3-1-2008

Computing Lightweight Spanning Subgraphs Locally

Iyad A. Kanj
DePaul

Ljubomir Perkovic
DePaul

Ge Xia
Lafayette College

Computing lightweight spanning subgraphs locally

Iyad A. Kanj and Ljubomir Perković

School of CTI, DePaul University, 243 S. Wabash Ave., Chicago, IL 60604.
Email: {ikanj, lperkovic}@cs.depaul.edu.

Ge Xia

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

Abstract

We consider the problem of computing bounded-degree lightweight plane spanning subgraphs of unit disk graphs in the local distributed model of computation. We are motivated by the hypothesis that such subgraphs can provide the underlying network topology for efficient unicasting and/or multicasting in wireless distributed systems. We start by showing that, for any integer $k \geq 2$, there exists a k-local distributed algorithm that, given a unit disk graph U embedded in the plane, constructs a plane subgraph of U containing a Euclidean Minimum Spanning Tree (EMST) of $V(U)$, whose degree is at most 6, and whose total weight is at most $(1 + \frac{2}{k-1})$ times the weight of an EMST of $V(U)$. We show that this bound is tight by proving that, for any $\epsilon > 0$, there exists a unit disk graph U such that no k-local distributed algorithm can construct a spanning subgraph of U whose total weight is at most $(1 + \frac{2}{k-1} - \epsilon)$ times the weight of an EMST of $V(U)$. We then go further and present the first k-local distributed algorithm, where k is a constant, that computes a bounded-degree plane lightweight *spanner* of a given unit disk graph. The upper bounds on the number of communication rounds of the algorithm, the degree, the stretch factor, and the weight of the spanner, are very small. For example, our results imply an 18-local distributed algorithm that computes a plane spanner of a given unit disk graph U, whose degree is at most 14, stretch factor at most 8.81, and weight at most 8.81 times the weight of an EMST of $V(U)$.

All the obtained results rely on an elegant structural result that we develop for weighted planar graphs. We show a wider application of this result by giving an $O(n \lg n)$ time centralized algorithm that constructs bounded-degree plane lightweight spanners of unit disk graphs (which include Euclidean graphs), with the best upper bounds on the spanner degree, stretch factor, and weight.

1 Introduction

A central issue in distributed systems is the issue of scalability. Algorithms that are centralized or, if distributed, use global propagation of information, are typically not scalable. We focus our attention in this paper on developing efficient scalable algorithms for fundamental problems in emerging distributed systems technologies, such as wireless ad-hoc and sensor networks. For these applications, the network is 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. The algorithms designed in this paper conform to the local distributed model defined by Linial [18] and Peleg [19]. An algorithm in this model is *k-local* if, "intuitively", the computation at each point in the graph (or network) depends solely on the information about the points that are at most k hops (edges) away. Wattenhofer [21] formalizes this notion as follows: a distributed algorithm is *k-local* if it runs in at most k synchronous communication rounds for some integer parameter $k > 0$. Distributed algorithms that are k-local are naturally scalable and robust to local changes because "change" can be handled locally. Therefore, it is natural to study what problems can or cannot be solved under this model, as did Kuhn, Moscibroda, and Wattenhofer in [12].

The fundamental problems under consideration in this paper are the construction of *lightweight* spanning subgraphs and *lightweight spanners* of a UDG U. The weight of each edge in U is defined to be its Euclidean distance, and the weight of a subgraph of U is the sum of the weights of its edges. It is well-known that a connected UDG contains a Euclidean Minimum Spanning Tree (EMST) of its point-set. A spanning subgraph of U 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. A subgraph H of U is said to be a *spanner* of U if there exists a constant ρ such that: for every two points $A, B \in U$, the weight of a shortest path between A and B in H is at most ρ times the weight of a shortest path between A and B in U. The constant ρ is called the *stretch factor* of H (with respect to U). Lightweight spanning subgraphs and spanners of UDGs are fundamental to wireless distributed systems because they represent topologies that can be used for both efficient unicasting *and* efficient broadcasting. Lightweight spanners are also important in computational geometry, and much of the early work on lightweight spanners was done from that perspective under the centralized model [1, 2, 6, 7, 8, 9, 13]. Additional requirements on spanning subgraphs and spanners that have been considered are planarity and bounded degree [2, 9, 10, 16, 20]. These requirements are usually motivated by applications in wireless and sensor networks, whose devices have limited resources. For example, the planarity of the topology is often a requirement for efficient routing (see [3, 10, 11, 16, 20]).

The problems we are thus considering are the design of k-local distributed algorithms that construct bounded-degree plane lightweight spanning subgraphs and spanners of unit disk graphs.

We start by showing a general graph-theoretic result—which is of independent interest—about planar graphs with arbitrary positive edge weights. We prove that in any such graph, if the weight of every cycle is at least a constant (λ) times the weight of any edge on the cycle, then the total weight of the graph is at most a constant $(1 + \frac{2}{\lambda-2})$ times the weight of a Minimum Spanning Tree (MST) of the graph. (We also show these bounds are essentially tight.) In other words, the elimination of short cycles (a local property that can be ensured through local changes) leads to a lightweight graph (a global property).

We first consider the problem of constructing a bounded-degree plane lightweight spanning subgraph of the UDG U, which is not necessarily a spanner, using a k-local distributed algorithm. As was already mentioned, the connected UDG U contains an EMST of $V(U)$—the set of vertices of U. It is also well-known that any EMST of a set of points in the plane has degree bounded by 6. Unfortunately, it is impossible to construct an EMST, or even a spanning tree, of a UDG using a k-local distributed algorithm (for any $k < n/2$) because long cycles cannot be detected locally. This raises the question of what kind of bounded-degree lightweight subgraph of U can be constructed using a k-local distributed algorithm. Such a subgraph is an "approximation" of the EMST, both in terms of weight and degree bound, but on the other hand allows cycles—at least long ones.

The above problem has been considered by Li, Houa and Sha [14] who were the first to propose a 1-local algorithm for the problem. The subgraph they obtain is connected and has degree at most 6; however, this subgraph may not be of light weight. Li [15] showed, in fact, that it is necessary for every point to use information about its 2-hop neighbors in the construction of any lightweight spanning subgraph of U. Li [15] and Li, Wang, and Song [17] generalized the approach in [14] to obtain a k-local algorithm for the problem. Their algorithm constructs a plane spanning subgraph of U that has degree bounded by 6 and weight bounded by $O(wt(\text{EMST}))$; however, the constant in the asymptotic notation is undetermined, and was imported from a more general result by Das, Narasimhan, and Salowe [8].

We employ the structural graph-theoretic results to develop a k-local distributed algorithm that constructs a plane subgraph of the unit disk graph U containing an EMST of $V(U)$, whose degree is at most 6, and whose total weight is at most $(1 + \frac{2}{k-1})$ times the weight of the EMST of $V(U)$ (Theorem 4.6), for any integer parameter $k \geq 2$. We prove (in Theorem 4.7) that this "approximation ratio" is tight by showing that, for any $\epsilon > 0$, there exists a unit disk graph U such that no k-local distributed algorithm can construct a spanning subgraph of U whose total weight is at most $(1 + \frac{2}{k-1} - \epsilon)$ times the weight of an EMST of $V(U)$.

Theorem 4.6 significantly improves the previous results [14, 17]. In particular, for $k = 2$, Theorem 4.6 gives a 2-local distributed algorithm that constructs a plane subgraph of U containing the EMST of $V(U)$, whose degree is at most 6, and whose weight is at most $3 \cdot wt(\text{EMST})$. Moreover, Theorem 4.7 shows that, for any $\epsilon > 0$, there exists a unit disk graph U such that no 2-local distributed algorithm can construct a spanning subgraph of U whose weight is at most $(3 - \epsilon) \cdot wt(\text{EMST})$. Given that no 1-local distributed algorithm can construct a lightweight spanning subgraph of U [15], these results essentially close the book on this problem.

We then consider the problem of constructing a bounded-degree lightweight plane spanner of U. Levcopoulos and Lingas [13] developed the first centralized algorithm for this problem on Euclidean graphs (i.e., the complete graph on n points in the plane). Their $O(n \log n)$ time algorithm, given a positive rational r, produces a plane spanner with stretch factor $(1 + \frac{1}{r}) \cdot C_{del}$ and total weight $(2r + 1) \cdot wt(EMST)$, where the constant $C_{del} \approx 2.42$ is the stretch factor of the Delaunay subgraph of the Euclidean graph. The degree of the lightweight spanner in [13], however, may be unbounded: it is not possible to bound the degree without significantly worsening the stretch factor or the weight. A more recent $O(n \log n)$ time algorithm by Bose, Gudmundsson, and Smid [2] for Euclidean graphs, succeeded in bounding the degree of the spanner by 27 but at a large cost: the stretch factor of the obtained plane spanner is approximately 10.02, and its weight is $O(wt(EMST))$, where the hidden constant in the asymptotic notation is undetermined.

Our first contribution with regard to this problem is a centralized algorithm for unit disk graphs, which include Euclidean graphs, that improves the previous algorithms. We again employ the structural results we developed to design a centralized algorithm that, for any integer constant $\Delta \geq 14$ and (real) constant $\lambda > 2$, constructs a plane spanner of a unit disk graph (or a Euclidean graph) having degree at most Δ, stretch factor $(\lambda - 1) \cdot (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$, and weight at most $(1 + \frac{2}{\lambda - 2}) \cdot wt(EMST)$ (Theorem 5.2). To compare these results to the previous ones, if we let $\Delta = 14$ and $\lambda \approx 2.475$ in Theorem 5.2, we obtain an $O(n \lg n)$ time algorithm that, given a unit disk graph (or a 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 finally consider the problem of computing bounded-degree plane lightweight spanners of unit disk graphs using a k-local distributed algorithm. To the best of our knowledge, the only distributed algorithm for this problem works for a generalization of unit disk graphs, called quasi-unit disk graphs, in higher dimensional Euclidean spaces [5]. While the algorithm developed in [5] is general, it runs in poly-logarithmic number of communication rounds, and the weight and the degree of the spanner are bounded asymptotically. We note that distributed algorithms for computing lightweight spanners of general graphs have been extensively considered in the literature; see for example [19] for a survey on some of these results. In this paper we show that (Theorem 5.8): for any $\Delta \geq 14$ and $\lambda > 2$, there exists a k-local distributed algorithm, where $k = \lceil 2\lambda^2 \rceil + 7$, that computes a plane spanner of a given unit disk graph containing the EMST on its point-set, of degree at most Δ, weight at most $(1 + \frac{2}{\lambda - 2}) \cdot wt(\text{EMST})$, and stretch factor $(\lambda - 1)^4 \cdot (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$. This is the first local algorithm for this problem that runs in constant number of communication rounds. This result is yet another application of the structural results we develop. If we set $\Delta = 14$ and $\lambda \approx 2.256$, we obtain a local algorithm that runs in at most 18 rounds, and computes a plane spanner of degree at most 14, stretch factor at most 8.81, and weight at most $8.81 \cdot wt(\text{EMST})$, of the given unit disk graph.

To summarize, we have developed in this paper robust scalable algorithms for fundamental communication problems in systems modeled as unit disk graphs. The bounds on the parameters of the algorithms and the constructed topologies are small, and suggest that the algorithms and the topologies are practical.

2 Preliminaries

Given a set of points S in the plane, the Euclidean graph E on S is defined to be the complete graph whose point-set is S. The *unit disk graph* U on S is the subgraph of E with the same point-set as E, and such that AB is an edge of U if and only if $|AB| \leq 1$, where $|AB|$ is the Euclidean length of edge AB. We assume in this paper that the unit disk graph U is connected. Except for the general results in Section 3 where we allow the weight of an edge to be an arbitrary positive number, we define the weight of an edge AB to be the Euclidean distance between points A and B, that is $wt(AB) = |AB|$. For a subgraph $H \subseteq E$, we denote by $V(H)$ and $E(H)$ the set of vertices 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, $wt(H) = \sum_{XY \in E(H)} wt(XY)$.

For two distinct points A and B in U, we define $lune(A, B)$ to be the intersection region of the two disks centered at A and B and of radius $|AB|$. The *relative neighborhood graph* of U, denoted $RNG(U)$, is defined to be the subgraph of U having the same point-set as U, and such that an edge $AB \in E(U)$ is an edge in $RNG(U)$ if and only if there is no point in the interior of $lune(A, B)$. A *Euclidean Minimum Spanning Tree* (EMST) of a set of points S in the plane is a minimum spanning tree of the Euclidean graph on S. It is easy to see that if U is connected, then $RNG(U)$ contains an EMST of $V(U)$. Without loss of generality, we assume that a strict ordering on the weights of the edges in E has been predefined (for example, a lexicographic ordering), as in [17], so that ties among edges of equal weight can be broken. Then we can speak of the *unique* EMST of $V(U) = V(E)$, which is a subgraph of $RNG(U)$. Therefore, whenever we speak of an EMST, we will be referring to the unique EMST of $V(U)$ or $V(E)$. Extending the standard characterization of an EMST, for any edge AB in the unique EMST the following holds: There is no cycle C in E such that AB is the edge of maximum weight on C.

A point in $RNG(U)$ can have an unbounded number of incident edges, and hence unbounded degree. This can happen when there is an unbounded number of edges of equal weight incident to a point. To bound the degree of $RNG(U)$, Li et al. [17] defined a subgraph of $RNG(U)$, referred to as the *restricted relative neighborhood graph* of U, and denoted $RRNG(U)$, where edges incident on a point A and having equal weight are pruned according to some predefined ordering. It was shown in [17] that $RRNG(U)$ has degree bounded by 6, contains the EMST of $V(U)$, and that it can be computed locally (by a 1-local distributed algorithm).

The length of a path P (resp. cycle C) in a subgraph $H \subseteq E$, denoted $|P|$ (resp. $|C|$), is the number of edges in P (resp. C). A point B is said to be a k-neighbor of A in a subgraph $H \subseteq E$, if there exists a path P from A to B in H satisfying $|P| \leq k$.

The complexity of a k-local algorithm can be described in terms of the total number of bits exchanged during all communications. Every one of the k synchronous communication rounds consists of two phases: phase 1, in which every point receives messages sent to it in the preceding phase, and phase 2, in which every point sends messages to its neighbors. The local computation in a round occurs between the two phases. If message length must be bounded, then messages can be decomposed into smaller ones. Since our focus is on wireless systems, we will assume that a message broadcast by a point in U will be received by all its neighbors.

The k-local distributed algorithms we develop in this paper construct subgraphs of U and take two steps. In the first step, all points learn about their k-hop neighbors using a k-local distributed algorithm. In the second step, each point runs a local computation to make a decision on what incident edges to select in the final spanning subgraph/spanner (no messages are exchanged in this step). The message complexity of all the k-local distributed algorithms we develop is therefore the message complexity of the "best" *k-local k-neighborhood algorithm*—a k-local algorithm in which each point learns about the coordinates of its k-hop neighbors. A basic k-local k-neighborhood algorithm runs as follows. In the first round, every point broadcasts its ID and coordinates to its neighbors in U. In the remaining $k-1$ rounds, every point broadcasts the ID and coordinates of every point it learned about in the previous round. The total number of bits *broadcast* by every point X is bounded above by $|N_{k-1}(X)| \cdot O(\log n)$—where $N_i(X)$ is the set of points within i hops from point X— assuming that IDs and coordinates can be encoded with $O(\log n)$ bits. We are not aware of any other k-local k-neighborhood algorithm, and improving the total message complexity of the basic algorithm is a fundamental problem that is worth investigating. We note that Calinescu [4] developed a distributed algorithm for computing 2-hop neighborhoods using $O(n \log n)$ bits; however, his algorithm uses global propagation of information, and thus, is not a local distributed algorithm.

3 A structural result about weighted planar graphs

In this section we will prove an elegant and tight structural result about weighted planar graphs, where the weight is not restricted to the Euclidean length. This result is pivotal to the results in the following sections of the paper, and is of independent interest on its own.

Theorem 3.1. *Let G be a weighted connected 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 an MST of G.*

To prove Theorem 3.1, we embed G in the plane and consider an MST T of G. We will bound the total weight of the set of edges $R = E(G) - E(T)$ in terms of $wt(T)$. Call an edge $e \in E(T)$ a *tree edge* and an edge $e \in R$ 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 $T + e$).

Definition 3.2. Define a relationship $\preceq$ on the set $E(G)$ as follows. For every edge e, $e \preceq e$. For two edges e and e' in $E(G)$, $e \preceq e'$ 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 $\mathcal{L}$. We shall henceforth assume that the edges in $E(G)$ are indexed by their indices in $\mathcal{L}$.

Lemma 3.3. *Let e_i be a non-tree edge. Then there exists a unique face (in terms of the edges on the face) F_i in G 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. □

Definition 3.4. Let e_i be a non-tree edge. The *fundamental face* of e_i is defined to be the unique face F_i satisfying: for every edge e_j on F_i, $e_j \preceq e_i$.

To bound the the total weight of the edges in $E(G)$ in terms of $wt(T)$, we "charge" the weight of every edge in R to the edges of T using the following charging scheme. We note that the method of charging the weight of the edges of a plane graph to the edges of a subgraph of it (or an MST) was used informally and in a more specific setting (for triangulations) by Levcopoulos and Lingas [13].

Definition 3.5. The *charging scheme* is defined inductively as follows.

Round 1: The initialization phase. The charge of every edge e_i, denoted $charge(e_i)$, is initialized to $wt(e_i)$ if $e_i \in R$, and to 0 if $e_i \in T$.

Round ℓ ($\ell > 1$): The charging phases. Suppose inductively that the charging scheme has been defined for round $\ell - 1$. Then for every edge $e_i \in R$ that was charged in round $\ell - 1$, let F_i be the fundamental face of

e_i. For every edge $e_j \neq e_i$ on F_i, e_i *charges* e_j, or alternatively e_j *is charged by* e_i, the value $wt(e_j)/(\lambda-1)^{\ell-1}$. The charging scheme halts if no edge $e_i \in R$ was charged in round ℓ by any other edge.

Lemma 3.6. *Let $e \in E(G)$ be an edge on a face F of G. Then e can be charged by at most one edge on F.*

Proof. Suppose not, and let e_i and e_j be two distinct edges on F that charge e. By the definition of the charging scheme, F must be the fundamental face of e_i and e_j. It follows that $e_i \preceq e_j$ and $e_j \preceq e_i$. This contradicts the fact that $\preceq$ is a partial order. $\square$

Lemma 3.7. *(See parts (iv), (v), and (vi) of Lemma 7.1, Appendix A) The following are true.*

(a) At the end of round $\ell \geq 2$, every tree edge e has been charged a total value of at most $2wt(e)/(\lambda-1)^{\ell-1}$.

(b) At the end of round $\ell \geq 2$, every non-tree edge e charges other edges in G a total value at least equals to the total value charged to edge e at the end of round $\ell - 1$.

(c) The charging scheme is well defined and will eventually halt. Moreover, when the charging scheme halts, the only edges that could possess non-zero charges are the tree edges.

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Since $wt(G) = wt(R) + wt(T)$, it suffices to show that $wt(R) \leq (2/(\lambda-2)) \cdot wt(T)$.

At the initialization phase of the charging scheme in Definition 3.5, the charge of each non-tree edge e is equal to its weight, and the charge of each tree edge is equal to 0. Let $charge_\ell(e)$ be the charge of edge e at the end of round ℓ. Let $total_\ell = \sum_{e \in E(G)} charge_\ell(e)$. Then it follows that $total_1 = wt(R)$.

By part (b) of Lemma 3.7, at each round $\ell \geq 2$, each edge $e \in R$ charges a total value not smaller than the total value charged to it at the end of round $\ell - 1$. It follows that for every round $\ell \geq 1$, we have $total_\ell \geq total_{\ell-1} \geq wt(R)$. Let d be the integer such that the charging scheme halts at round d. Then we have $total_d \geq wt(R)$.

By part (c) of Lemma 3.7, the charging scheme terminates with all the charges at the tree edges. Therefore, we have $total_d = \sum_{e \in T} \sum_{\ell=1}^{d} charge_\ell(e) \geq wt(R)$. By part (a) of Lemma 3.7, each edge $e \in T$ is charged a value at most $2wt(e)/(\lambda-1)^{\ell-1}$ at the end of round ℓ, for any $\ell \geq 2$. Since the charge for an edge $e \in T$ at the end of round $\ell = 1$ is zero, it follows that the total accumulated charges for an edge $e \in T$ at the end of the scheme is bounded by $2wt(e) \sum_{\ell=2}^{d} 1/(\lambda-1)^{\ell-1} \leq 2wt(e) \sum_{\ell=2}^{\infty} 1/(\lambda-1)^{\ell-1} \leq 2wt(e)/(\lambda-2)$. Consequently, at the end of round d we have $wt(R) \leq \sum_{e \in T} 2wt(e)/(\lambda-2) \leq (2/(\lambda-2)) \sum_{e \in T} wt(e) = (2/(\lambda-2)) \cdot wt(T)$. This completes the proof. $\square$

Note that in the charging scheme we only charge the weight of non-tree edges to edges on their fundamental faces. Therefore, the condition $wt(C) \geq \lambda \cdot wt(e)$, for any cycle C in G and any edge $e \in C$, in the statement of Theorem 3.1, is not necessary, and can be relaxed by requiring this condition to be true only for cycles forming the boundary of the faces in the graph. We state this relaxed version of Theorem 3.1 as a corollary; we will be using it later in Section 5.

Corollary 3.8. *Let G be a 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 $wt(G) \leq (1 + \frac{2}{\lambda-2}) \cdot wt(T)$.*

The following theorem, whose proof is delegated to Appendix B, shows that the bound in Theorem 3.1 is essentially[1] tight:

Theorem 3.9. *(Theorem 8.3, Appendix B) For any integer constant $\lambda > 2$ and any $\epsilon > 0$, there exists a weighted planar graph G satisfying $wt(C) \geq \lambda \cdot wt(e)$ for any cycle C in G and any edge e on C, and such that $wt(G) \geq (1 + \frac{2}{\lambda-2} - \epsilon) \cdot wt(T)$, where T is an MST of G.*

[1]This lower-bound result applies only to integer constants $\lambda > 2$, as opposed to the upper-bound result in Theorem 3.1 which applies to any (real) constant $\lambda > 2$.

4 An optimal local approximation scheme for the Euclidean MST problem

We use the structural results in the previous section to develop a k-local distributed algorithm that, given a UDG U embedded in the plane, constructs a lightweight planar, bounded-degree spanning subgraph G of U. We start with the following definition.

Definition 4.1. *Let U be a unit disk graph in the plane and let $k \geq 1$ be an integer. We define the k-relative neighborhood graph of U, denoted $RRNG_k(U)$, to be the subgraph of $RRNG(U)$ with the same point set, and satisfying that XY in $RRNG(U)$ is an edge in $RRNG_k(U)$ if and only if there is no cycle C in $RRNG(U)$ on which XY is the edge of maximum weight, and on which every point is a k-neighbor of X (or Y).*

(Recall that a strict ordering on the weights of the edges in E is assumed; see Section 2.)

Theorem 4.2. *The graph $RRNG_k(U)$ is plane, has degree at most 6, contains the EMST of $V(U)$ (and hence is a spanning subgraph of U), and is computable by a k-local distributed algorithm.*

Proof. Since $RRNG_k(U)$ is a subgraph of $RRNG(U)$, $RRNG_k(U)$ is plane and has degree at most 6. By the definition of $RRNG_k(U)$, every edge in $RRNG(U)$ that is not in $RRNG_k(U)$ must be the edge of maximum weight on some cycle in $RRNG(U)$, and hence is not part of the EMST of $V(U)$ contained in $RRNG(U)$. It follows that $RRNG_k(U)$ contains the EMST of $V(U)$. $RRNG_k(U)$ can be computed by applying a k-local k-neighborhood algorithm, and then having each point decide (locally) which incident edges to keep (based on whether or not an incident edge is the edge of maximum weight on some cycle, in the subgraph of U induced by the k-neighborhood of the point). We note that if a point X reaches the decision of removing an incident edge XY after the application of this algroithm, then the same decision is reached by point Y, and hence no further communication is needed. ☐

Lemma 4.3. *Let A and M be two points in U such that there exists a path P_{AM} from A to M satisfying $wt(P_{AM}) \leq \lceil k/2 \rceil$, for some non-negative integer k. Then M is a k-neighbor of A in U.*

Proof. Proceed by contradiction. Assume that M is not a k-neighbor of A in U, and let $P_{min} : (A = X_0, \ldots, X_\ell = M)$ be a path from A to M with the minimum length, over all paths P from A to M satisfying $wt(P) \leq \lceil k/2 \rceil$. Since M is not a k-neighbor of A in U, we have $\ell \geq k+1$. By the choice of P_{min}, we have $wt(X_iX_{i+1}) + wt(X_{i+1}X_{i+2}) > 1$, for $i = 0, \ldots, \ell-1$; otherwise, we can take a short cut around point X_{i+1} to obtain a shorter path than P_{min} and of weight at most that of P_{min} (by the triangular inequality), contradicting the choice of P_{min}. It follows that $wt(P_{min}) = \sum_{i=0}^{\ell-1} wt(X_iX_{i+1}) > \lfloor \ell/2 \rfloor \geq \lfloor (k+1)/2 \rfloor \geq \lceil k/2 \rceil$. This contradicts the hypothesis that $wt(P_{AM}) \leq \lceil k/2 \rceil$. ☐

Corollary 4.4. *Let C be a cycle in $RRNG_k(U)$ and let AB be an edge on C. Then $wt(C) \geq (k+1) \cdot wt(AB)$.*

Proof. Without loss of generality, we can assume that AB is the edge of maximum weight on C.

By the definition of $RRNG_k(U)$, there must exist a point M on C such that M is not a k-neighbor of A; otherwise, the edge AB would not be in $RRNG_k(U)$. Since $AB \in U$, it follows that M is not a $(k-1)$-neighbor of B in U. Let P_{AM} and P_{BM} be the subpaths on C from A to M and from B to M, respectively, that do not contain AB.

Since M is not a k-neighbor of A, it follows from Lemma 4.3 that $wt(P_{AM}) > \lceil k/2 \rceil > \lceil k/2 \rceil \cdot wt(AB)$ (since $wt(AB) \leq 1$). Similarly, $wt(P_{BM}) > \lceil (k-1)/2 \rceil \cdot wt(AB)$. It follows that $wt(C) = wt(AB) + wt(AM) + wt(BM) > (\lceil k/2 \rceil + \lceil (k-1)/2 \rceil + 1) \cdot wt(AB) \geq (k+1) \cdot wt(AB)$. ☐

Theorem 4.5. *Let U be a unit disk graph in the plane. For any integer $k \geq 2$, we have $wt(RRNG_k(U)) \leq (1 + 2/(k-1)) \cdot wt(EMST)$.*

Proof. By Theorem 4.2, the subgraph $RRNG_k(U)$ of U contains the EMST of $V(U)$, which is also an MST of $RRNG_k(U)$. Since $RRNG_k(U)$ is plane and satisfies Corollary 4.4, it follows from Theorem 3.1 (applied with $\lambda = k+1$) that $wt(RRNG_k(U)) \leq (1 + 2/(k-1)) \cdot wt(\text{EMST})$. □

We have the following theorem whose proof follows directly from Theorem 4.2 and Theorem 4.5:

Theorem 4.6. *For any integer $k \geq 2$, there exists a k-local distributed algorithm that, given a unit disk graph U in the plane, constructs a plane spanning subgraph of U containing the EMST of $V(U)$, whose degree is at most 6, and whose total weight is at most $(1 + 2/(k-1)) \cdot wt(EMST)$.*

The following theorem, whose proof is given in Appendix B, shows that the above approximation scheme is tight:

Theorem 4.7. *(Theorem 8.7, Appendix B) For any integer $k \geq 2$ and any constant $\epsilon > 0$, there exists a unit disk graph U such that any spanning subgraph of U constructed by a k-local distributed algorithm has weight at least $(1 + \frac{2}{k-1} - \epsilon) \cdot wt(EMST)$.*

5 A lightweight spanner

In this section we present two algorithms, a centralized algorithm and a k-local algorithm, that construct lightweight bounded degree plane spanners of a given unit disk graph U.

5.1 The centralized algorithm

Kanj and Perlović [10] gave an $O(n \lg n)$ time centralized algorithm that, given a Euclidean graph E on a set of n points in the plane, and an integer parameter $\Delta \geq 14$, constructs a plane spanner G' of E containing the EMST of $V(E)$, of degree at most Δ, and of stretch factor $\rho = (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$, where $C_{del} \approx 2.42$ is the stretch factor of the Delaunay subgraph of E. This result can be extended to unit disk graphs:

Lemma 5.1. *(Lemma 9.1, Appendix C) For any $\Delta \geq 14$, the subgraph G'_U of the spanner G' described in [10], consisting of those edges in G' of weight at most 1, is a plane spanner of the unit disk graph U on $V(E)$ of degree at most Δ, and of stretch factor $\rho = (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$ (with respect to U). Moreover, G'_U contains the EMST of $V(U)$.*

The spanner G'_U, however, may not be of light weight. Therefore, we need to discard edges from G'_U so that the resulting subgraph is of light weight, while at the same time not affecting the stretch factor of G'_U by much. To do so, since G'_U is a plane graph containing the EMST of $V(U)$, we would like to employ the results in Corollary 3.8. However, there is one technical problem: the fundamental faces of G'_U may not satisfy the condition in Corollary 3.8, namely that the weight of every fundamental face F_e of a non-EMST edge e in G'_U satisfies $wt(F_e) \geq \lambda \cdot wt(e)$ ($\lambda > 2$ is a constant). We will show next how to prune the set of edges in G'_U so that this condition is satisfied.

Let T be the EMST of $V(U)$ contained in G'_U. As described in Section 3, we can order the non-tree edges in G'_U with respect to the partial order $\preceq$ described in Definition 3.2. Let $\mathcal{L}' = \langle e_1, e_2, \ldots, e_s \rangle$ be the sequence of non-tree edges in G'_U sorted in a 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'_U 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. We have the following theorem:

Theorem 5.2. *(Theorem 9.3, Appendix C) For any integer parameter $\Delta \geq 14$ and any (real) constant $\lambda > 2$, the subgraph G of the unit disk graph U constructed above is a plane spanner of U containing the EMST of $V(U)$, whose degree is at most Δ, whose stretch factor is $(\lambda - 1) \cdot \rho$, where $\rho = (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$, and whose weight is at most $(1 + \frac{2}{\lambda-2}) \cdot wt(EMST)$. Moreover, G can be constructed in $O(n \lg n)$ time.*

Note that since a Euclidean graph is a unit disk graph with radius equal to ∞, the above theorem holds with U replaced by the Euclidean graph E.

5.2 The k-local algorithm

In this subsection we present a k-local distributed algorithm that constructs a bounded-degree plane lightweight spanner of U.

The same paper by Kanj and Perlović [10], described above, presents a 3-local distributed algorithm that, given a unit disk graph U and an integer parameter $\Delta \geq 14$, constructs a plane spanner G' of U containing the EMST of $V(U)$, of degree at most Δ and stretch factor $\rho = (1+2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$. 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 the structural results in Theorem 3.1. 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 k-local distributed algorithm (for any constant k) is capable of computing the global partial order described in Definition 3.2. To coordinate the removal of edges, we use an idea that, at its core, sits 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 coordinates 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 5.3. *Let C be a cycle of weight at most ℓ. Then the projection of C on any horizontal (resp. vertical) 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 5.4. *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. (Sketch) 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 5.3 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 5.3, 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 5.4 shows that by affecting an appropriate 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 translate under T reside in a single tile will form a separate 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 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 $\alpha \cdot wt(AB)$ exists in the growing graph. We will call this algorithm **Centralized Greedy**. The following facts about this greedy algorithm are known:

Fact 5.5. *Let H be a subgraph of the 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 an MST of H.

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

We now present the k-local distributed algorithm formally and prove that it constructs the desired lightweight spanner. We first need the following lemma, whose proof can be verified by the reader.

Lemma 5.6. *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 $\lceil 2\ell^2 \rceil$-hop neighbors in U (i.e., A and B are at most $\lceil 2\ell^2 \rceil$ hops away from one another in U).*

The input to the algorithm is the spanner G' of U constructed in [10], 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 $\lceil 2\lambda^2 \rceil$-hop neighbors in U by applying the k-local k-neighborhood algorithm described in Section 2, where $k = \lceil 2\lambda^2 \rceil$. By Lemma 5.6, this ensures that every point knows all the points in its connected component residing with it in the same tile under any translation.[2] 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 5.7. *The subgraph G of G' is a spanner of U containing the EMST of $V(U)$, with stretch factor $\rho \cdot (\lambda - 1)^4$, and satisfying $wt(G') \le (1 + \frac{2}{\lambda - 2}) \cdot wt(EMST)$, 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 3.1. We show first that G contains the EMST of $V(U)$.

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

Since G' contains the EMST of $V(U)$, it suffices to show that after each round of the algorithm **Local-LightSpanner**, no EMST edge is removed from the graph. 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 the EMST of $V(U)$. Since an edge removed from G'^{+} in round j must have its translate contained within a single tile in $\mathcal{T}$, it is enough to show that no translate of an EMST edge is removed from any tile of $\mathcal{T}$ during round j. 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 vertices $S_j(p)$ defined in the algorithm **Local-LightSpanner**. By part *(ii)* of Fact 5.5, 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 the EMST of G' whose translate $T_j(e)$ is in $H_j(p)$, its translate $T_j(e)$ must be an edge of τ_0. Otherwise, by adding $T_j(e)$ to τ_0, we create a cycle in τ_0 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 the EMST of G'. Therefore, no edge in the EMST of G'^{+} is removed during round j, and G'^{-} contains the EMST of G'^{+} as claimed. It follows that G contains the EMST of $V(U)$.

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 5.4, 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 5.5, 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 3.1, 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 5.5 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. □

We conclude with the following theorem:

Theorem 5.8. *Let U be a connected unit disk graph, $\Delta \geq 14$ be an integer constant, and $\lambda > 2$ be a constant. Then there exists a k-local distributed algorithm with $k = \lceil 2\lambda^2 \rceil + 7$, that computes a plane spanner of U containing the EMST of $V(U)$, of degree at most Δ, weight at most $(1 + \frac{2}{\lambda-2}) \cdot wt(EMST)$, and stretch factor $(\lambda - 1)^4 \cdot (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$, where $C_{del} \approx 2.42$.*

Note that the $+7$ term in the expression for the number of communication rounds k in Theorem 5.8, is to account for the number of communication rounds in the 4-local distributed algorithm **Local-LightSpanner** plus the number of communication rounds in the 3-local distributed algorithm in [10].

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

[3] 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.

[4] G. Calinescu. Computing 2-hop neighborhoods in Ad Hoc wireless networks. In *Proceedings of ADHOC-NOW*, volume 2865 of *Lecture Notes in Computer Science*, pages 175–186. Springer, 2003.

[5] M. Damian, S. Pandit, and S. Pemmaraju. Local approximation schemes for topology control. In *Proceedings of PODC*, pages 208–217, 2006.

[6] G. Das, P. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proceedings of SoCG*, pages 53–62, 1993.

[7] G. Das and G. Narasimhan. A fast algorithm for constructing sparse euclidean spanners. In *Proceedings of SoCG*, pages 132–139, 1994.

[8] G. Das, G. Narasimhan, and J. Salowe. A new way to weigh malnourished euclidean graphs. In *Proceedings of SODA*, pages 215–222, 1995.

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

[10] I. Kanj and L. Perković. On geometric spanners of euclidean and unit disk graphs. In *Proceedings of STACS*, 2008.

[11] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceeding of CCCG*, volume 11, pages 51–54, 2005.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What cannot be computed locally! In *Proceedings of PODC*, pages 300–309, 2004.

[13] 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.

[14] N. Li, J. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *Proceedings of INFOCOM*, 2003.

[15] X.-Y. Li. Localized construction of low weighted structure and its applications in wireless ad hoc networks. *Wireless Networks*, 11(6):697–708, 2005.

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

[17] X.-Y. Li, Y. Wang, and W.-Z. Song. Applications of k-local MST for topology control and broadcasting in wireless ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(12):1057–1069, 2004.

[18] N. Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

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

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

[21] R. Wattenhofer. Sensor networks: distributed algorithms reloaded - or revolutions? In *Proceedings of SIROCCO*, pages 24–28, 2006.

6 Figures

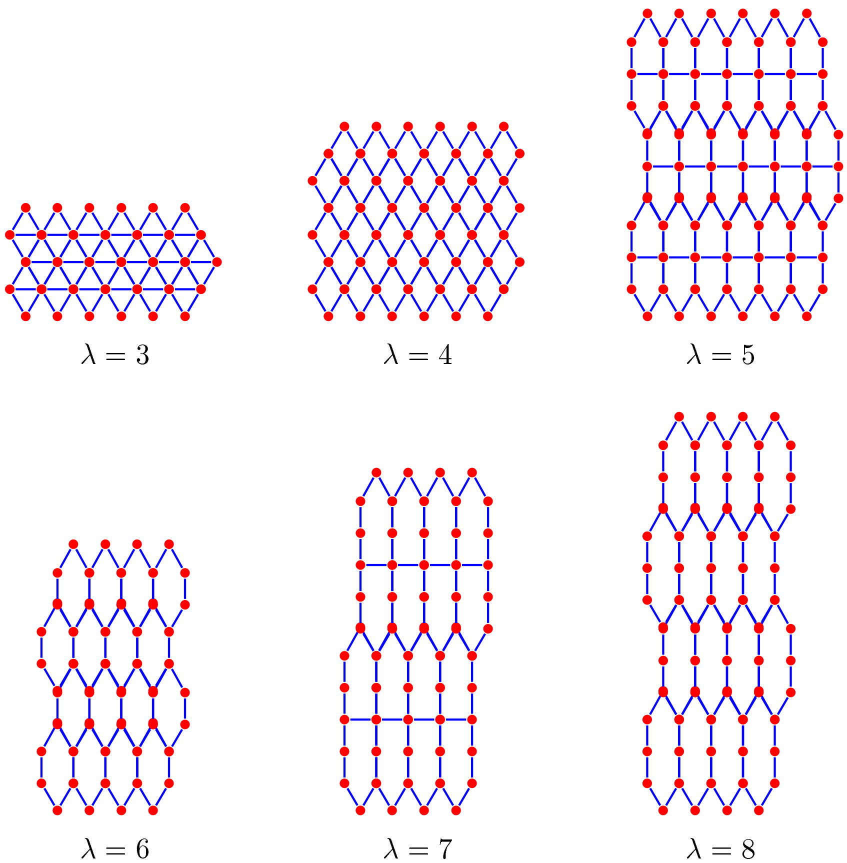

Figure 1: Regions of λ-tiling for $\lambda = 3, 4, 5, 6, 7, 8$.

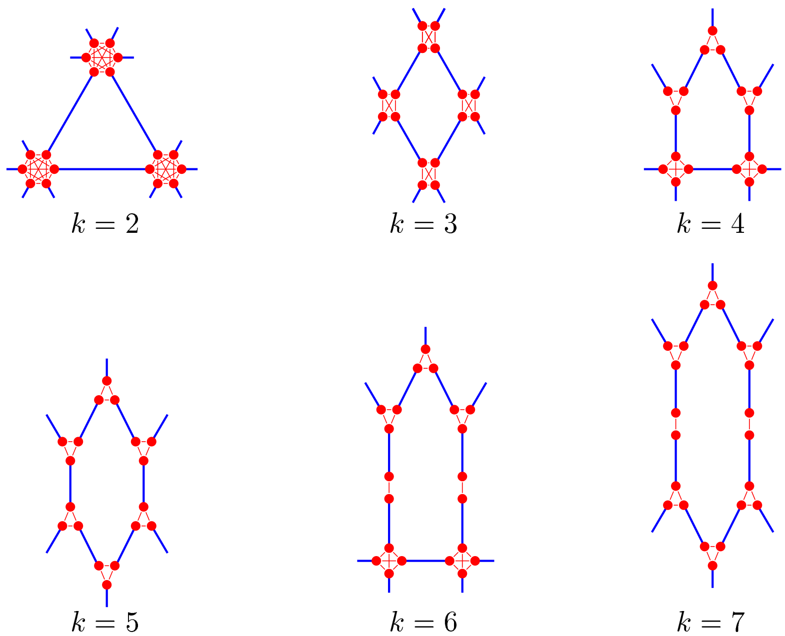

Figure 2: Portions of the resulting unit disk graph U from the tilting R for $k = 2, \ldots, 7$.

7 Appendix A

Lemma 7.1. *The following are true.*

(i) A tree edge can be charged by at most two non-tree edges.

(ii) A non-tree edge can be charged by at most one non-tree edge.

(iii) At the end of round $\ell \geq 1$, every non-tree edge e has been charged in round ℓ a total value of at most $wt(e)/(\lambda - 1)^{\ell-1}$.

(iv) At the end of round $\ell \geq 2$, every tree edge e has been charged a total value of at most $2wt(e)/(\lambda-1)^{\ell-1}$.

(v) At the end of round $\ell \geq 2$, every non-tree edge e charges other edges in G a total value at least equals to the total value charged to edge e at the end of round $\ell - 1$.

(vi) The charging scheme is well defined and will eventually halt. Moreover, when the charging scheme halts, the only edges that could possess non-zero charges are the tree edges.

Proof. (i) Let e be an edge in T. Then e appears on at most two faces of G, say F_i and F_j. By Lemma 3.6, at most one edge on F_i and at most one edge on F_j can charge e. It follows that e can be charged by at most two non-tree edges.

(ii) Let e_i be an edge in R and let F_i be its fundamental face in G. Clearly e_i is not charged by any edge e_j on F_i because $e_j \prec e_i$ for every edge $e_j \neq e_i$ on F_i. Since e_i belongs to at most two faces in G including F_i, it follows by Lemma 3.6 that e_i can be charged by at most one non-tree edge.

(iii) The statement is true in round 1 since the charge of every non-tree edge e is initialized to $wt(e)$. In round $\ell > 1$, by part (ii) above, every non-tree edge e can be charged by at most one non-tree edge, say e'. By Definition 3.5, edge e' charges e the value $wt(e)/(\lambda - 1)^{\ell-1}$.

(iv) By part (i) above, every tree edge e can be charged by at most two non-tree edges, each of which charges e the value $wt(e)/(\lambda - 1)^{\ell-1}$ at the end of round ℓ. It follows that e can be charged a total value of at most $2wt(e)/(\lambda - 1)^{\ell-1}$ at the end of round ℓ, for any $\ell \geq 2$.

(v) Let e be a non-tree edge and let F be its fundamental face. By part (iii) above, the value charged to e at the end of round $\ell - 1$ is at most $wt(e)/(\lambda - 1)^{\ell-2}$. Since F is a cycle containing e, by the hypothesis of Theorem, $wt(F) \geq \lambda \cdot wt(e)$. Therefore, $wt(F - e) \geq (\lambda - 1) \cdot wt(e)$. Now every edge e' on $F - e$ is charged at the end of round ℓ a value equals $wt(e')/(\lambda - 1)^{\ell-1}$ by e. Therefore, the total charge induced by e at the end of round ℓ is at least $wt(F - e)/(\lambda - 1)^{\ell-1} \geq wt(e)/(\lambda - 1)^{\ell-2}$. It follows that edge e in round ℓ dispenses with all the value charged to it at the end of round $\ell - 1$.

(vi) By the definition of the charging scheme, a non-tree edge e_i can only charge other edges e_j on its fundamental face, and hence can only charge edges e_j satisfying $e_j \prec e_i$. Therefore, the charging scheme is acyclic. By part (v) above, at the end of every round ℓ, every non-tree edge has dispensed with the charges charged to it at the end of round $\ell - 1$. It follows from what precedes that the charging scheme must eventually halt when no non-tree edge (still) possesses a non-zero charge. At that point, the only edges that could possess non-zero charges are the minimal elements in $(E(G), \preceq)$, which are the tree edges of G.

□

8 Appendix B

In this section we will prove Theorem 3.9 and Theorem 4.7.

It is easy to verify that for any integer $\lambda > 2$, there exists a tiling of the plane that uses λ-gons whose sides are of equal length and whose internal angles are at least $\pi/3$. For any $\lambda > 2$, we call such a tiling a λ-tiling of the plane. Consider a λ-tiling T of the plane. Let R be a $2n \times 2n$ region of T consisting of $2n$ continuous rows of tiles, in which each row has $2n$ tiles, and such that the overlap between two consecutive rows is maximized. Figure 1 in Section 6 shows examples of such region for $\lambda = 3, \ldots, 8$. For $\lambda > 8$, we can simply repeat the pattern of $\lambda = 7$ and $\lambda = 8$.

Let $|V(R)|$ be the number of vertices in R, and $|E(R)|$ be the number of segments in R. Observing that the number of faces in the R is exactly $4n^2 + 1$, and using Euler's formula and a basic counting argument, it can be verified that:

Fact 8.1. $|V(R)| = n(2n+1)(\lambda-2) + 4n$ *and* $|E(R)| = n(2n+1)\lambda + 2n - 1$.

Lemma 8.2. *For any* $\lambda > 2$ *and* $n > 1$, $|E(R)| \geq (1 + \frac{2}{\lambda-2})(1 - \frac{2}{n})|V(R)|$.

Proof. From Fact 8.1 we have:

$$\begin{aligned} \frac{|E(R)|}{|V(R)|} &= \frac{n(2n+1)\lambda + 2n - 1}{n(2n+1)(\lambda-2) + 4n} \\ &\geq \frac{n(2n+1)\lambda}{n(2n+1)(\lambda-2) + 4n} \\ &= (1 + \frac{2}{\lambda-2})(1 - \frac{4}{(2n+1)(\lambda-2)+4}) \\ &\geq (1 + \frac{2}{\lambda-2})(1 - \frac{2}{n}). \end{aligned}$$

□

Theorem 8.3. *[Theorem 3.9] For any integer* $\lambda > 2$ *and any* $\epsilon > 0$, *there exists a weighted planar graph* G *satisfying* $wt(C) \geq \lambda \cdot wt(e)$ *for any cycle* C *in* G *and any edge* e *on* C, *and such that* $wt(G) \geq (1 + \frac{2}{\lambda-2} - \epsilon) \cdot wt(T)$, *where* T *is an MST of* G.

Proof. Let G be the graph whose vertices are the points of R and whose edges are the segments of R. For any edge $e \in G$, let $wt(e) = 1$. Observe that any MST T of G has weight $|V(R)| - 1$. Moreover, since any cycle in G has length at least λ, and all edges in G have weight 1, it follows that for any cycle C in G and any edge e on C we have $wt(C) \geq \lambda \cdot wt(e)$.

Noting that $wt(G) = |E(R)|$ and $wt(T) = |V(R)| - 1$, we have:

$$\frac{wt(G)}{wt(T)} = \frac{|E(R)|}{|V(R)|-1} \geq \frac{|E(R)|}{|V(R)|} \geq (1 + \frac{2}{\lambda-2})(1 - \frac{2}{n}).$$

The last inequality follows from Lemma 8.2.

Given $\epsilon > 0$, we can choose n large enough so that $(1 + \frac{2}{\lambda-2})(1 - \frac{2}{n}) \geq (1 + \frac{2}{\lambda-2} - \epsilon)$. This completes the proof. □

Now we proceed to prove Theorem 4.7.

Let $\delta < 1/10$ be a positive constant whose value is to be determined later. For any $k \geq 2$, let $\lambda = k + 1$ and consider the $2n \times 2n$ λ-tiling R described in above and illustrated in Figure 1 in Section 6. Let the length of each side of the λ-gons in R be $1 + 2\delta$. We will construct a unit disk graph U based on the tiling R as follows. To avoid confusion, we will refer to the elements of R as vertices and segments, and those of U as points and edges.

For each vertex u in R, and each segment (u, v) incident on it, place a point $A_{u,v}$ in U at distance δ from u on the segment (u, v). After all the points are added, any two points in U are connected by an edge if and only if the distance between them is at most 1. This completes the construction of U.

Figure 2 in Section 6 shows a portion of the resulting unit disk graph U that corresponds to a face of the tilting R for $k = 2, \ldots, 7$.

There are two types of edges in the resulting unit disk graph U: *long edges* whose length is exactly 1, and *short edges* whose length is at most 2δ. Note that the subgraph of U induced by the short edges consists of disjoint components, each of which is a clique of at most 6 points (because the angle between any two edges is at least $\pi/3$). Every cycle that contains a long edge has length at least $2k + 2$. Moreover, there is a natural correspondence (bijection) between the set of cliques of U and the set of vertices of R, and also a correspondence between the set of long edges of U and the set of segments of R.

Lemma 8.4. *The total weight of long edges in U is $|E(R)|$.*

Proof. This follows from the bijection between the set of long edges of U and the set of segments of R, and the fact that the length of any long edge is exactly 1. □

Lemma 8.5. *The weight of a minimum spanning tree of U is at most $(1 + 10\delta)|V(R)|$.*

Proof. Consider a spanning tree of U formed as follows. From every clique of U pick a spanning tree of the clique. Then, connect the spanning trees of these $|V(R)|$ cliques using $|V(R)| - 1$ long edges to form a spanning tree of U. Since every clique has at most 6 points, and every clique-edge has length at most 2δ, the weight of the spanning tree of a clique is at most 10δ. Therefore, the spanning tree of U formed above has weight at most $|V(R)| - 1 + 10\delta|V(R)| \leq (1 + 10\delta)|V(R)|$. This proves the lemma. □

Lemma 8.6. *Let A be any k-local distributed algorithm that constructs a spanning subgraph of a unit disk graph. Let G' be spanning subgraph of U constructed by A when applied to U. Then G' contains all long edges of U.*

Proof. Suppose that a long edge (u, v) of U is not in G'. Consider the subgraph H of U induced by the k-hop neighbors of u. The edge (u, v) must be a bridge in H because every cycle in U involving (u, v) has at least $2k + 2$ edges, and hence at least one point on that cycle is not a k-hop neighbor of u. Apply the same algorithm A to H, which is a unit disk graph. Since A is a k-local distributed algorithm, u will remove (u, v) in H and hence disconnects the graph. This contradicts the fact that A constructs a spanning subgraph of a unit disk graph. □

Theorem 8.7. *[Theorem 4.7] For any integer $k \geq 2$ and any constant $\epsilon > 0$, there exists a unit disk graph U such that any spanning subgraph of U that is constructed by a k-local distributed algorithm has weight at least $(1 + \frac{2}{k-1} - \epsilon) \cdot wt(EMST)$. (EMST is the EMST of $V(U)$.)*

Proof. Let U be the unit disk graph defined above. Let G' be any spanning subgraph of U that is constructed by a k-local distributed algorithm. By Lemma 8.6, G' must contain all long edges in U. By Lemma 8.4 we have:

$$wt(G') \geq |E(R)|. \tag{1}$$

Combining (1) with Lemma 8.2 we get:

$$wt(G') \geq (1 + \frac{2}{k-1})(1 - \frac{2}{n})|V(R)|. \tag{2}$$

Combining (2) with Lemma 8.5 we get:

$$\begin{aligned} wt(G') &\geq (1+\frac{2}{k-1})(1-\frac{2}{n})(\frac{1}{1+10\delta})\cdot wt(\text{EMST}) \\ &\geq (1+\frac{2}{k-1})(1-\frac{2}{n})(1-10\delta)\cdot wt(\text{EMST}). \end{aligned} \tag{3}$$

It can be easily seen from inequality (3) that, for any fixed $k \geq 2$ and $\epsilon > 0$, we can choose δ small enough and n large enough so that:

$$wt(G') \geq (1+\frac{2}{k-1}-\epsilon)\cdot wt(\text{EMST}).$$

This completes the proof. □

9 Appendix C

Lemma 9.1. *For any $\Delta \geq 14$, the subgraph G'_U of the spanner G' described in [10], consisting of those edges in G' of weight at most 1, is a plane spanner of the unit disk graph U on $V(E)$ of degree at most Δ, and of stretch factor $\rho = (1+2\pi(\Delta\cos\frac{\pi}{\Delta})^{-1})\cdot C_{del}$ (with respect to U). Moreover, G'_U contains the EMST of $V(U)$.*

Proof. We need to verify that the subgraph of G' obtained by removing every edge of weight greater than 1 from G', is also a spanner of the unit disk graph U on S satisfying the same properties as G'.

It was shown in [10] that the spanner G' satisfies the property that, for every edge $AB \in E$ that is not in G', there exists a path P_{AB} from A to B in G' of weight at most $\rho \cdot wt(AB)$, and such that AB has maximum weight among all edges on P_{AB} (see Theorem 2.10 in [10]). Since the unit disk graph U is the subgraph of E consisting precisely of those edges in E of weight at most 1, by discarding from G' every edge of weight greater than 1, we obtain a subgraph G'_U of U that is plane and of degree at most Δ. Since U is connected, U contains the EMST of $V(U)$, and hence every edge in the EMST has weight at most 1. Since G' contains the EMST of $V(U)$, it follows from the preceding statement, and from the definition of G'_U, that G'_U contains the EMST of $V(U)$ as well. If an edge $AB \in U$ is not in G', from the properties of the spanner G' described above, there exists a path P_{AB} in G' whose weight is at most $\rho \cdot wt(AB)$, and on which AB is the edge of maximum weight. From the definition of G'_U, the path P_{AB} is also in G'_U. It follows that the same algorithm described in [10] computes a plane spanner G'_U of the unit disk graph U, of degree at most Δ, and of stretch factor $(1+2\pi(\Delta\cos\frac{\pi}{\Delta})^{-1})\cdot C_{del}$, for any integer parameter $\Delta \geq 14$. □

Lemma 9.2. *Given the set of n points $V(E)$ in the plane, the graph G can be constructed in $O(n \lg n)$ time.*

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

The bounded-degree plane spanner G' of E can be constructed in $O(n \lg n)$ time [10], and obviously so can G'_U. Since every point in G'_U has bounded degree, and since G'_U is a geometric plane graph, in $O(1)$ time we can compute a rotation system for the points in G'_U (for example, for every point in G'_U, we can list its incident edges in clockwise order). Moreover, since G'_U has $O(n)$ edges, the EMST T contained in G'_U can be computed in $O(n \lg n)$ time by a standard MST algorithm. Now using the rotation system of G'_U, we can traverse the edges on the boundary face of G'_U. As we traverse these edges, we push the non-tree (with respect to T) edges into a stack, while we discard the edges in T together with any isolated points resulting from this process. Note that, as we remove edge from G'_U, the non-tree edges on the outer face of G'_U are the maximal edges (among the remaining edges in G'_U) with respect to the ordering $\preceq$. The process terminates when G'_U is empty, and at that point, the stack contains the sequence of non-tree edges, sorted according to the partial order $\preceq$; this stack constitutes the list $\mathcal{L}'$. 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 $O(n \lg n)$ time, and the proof is complete. □

Theorem 9.3. *For any integer parameter $\Delta \geq 14$ and any (real) constant $\lambda > 2$, the subgraph G of the unit disk graph U constructed above is a plane spanner of U containing the EMST of $V(U)$, whose degree is at most Δ, whose stretch factor is $(\lambda - 1) \cdot \rho$, where $\rho = (1 + 2\pi(\Delta \cos \frac{\pi}{\Delta})^{-1}) \cdot C_{del}$, and whose weight is at most $(1 + \frac{2}{\lambda - 2}) \cdot wt(EMST)$. Moreover, G can be constructed in $O(n \lg n)$ time.*

Proof. The planarity and degree bound of G follow from the fact that G is a subgraph of G'_U. By construction, G contains the EMST of $V(U)$, 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 3.8, we have $wt(G) \leq (1 + \frac{2}{\lambda - 2}) \cdot wt(EMST)$. Since by Lemma 9.2 G can be constructed in $O(n \lg n)$ time, it suffices to show that the stretch factor of G with respect to U is $(\lambda - 1) \cdot \rho$.

Note that G'_U has stretch factor ρ with respect to U. If an edge e_i is in G'_U 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 $E(G'_U) - E(G)$ is stretched by a factor at most $\lambda - 1$. Since G'_U has stretch factor ρ with respect to U, it follows that the stretch factor of G with respect to U is $(\lambda - 1) \cdot \rho$. This completes the proof. □