\}\cdot C_{del}$ and degree bounded by $14 + 2\pi/\alpha$, for any $0 < \alpha \le \pi/3$. This was the best bound on the stretch factor and the degree.

We present two new results on constructing geometric spanners. We prove structural results about Delaunay graphs that allow us to develop a very simple linear-time algorithm that, given a Delaunay graph, constructs a subgraph of the Delaunay graph with stretch factor $1 + 2\pi(k\cos(\pi/k))^{-1}$ (with respect to the Delaunay graph) and degree at most k, for any integer parameter $k \ge 14$. This result immediately implies an $O(n\lg n)$ algorithm for constructing a plane geometric spanner of a Euclidean graph with stretch factor of $(1 + 2\pi(k\cos(\pi/k))^{-1})\cdot C_{del}$ and degree at most k, for any integer parameter $k \ge 14$ (n is the number of points in the graph). We then translate our work to unit disk graphs and present our second result: a very simple, *3-local* distributed algorithm that, given a unit-disk graph embedded in the plane, constructs a plane geometric spanner of the unit disk graph with stretch factor $(1 + 2\pi(k\cos(\pi/k))^{-1})\cdot C_{del}$ and degree bounded by k, for any integer parameter $k \ge 14$. This efficient distributed algorithm exchanges no more than $O(n)$ messages in total, and in which the local processing time (at any point in the graph) is $O(n\lg n)$.

Both algorithms significantly improve the previous results in terms of stretch factor and degree bound. To show this, we compare our results with previous results in more detail. For a degree bound $k = 14$, our result on Euclidean graphs imply a bound of at most 3.54 on the stretch factor. As the degree bound k approaches ∞, our bound on the stretch factor approaches C_{del}. The very recent results of Bose et al. [7] achieve a lowest degree bound of 17 which corresponds to a bound on the stretch factor of at least 23. If Bose et al. [7] allow the degree bound to be arbitrarily large (i.e., approach ∞), their bound on the stretch factor approaches $(\pi/2)\cdot C_{del} > 3.75$. Our stretch factor and degree bounds for unit disk graphs are the same as our results for complete Euclidean graphs. The smallest degree bound derived by Bose et al. [7] is 20 which corresponds to a stretch factor of at least 6.19. If Bose et al. [7] allow the degree bound to be arbitrarily large, then their bound on the stretch factor approaches $(\pi/2)\cdot C_{del} > 3.75$. On the other hand, the smallest degree bound derived in Wang et al. [28] is 25, and that corresponds to a bound of 6.19 on the stretch factor. If Wang et al. [28] allow the degree bound to be arbitrarily large, then their bound on the stretch factor approaches $(\pi/2)\cdot C_{del} > 3.75$. Therefore, even the worst bound of at most 3.54 on the stretch factor corresponding to our lowest bound on the degree $k = 14$, is better than the best bound on the stretch factor of at least 3.75 corresponding to arbitrarily large degree in both Bose et al. [7] and Wang et al. [28].

1.2 Lightweight spanners

Levcopoulos and Lingas [22] developed the first centralized algorithm for the problem of constructing lightweight spanners of complete Euclidean graphs. Their $O(n\log n)$ time algorithm, given a rational $\lambda > 2$, produces a plane spanner with stretch factor $(\lambda - 1)\cdot C_{del}$ and total weight $(1 + \frac{2}{\lambda-2})\cdot wt(\text{EMST})$, where $wt(\text{EMST})$ is the weight of a Euclidean Minimum Spanning Tree on the point-set of the graph. Althöfer et al. [1] gave a polynomial time greedy algorithm that constructs a lightweight plane spanner of a Euclidean graph having the same upper bound on the stretch factor and weight as the algorithm by Levcopoulos and Lingas [22]. The degree of the lightweight spanner in both [22] and [1], however, may be unbounded: it is not possible to bound the degree without worsening the stretch factor. A more recent $O(n \log n)$ time algorithm by Bose, Gudmundsson, and Smid [4] for complete Euclidean graphs, succeeded in bounding the degree of the plane spanner by 27 but at a large cost: the stretch factor of the obtained plane spanner is approximately 10.02, and its weight is $O(wt(\text{EMST}))$, where the hidden constant in the asymptotic notation is undetermined.

Our contribution with regard to this problem is a centralized algorithm for complete Euclidean graphs that improves the above algorithms. We design a centralized algorithm that, for any integer constant $k \geq 14$ and constant $\lambda > 2$, constructs a plane spanner of a complete Euclidean graph having degree at most k, stretch factor $(\lambda - 1) \cdot (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$, and weight at most $(1 + \frac{2}{\lambda-2}) \cdot wt(\text{EMST})$. We can compare our algorithm with the algorithm by Bose, Gudmundsson, and Smid [4] if we let $k = 14$ and $\lambda \approx 2.475$ in the above result: we obtain an $O(n \log n)$ time algorithm that, given a complete Euclidean graph on n points, computes a plane spanner of the given graph having degree at most 14, stretch factor at most 5.22, and weight at most $5.22 \cdot wt(\text{EMST})$.

We then consider the problem of computing bounded-degree plane lightweight spanners of unit disk graphs using a local distributed algorithm. To the best of our knowledge, the only distributed algorithm for this problem is the algorithm in [10]. While the distributed algorithm in [10] solves the problem for a generalization of unit disk graphs, called quasi-unit ball graphs, in higher dimensional Euclidean spaces, the algorithm is not local (it runs in a poly-logarithmic number of rounds), and the weight and the degree of the spanner are only bounded asymptotically. We note that distributed algorithms for computing lightweight spanners of general graphs have been extensively considered in the literature; see for example [27] for a survey on some of these results. In this paper we show that: for any integer constant $k \geq 14$ and constant $\lambda > 2$, there exists an i-local distributed algorithm, where $i = \lfloor (8/\pi) \cdot (\lambda + 1)^2 \rfloor$, that computes a plane spanner of a given unit disk graph containing a EMST on its point-set, of degree at most k, weight at most $(1 + \frac{2}{\lambda-2}) \cdot wt(\text{EMST})$, and stretch factor $(\lambda - 1)^4 \cdot (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$. This is the first local algorithm for this problem.

Table 1: A comparison of lightweight spanner algorithms given the constant $\lambda > 2$ and the maximum degree bound k; the following notations are used: $\rho^* = (\lambda - 1) \cdot C_{del}$, $c^* = (1 + \frac{2}{\lambda-2})$, and $a^* = 1 + 2\pi(k \cos \frac{\pi}{k})^{-1}$.

Algorithm	LL92 [22]	ADDJS93 [1]	BGS05 [4]	KPX08	KPXLoc08
Stretch factor	ρ^*	ρ^*	10.02	$a^* \cdot \rho^*$	$a^* \cdot (\lambda - 1)^3 \cdot \rho^*$
Weight factor	c^*	c^*	$O(1)$	c^*	c^*
Max. degree	∞	∞	27	k	k
Running time	$O(n \log n)$	$O(n^2 \log n)$	$O(n \log n)$	$O(n \log n)$	N/A

In Table 1, we compare the centralized complete Euclidean graph lightweight spanner algorithms by Levcopoulos and Lingas [22] (denoted LL92), by Althöfer et al. [1] (denoted ADDJS93), and by Bose, Gudmundsson, and Smid [4] (denoted BGS05) with our centralized algorithm (denoted KPX08) and our local distributed algorithm (denoted KPXLoc08) developed to compute lightweight spanners of the more general unit disk graphs. The table gives the bounds on the stretch factor, the weight factor (the constant c^* such that the weight of the spanner is at most $c^* \cdot wt(\text{EMST})$), the maximum degree and the running time. Note that the first two algorithms (LL92 and ADDJS93) do not guarantee an upper bound on the degree of the spanner. Our algorithms match their bounds on the weight factor to provide a maximum degree bound at a small multiplicative cost in the stretch factor (a^* for our centralized algorithm and $(\lambda - 1)^3 \cdot a^*$ for our local distributed algorithm). For example, for a degree bound of 14, our upper bound on the stretch factor increases (with respect to [22] and [1]) by a multiplicative constant of 1.47 for the centralized algorithm, and of 2.92 (corresponding to $\lambda = 2.256$) for the local distributed algorithm. For larger values of k, the multiplicative factors are even smaller.

In Table 2 we use some concrete values for k and λ in order to compare our algorithms with the algorithm BGS05 by Bose, Gudmundsson, and Smid [4]. Their algorithm only guarantees a maximum degree bound of 27. The listed bounds for stretch factor ρ^* and weight factor c^* for $k = 27$ are obtained by setting $\lambda = 2.551$ in KPX08 and $\lambda = 2.282$ in KPXLoc08. The bounds for stretch factor ρ^* and weight factor c^* when $k = 14$ are obtained by setting $\lambda = 2.475$ in KPX08 and $\lambda = 2.256$ in KPXLoc08.

Table 2: Comparison between algorithm BGS05 [4] and our algorithms KPX08 and KPXLoc08 for different values of k.

$k =$	14	27
BGS05	N/A	$\rho^* = 10.02, c^* = O(1)$
KPX08	$\rho^*, c^* = 5.22$	$\rho^*, c^* = 4.63$
KPXLoc08	$\rho^*, c^* = 8.81$	$\rho^*, c^* = 8.08$

1.3 Overview of the paper

The rest of the paper is organized as follows. In the next section we review the necessary terminology and background. In Section 3, we present a centralized algorithm for computing a bounded-degree plane spanner of a complete Euclidean graph. We then generalize this in Section 4 to a local distributed algorithm for computing a bounded-degree plane spanner of a unit disk graph. In Section 5 we present a centralized algorithm for computing a bounded-degree lightweight plane spanner of a complete Euclidean graph. Finally, in Section 6, we present a local distributed algorithm for computing a bounded-degree lightweight plane spanner of a unit disk graph.

2 Definitions and Background

2.1 Graphs embedded in the two-dimensional Euclidean plane

Given a set of points $\mathcal{P}$ in the 2-dimensional Euclidean plane, the complete Euclidean graph $\mathcal{E}$ on $\mathcal{P}$ is defined to be the complete graph whose point-set is $\mathcal{P}$. Each edge ab connecting points a and b is assumed to be embedded in the plane as the straight line segment ab; we define its *weight* to be the Euclidean distance $|ab|$. We define the unit disk graph U to be the subgraph of $\mathcal{E}$ consisting of all edges ab with $|ab| \leq 1$. We assume in this paper that the unit disk graph U is connected. It is well-known that a connected unit disk graph contains a Euclidean minimum spanning tree of its point-set.

For a subgraph $H \subseteq \mathcal{E}$, we denote by $V(H)$ and $E(H)$ the set of points and the set of edges of H, respectively, and by $wt(H)$ the sum of the weights of all the edges in H, that is, $wt(H) = \sum_{xy \in E(H)} |xy|$. The *length* of a path P (resp. cycle C) in a subgraph $H \subseteq \mathcal{E}$, denoted $|P|$ (resp. $|C|$), is the number of edges in P (resp. C). A point b is said to be an *i-hop neighbor* of a in a subgraph $H \subseteq E$, if there exists a path P from a to b in H satisfying $|P| \leq i$.

Let G be a subgraph of $\mathcal{E}$. The weight of a simple path $a = m_0, m_1, ..., m_r = b$ in G is $\sum_{j=0}^{r-1} |m_j m_{j+1}|$. A spanning subgraph H of G is said to be a *geometric spanner* of G if there is a constant ρ such that, for every two points $a, b \in G$, the weight of a shortest path from a to b in H is at most ρ times the weight of a shortest path from a to b in G. The constant ρ is called the *stretch factor* of H (with respect to G). The following is a well known—and obvious—fact:

Fact 2.1. *A subgraph H of graph G has stretch factor ρ with respect to G if and only if for every edge $xy \in G$: the weight of a shortest path in H from x to y is at most $\rho \cdot |xy|$.*

A spanning subgraph of $\mathcal{E}$ is said to have *low* weight, or to be *lightweight*, if its weight is at most $c \cdot wt(\text{EMST})$, for some constant c.

For three non-collinear points x, y, z in the plane we denote by $\bigcirc xyz$ the circumscribed circle of $\triangle xyz$. A *Delaunay triangulation* of $\mathcal{P}$ is a triangulation of $\mathcal{P}$ such that the circumscribed circle of every triangle in this triangulation (i.e., every triangular face) contains no point of $\mathcal{P}$ in its interior [14]. It is well known that if the points in $\mathcal{P}$ are *in general position* (i.e., no four points in $\mathcal{P}$ are cocircular) then the Delaunay triangulation of $\mathcal{P}$ is unique [14]. In this paper—as in most papers in the literature—we shall assume that the points in $\mathcal{P}$ are in general position; otherwise, the input can be slightly perturbed so that this condition is satisfied. The *Delaunay graph* of $\mathcal{P}$ is defined as the plane graph whose point-set is $\mathcal{P}$ and whose edges are the edges of the Delaunay triangulation of $\mathcal{P}$. An alternative equivalent definition that we end up using is:

Definition 2.2. ([14]) An edge xy is in the Delaunay graph of $\mathcal{P}$ if and only if there exists a circle through points x and y whose interior contains no point in $\mathcal{P}$.

It is well known that the Delaunay graph of $\mathcal{P}$ is a spanning subgraph of $\mathcal{E}$ whose stretch factor is at most $C_{del} = 4\sqrt{3}\pi/9 < 2.42$ [20].

Given integer parameter $k > 6$, the *Yao subgraph* [30] of a plane graph G is constructed by performing the following *Yao step*: For each point m in G partition the space (arbitrarily) into k cones of equal measure whose apex is at m, thus creating k closed cones of angle $2\pi/k$ each, and choose the shortest edge in G out of m (if any) in each cone. The Yao subgraph consists of edges in G chosen by *either* endpoint. Note that the degree of a point in the Yao subgraph of G may be unbounded.

Two edges mx, my incident to a point m in a subgraph G of $\mathcal{E}$ are said to be *consecutive* if one of the angular sectors determined by the two segments mx and my in the plane contains no neighbors of m.

Let x and y be two points in the plane and let (O) be any circle passing through x and y. The chord xy subtends two regions in the interior of (O). If z is a point in the plane that does not belong to the straight line through x and y, then one of the two regions interior to (O) subtended by the chord xy is on the same side of the straight line passing through x and y as z, whereas the other region is on the opposite side. For convenience, we will refer to the former as the region interior to (O) subtended by chord xy and *closer* to z, and to the latter as the region interior to (O) subtended by chord xy and *farther* or *away* from z. We will be using the following simple fact (see Figure 1 for illustration):

Fact 2.3. *Let x and y be two points in the plane and let (O) be a circle passing through x and y. Let z be any point exterior to (O), and let $(O') = \bigcirc xyz$. Then the region interior to (O') subtended by chord xy and away from z is inside the region interior to (O) subtended by xy and away from z.*

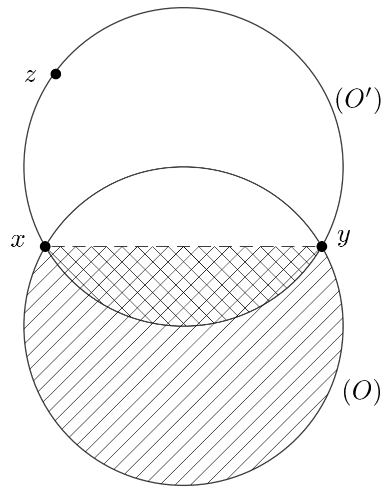

Figure 1: Illustration for Fact 2.3.

2.2 Message complexity of local distributed algorithms

Assuming that the distributed system is modeled as a graph, a distributed algorithm is said to be i-*local* if, "intuitively", the computation at each point of the graph depends solely on the information about its i-hop neighbors. This notion can be formalized as follows [25, 27, 29]: a distributed algorithm is i-*local* if it can be simulated to run in at most i synchronous communication rounds for some integer parameter $i \geq 0$. A distributed algorithm is called *local* if it is i-local for some integer constant i.

Each point in the local distributed algorithms presented in this paper starts by collecting the IDs and coordinates of its i-hop neighbors for some fixed i; then it performs only local computations afterwards. For a fixed i, it was shown in [19] that the i-hop neighborhoods of the points in a UDG U can be computed by a local distributed algorithm in which the total number of messages sent is $O(n)$, where $n = |V(U)|$, and where the message length is $O(\lg n)$ bits. Therefore, the message complexity of the i-local distributed algorithms in this paper is $O(n)$.

3 Computing Spanners of Delaunay and Euclidean Graphs

Let $\mathcal{P}$ be a set of points in the plane and let $\mathcal{E}$ be the complete Euclidean graph defined on point-set $\mathcal{P}$. Let G be the Delaunay graph of $\mathcal{P}$. This section is devoted to proving the following theorem:

Theorem 3.1. *For every integer $k \geq 14$, there exists a subgraph G' of G such that G' has maximum degree k and stretch factor $1 + 2\pi(k \cos \frac{\pi}{k})^{-1}$.*

A linear time algorithm that computes G' from G is the key component of our proof. This very simple algorithm essentially performs a *modified Yao step* (see Section 2) and selects up to k edges out of every point of G. G' is simply the spanning subgraph of G consisting of edges chosen by *both* endpoints.

In order to describe the modified Yao step, we must first develop a better understanding of the structure of the Delaunay graph G. Let ca and cb be edges incident on point c in G such that $\angle bca \leq 2\pi/k$ and ca is the shortest edge within the angular sector $\angle bca$. We will show how the above theorem easily follows if, for every such pair of edges ca and cb:

1. we show that there exists a path P from a to b in G such that: $|ca| + wt(P) \leq (1 + 2\pi(k \cos \frac{\pi}{k})^{-1})|cb|$, and
2. we modify the standard Yao step to include the edges of this path in G', in addition to including the edges picked by the standard Yao step, and without choosing more than k edges at any point.

This will ensure that: for any edge $cb \in G$ that is not included in G' by the modified Yao step, there exists a path from c to b in G', whose edges are all included in G' by the modified Yao step, and whose weight is at most $(1 + 2\pi(k \cos \frac{\pi}{k})^{-1})|cb|$. In the lemma below, we prove the existence of this path and show some properties satisfied by edges of this path. We will then modify the standard Yao step to include edges satisfying these properties.

Lemma 3.2. *Let $k \geq 14$ be an integer, and let ca and cb be edges in G such that $\angle bca \leq 2\pi/k$ and ca is the shortest edge in the angular sector $\angle bca$. There exists a path $P : (a = m_0, m_1, \ldots, m_r = b)$ in G such that:*

(i) $|ca| + \sum_{i=0}^{r-1} |m_i m_{i+1}| \leq (1 + 2\pi(k \cos \frac{\pi}{k})^{-1})|cb|$.

(ii) There is no edge in G between any pair m_i and m_j lying in the closed region enclosed by ca, cb and the edges of P, for any i and j satisfying $0 \leq i < j - 1 \leq r$.

(iii) $\angle m_{i-1} m_i m_{i+1} > (\frac{k-2}{k})\pi$*, for $i = 1, \ldots, r-1$.*

(iv) $\angle cam_1 \geq \frac{\pi}{2} - \frac{\pi}{k}$.

We break down the proof of the above lemma into two separate cases: when $\triangle abc$ contains no point of G in its interior, and when it does. We define some additional notation and terminology first. We will denote by o the center of $\bigcirc abc$, and by Θ the measure of $\angle bca$. Note that $\angle aob = 2\Theta \leq 4\pi/k$. We will use $\overset{\frown}{ab}$ to denote the arc of $\bigcirc abc$ determined by points a and b and facing $\angle aob$. We will make use of the following proposition (see Figure 2 for illustration):

Proposition 3.3. *If there are two circles through c and a and through c and b, respectively, that do not contain any points of G in their interior, then the region interior to $\bigcirc abc$ subtended by chord ca and away from b and the region interior to $\bigcirc abc$ subtended by chord cb and away from a contain no points of G.*

Proof. Let C_a be a circle passing through c and a whose interior is devoid of points of G. Then b is not interior to C_a. By Fact 2.3, the region interior to $\bigcirc abc$ subtended by chord ca and away from b is inside the region interior to C_a subtended by chord ca and away from b, and hence is devoid of points of G. The proof that the region interior to $\bigcirc abc$ subtended by chord cb and away from a is devoid of points of G is analogous. □

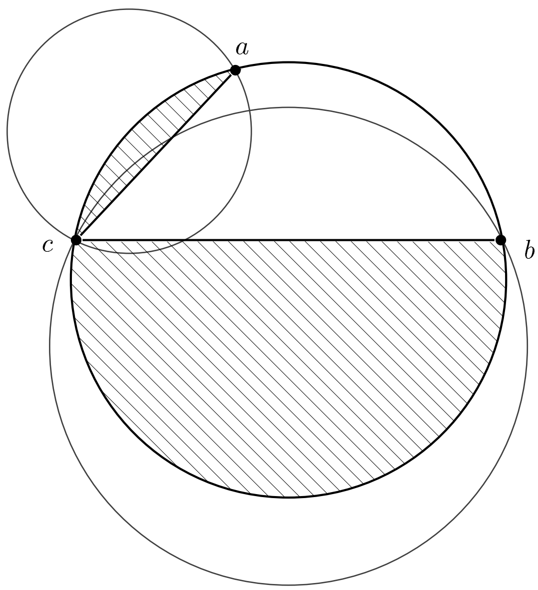

Figure 2: Illustration for Proposition 3.3. The interior of the shaded regions are devoid of points of G.

3.1 The outward path

We consider first the case when no points of G are inside $\triangle abc$. Since ca and cb are edges in G, by Definition 2.2 and Proposition 3.3, it follows that the region interior to $\bigcirc abc$ subtended by chord ab and closer to c is devoid of points of G. Keil and Gutwin [20] showed that, in this case, there exists a path between a and b in G in the region interior to $\bigcirc abc$ subtended by chord ab away from c whose weight is bounded by the length of the arc $\widehat{ab}$ (see Lemma 1 in [20]). To prove some properties that this path satisfies, we find it convenient to use an alternative recursive definition of this path, one based on hypotheses $\mathcal{H}$ and $\mathcal{H}'$ and on Proposition 3.5 described next. Moreover, this definition serves another purpose: generalizing the results of this section to unit disk graphs in Section 4.

Let c be a point of G. For a point x in G distinct from c, we define the following hypothesis:

$\mathcal{H}$: there exists a circle passing through c and x containing no point of G in its interior.

For a pair of points (x, y) in G, where $x \neq c$ and $y \neq c$, we define the following hypothesis:

$\mathcal{H}'$: the interior of $\triangle cxy$ is devoid of points of G.

Since both ca and cb are edges in G, there are circles C_a and C_b passing through c, a and c, b, respectively, that contain no point of G in their interior. Therefore, both a and b satisfy hypothesis $\mathcal{H}$ with respect to c, and since—by our assumption—no point of G is interior to $\triangle abc$, the pair (a, b) satisfies hypothesis $\mathcal{H}'$ with respect to c.

Then, given points a and b satisfying hypothesis $\mathcal{H}$ such that the pair (a, b) satisfies hypothesis $\mathcal{H}'$ with respect to c, the path in [20] can be defined recursively as follows:

1. **Base case:** If $ab \in G$, the path consists of edge ab.

2. **Recursive step:** Otherwise, a *point* must reside in the interior of $\bigcirc abc$. Since a and b satisfy hypothesis $\mathcal{H}$ and the pair (a, b) satisfies hypothesis $\mathcal{H}'$ with respect to c, it follows from Proposition 3.3 that no *point* of G is in the region R interior to $\bigcirc cab$ subtended by ab and closer c. Since $ab \notin G$, the region interior to $\bigcirc cab$ subtended by ab and away from c, plus the open segment ab, is not empty; let m be a point in that region with the property that the region R' interior to $\bigcirc amb$ subtended by chord ab and closer to m is empty.[1] We call m an *intermediate point* with respect to the pair of points (a, b). As we show in part (b) of Proposition 3.5 below, point m will satisfy hypothesis $\mathcal{H}$ with respect to c. Moreover, by part (a) of Proposition 3.5, the pair (a, m) satisfies hypothesis $\mathcal{H}'$. Therefore, we can recurse on points m and a. Since G is finite, the recursion must terminate and a path P_{am} between a and m in G is obtained. Similarly, both points m and b satisfy hypothesis $\mathcal{H}$, and the pair (b, m) satisfies hypothesis $\mathcal{H}'$. Therefore, recursing on the points m and b will eventually result in a path P_{mb} between m and b. We concatenate P_{am} and P_{mb} to obtain a path between a and b in G.

See Figure 3 for an illustration of the recursive step above.

Definition 3.4. We call the path constructed above the *outward path* between a and b.

[1] In the case when m belongs to the interior of segment ab, the circle $\bigcirc amb$ becomes an infinite circle (i.e., degenerates to a straight line), and all the arguments still hold.

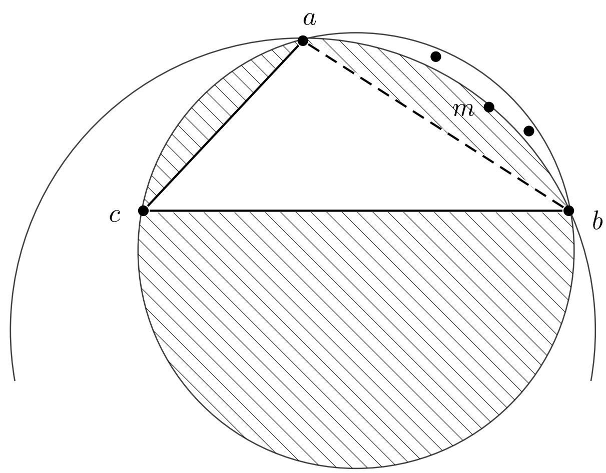

Figure 3: Illustration for the recursive step in the definition of the outward path between a and b. The interior of the shaded regions are devoid of points of G.

Figure 4 illustrates an outward path between a and b.

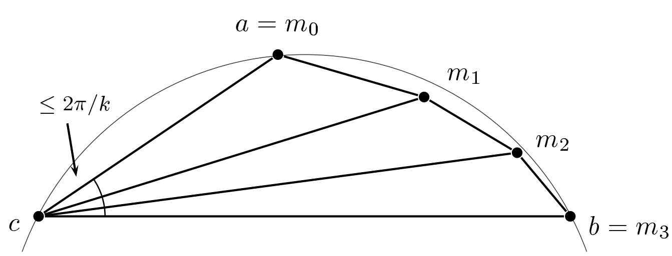

Figure 4: Illustration of an outward path.

Proposition 3.5. *In the recursive construction:*

(a) The pairs (a, m) and (b, m) satisfy hypothesis $\mathcal{H}'$.

(b) Point m satisfies hypothesis $\mathcal{H}$.

Proof. The triangle $\triangle cam$ can be partitioned into two regions: the first region is contained in $\triangle cab$, and hence its interior is empty (since (a, b) satisfies $\mathcal{H}'$), and the second region is contained in the region interior to $\bigcirc amb$ subtended by chord ab and closer to m, which is devoid of points of G by the choice of m. It follows that the pair (a, m) satisfies hypothesis $\mathcal{H}'$. Similarly, it can be shown that the pair (m, b) satisfies hypothesis $\mathcal{H}'$. This proves part (a).

Since a satisfies $\mathcal{H}$, there exists a circle C_a passing through c and a and containing no points of G in its interior. Similarly, there exists a circle C_b passing through b and c whose interior is devoid of points of G. Since m is interior to $\bigcirc cab$, c is interior to $\bigcirc amb$. Therefore, the circle C_m passing through c and m and internally tangent to circle $\bigcirc amb$ at m is well defined. Note that since m is exterior to C_a, C_m intersects the interior of chord ca, and hence, the region R_a inside C_m determined by segment ca and away from m is contained within C_a, and hence is devoid of points of G. The same holds true for the region R_b inside C_m determined by segment C_b and away from m. Circle C_m is contained in $C_a \cup C_b \cup R \cup R'$, and therefore its interior is devoid of points of G. This proves part (b). □

We are now ready to prove Lemma 3.2 for the case when no point of G lies inside $\triangle abc$. In this case we define the path in Lemma 3.2 to be the outward path between a and b.

Proof. (Proof of Lemma 3.2 for the case of the outward path.)

(*i*) It was proved in [20] that the weight of the outward path P between a and b, i.e., $\sum_{i=0}^{r-1} |m_i m_{i+1}|$, is bounded by $|\widehat{ab}|$. Therefore, it suffices to show that $|ca| + |\widehat{ab}| \leq (1 + 2\pi(k \cos \frac{\pi}{k})^{-1})|cb|$. With $\Theta = \angle bca$, we have $|\widehat{ab}| = 2\Theta \cdot |oa|$ and $\sin \Theta = |ab|/(2|oa|)$. We note that, since $|ca| \leq |cb|$, $|ca| + |\widehat{ab}|$ is largest when $|ca| = |cb|$, i.e., when ca and cb are symmetrical with respect to the diameter of circle $\bigcirc cab$. Therefore, we can assume that $|ca| = |cb|$. Since $|ca| = |cb|$, $\sin \frac{\Theta}{2} = \frac{|ab|}{2|cb|}$. It follows from the above facts that:

$$
\begin{aligned}
|ca| + |\widehat{ab}| &= |cb| + 2\Theta \cdot |oa| \\
&= |cb| + (\frac{\Theta}{\sin \Theta}) \cdot |ab| \\
&= |cb| + (\frac{\Theta}{\cos \frac{\Theta}{2}}) \cdot |cb| \qquad (1) \\
&\leq (1 + 2\pi(k \cos \frac{\pi}{k})^{-1})|cb|. \qquad (2)
\end{aligned}
$$

Equality (1) follows from the fact that $\sin \frac{\Theta}{2} = \frac{|ab|}{2|cb|}$ and the trigonometric identity $\sin \Theta = 2 \sin \frac{\Theta}{2} \cos \frac{\Theta}{2}$. Inequality (2) follows from $\Theta \leq 2\pi/k$ and $k > 2$.

(*ii*) If $m_i m_j$ is an edge in G in the region enclosed by ca, cb, and the edges of P, then there exists a circle passing through m_i and m_j and devoid of points of G; in particular, every point m_p, where $i < p < j$, is exterior to this circle. This, however, contradicts part (b) of Proposition 3.5 applied to point m_p, stating that there exists a circle passing through m_p and c and devoid of points in G (since such a circle would necessarily have to contain m_i or m_j).

(*iii*) By part (b) of Proposition 3.5, there exists a circle passing through c and m_i whose interior is devoid of points of G, in particular, of points m_{i-1} and m_{i+1}. It follows that $\angle m_{i-1} m_i m_{i+1} \geq \pi - \angle m_{i-1} c m_{i+1} \geq \pi - \angle bca \geq \pi - 2\pi/k$.

(*iv*) This follows from the fact that $\angle cam_1 \geq \angle cab \geq \pi/2 - \pi/k$. The last inequality is true because $|ca| \leq |cb|$ and $\angle bca \leq 2\pi/k$ in $\triangle cab$.

□

3.2 The inward path

We consider now the case when the interior of $\triangle abc$ contains points of G. Recall that ca and cb are edges of G such that ca is the shortest edge in the angular sector $\angle bca$, and such that $\angle bca \leq 2\pi/k$.

Let S be the set of points consisting of points a and b plus all the points interior to $\triangle abc$ (note that $c \notin S$). Let $CH(S)$ be the set of points on the convex hull of S. Then $CH(S)$ consists of points $n_0 = a$ and $n_s = b$, and points $n_1, \ldots, n_{s-1}$ of G interior to $\triangle abc$.

Proposition 3.6. *The following are true:*

(a) For every $i = 0, \ldots, s-1$: $|cn_i| \leq |cn_{i+1}|$, and

(b) For every $i = 0, \ldots, s-2$: $\angle n_i n_{i+1} n_{i+2} \geq \pi$, where $\angle n_i n_{i+1} n_{i+2}$ is the angle facing point c.

Proof. Part (a) follows from the facts that ca is the shortest edge in the sector $\angle bca$ (and hence $|ca| \leq |cn_i|$, for $i = 0, \ldots, s$), and points $n_0, \ldots, n_s$ are on $CH(S)$ in the listed order.

Part (b) follows from the convexity of $CH(S)$ because all these angles are exterior angles to $CH(S)$, and each interior angle to $CH(S)$ measures at most π. □

Proposition 3.7. *The following are true (please refer to Figures 5 and 6 for illustration):*

(a) For every $i = 0, \ldots, s-1$, the interior of $\triangle cn_i n_{i+1}$ is devoid of points of G.

(b) For every $i = 0, \ldots, s$, there exists a circle passing through cn_i whose interior is devoid of points of G.

Proof. Part (a) follows from the fact that the points $n_0, \ldots, n_s$ are on $CH(S)$, and hence the interior of the region enclosed by ca, cb, and the polygonal curve of $CH(S)$ determined by these points, is empty.

To prove part (b), let C_a, C_b be two circles passing through points c and a, and points c and b, respectively, whose interior is devoid of points of G. Note that the only points in the interior of region $R = C_a \cup C_b \cup \triangle cab$ are the points inside the region defined by the convex hull $CH(S)$. Since $n_i \in CH(S)$, there exists a line (L_i) passing through n_i such that all the points on $CH(S)$ reside in the closed half plane determined by (L_i) that does not contain point c. Let C_i be the circle passing through points c and n_i and tangent to (L_i) (at n_i).

If C_i intersects ca at a point c', then using Fact 2.3, it is not difficult to see that the region interior to C_i subtended by chord cc' and away from n_i is inside C_a and is thus devoid of points of G. Similarly, if C_i intersects cb. The remaining region inside C_i is contained in the region of $\triangle abc$ delimited by the boundary of $CH(S)$, and hence is also devoid of points of G.

It follows that the interior of C_i is devoid of point of G, and this proves part (b). □

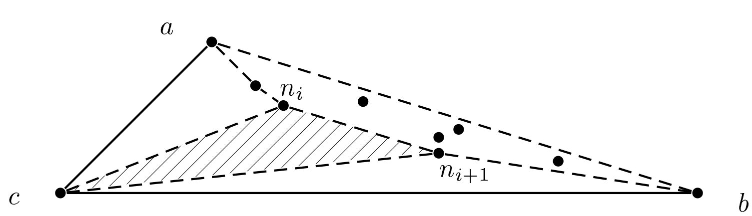

Figure 5: Illustration for part (a) of Proposition 3.7. $n_0 = a$, $n_s = b$, and points $n_1, \ldots, n_{s-1}$ form $CH(S)$. The interior of the shaded region is devoid of points of G.

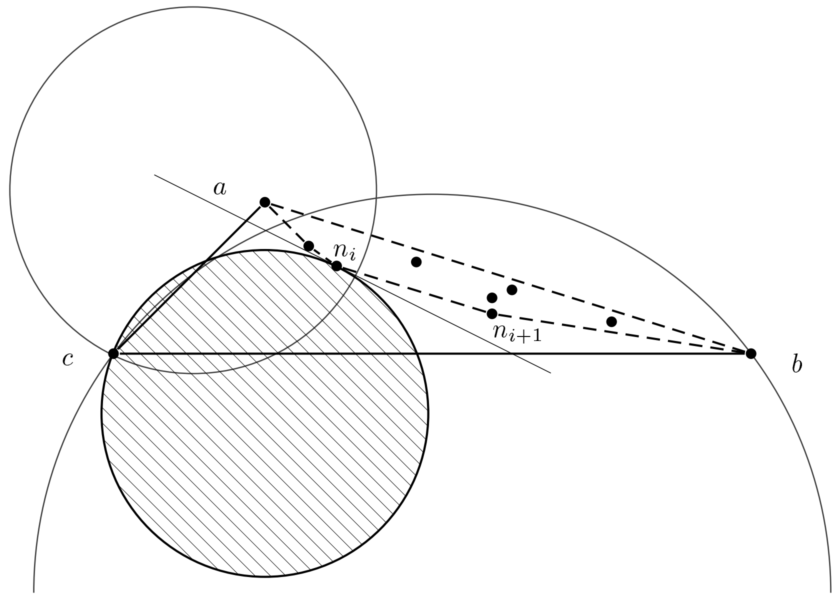

Figure 6: Illustration for part (b) of Proposition 3.7. The interior of the shaded circle C_i (passing through c and n_i) is devoid of points of G.

From part (b) of Proposition 3.7, every point n_i, $i = 0, \ldots, s$ satisfies hypothesis $\mathcal{H}$ in the previous subsection, and from part (a) of Proposition 3.7, every pair of points (n_i, n_{i+1}), $i = 0, \ldots, s-1$, satisfies hypothesis $\mathcal{H}'$. Therefore, from the previous subsection, for every pair of points (n_i, n_{i+1}), $i = 0, \ldots, s-1$, the outward path P_i between points n_i and n_{i+1} is well defined. Let $a = m_0, m_1, \ldots, m_r = b$ be the concatenation of the paths P_i, for $i = 0, \ldots, s-1$.

Definition 3.8. We call the path $a = m_0, m_1, \ldots, m_r = b$ constructed above the *inward path* between a and b.

Figure 7 illustrates an inward path between a and b.

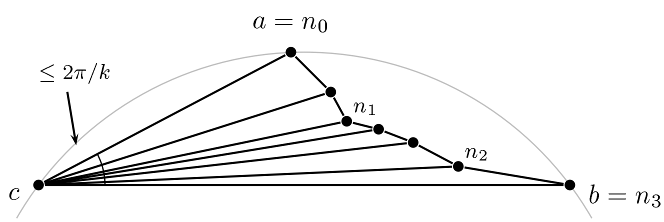

Figure 7: Illustration of an inward path.

We now prove Lemma 3.2 in the case when there are points of G interior to $\triangle abc$. In this case we define the path in Lemma 3.2 to be the inward path between a and b.

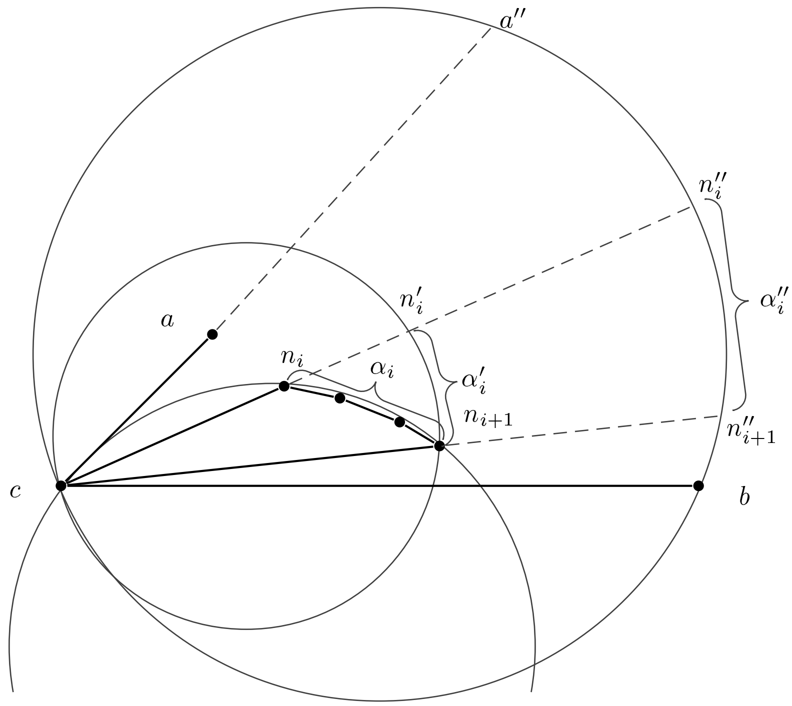

Figure 8: Illustration for the proof of part (i) of Lemma 3.2.

Proof. (Proof of Lemma 3.2 for the case of the inward path.)

(i) For two points x and y we denote by $[xy$ the half line (of the straight line xy) that starts at x and contains y.

Define a'' to be a point on the half-line $[ca$ such that $|ca''| = |cb|$, and let $(O'') = \bigcirc ca''b$. Denote by α'' the length of the arc of $\bigcirc ca''b$ subtended by chord $a''b$ and facing $\angle a''cb$. For every $i = 0, 1, \ldots, s-1$, we define arc α_i to be the arc of $\bigcirc cn_in_{i+1}$ subtended by chord n_in_{i+1} and facing $\angle n_icn_{i+1}$. For every $i = 0, 1, ..., s-1$, we define n_i' to be the point on the half-line $[cn_i$ such that $|cn_i'| = |cn_{i+1}|$ (recall that $|cn_i| \leq |cn_{i+1}|$ by part (a) of Proposition 3.6), (O_i) to be the circle $\bigcirc cn_i'n_{i+1}$, and α_i' to be the arc of (O_i) subtended by chord $n_i'n_{i+1}$ and facing $\angle n_i'cn_{i+1}$. Finally, for every $i = 0, \ldots, s-1$, we define n_i'' to be the point of intersection of the half-line $[cn_i$ and circle (O''), and α_i'' to be the arc of (O'') subtended by chord $n_i''n_{i+1}''$ and facing $\angle n_i''cn_{i+1}''$. Please refer to Figure 8 for illustration.

As shown in Subsection 3.1, the length of the outward path P_i between n_i and n_{i+1} is bounded by the length of α_i. Since the convex body C_1 delimited by cn_i, cn_{i+1} and α_i is contained inside the convex body C_2 delimited by cn_i', cn_{i+1} and α_i', by [2], the perimeter of C_1 is not larger than that of C_2. We get:

$$wt(P_i) \leq |n_in_i'| + \alpha_i', \qquad i = 1, \ldots, s-1. \tag{3}$$

The arc α_i' faces the inscribed angle $\angle n_i'cn_{i+1}$ in (O_i), and the arc α_i'' faces the inscribed angle $\angle n_i''cn_{i+1}''$ in (O''). Since the two angles $\angle n_i'cn_{i+1}$ and $\angle n_i''cn_{i+1}''$ have the same measure, and since the radius of (O_i) is not larger than that of (O''), we have $\alpha_i' \leq \alpha_i''$, for $i = 0, \ldots, s-1$.

(The radius of (O_i) can be easily computed to be $|cn_{i+1}|/(2\cos(\gamma/2))$, where $\gamma = \angle n_i'cn_{i+1}$, and the radius of (O'') can be computed to be $|cb|/(2\cos(\Theta/2))$, where $\Theta = \angle bca$. Since $\gamma \leq \Theta$ and $|cn_{i+1}| \leq |cb|$, the radius of (O_i) is not larger than that of (O'').)

It follows from Inequality (3) that:

$$wt(P_i) \leq |n_i n_i'| + \alpha_i'', \qquad i = 1, \ldots, s-1. \tag{4}$$

Using Inequalities (3) and (4) we get:

$$|ca| + \sum_{i=0}^{s-1} wt(P_i) \leq |ca| + \sum_{i=0}^{s-1} |n_i n_i'| + \sum_{i=0}^{s-1} \alpha_i''. \tag{5}$$

Noting that $\sum_{i=0}^{s-1} |n_i n_i'| = |cb| - |ca|$ and that $\sum_{i=0}^{s-1} \alpha_i'' = \alpha''$, it follows from Inequality (5) that:

$$\begin{aligned} |ca| + \sum_{i=0}^{s-1} wt(P_i) &\leq |cb| + \alpha'' \\ &\leq (1 + 2\pi(k\cos\frac{\pi}{k})^{-1})|cb|. \end{aligned} \tag{6}$$

The last inequality is true by the same argument used in the proof of part (i) in Lemma 3.2 for the case of outward path.

(ii) Since G is plane and the points $n_0, \ldots, n_s$ belong to $CH(S)$, if an edge between two points m_i and m_j exists, then m_i and m_j must belong to an outward path between two points n_p and n_{p+1} of $CH(S)$. However, this contradicts part (ii) of Lemma 3.2 for the case of the outward path applied to n_p and n_{p+1}.

(iii) For each $i = 0, \ldots, r$, either $m_i = n_j \in CH(S)$, or m_i is an intermediate point on the outward path between two points n_p and n_q in $CH(S)$. In the former case $\angle m_{i-1}m_i m_{i+1} \geq \angle n_{j-1}m_i n_{j+1} \geq \pi \geq (k-2)\pi/k$ for $k \geq 14$ (n_{j-1} and n_j are points before and after $m_i = n_j$ on $CH(S)$), by part (b) of Proposition 3.6. In the latter case, since $|cn_p| \leq |cn_q|$ and $\angle n_p c n_q \leq \angle bca \leq 2\pi/k$, it follows that $\angle m_{i-1}m_i m_{i+1} \geq (k-2)\pi/k$ by the proof of part (iii) of Lemma 3.2 applied to the outward path between n_p and n_q.

(iv) This follows from $|ca| = |cm_0| \leq |cm_1|$ and $\angle acm_1 \leq \angle bca \leq 2\pi/k$, in triangle $\triangle cam_1$.

□

3.3 The modified Yao step

We now augment the *Yao step* so edges forming the paths described in Lemma 3.2 are included in G'. Lemma 3.2 (parts (iii) and (iv)) says that consecutive edges on such paths form moderately large angles. The modified Yao step will ensure that consecutive edges forming large angles are included in G'. The algorithm is described in Figure 9.

Since the algorithm selects at most k edges incident on any point m and since only edges chosen by both endpoints are included in G', each point has degree at most k in G'.

Algorithm **Modified Yao step**

INPUT: *A Delaunay graph G; integer $k \geq 14$*
OUTPUT: *A subgraph G' of G of maximum degree k*

1. For every point $m \in G$ we do the following:
   1.1. define k disjoint cones of angle $2\pi/k$ with apex at m;
   1.2. in every non-empty cone, select the shortest edge incident on m in this cone;
   1.3. **for** every maximal sequence of $\ell \geq 1$ consecutive empty cones with apex at m:
      1.3.1. **if** $\ell > 1$ **then** select the first $\lfloor \ell/2 \rfloor$ unselected (during previous steps of the algorithm) incident edges on m clockwise from the sequence of empty cones and the first $\lceil \ell/2 \rceil$ unselected edges incident on m counterclockwise from the sequence of empty cones;
      1.3.2. **else** (i.e., $\ell = 1$) let mx and my be the incident edges on m clockwise and counterclockwise, respectively, from the empty cone; **if** either mx or my is selected **then** select the other edge (in case it has not been selected); **otherwise** select the shorter edge between mx and my breaking ties arbitrarily;
2. G' is the spanning subgraph of G consisting of edges selected by both endpoints.

Figure 9: The modified Yao step.

Before we complete the proof of Theorem 3.1, we show that the running time of the algorithm is linear. Note first that all edges incident on point m of degree Δ can be mapped to the k cones around m in linear time in Δ. Then, the shortest edge in every cone can be found in time $O(\Delta)$ (step 1.2 in the algorithm). Since k is a constant, selecting the $\ell/2$ edges clockwise (or counterclockwise) from a sequence of $\ell < k$ empty cones around m (step 1.3.1) can be done in $O(\Delta)$ time. Noting that the total number of edges in G is linear in the number of points (G is planar) completes the analysis.

To complete the proof of Theorem 3.1, all we need to do is show:

Lemma 3.9. *If edge $cb \in G$ is not selected by the algorithm, let $ca \in G$ be the shortest edge in the cone out of c to which cb belongs. Then the edges of the path P described in Lemma 3.2 are included in G' by the algorithm.*

Proof. For brevity, instead of saying that the algorithm **Modified Yao Step** selects an edge mx out of a point m, we will say that m selects edge mx. To get started, it is obvious that c will select edge $ca = cm_0$ ($a = m_0$).

By part (*iv*) of Lemma 3.2, the angle $\angle cam_1 \geq \pi/2 - \pi/k \geq 6\pi/k$ for $k \geq 14$. Therefore, at least two empty cones must fall within the sector $\angle cam_1$ determined by the two consecutive edges ca and am_1, and edges ac and am_1 will both be selected by a. Since edge ca is also selected by point c, edge $ac \in G'$.

By part (*iii*) of Lemma 3.2, for every $i = 1, 2, \ldots, r-1$, the angle $\angle m_{i-1}m_im_{i+1} \geq (k-2)\pi/k \geq 10\pi/k$ for $k \geq 12$, and hence at least four cones fall within the angular sector $\angle m_{i-1}m_im_{i+1}$. Since by part *(ii)* of Lemma 3.2 m_ic is the only possible edge inside the angular sector $\angle m_{i-1}m_im_{i+1}$, it is easy to see that regardless of the position of these four cones with respect to edge m_ic, m_i ends up selecting all edges m_im_{i-1}, m_im_{i+1} and m_ic in steps 1.2 and/or 1.3 of the algorithm. Since we showed above that a selects edge am_1, this shows that all edges m_im_{i+1}, for $i = 0, \ldots, r-2$, are selected by both their endpoints, and hence must be in G'. Moreover, edge $m_{r-1}m_r = m_{r-1}b$ is selected by point m_{r-1}.

We now argue that edge bm_{r-1} will be selected by b. First, observe that $|bm_{r-1}| \leq |\widehat{\text{ab}}| < |cb|$. Let cd be the other consecutive edge to cb in G (other than cm_{r-1}). Because c does not select b, it follows that $\angle m_{r-1}cd \leq 6\pi/k$. Otherwise, since cm_{r-1} and cb are in the same cone, two empty cones would fall within the sector $\angle bcd$ and c would select b. Since cb is an edge in G, by the characterization of Delaunay edges [14], $\angle cm_{r-1}b + \angle cdb \leq \pi$. By considering the quadrilateral $cdbm_{r-1}$, we have $\angle m_{r-1}cd + \angle dbm_{r-1} \geq \pi$. This, together with the fact that $\angle m_{r-1}cd \leq 6\pi/k$, imply that $\angle dbm_{r-1} \geq (k-6)\pi/k \geq 8\pi/k$, for $k \geq 14$. Therefore, $\angle dbm_{r-1}$ contains at least three cones of angle $2\pi/k$ out of b. If one of these cones falls within the angular sector $\angle cbm_{r-1}$ then, since $|m_{r-1}b| < |cb|$, bm_{r-1} must have been selected out of b.

Suppose now that $\angle cbm_{r-1}$ contains no cone inside and hence $\angle cbm_{r-1} < 4\pi/k$. If one of these three cones within sector $\angle dbm_{r-1}$ contains edge cb, then the remaining two cones must fall within $\angle dbc$ and bm_{r-1} will get selected out of b when considering the sequence of at least two empty cones contained within $\angle cbd$. Suppose now that all three empty cones fall within $\angle cbd$. Then we have $\angle cbd \geq 6\pi/k$.

If $\angle m_{r-1}cd \geq 4\pi/k$, then since $m_{r-1}c$ and cb belong to the same cone, the sector $\angle bcd$ must contain an empty cone. Because d is exterior to $\bigcirc cbm_{r-1}$, $\angle cbm_{r-1} < 4\pi/k$, and $\angle m_{r-1}cb \leq 2\pi/k$, it follows that $\angle cdb < \angle m_{r-1}cb + \angle cbm_{r-1} < 6\pi/k < \angle dbc$. Therefore, by considering the triangle $\triangle cdb$, we note that $|cb| < |cd|$. But then edge cb would have been selected by c in step 1.3 since the sector $\angle bcd$ contains an empty cone, a contradiction.

It follows that $\angle m_{r-1}cd \leq 4\pi/k$, and therefore $\angle m_{r-1}bd \geq (k-4)\pi/k \geq 10\pi/k$ for $k \geq 14$. This means that at least four cones are contained inside sector $\angle dbm_{r-1}$. It is easy to check now that regardless of the placement of the edge bc with respect to these cones, edge bm_{r-1} is always selected out of b by the algorithm. This completes the proof. ☐

Corollary 3.10. *A Euclidean minimum spanning tree (EMST) on $\mathcal{P}$ is a subgraph of G'.*

Proof. It is well known that a Delaunay graph (G) contains a EMST on its point-set [14]. If an edge cb is not in G', then, by Lemma 3.9, a path from c to b is included in G'. All edges on this path are no longer than cb; this is because $|ca| \leq |cb|$, and the weight of the canonical path between a and b is bounded by the length of $\overset{\frown}{ab}$, which is, in turn, bounded by $|cb|$. This latter fact follows from the facts that $|ac| \leq |bc|$, $\angle bca \leq 2\pi/k$, and $k \geq 14$.

Since G contains a EMST, and since whenever an edge cb is not included in G', a path between c and b consisting of edges each of length at most $|cb|$ is included in G', it follows that G' contains a EMST on $\mathcal{P}$. ☐

The fact that G' contains a EMST on $\mathcal{P}$ is crucial to the results in Section 5.

Since a Delaunay graph of a complete Euclidean graph of n points can be computed in time $O(n \lg n)$ [14] and has stretch factor at most C_{del}, we have the following theorem:

Theorem 3.11. *There exists an algorithm that, given a set $\mathcal{P}$ of n points in the plane, computes a plane geometric spanner of the complete Euclidean graph on $\mathcal{P}$ that contains a EMST, has maximum degree k, and has stretch factor $(1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$, where $k \geq 14$ is an integer. Moreover, the algorithm runs in time $O(n \lg n)$.*

4 Computing Spanners of UDGs Locally

In this section we generalize the centralized algorithm for complete Euclidean graphs from the previous section to a local distributed algorithm on UDGs. The results in the previous section do not carry over to unit disk graphs because not all the Delaunay graph edges of a point-set $\mathcal{P}$ are unit disk edges. However, if U is the unit disk graph on the points in $\mathcal{P}$ and $UDel(U)$ is the subgraph of the Delaunay graph on $\mathcal{P}$ obtained by deleting edges of length greater than one unit, then $UDel(U)$ is a connected, plane, spanning subgraph of U with stretch factor bounded by C_{del} (see [5, 23]). Therefore, if we apply the results from the previous section to $UDel(U)$ and observe that all edges on the path defined in Lemma 3.2 must be unit disk edges (given that edges ca and cb are), it is easy to see that Theorem 3.1 and Theorem 3.11 carry over to unit disk graphs. The only problem, however, is that the construction of $UDel(U)$ cannot be done locally.

To solve this problem, Wang et al. [23, 24] introduced a subgraph of U called $LDel^{(2)}(U)$, defined in a style similar to Definition 2.2 as follows:

Definition 4.1. ([8, 28]) An edge xy of U is in $LDel^{(2)}(U)$ if and only if there exists a circle through points x and y whose interior contains no point of U that is a 2-hop neighbor (in U) of x or y.

It was shown in [23, 24] that $LDel^{(2)}(U)$ is a plane supergraph of $UDel(U)$, and hence also has stretch factor bounded by C_{del}. Moreover, $LDel^{(2)}(U)$ can be computed by a 3-local distributed algorithm as follows:

First, every point learns its 3-hop neighborhood using the local distributed algorithm given in [19]. Then every point x will decide, for every incident edge xy, whether it is in $LDel^{(2)}(U)$ as follows: xy is in $LDel^{(2)}(U)$ if and only if there exists a point z in the 3-hop neighborhood of x such that the interior of $\bigcirc xyz$ is devoid of 2-hop neighbors of x or y.

We will use $G = LDel^{(2)}(U)$ as the underlying subgraph of U to replace the Delaunay graph G used in the previous section. We note that G is plane and a supergraph of $UDel(U)$, and hence has stretch factor C_{del}. To translate our results to unit disk graphs, we need to show that the inward and outward paths are still well defined in G. We first need some structural results.

Lemma 4.2. *Let ca and cb be two edges in U such that $\angle bca \leq 2\pi/k$, where $k \geq 14$ is an integer. Let m be any point in the region inside $\bigcirc bca$ enclosed by the angular sector $\angle bca$ (the sector which measures at most $2\pi/k$). Then $|cm| \leq 1/\cos(\pi/k)$.*

Proof. Let o be the center of $\bigcirc bca$. If o is in the region interior to $\bigcirc bca$ subtended by ca and away from b, then ca is the longest chord in the region interior to $\bigcirc bca$ subtended by ca and closer to b. In particular, $|cm| \leq |ca| \leq 1 \leq 1/\cos(\pi/k)$. The case is analogous if o belongs to the region interior to $\bigcirc bca$ subtended by cb and away from a. Therefore, we can assume that o belongs to the region inside $\bigcirc bca$ facing the angular sector $\angle bca$. Clearly, cm is longest in this case when it is a diameter of $\bigcirc bca$. One of the two angles $\angle mca$ and $\angle mcb$ is at most π/k; assume, without loss of generality, that $\angle mca \leq \pi/k$. By considering the right angle triangle $\triangle mac$, we have $\cos(\angle mca) = |ca|/|cm| \geq \cos(\pi/k)$. Since $|ca| \leq 1$, it follows that $|cm| \leq 1/\cos(\pi/k)$. □

Lemma 4.3. *Let x and y be two points in U such that $|xy| \leq 1/\cos(\pi/k)$, where $k \geq 14$ is an integer. Let (O_{xy}) be any circle passing through x and y whose interior is devoid of any 2-hop neighbors of x and of y, and let z be any point on circle (O_{xy}). Then the region R interior to (O_{xy}) subtended by xy and away from z is devoid of any 2-hop neighbors of z. (Please refer to Figure 10 for illustration.)*

Proof. Let o_{xy} be the center of (O_{xy}).

If o_{xy} does not belong to R, then it is easy to see that any point m in region R is of distance at most $(\sqrt{2}/2)|xy|$ from either x or y (this upper bound corresponds to the case when xy is a diameter of (O_{xy}) and m is the point on the boundary of R equidistant from x and y). Since by the hypothesis we have $|xy| \leq 1/\cos(\pi/k)$, it follows that the distance from m to either x or y is at most $\sqrt{2}/(2\cos(\pi/k)) < 1$ for $k \geq 14$. Consequently, m is a neighbor of either x or y, contradicting the hypothesis that the interior of (O_{xy}) is devoid of any 2-hop neighbors of x or y.

Therefore, if o_{xy} is not in R, then R does not contain any point of U, and in particular, R does not contain any 2-hop neighbors of z.

Suppose now that o_{xy} belongs to R. Let R' be the region interior to (O_{xy}) subtended by xy and closer to z. By the same argument made above, z must be a neighbor of x or y (or both). Consequently, region R cannot contain any 1-hop neighbor of z since such a neighbor would be a 2-hop neighbor of x or y, contradicting the hypothesis. Therefore, it suffices to show that R does not contain any point q whose hop-distance from z is exactly 2. Proceed by contradiction. Assume that $q \in R$ is a 2-hop neighbor of z, and let m be a common neighbor in U of both z and q. Note that m and q cannot be neighbors of x nor of y. Therefore, xq, xm, yq, and ym are not edges of U.

Since z is a neighbor of x or y, m is a 2-hop neighbor of x or y, and consequently, m is not inside (O_{xy}). Moreover, since mz is an edge, m is closer to z than to x and y. Therefore, m belongs to the angular sector determined by the two perpendicular bisectors of zy and zx starting at o_{xy}. Since m is not interior to (O_{xy}), m and o_{xy} must be on opposite side of line xy, and since q and o_{xy} are on the same side of line xy (they both belong to R), segment mq must intersect the line xy. We distinguish two cases based on whether mq intersects line xy internally (inside segment xy) or externally.

Suppose first that mq intersects the interior of xy. Consider the convex quadrilateral $xqym$. Note that xy cannot be an edge of U in this case because otherwise, since mq is an edge of U, the two diagonals mq and xy of the quadrilateral are shorter than all the sides of this quadrilateral (since none of the the sides is an edge of U). Since mq is the shortest edge in the two triangles $\triangle qmy$ and $\triangle mqx$, it is easy to see that $\angle yqx + \angle ymx \geq \angle qxm + \angle qym$. Consequently, $\angle yqx + \angle ymx \geq \pi$. It follows that one of the two angles $\angle yqx$ and $\angle ymx$ is at least $\pi/2$; assume $\angle ymx \geq \pi/2$ and the proof is similar in the other case. Consider $\triangle xmy$ and note that xy is a longest side in this triangle. Consequently, either xm or my has length bounded by $(\sqrt{2}/2)|xy| \leq 1$, a contradiction.

Suppose now that mq intersects line xy exterior to segment xy. Assume, without loss of generality, that the intersection point is closer to point y. Then point y must be interior to $\triangle zqm$. Since mq is an edge of U and yq and ym are not, mq is the shortest side in $\triangle qym$. Consequently, $\angle qym \leq \pi/3$. It follows that $\angle myz \geq 2\pi/3$ (since $\angle qyz \leq \pi$). However, this contradicts the fact that $|mz| \leq |my|$ (since mz is an edge in U and my is not).

This completes the proof. $\square$

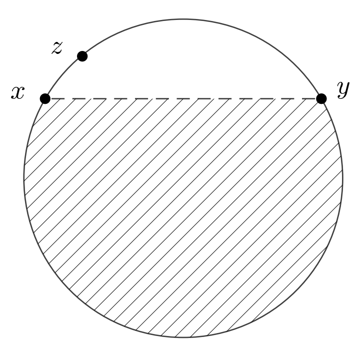

Figure 10: Illustration for Lemma 4.3. If the interior of circle (O_{xy}) is devoid of 2-hop neighbors of x and y, then the interior of the shaded region is devoid of 2-hop neighbors of z.

The following proposition is parallel to Proposition 3.3:

Proposition 4.4. *Let c, a, b be points of G such that:*

- $|ca|$ *and* $|cb|$ *are at most* $1/\cos(\pi/k)$*, for* $k \geq 14$*,*
- *there is a circle* C_a *through* c *and* a *whose interior contains no two-hop neighbor of* c *or* a*, and*
- *there is a circle* C_b *through* c *and* b *whose interior contains no two-hop neighbor of* c *or* b*.*

Then the region interior to $\bigcirc abc$ *subtended by chord* ca *and away from* b *and the region interior to* $\bigcirc abc$ *subtended by chord* cb *and away from* a *contain no 2-hop neighbors (in* U*) of* a*,* b *or* c*.*

Proof. By symmetry, it is enough to prove the proposition for the region interior to $\bigcirc abc$ subtended by chord ca and away from b. By Fact 2.3, the region interior to $\bigcirc abc$ subtended by chord ca and away from b is inside C_a, and hence contains no 2-hop neighbors of a or c. Since $|ca| \leq 1/\cos(\pi/k)$, it follows from Lemma 4.3 that the region interior to $\bigcirc abc$ subtended by chord ca and away from b contains no 2-hop neighbors of b as well. □

Let ca and cb be two edges in G such that $\angle bca \leq 2\pi/k$, where $k \geq 14$ is an integer, and such that ca is the shortest edge in the angular sector $\angle bca$. Note that by Lemma 4.2, for any point m residing in the angular sector $\angle bca$, we have $|cm| \leq 1/\cos(\pi/k)$.

We start by showing that the results in Subsection 3.1 about the existence of an outward path between a and b translate to the UDG case.

4.1 The outward path

We assume in this subsection that no point of G is interior to $\triangle abc$. As we did in the previous section, we will define two hypotheses with respect to a point c of G. For a point x in G, we define the following hypothesis, which is analogous to hypothesis $\mathcal{H}$ in Subsection 3.1:

$\mathcal{H}_{\mathcal{U}}$: there exists a circle passing through c and x containing no 2-hop neighbors of c or x in its interior.

For a pair of points (x, y) in the sector $\angle bca$, we define the following hypothesis which is analogous to hypothesis $\mathcal{H}'$:

$\mathcal{H}'_{\mathcal{U}}$: the interior of $\triangle cxy$ is devoid of points of G.

Since both ca and cb are edges in G, there are circles C_a and C_b passing through c, a and c, b, respectively, whose interiors contain no two-hop neighbor (in U) of c, a and c, b, respectively. Therefore, both a and b satisfy hypothesis $\mathcal{H}_{\mathcal{U}}$ with respect to c, and since—by our assumption—no point of G is interior to $\triangle abc$, the pair (a, b) satisfies hypothesis $\mathcal{H}'_{\mathcal{U}}$ with respect to c.

Then, given points a and b that satisfy hypothesis $\mathcal{H}_{\mathcal{U}}$ and that the pair (a, b) satisfies hypothesis $\mathcal{H}'_{\mathcal{U}}$ with respect to c, the canonical path between a and b is constructed exactly as in the previous section (with the two instances of "point" changed to "two-hop neighbor of a and b"). So, to make the construction work, we need to prove an analogous proposition to Proposition 3.5. The existence of the outward path satisfying the properties of Lemma 3.2 will then follow immediately.

Proposition 4.5. *In the recursive construction:*

(a) *The pairs* (a, m) *and* (b, m) *satisfy the hypothesis* $\mathcal{H}'_{\mathcal{U}}$.

(b) *Point* m *satisfies the hypothesis* $\mathcal{H}_{\mathcal{U}}$.

Proof. The proof of part (a) is exactly the same as that of part (a) of Proposition 3.5, and is omitted.

Since a satisfies $\mathcal{H}_{\mathcal{U}}$, there exists a circle C_a passing through c and a and containing no 2-hop neighbors of c or a in its interior. Similarly, there exists a circle C_b passing through b and c whose interior is devoid of any 2-hop neighbors of b or c. Since m is interior to $\bigcirc cab$, c is interior to $\bigcirc amb$. Therefore, the circle C_m passing through c and m and internally tangent to circle $\bigcirc amb$ at m is well defined. Note that since m is exterior to C_a, C_m intersects the interior of chords ca, and hence, the region R_a of C_m subtended by segment ca and away from m is contained in the region R'_a interior to $\bigcirc cma$ subtended by chord ca and away from m, which in turn is contained in the region interior to C_a subtended by chord ca and away from m by Fact 2.3. Therefore, the region R'_a, and consequently R_a, is devoid of 2-hop neighbors of c and a, and by Lemma 4.3 (since $|ca| \leq 1/\cos(\pi/k)$) of 2-hop neighbors of m. The same holds true for the region R_b interior to C_m determined by segment C_b and away from m. Now circle C_m is contained in $R_a \cup R_b \cup R \cup R'$ (refer to the previous section for the definition of R and R'), and therefore its interior is devoid of 2-hop neighbors of c and of m. This proves part (b). □

With the above proposition, the definition of the outward path between a and b translates transparently to the UDG case, of course under the assumption that $\angle bca \leq 2\pi/k$. We prove that the properties of this path listed in Lemma 3.2 translate as well to the case of UDG.

Proof. (Proof of Lemma 3.2 for the case of outward path.)

(*i*) The proof of this part is exactly the same as that of part (*i*) of Lemma 3.2 in Subsection 3.1.

(*ii*) The proof of this part is exactly the same as that of part (*ii*) of Lemma 3.2 in Subsection 3.1, after noting that the points on the canonical path form a clique, and hence any two points are 2-hop neighbors in U. The latter statement is true because all these points reside in the region of $\bigcirc bca$ subtended by chord ab and away from c, and $|ab| \leq 1$.

(*iii*) The proof of this part is exactly the same as that of part (*iii*) of Lemma 3.2 in Subsection 3.1, after noting that any two points on the canonical path are 2-hop neighbors in U.

(*iv*) The proof of this part is exactly the same as that of part (*iv*) of Lemma 3.2 in Subsection 3.1.

□

4.2 The inward path

The definition of the inward path translates directly from that in Subsection 3.2. The properties of the inward path proved in Lemma 3.2 in Subsection 3.2 would also follow directly if we can prove Proposition 3.6 and Proposition 3.7 for the UDG case. The proof of Proposition 3.6 is exactly the same as that in Subsection 3.2. The proof of part (a) of Proposition 3.7 is also exactly the same as that in Subsection 3.2. The proof of part (b) is the same as that in Subsection 3.2 after noting that any two points in $n_0, \ldots, n_s$ are 2-hop neighbors of each other since they reside in $\triangle bca$, and that the distance between c and any point in $\angle bca$ is at most $1/\cos(\pi/k)$ (and hence Lemma 4.3 can be applied).

Finally, the same **Modified Yao Step** algorithm in Section 3.3, after setting $G = LDel^{(2)}(U)$, gives a 3-local distributed algorithm. Therefore, we have the following theorem:

Theorem 4.6. *There exists a 3-local distributed algorithm that, given a set $\mathcal{P}$ of n points in the plane, computes a plane geometric spanner of the unit disk graph on $\mathcal{P}$ that contains a EMST on $\mathcal{P}$, has maximum degree k, and has stretch factor $(1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$, for any integer $k \geq 14$. Moreover, the algorithm exchanges no more than $O(n)$ messages in total, and has a local processing time of $O(n \lg n)$.*

5 Computing Lightweight Spanners of Euclidean Graphs

In this section we present a centralized algorithm that constructs a bounded-degree plane lightweight spanner of the complete Euclidean graph $\mathcal{E}$. We first need the following structural results.

Let G be a plane graph. Fix a spanning tree T of G. Call an edge $e \in E(T)$ a *tree edge* and an edge $e \in E(G) - T$ a *non-tree edge*. Every non-tree edge induces a unique cycle in the graph $T + e$ called the *fundamental cycle* of e. Since T is embedded in the plane, we can talk about the *fundamental region* of e, which is the closed region in the plane enclosed by the fundamental cycle of e (other than the outer face of $T + e$). See Figure 11 for an illustration of fundamental regions and fundamental cycles.

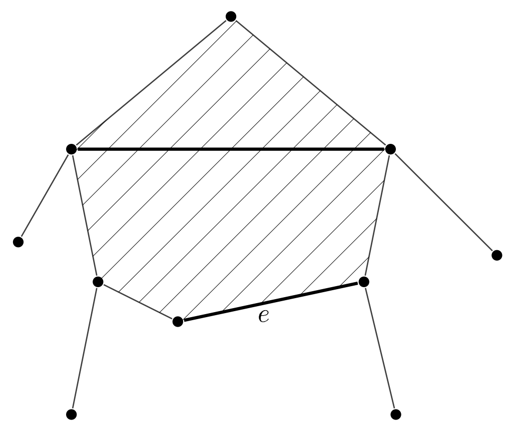

Figure 11: Illustration for fundamental regions and fundamental cycles. The light edges are tree edges and the thick/bold edges are non-tree edges. The shaded region is the fundamental region of a non-tree edge e and its boundary is the fundamental cycle of e. Note that a fundamental region is not necessarily a face.

Definition 5.1. Define a relationship $\preceq$ on the set $E(G)$ as follows. For every edge e, $e \preceq e$. If e is a tree edge and e' is a non-tree edge then $e \preceq e'$. For two non-tree edges e and e', $e \preceq e'$ if and only if e is contained in the fundamental region of e'.

It is not difficult to verify that $\preceq$ is a partial order relation on $E(G)$, and hence $(E(G), \preceq)$ is a partially ordered set (POSET). Note that any two distinct tree edges are not comparable by $\preceq$, and that every tree edge is a minimal element in $(E(G), \preceq)$. Therefore, we can topologically sort the edges in $E(G)$ to form a list $\mathcal{L} = \langle e_1, \ldots, e_r \rangle$, in which no non-tree edge appears before a tree edge, and such that if $e_i \preceq e_j$ then e_i does not appear after e_j in $\mathcal{L}$.

Lemma 5.2. *Let e_i be a non-tree edge. Then there exists a unique face F_i in G containing e_i such that every edge e_j of F_i satisfies $e_j \preceq e_i$.*

Proof. Let F_i be the face of G containing e_i and residing in the fundamental region of e_i, and let e_j be an edge on F_i. Since e_j is on F_i, e_j is contained in the fundamental region of e_i. By the definition of $\preceq$, we have $e_j \preceq e_i$. This shows the existence of such a face F_i.

To prove the uniqueness of F_i, suppose that there is another distinct face F_i' with the above properties. Since every edge e_j on F_i' satisfies $e_j \preceq e_i$, every edge on F_i' is contained in the fundamental region of e_i, and hence the whole face F_i' is contained in the fundamental region of e_i. This means that there are two distinct faces containing e_i that are enclosed within the fundamental cycle of e_i. This contradicts the planarity of G. $\square$

We will call the unique face associated with a non-tree edge e_i, described in Lemma 5.2, the *fundamental face* of e_i. See Figure 12 for an illustration of fundamental faces and the partial ordering among the corresponding non-tree edges.

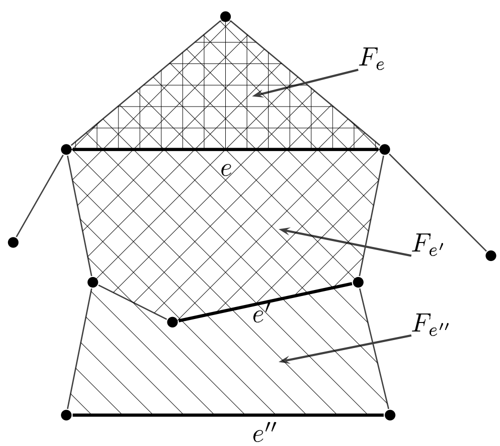

Figure 12: Illustration for the fundamental faces of e, e', and e''. The fundamental region of e is contained in the fundamental region of e', which is in turn, contained in the fundamental region of e''. Hence, $e \preceq e' \preceq e''$.

The following result is a consequence of the proof of Theorem 2 in [1]. A similar, but less general result, was also proved earlier by Levcopoulos and Lingas [22].

Theorem 5.3. *(Theorem 2 in [1]) Let G be a connected weighted planar graph with nonnegative weights satisfying the following property: for every cycle C in G and every edge $e \in C$, $wt(C) \geq \lambda \cdot wt(e)$ for some constant $\lambda > 2$. Then $wt(G) \leq (1 + \frac{2}{\lambda-2}) \cdot wt(T)$, where T is a minimum spanning tree of G.*

The following corollary can be proved using the same techniques used in the proof of Theorem 2 in [1]. In order not to repeat the complete proof of Theorem 2 in [1], we describe the result and refer the reader to Corollary 3.8 in [18] for a complete proof.

Corollary 5.4. *Let G be a connected weighted plane graph with nonnegative weights, and let T be a spanning tree in G. Let $\lambda > 2$ be a constant. Suppose that for every edge $e \in E(G) - T$ we have $wt(F_e) \geq \lambda \cdot wt(e)$, where F_e is the boundary cycle of the fundamental face of e in G. Then $wt(G) \leq (1 + \frac{2}{\lambda-2}) \cdot wt(T)$.*

Now we are ready to show how to construct a bounded-degree lightweight spanner of $\mathcal{E}$. By Theorem 3.11, given an integer parameter $k \geq 14$, we can construct in $O(n \log n)$ time a plane spanner G' of $\mathcal{E}$ containing a EMST on $\mathcal{P} = V(\mathcal{E})$, of degree at most k, and of stretch factor $\rho = (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$. The spanner G', however, may not be of light weight. Therefore, we need to discard edges from G' so that the resulting subgraph is of light weight, while at the same time not affecting the stretch factor of G' by much. To do so, since G' is a plane graph containing a EMST on $\mathcal{P}$, we would like to employ Corollary 5.4. We will show next how to prune the set of edges in G' so that the weight of every fundamental face F_e of a non-EMST edge e in G' satisfies $wt(F_e) \geq \lambda \cdot wt(e)$ ($\lambda > 2$ is a constant).

Let T be a EMST on $\mathcal{P}$ that is contained in G'. As described above, we can order the non-tree edges in G', with respect to the fixed tree T, by the partial order $\preceq$ described in Definition 5.1. Let $\mathcal{L}' = \langle e_1, e_2, \ldots, e_s \rangle$ be the sequence of non-tree edges in G' sorted in non-decreasing order with respect to the partial order $\preceq$. Note that, by the definition of the partial order $\preceq$, if we add the edges in $\mathcal{L}'$ to T in the respective order they appear in $\mathcal{L}'$, once an edge e_i is added to form a fundamental face in the partially-grown graph, this fundamental face will remain a face in the resulting graph after all the edges in $\mathcal{L}'$ have been added to T. That is, the face will not be affected (i.e., changed/split) by the addition of any later edge in this sequence.

Given a constant $\lambda > 2$, to construct the desired lightweight spanner G, we first initialize G to the EMST T. We consider the non-tree edges of G' in the order that they appear in $\mathcal{L}'$. Inductively, suppose that we have processed the edges $e_1, \ldots, e_{i-1}$ in $\mathcal{L}'$. To process edge e_i, let F_i be the fundamental face of e_i in $G + e_i$. If $wt(F_i) > \lambda \cdot wt(e_i)$, we add e_i to G; otherwise, e_i is not added to G. This completes the description of the construction process. Let G be the resulting graph at the end of the construction process.

Lemma 5.5. *Given the set of n points $\mathcal{P}$ in the plane, the graph G can be constructed in $O(n \log n)$ time.*

Proof. We first describe how to compute the sequence $\mathcal{L}'$.

The bounded-degree plane spanner G' of $\mathcal{E}$ can be constructed in $O(n \log n)$ time by Theorem 3.11. Since every point in G' has bounded degree, and since G' is a geometric plane graph, in $O(1)$ time per point, and hence in $O(n)$ time, for every point in G', we can list its incident edges in clockwise order. Moreover, since G' has $O(n)$ edges, the EMST T contained in G' can be computed in $O(n \log n)$ time by a standard MST algorithm.

Note that if we can properly contract the edges in T to obtain a single point with self-loops corresponding to all the non-tree edges in G', then the incidence ordering of these self-loops around the point reveals a sequence $\mathcal{L}'$. (See Figure 13 for an illustration.) We can accomplish this by traversing the tree edges of G' starting at a point on the outer face of G' (e.g., the point with the smallest x-coordinate). As a point v is visited in this traversal, the non-tree edges incident on v that appear between the entering edge and the exiting edge are appended to a list L according to their clockwise order fixed above. At the end of the traversal, the list L will correspond to an incidence ordering in which the self-loops appear around a point that results from contracting the edges in T. Finally the sequence $\mathcal{L}'$ can be derived from the list L by removing the first occurrences of each edge in L. Clearly, this process can be carried out in $O(n)$ time.

After computing $\mathcal{L}'$, we initialize G to the EMST T. As we consider the edges in $\mathcal{L}'$, when we add an edge e in $\mathcal{L}'$ to form a fundamental face F_e in $G + e$, we need to check whether the fundamental face F_e satisfies the condition $wt(F_e) > \lambda \cdot wt(e)$. To do so, we need to traverse the edges on F_e. If e is not subsequently added to G, we might need to traverse some edges on F_e multiple times when we later consider edges that are larger than e in the ordering $\preceq$. To avoid this problem, we can do the following. If we decide to add an edge to G, we add this edge and mark it as a "real" edge of G. On the other hand, if e is not to be added to G, we still add e to G but we mark it as a "virtual" edge of G, and assign it a weight equal to the weight of its fundamental face. The graph G will consist of the tree T plus the set of edges that were marked as real edges. This way each edge in G is traversed at most twice (as every edge appears in at most two faces), and the running time is kept $O(n)$.

It follows that G can be constructed in $O(n \log n)$ time, and the proof is complete. $\square$

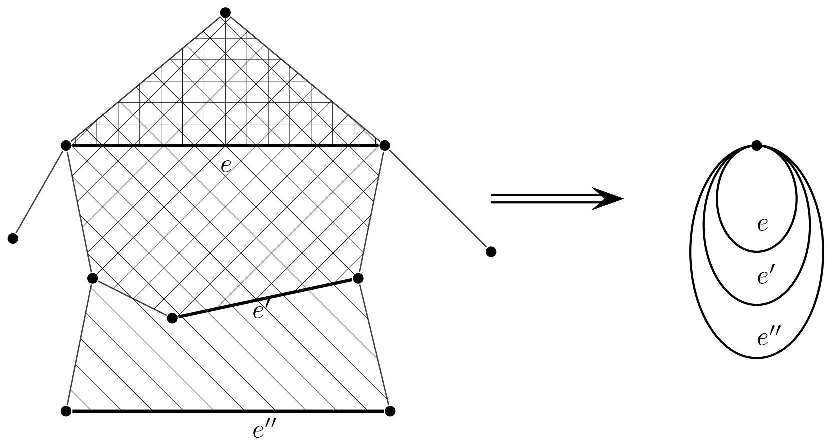

Figure 13: Illustration for the contraction of the tree edges in the proof of Lemma 5.5. The incidence ordering of the self-loops around the point reveals a sequence $\mathcal{L}'$.

Theorem 5.6. *For any integer parameter $k \geq 14$ and any constant $\lambda > 2$, the subgraph G of $\mathcal{E}$ constructed above is a plane spanner of $\mathcal{E}$ containing a EMST on $\mathcal{P}$, whose degree is at most k, whose stretch factor is $(\lambda - 1) \cdot \rho$, where $\rho = (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$, and whose weight is at most $(1 + \frac{2}{\lambda-2}) \cdot wt(EMST)$. Moreover, G can be constructed in $O(n \log n)$ time.*

Proof. The planarity and degree bound of G follow from the fact that G is a subgraph of G'. By construction, G contains a EMST on $\mathcal{P}$, and every fundamental face F_e of a non-tree edge e in G satisfies $wt(F_e) \geq \lambda \cdot wt(e)$. Therefore, by Corollary 5.4, we have $wt(G) \leq (1 + \frac{2}{\lambda-2}) \cdot wt(\text{EMST})$. Since by Lemma 5.5 G can be constructed in $O(n \log n)$ time, it suffices to show that the stretch factor of G with respect to $\mathcal{E}$ is $(\lambda - 1) \cdot \rho$.

Note that G' has stretch factor ρ with respect to $\mathcal{E}$. If an edge e_i is in G' but not in G, then by the construction of G, when the edge e_i is considered, the fundamental face F_i of e_i in $G + e_i$ satisfies $wt(F_i) \leq \lambda \cdot wt(e_i)$ (otherwise, the edge e_i would have been added). Therefore, when edge e_i was considered, G contained a path between the endpoints of e_i whose weight is at most $(\lambda - 1) \cdot wt(e_i)$. This path will remain in G after all edges in $\mathcal{L}'$ have been considered. Therefore, every edge in $E(G') - E(G)$ is stretched by a factor at most $\lambda - 1$. Since G' has stretch factor ρ with respect to $\mathcal{E}$, it follows that the stretch factor of G with respect to $\mathcal{E}$ is $(\lambda - 1) \cdot \rho$. This completes the proof. $\square$

6 Computing Lightweight Spanners of UDGs Locally

In this section we present a local distributed algorithm that constructs a bounded-degree plane lightweight spanner of a unit disk graph U.

By Theorem 4.6, there exists a 3-local distributed algorithm that, given a unit disk graph U and an integer parameter $k \geq 14$, constructs a plane spanner G' of U containing a EMST on $V(U)$, of degree at most k and stretch factor $\rho = (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$. Again, G' might not be of light weight, and we need to discard edges from G' so that the obtained subgraph is of light weight. Ultimately, we would like to be able to apply Theorem 5.3. However, a serious problem, which was not present previously in the centralized model, poses itself here in the local model: the removal of the edges from the spanner by different points in the graph needs to be coordinated. This problem was overcome in the centralized model by using a global ordering among the edges of the spanner. Clearly, no local distributed algorithm is capable of computing the global partial order described in Definition 5.1. To coordinate the removal of edges, we use a clustering technique.

Fix an infinite rectilinear tiling $\mathcal{T}$ of the plane whose tiles are $\ell \times \ell$ squares, for some positive constant ℓ to be determined later. Assume, without loss of generality, that one of the tiles in $\mathcal{T}$ has its bottom-left corner coinciding with the origin $(0, 0)$, and that this fact is known to the points in U. Note that this assumption is justifiable in practice because an absolute reference system usually exists (a coordinate system, for example). Therefore, any point in U can determine (using simple arithmetic operations) which tile of $\mathcal{T}$ it resides in. We start with the following simple fact whose proof is easy to verify.

Fact 6.1. *Let C be a cycle of weight at most ℓ. The orthogonal projection [2] of C on any straight line has weight at most $\ell/2$.*

Let T_I be the translation with vector $(0, 0)$ (the identity translation), T_H the translation of vector $(\ell/2, 0)$ (horizontal translation), T_V the translation of vector $(0, \ell/2)$ (vertical translation), and T_D the translation of vector $(\ell/2, \ell/2)$ (diagonal translation). We have the following simple lemma.

Lemma 6.2. *Let C be any cycle of weight at most ℓ. There exists a translation T in $\{T_I, T_H, T_V, T_D\}$ such that the translate of C, $T(C)$, resides in a single tile of $\mathcal{T}$.*

Proof. If C resides within a single tile of $\mathcal{T}$ then clearly translation T_I serves the purpose. If C resides within exactly two horizontal (resp. vertical) tiles of $\mathcal{T}$, then these two tiles must be adjacent, and it is easy to verify using Fact 6.1 that translation T_H (resp. T_V) serves the purpose. Finally, if C resides within more than two tiles of $\mathcal{T}$, then again, using Fact 6.1, it can be easily verified that translation T_D serves the purpose. □

Even though a cycle of weight ℓ may not reside within a single tile of $\mathcal{T}$, Lemma 6.2 shows that by affecting some translation T in $\{T_I, T_H, T_V, T_D\}$, the translate of C under T will reside in a single tile. For each translation T in $\{T_I, T_H, T_V, T_D\}$, the points in G whose translates under T reside in a single tile will form a *cluster*. Then, these points will coordinate the detection and removal of the low-weight cycles residing in the cluster by applying a centralized algorithm to the cluster. Since the clusters do not overlap, and since each cluster works as a centralized unit, this maintains the stretch factor under control, while ensuring the removal of every low weight

[2] By the orthogonal projection of C on a given line we mean the points that are the orthogonal projections of the points in C on the given line. Note that, by the continuity of the curve C, this set of points is a line segment.

cycle. The centralized algorithm that we apply to each cluster is the standard greedy algorithm that has been extensively used (see for example [1]) to compute lightweight spanners. Given a graph H and a parameter $\alpha > 1$, this greedy algorithm sorts the edges in H in a non-decreasing order of their weight, and starts adding these edges to an empty graph in the sorted order. The algorithm adds an edge ab to the growing graph if and only if no path between a and b whose weight is at most $\alpha \cdot |ab|$ exists in the growing graph. We will call this algorithm **Centralized Greedy**. The following properties about this greedy algorithm are known:

Fact 6.3. *Let H be a subgraph of the complete Euclidean graph E, and let $\alpha > 1$ be a constant. Let H' be the subgraph of H constructed by the algorithm* **Centralized Greedy** *when applied to H with parameter α. Then:*

(i) H' is a spanner of H with stretch factor α.

(ii) H' contains a MST of H.

(iii) For any cycle C in H' and any edge e on C, $wt(C) > (1+\alpha) \cdot wt(e)$.

Lemma 6.4. *If H is a plane graph, then the algorithm* **Centralized Greedy** *can be implemented to run in $O(n^2 \lg n)$ time on H, where n is the number of points in H.*

Proof. Since H is plane, it has $O(n)$ edges and they can be sorted in $O(n \lg n)$ time. Moreover, for the same reason, a shortest-path query between any two points in H can be answered in $O(n \lg n)$ time. It follows that the algorithm **Centralized Greedy** can be implemented to run in $O(n^2 \lg n)$ time on H. □

Lemma 6.5. *Let t_0 be a tile in $\mathcal{T}$, and let U_{t_0} be the subgraph of U induced by all the points of U residing in tile t_0. If a and b are two points in the same connected component of U_{t_0}, then a and b are $(\lfloor (8/\pi) \cdot (\ell+1)^2 \rfloor)$-hop neighbors in U (i.e., a and b are at most $\lfloor (8/\pi) \cdot (\ell+1)^2 \rfloor$ hops away from one another in U).*

Proof. Let $P_{min} = (a = p_0, p_1, \ldots, p_x = b)$ be a path between a and b in U_{t_0} of minimum length. Let D_i, for $i = 0, \ldots, x$, be the disk centered at p_i and of radius $1/2$, and observe that all the disks D_i are contained within a bounding square-box b of dimensions $(\ell + 1) \times (\ell + 1)$, whose center coincides with the center of t_0. Observe also that the disks D_i, for even i, are mutually disjoint; that is, the points p_i, for even i, form an independent set in U (otherwise, P_{min} would not be a minimal-length path between a and b). Therefore, the area of the region R, denoted a, determined by the union of the disks D_i, for even i, is the sum of the areas determined by these individual disks. The value of a is precisely $(\pi/4) \cdot \lceil x/2 \rceil$. Since the region R is contained in the bounding box b of area $(\ell+1) \times (\ell+1)$, we have $a \leq (\ell+1)^2$. Consequently, $(\pi/4) \cdot \lceil x/2 \rceil \leq (\ell+1)^2$. Solving for the integer x in the previous equation we obtain $x \leq \lfloor (8/\pi) \cdot (\ell+1)^2 \rfloor$. This shows that the length of the path P_{min}, which is x, is bounded by $\lfloor (8/\pi) \cdot (\ell+1)^2 \rfloor$, and the proof is complete. □

We now present the local distributed algorithm formally and prove that it constructs the desired lightweight spanner. The input to the algorithm is the spanner G' of U constructed by Theorem 4.6, and a constant $\lambda > 2$. We set $\ell = \lambda$ in the above tiling $\mathcal{T}$. We assume that each point in U has computed its $(\lfloor (8/\pi) \cdot (\lambda+1)^2 \rfloor)$-hop neighbors in U by applying the local distributed algorithm described in Subsection 2.2, where $i = \lfloor (8/\pi) \cdot (\lambda+1)^2 \rfloor$. By Lemma 6.5, this ensures that every point knows all the points in its connected component residing with it in the same tile under any translation.[3] After that, for every round $j \in \{I, H, V, D\}$, each point $p \in U$ executes the following algorithm **Local-LightSpanner**:

(i) p applies translation T_j to compute its virtual coordinates under T_j; Suppose that the translate of p under T_j, $T_j(p)$, resides in tile $t_0 \in \mathcal{T}$;

(ii) p determines the set $S_j(p)$ of all the points in the resulting subgraph of G' (prior to round j) whose translates under T_j reside in the same connected component as $T_j(p)$ in tile t_0;

(iii) p applies the algorithm **Centralized Greedy** to the subgraph $H_j(p)$ of the resulting graph of G' induced by $S_j(p)$ with parameter $\alpha = \lambda - 1$; **if** p decides to remove an edge (p, q) from $H_j(p)$ **then** p removes (p, q) from its adjacency list in G';

Note that since all the points whose translate reside in a single tile apply the same algorithm to the same subgraph during any round j, if a point p decides to remove an edge (p, q), then point q must reach the same decision of removing edge (p, q).

Let G be the subgraph of G' consisting of the set of remaining edges in G' after each point $p \in G'$ applies the algorithm **Local-LightSpanner**.

Theorem 6.6. *The subgraph G of G' is a spanner of U containing a EMST of $V(U)$, with stretch factor $\rho \cdot (\lambda - 1)^4$, and satisfying $wt(G') \leq (1 + \frac{2}{\lambda - 2}) \cdot wt(EMST)$, where ρ is the stretch factor of G'.*

Proof. We first show that G is of light weight. To do so, we need to show that G satisfies the conditions of Theorem 5.3. We show first that G contains a EMST on $V(U)$.

Since G' contains a EMST on $V(U)$, it suffices to show that after each round of the algorithm **Local-LightSpanner**, the resulting graph still contains a EMST on $V(U)$. Fix a round $j \in \{I, H, V, D\}$, and let G'^{+} be the graph resulting from G' just before the execution of round j, and G'^{-} that resulting from G' after the execution of round j. Assume inductively that G'^{+} contains a EMST on $V(U)$. Note that any edge removed from G'^{+} in round j must have its translate contained within a single tile in $\mathcal{T}$. Let t_0 be a tile in $\mathcal{T}$. In round j, each point p whose translate $T_j(p)$ is in t_0, applies the algorithm **Centralized Greedy** to the subgraph of G'^{+}, $H_j(p)$, induced by the set of points $S_j(p)$ defined in the algorithm **Local-LightSpanner**. By part (ii) of Fact 6.3, this algorithm computes a spanner for $H_j(p)$ containing a"local" EMST τ_0 of $H_j(p)$. It is easy to see that an edge e in a EMST of G'^{+} whose translate $T_j(e)$ is in $H_j(p)$, its translate $T_j(e)$ is either an edge of τ_0, or is contained in a cycle whose edges other than e have the same weight as e and are in τ_0. Otherwise, by adding $T_j(e)$ to τ_0, we create a cycle on which $T_j(e)$ is the edge of maximum weight (if not, $T_j(e)$ could replace an edge of τ_0 of larger weight than e, contradicting the minimality of τ_0), and this means that $T_j(e)$ would be the edge of maximum weight on some cycle of G'; since a translation is an isometric transformation—and hence preserves length, this contradicts the fact that e is an edge in a EMST of G'^{+}. Therefore, if an edge in a EMST of G'^{+} is removed during round j, then G'^{-} will still contain a path between the endpoints of e all of whose edges have the same weight as e. Consequently, G'^{-} will still contain a EMST on $V(U)$. It follows that G contains a EMST on $V(U)$.

Now we show that for every cycle C in G, and for every edge e on C, we have $wt(C) \geq \lambda \cdot wt(C)$. Suppose not, and let cycle C and edge $e \in C$ be a counter example. Since every edge in U has weight at most 1, and $wt(C) < \lambda \cdot wt(e)$, it follows that $wt(C) < \lambda$, and by Lemma 6.2,

[3]Note that the subgraph of G' induced by the set of points in a single tile may not be connected.

there exists a round j in which the translate of C resides in a single tile t_0 of $\mathcal{T}$. By part (*iii*) of Fact 6.3, after the application of the algorithm **Centralized Greedy** to the connected component κ containing the translate of C in tile t_0 in round j, no cycle of weight smaller or equals to $(1+\alpha) \cdot wt(e) = (1+\lambda-1) \cdot wt(e) = \lambda \cdot wt(e)$ in the inverse translation of κ remains; in particular, the cycle C will no longer be present in the resulting graph. This is a contradiction. It follows that G satisfies the conditions of Theorem 5.3, and $wt(G) \leq (1 + \frac{2}{\lambda-2}) \cdot wt(\text{EMST})$.

Finally, it remains to show that the stretch factor of G, with respect to U, is at most $\rho \cdot (\lambda-1)^4$. Since G' has stretch factor ρ, it suffices to show that after each round of the algorithm **Local-LightSpanner**, the stretch factor of the resulting graph increases from the previous round by a multiplicative factor of at most $(\lambda - 1)$. Fix a round $j \in \{I, H, V, D\}$, and let G'^{+} and G'^{-} be as above. Suppose that an edge e is removed by the algorithm in round j. Then the translate of e in round j must reside in a single tile t_0 of $\mathcal{T}$. Since by part (*i*) of Fact 6.3 the algorithm **Centralized Greedy** has stretch factor $\alpha = \lambda - 1$, and since a translation is an isometric transformation, a path of weight at most $(\lambda - 1) \cdot wt(e)$ remains between the endpoints of e in G'^{-}. Therefore, the stretch factor of G'^{-} with respect to G'^{+} increases by a multiplicative factor of at most $(\lambda - 1)$ during round j. This completes the proof. □

Lemma 6.7. *The local processing time of a point p in the algorithm* **Local-LightSpanner** *is $O(n^2 \lg n)$, where $n = |V(U)|$.*

Proof. Clearly steps (*i*) and (*ii*) can be carried by point p in $O(n)$ time. Since for each $j \in \{I, H, V, D\}$ the subgraph $H_j(p)$ of G' is plane, the algorithm **Centralized Greedy** runs in $O(n^2 \lg n)$ time on $H_j(p)$ by Lemma 6.4. This completes the proof. □

We conclude with the following theorem:

Theorem 6.8. *Let U be a connected unit disk graph on n points, $k \geq 14$ be an integer constant, and $\lambda > 2$ be a constant. Then there exists an i-local distributed algorithm with $i = \lfloor (8/\pi) \cdot (\lambda+1)^2 \rfloor$, that computes a plane spanner of U containing a EMST on $V(U)$, of degree at most k, weight at most $(1 + \frac{2}{\lambda-2}) \cdot wt(EMST)$, and stretch factor $(\lambda - 1)^4 \cdot (1 + 2\pi(k \cos \frac{\pi}{k})^{-1}) \cdot C_{del}$. Moreover, the local processing time of a point in the algorithm is $O(n^2 \lg n)$.*

References

[1] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[2] R. Benson. *Euclidean Geometry and Convexity.* Mc-Graw Hill, New York, 1966.

[3] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. In *proceedings of the 10th Annual European Symposium on Algorithms*, volume 2461 of *Lecture Notes in Computer Science*, pages 234–246. Springer, 2002.

[4] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. *Algorithmica*, 42(3-4):249–264, 2005.

[5] P. Bose, A. Maheshwari, G. Narasimhan, M. Smid, and N. Zeh. Approximating geometric bottleneck shortest paths. *Computational Geometry: Theory and Applications*, 29:233–249, 2004.

[6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[7] P. Bose, M. Smid, and D. Xu. Diamond triangulations contain spanners of bounded degree. In *proceedings of the 17th International Symposium on Algorithms and Computation*, volume 4288 of *Lecture Notes in Computer Science*, pages 173–182. Springer, 2006. To appear in *Algorithmica*.

[8] G. Calinescu. Computing 2-hop neighborhoods in Ad Hoc wireless networks. In *proceedings of the 2nd International Conference on Ad-Hoc, Mobile, and Wireless Networks*, volume 2865 of *Lecture Notes in Computer Science*, pages 175–186. Springer, 2003.

[9] P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computers and System Sciences*, 39(2):205–219, 1989.

[10] M. Damian, S. Pandit, and S. Pemmaraju. Local approximation schemes for topology control. In *proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing*, pages 208–217, 2006.

[11] G. Das, P. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional Euclidean space. In *proceedings of the 19th ACM Symposium on Computational Geometry*, pages 53–62, 1993.

[12] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *proceedings of the 20th ACM Symposium on Computational Geometry*, pages 132–139, 1994.

[13] G. Das, G. Narasimhan, and J. Salowe. A new way to weigh malnourished Euclidean graphs. In *proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 215–222, 1995.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, third edition, 2008.

[15] D. Dobkin, S. Friedman, and K. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete Computational Geometry*, 5(4):399–407, 1990.

[16] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal on Computing*, 31(5):1479–1500, 2002.

[17] I. Kanj and L. Perković. On geometric spanners of Euclidean and unit disk graphs. In *proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science*, pages 409–420, 2008.

[18] I. Kanj, L. Perkovic, and G. Xia. Computing lightweight spanning subgraphs locally. Technical report # 08-002 at: http://www.cdm.depaul.edu/research/Pages/TechnicalReports.aspx.

[19] I. Kanj, A. Wiese, and F. Zhang. Computing the k-hop neighborhoods locally. Technical report # 08-007 at: http://www.cdm.depaul.edu/research/Pages/TechnicalReports.aspx.

[20] J. Keil and C. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.

[21] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[22] C. Levcopoulos and A. Lingas. There are planar graphs almost as good as the complete graphs and almost as cheap as minimum spanning trees. *Algorithmica*, 8(3):251–256, 1992.

[23] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. In *proceedings of the IEEE Conference on Computer Communications*, 2002.

[24] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang. Localized delaunay triangulation with application in Ad Hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.

[25] N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[26] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

[27] D. Peleg. *Distributed computing: A Locality-Sensitive Approach*. SIAM Monographs on Discrete Mathematis and Applications, 2000.

[28] Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. *Mobile Networks and Applications*, 11(2):161–175, 2006.

[29] R. Wattenhofer. Sensor networks: distributed algorithms reloaded - or revolutions? In *proceedings of the 13th Colloquium on Structural Information and Communication Complexity*, volume 4056 of *Lecture Notes in Computer Science*, pages 24–28. Springer, 2006.

[30] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.