The Euclidean Bottleneck Steiner Path Problem and Other Applications of (α, β)-Pair Decomposition*

A. Karim Abu-Affash[†] Paz Carmi[‡] Matthew J. Katz[§] Michael Segal[¶]

August 27, 2013

Abstract

We consider a geometric optimization problem that arises in network design. Given a set P of n points in the plane, source and destination points $s, t \in P$, and an integer $k > 0$, one has to locate k Steiner points, such that the length of the longest edge of a bottleneck path between s and t is minimized. In this paper, we present an $O(n \log^2 n)$-time algorithm that computes an optimal solution, for any constant k. This problem was previously studied by Hou et al. [17], who gave an $O(n^2 \log n)$-time algorithm. We also study the dual version of the problem, where a value $\lambda > 0$ is given (instead of k), and the goal is to locate as few Steiner points as possible, so that the length of the longest edge of a bottleneck path between s and t is at most λ.

Our algorithms are based on two new geometric structures that we develop — an (α, β)-pair decomposition of P and a floor $(1 + \varepsilon)$-spanner of P. For real numbers $\beta > \alpha > 0$, an (α, β)-pair decomposition of P is a collection $\mathcal{W} = \{(A_1, B_1), \ldots, (A_m, B_m)\}$ of pairs of subsets of P, satisfying: (i) For each pair $(A_i, B_i) \in \mathcal{W}$, both minimum enclosing circles of A_i and B_i have a radius at most α, and (ii) For any $p, q \in P$, such that $|pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$, or vice versa. We construct (a compact representation of) an (α, β)-pair decomposition of P in time $O((\beta/\alpha)^3 n \log n)$. In some applications, a simpler (though weaker) grid-based version of an (α, β)-pair decomposition of P is sufficient. We call this version a weak (α, β)-pair decomposition of P.

[*] A preliminary version of this paper appeared in the Proceedings of the 2011 ACM Symposium on Computational Geometry.

[†] Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, abuaffas@cs.bgu.ac.il. Work by A.K. Abu-Affash was partially supported by the Lynn and William Frankel Center for Computer Sciences, by the Robert H. Arnow Center for Bedouin Studies and Development, by a fellowship for outstanding doctoral students from the Planning & Budgeting Committee of the Israel Council for Higher Education, and by a scholarship for advanced studies from the Israel Ministry of Science and Technology.

[‡] Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, carmip@cs.bgu.ac.il. Work by P. Carmi was partially supported by a grant from the German-Israeli Foundation.

[§] Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, matya@cs.bgu.ac .il. Work by M.J. Katz was partially supported by grant 1045/10 from the Israel Science Foundation, and by the Israel Ministry of Industry, Trade and Labor (consortium CORNET).

[¶] Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, segal@cse.bgu.ac.il. Work by M. Segal was partially supported by US Air Force European Office of Aerospace Research and Development, grant number FA8655-09-1-3016, Deutsche Telekom, European project ICT FP7 FLAVIA, and the Israel Ministry of Industry, Trade and Labor (consortium CORNET).

For $\varepsilon > 0$, a floor $(1+\varepsilon)$-spanner of P is a $(1+\varepsilon)$-spanner of the complete graph over P with weight function $w(p,q) = \lfloor |pq| \rfloor$. We construct such a spanner with $O(n/\varepsilon^2)$ edges in time $O((1/\varepsilon^2) n \log^2 n)$, even though w is not a metric.

Finally, we present two additional applications of an (α, β)-pair decomposition of P. In the first, we construct a strong spanner of the unit disk graph of P, with the additional property that the spanning paths also approximate the number of substantial hops, i.e., hops of length greater than a given threshold. In the second application, we present an $O((1/\varepsilon^2) n \log n)$-time algorithm for computing a one-sided approximation for distance selection (i.e., given k, $1 \le k \le \binom{n}{2}$, find the k'th smallest Euclidean distance induced by P), significantly improving the running time of the algorithm of Bespamyatnikh and Segal.

1 Introduction

Let P be a set of n points (terminals) in the Euclidean plane. A path in P is a path in the complete graph over P. A path in P between points $p, q \in P$ is a *bottleneck path* if it minimizes the length of the longest edge in the path. In the *Euclidean bottleneck Steiner path* (EBSP) problem, we are given P, two terminals $s, t \in P$ and an integer k, and we need to reduce, as much as possible, the length of the longest edge in a bottleneck path between s and t, by placing k new terminals, called Steiner points. The EBSP problem arises naturally in the design of networks, in particular, wireless communication networks. For example, when it is required to reduce the length of the longest edge (i.e., the maximum transmission range) in a bottleneck path between two transceivers, s and t, in an existing network consisting of n transceivers, by adding k new transceivers. See [17, 18, 25] for more applications in sensor networks.

The *Euclidean bottleneck Steiner tree* (EBST) problem is closely related to the EBSP problem. In the EBST problem, the goal is to find a spanning tree of P with k Steiner points, minimizing the length of the longest edge in the tree. The EBST problem is NP-hard; moreover, unless $P = NP$, there is no polynomial-time approximation algorithm for the problem with approximation ratio less than $\sqrt{2}$ [29]. Unlike the EBST problem, the EBSP problem is solvable in polynomial time. Ignoring (only for now) the dependence on k, Hou et al. [17] gave an $O(n^2 \log n))$-time exact algorithm for the problem, based on a binary search in the set of all potential lengths (of the longest edge). In this paper, we show that it suffices to consider a certain subset of potential lengths. This subset is of linear size and can be computed efficiently, via an appropriate (α, β)-*pair decomposition* of P (see below), leading to an $O(n \log^2 n)$-time algorithm for the EBSP problem.

We also study the dual version of the EBSP problem: Given P, two terminals $s, t \in P$ and a constant $\lambda > 0$, place as few Steiner points as possible, so that the length of the longest edge in a bottleneck path between s and t is at most λ. We present an $O({k^*}^2 n \log^2 n)$-time exact algorithm for this problem, where k^* is the number of required Steiner points. The algorithm is based on a *floor* $(1+\varepsilon)$-*spanner* of P (see below), which, in turn, is based on an appropriate (α, β)-pair decomposition of P. Again, the corresponding version of the EBST problem is NP-hard (see [24]).

Our algorithms are based on two new geometric structures, which are interesting on their own. For real numbers $\beta > \alpha > 0$, an (α, β)-*pair decomposition* (denoted $\mathrm{PD}_{\alpha,\beta}$) of P, is a collection $\mathcal{W} = \{(A_1, B_1), \ldots, (A_m, B_m)\}$ of pairs of subsets of P, satisfying the following two properties: (i) For each pair $(A_i, B_i) \in \mathcal{W}$, both minimum enclosing circles of A_i and B_i have a radius at most

α, and (ii) for any $p, q \in P$, such that $|pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$, or vice versa. The *weight* of a $\mathrm{PD}_{\alpha,\beta}$ is $\sum_{i=1}^{m}(||A_i|| + ||B_i||)$. We construct (a compact representation of) an (α, β)-pair decomposition of P of weight $O((\beta/\alpha)^2 n \log n)$ in time $O((\beta/\alpha)^3 n \log n)$. More precisely, we describe how to obtain a $\mathrm{PD}_{\alpha,\beta}$ of P from a, so called, semi-separated pair decomposition of P (with respect to β/α), see below. Sometimes, it is enough to require a weaker version of property (ii), that is, (ii') for any $p, q \in P$, such that $\alpha < |pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$, or vice versa. We refer to a collection of subsets of P satisfying properties (i) and (ii') as a *weak* (α, β)*-pair decomposition* of P. Such a weak decomposition of weight $O((\beta/\alpha)^2 n)$ can be easily constructed in $O((\beta/\alpha)^2 n \log n)$ time.

The second structure that we introduce is the *floor* $(1 + \varepsilon)$*-spanner*. For $\varepsilon > 0$, a floor $(1 + \varepsilon)$-spanner of P is a $(1 + \varepsilon)$-spanner of the complete graph over P with weight function $w'(p, q) = \lfloor |pq| \rfloor$, where $|pq|$ is the Euclidean distance between points p and q. Notice that w' is not a metric; in particular, the shortest path between p and q is not necessarily the edge between them. Nevertheless, we construct a floor $(1+\varepsilon)$-spanner of P with $O(n/\varepsilon^2)$ edges in time $O((1/\varepsilon^2) n \log^2 n)$. Informally, the edge set of a floor $(1 + \varepsilon)$-spanner is the union of the edge set of a minimum spanning tree of P, a set that is obtained from an appropriate (α, β)-pair decomposition of P, and an additional set that is obtained by picking a special edge in each cone around each point in P, as in the construction of a Θ-graph (see [13, 20]).

Finally, we present two additional applications of an (α, β)-pair decomposition. In the first, which demonstrates the full power of an (α, β)-pair decomposition, we construct a spanner of the unit disk graph of P (denoted $\mathrm{UDG}(P)$), with some desirable properties. That is, given constants $t > 1$ and $0 < \lambda < 1$, we construct a spanner G of $\mathrm{UDG}(P)$ with the following properties: For any two points $p, q \in P$, there exists a path $\Pi_G(p, q)$ in G between p and q, such that (i) the length of $\Pi_G(p, q)$ is bounded by t times the length of the shortest path between p and q in $\mathrm{UDG}(P)$, (ii) each edge in $\Pi_G(p, q)$ is of length at most $|pq|$, and (iii) the number of edges in $\Pi_G(p, q)$ of length greater than λ is at most the number of such edges in the shortest path between p and q in $\mathrm{UDG}(P)$. Properties (i) and (ii) simply say that G is a *strong* t*-spanner* of $\mathrm{UDG}(P)$, but property (iii) is new. One can think of λ as a threshold, such that the delay of an edge is substantial if and only if its length is above λ. Property (iii) guarantees that the delay caused by using $\Pi_G(p, q)$ is not greater than the delay caused by using the shortest path between p and q in $\mathrm{UDG}(P)$. The number of edges in the spanner G is only $O(s^4 n)$ and it can be constructed in time $O(s^4 n \log^2 n)$, where $s = s(t, \lambda)$ is an appropriate constant.

The second application deals with the well-known distance selection problem. Given an integer k, $1 \leq k \leq \binom{n}{2}$, compute the k-th smallest Euclidean distance, d_k, among the $\binom{n}{2}$ distances determined by pairs of points in P. The best known deterministic algorithm for computing a one-sided approximation of d_k (i.e., a value d such that $d_k \leq d \leq (1 + \varepsilon)d_k$) is due to Bespamyatnikh and Segal [6] and runs in time $O(n \log^3 n)$. We present a deterministic $O(n \log n)$-time algorithm for computing a one-sided approximation of d_k, using an appropriate (α, β)-pair decomposition of P.

1.1 Related work

The EBST problem has received much attention. The best approximation ratio to date is 1.866, achieved by an algorithm of Wang and Li [30]. Bae et al. [4] presented a $\Theta(n \log n)$-time exact algorithm and an $O(n^2)$-time exact algorithm for the special cases $k = 1$ and $k = 2$, respectively. Li et al. [23] presented a $(\sqrt{2} + \varepsilon)$-approximation algorithm, and a matching $\sqrt{2}$ inapproximability result, for a special case of the problem where paths may not contain edges between Steiner points. The dual version of the EBST problem has also received considerable attention, see, e.g., [11,12,24].

A *pair decomposition* of a set P of n points in the plane is a collection $\mathcal{W} = \{(A_1, B_1), \ldots, (A_m, B_m)\}$ of pairs of subsets of P, such that for any two points $p, q \in P$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$ such that $p \in A_i$ and $q \in B_i$, or vice versa. Callahan and Kosaraju [9] introduced the *well-separated* pair decomposition (WSPD). A pair decomposition $\mathcal{W}$ is *well-separated* with respect to a parameter $s > 0$, if, for each pair $(A_i, B_i) \in \mathcal{W}$, the distance between A_i and B_i is at least $s \cdot \max\{r(D_{A_i}), r(D_{B_i})\}$, where $r(D_{A_i})$ and $r(D_{B_i})$ are the radii of the minimum enclosing disks of A_i and B_i, respectively. Callahan and Kosaraju showed that a WSPD consisting of $m = O(s^2 n)$ pairs can be constructed in $O(s^2 n + n \log n)$ time; the weight of the decomposition can be $\Omega(n^2)$. This disadvantage has led Varadarajan [28] to define the *semi-separated* pair decomposition (SSPD). A pair decomposition $\mathcal{W}$ is *semi-separated* with respect to s, if, for each pair $(A_i, B_i) \in \mathcal{W}$, the distance between A_i and B_i is at least $s \cdot \min\{r(D_{A_i}), r(D_{B_i})\}$. Varadarajan showed how to compute a SSPD of weight $O(n \log^4 n)$ in $O(n \log^5 n)$ time, and used it to efficiently compute a min-cost perfect matching. Subsequently, Abam et al. [1] showed how to construct a SSPD of weight $O(s^2 n \log n)$ in $O(s^2 n \log n)$ time, and used it to construct region-fault tolerant geometric spanners. Since any pair decomposition has weight $\Omega(n \log n)$ (see [7]), the result of Abam et al. [1] is optimal. Recently, Abam and Har-Peled [2] gave alternative constructions of SSPDs.

Geometric spanners have been studied extensively in the last few years, see the book by Narasimhan and Smid [26] for a comprehensive survey of the topic. Θ-graphs were introduced by Clarkson [13] and Keil [20]; they are especially useful for navigation. Notice that any strong $(1 + \varepsilon)$-spanner of a set P of n points in the plane is also a $(1 + \varepsilon)$ spanner of $\textsc{udg}(P)$, and such a spanner with $O(n)$ edges can be constructed in $O(n \log n)$ time (see [31]). Unit disk graphs are often used to model wireless networks; see the book by X-Y Li [21] for results on spanners in the context of wireless networks. In particular, substantial research has been done on constructing spanners of unit disk graphs in a local manner, see, e.g., [3,8,22]. Roditty and Segal [27] considered spanners containing bounded leg short paths, i.e., short paths consisting of edges whose length does not exceed a prescribed value L.

The best deterministic algorithm to date for the distance selection problem runs in time $O(n^{4/3} \log^{2+\varepsilon} n)$ (see Katz and Sharir [19]), and the currently best deterministic algorithm for computing a one-sided approximation of the k-th smallest distance runs in time $O(n \log^3 n + n/\varepsilon^2)$ (see Bespamyatnikh and Segal [6]). Chan [10] presented an $O(n \log n + n^{2/3} k^{1/3} \log^{5/3} n)$ expected-time algorithm for distance selection, and Har-Peled and Raichel [16] recently presented an $O(n/\varepsilon^2)$ expected-time algorithm for computing a two-sided approximation of the k-th smallest distance.

1.2 Overview

The rest of the paper is organized as follows. In Section 2, we introduce the notion of an (α, β)-pair decomposition of P, and describe how to construct a compact representation of it of weight $O((\beta/\alpha)^2 n \log n)$ in time $O((\beta/\alpha)^3 n \log n)$. We also define the notion of a weak (α, β)-pair decomposition of P, which is essentially a grid-based construction. In Section 3, we demonstrate the full power of an (α, β)-pair decomposition, by constructing in $O(n \log^2 n)$ time a strong hop-sensitive spanner of $\text{UDG}(P)$ with $O(n)$ edges. In Section 4, we present solutions to the EBSP problem and to its dual version. For the former problem, we present an $O(k^2 n \log n + n \log^2 n)$-time algorithm, and for the latter problem, we present an $O(k^{*2} n \log^2 n)$-time algorithm. Section 5 deals with the construction of a floor spanner of P. We show how to construct a floor $(1 + \varepsilon)$-spanner of P with $O(n/\varepsilon^2)$ edges in time $O((1/\varepsilon^2) n \log^2 n)$, for any $\varepsilon > 0$. Finally, in Section 6, we employ an (α, β)-pair decomposition of P to obtain a more efficient implementation of Bespamyatnikh and Segal's algorithm for computing a one-sided approximation of the k'th smallest distance induced by P, for $1 \leq k \leq \binom{n}{2}$. The new running time is $O((1/\varepsilon^2) n \log n)$.

2 (α, β)-Pair Decomposition

Throughout this paper, $|pq|$ denotes the Euclidean distance between points p and q, $d(D_i, D_j)$ denotes the distance between the boundaries of two disjoint disks D_i and D_j, $r(D)$ denotes the radius of a disk D, and D_A denotes the smallest enclosing disk of a set of points A.

Given a set P of n points in the plane and two real numbers $\beta > \alpha > 0$, we are interested in a collection $\mathcal{W} = \{(A_1, B_1), (A_2, B_2), \ldots, (A_m, B_m)\}$ of pairs of subsets of P, satisfying the following two properties:

(i) For each pair $(A_i, B_i) \in \mathcal{W}$, $\max\{r(D_{A_i}), r(D_{B_i})\} \leq \alpha$, and

(ii) For any two distinct points $p, q \in P$ such that $|pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$, or vice versa.

We refer to such a structure as an (α, β)-*pair decomposition* (denoted, $\text{PD}_{\alpha,\beta}$) of P. The *size* of a $\text{PD}_{\alpha,\beta}$ is the number of pairs, m, and its *weight* is $\sum_{i=1}^{m}(\|A_i\| + \|B_i\|)$, where $\|A_i\|$ (resp. $\|B_i\|$) is the cardinality of A_i (resp. B_i). In this section, we show how to construct in $O(c^4 n \log n)$ time a $\text{PD}_{\alpha,\beta}$ of P of size $O(c^4 n)$ and weight $O(c^4 n \log n)$, where $c = \frac{\beta}{\alpha}$.

Sometimes, it is enough to require property (ii') below instead of property (ii).

(ii') For any two distinct points $p, q \in P$ such that $\alpha < |pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$ such that $p \in A_i$ and $q \in B_i$, or vice versa.

We refer to a collection of pairs of subsets of P satisfying properties (i) and (ii') as a *weak* (α, β)-*pair decomposition* (denoted, $\text{PD}^{\text{W}}_{\alpha,\beta}$) of P.

2.1 $\text{PD}_{\alpha,\beta}$ of P

Let A and B be two sets of points in the plane and let $s > 0$ be a constant. We say that A and B are *semi-separated* with respect to s (or with *separation* s), if $d(D_A, D_B) \geq s \cdot \min\{r(D_A), r(D_B)\}$.

Definition 2.1. Let P be a set of n points in the plane and let $s > 0$ be a constant. A *semi-separated pair decomposition* (SSPD) of P with respect to s is a collection $\{(A_1, B_1), \ldots, (A_m, B_m)\}$ of pairs of subsets of P, such that:

(i) A_i and B_i are *semi-separated* with respect to s, for $i = 1, \ldots, m$, and

(ii) For any two distinct points $p, q \in P$, there exists a single pair (A_i, B_i) in the collection, such that $p \in A_i$ and $q \in B_i$, or vice versa.

Abam et al. [1] showed that a SSPD of P of size $O(s^2 n)$ and weight $O(s^2 n \log n)$ can be constructed in $O(s^2 n \log n)$ time. We construct a $\mathrm{PD}_{\alpha,\beta}$ of P from this SSPD.

Let P be a set of n points in the plane and let $\beta > \alpha > 0$ be two real numbers. We first construct in $O(s^2 n \log n)$ time a SSPD $\mathcal{S}$ of P with respect to $s = \beta/\alpha$. Next, we remove all pairs $(A_i, B_i) \in \mathcal{S}$ for which $d(D_{A_i}, D_{B_i}) > \beta$. Denote by $\mathcal{W}$ the collection of pairs surviving this step, and notice that for any two distinct points $p, q \in P$ such that $|pq| \leq \beta$, there exists a single pair $(A_i, B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$ (or vice versa), and for each pair $(A, B) \in \mathcal{W}$, we have $\min\{r(D_A), r(D_B)\} \leq \beta/s = \alpha$.

Now, for each pair $(A, B) \in \mathcal{W}$, we either keep it in $\mathcal{W}$, if $\max\{r(D_A), r(D_B)\} \leq \alpha$, or we replace it by a collection of $O((\beta/\alpha)^2)$ pairs, otherwise. Consider a pair $(A, B) \in \mathcal{W}$ such that $\max\{r(D_A), r(D_B)\} > \alpha$, and assume without loss of generality that $r(D_A) < r(D_B)$. Put $\gamma_A = r(D_A) \leq \alpha$, and let o_A be the center of D_A. Let h be the half plane containing D_B that is defined by the line through o_A perpendicular to the line through o_A and the center of D_B. We partition h into $t = \lceil \pi/\theta \rceil$ cones $C_1, \ldots, C_t$, each with apex at o_A and angle at most θ, where $\theta = 2 \arcsin\left(\frac{\alpha}{2(\beta+\gamma_A)}\right)$; see Figure 1(a). Let $B^{(i)}$ be the subset consisting of points of B that lie in $D_B \cap C_i$ and whose distance from o_A is at most $\beta + \gamma_A$. We now explain how to partition $B^{(i)}$ into disjoint subsets $B_1^{(i)}, \ldots, B_l^{(i)}$, such that $B_j^{(i)}$ is contained in a disk of radius at most α, for $j = 1, \ldots, l$.

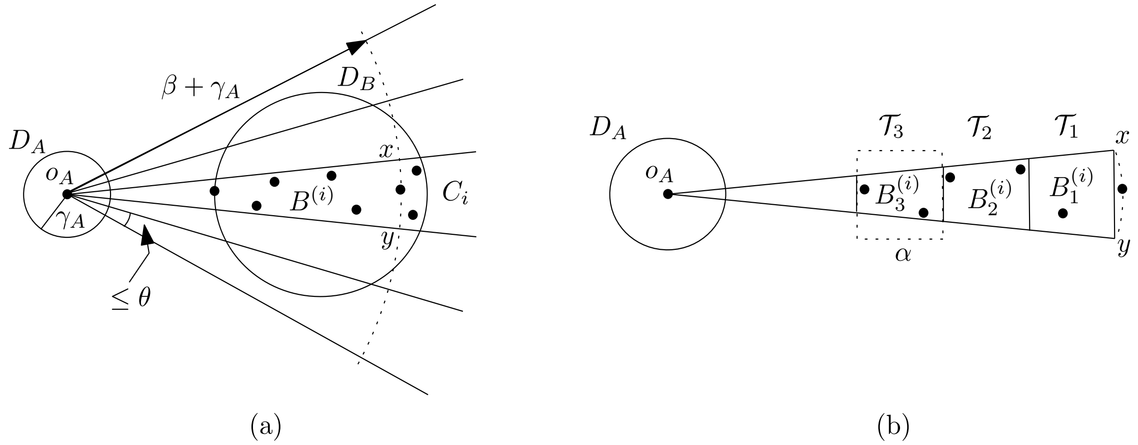


Figure 1: (a) The cones and the subset $B^{(i)}$. (b) Partitioning $B^{(i)}$ into $B_1^{(i)}, \ldots, B_l^{(i)}$.

Let x and y be the two points on C_i's boundary, whose distance from o_A is $\beta + \gamma_A$, and let Δ denote the triangle $x o_A y$. Since $\angle x o_A y \leq \theta$, we have $|xy| \leq 2(\beta+\gamma_A) \cdot \sin\frac{\theta}{2} = 2(\beta+\gamma_A) \cdot \frac{\alpha}{2(\beta+\gamma_A)} = \alpha$.

We now divide Δ into at most $l = \lceil \beta/\alpha \rceil$ trapezoids $\mathcal{T}_1, \ldots, \mathcal{T}_l$, each with base parallel to xy and height α; see Figure 1(b). Next, we partition $B^{(i)}$ into l disjoint subsets $B_1^{(i)}, \ldots, B_l^{(i)}$, such that $B_j^{(i)}$ is the set of points of $B^{(i)}$ that lie in $\mathcal{T}_j$, for $j = 2, \ldots, l$, and $B_1^{(i)}$ is the set of points of $B^{(i)}$ that lie either in $\mathcal{T}_1$ or to the right of xy.

Clearly, each subset $B_j^{(i)}$ is contained in a disk of radius at most α (actually, $\frac{\sqrt{2}}{2}\alpha$). Therefore, for each pair $(A, B_j^{(i)})$, we have $\max\{r(D_A), r(D_{B_j^{(i)}})\} \leq \alpha$. In addition, for any two points $p \in A$ and $q \in B^{(i)}$, there exists a single pair $(A, B_j^{(i)})$, such that $q \in B_j^{(i)}$. After repeating the above procedure for each of the cones $C_1, \ldots, C_t$, we replace the pair (A, B) with the collection of $t \cdot l$ pairs $\{(A, B_1^{(1)}), \ldots, (A, B_l^{(1)}), \ldots, (A, B_1^{(t)}), \ldots, (A, B_l^{(t)})\}$.

Clearly, we have obtained a $\mathrm{PD}_{\alpha,\beta}$ of P. The following theorem summarizes its properties.

Theorem 2.2. *Given a set P of n points in the plane and two real numbers $\beta > \alpha > 0$, a $PD_{\alpha,\beta}$ of P of size $O(c^4 n)$ and weight $O(c^4 n \log n)$ can be computed in $O(c^4 n \log n)$ time, where $c = \beta/\alpha$. A compact representation of $PD_{\alpha,\beta}$ of weight $O(c^2 n \log n)$ can be computed in $O(c^3 n \log n)$ time.*

Proof: A SSPD of size $O((\beta/\alpha)^2 n)$ and weight $O((\beta/\alpha)^2 n \log n)$ can be computed in $O((\beta/\alpha)^2 n \log n)$ time [1]. For each pair $(A, B) \in \mathcal{W}$ (i.e., for each (A, B) of the SSPD such that $d(D_A, D_B) \leq \beta$), we first partition B into t disjoint subsets $B^{(1)}, \ldots, B^{(t)}$, and then partition each $B^{(i)}$ into at most l disjoint subsets $B_1^{(i)}, \ldots, B_l^{(i)}$. That is, B is partitioned into $t \cdot l$ disjoint subsets, and this can clearly be done in time $(t + l) \cdot \|B\|$. Since $t = \lceil \pi/\theta \rceil = O(\beta/\alpha)$ [1] and $l = O(\beta/\alpha)$, the resulting $\mathrm{PD}_{\alpha,\beta}$ is of size[2] $O(t \cdot l \cdot (\beta/\alpha)^2 n) = O((\beta/\alpha)^4 n)$ and weight

$$\begin{aligned}
\sum_{(A,B)\in\mathcal{W}} \sum_{i=1}^{t} \sum_{j=1}^{l} (\|A\| + \left\|B_j^{(i)}\right\|) &= \sum (t \cdot l \cdot \|A\| + \|B\|) \\
&\leq t \cdot l \cdot \sum (\|A\| + \|B\|) \\
&= O(t \cdot l \cdot (\beta/\alpha)^2 n \log n) = O((\beta/\alpha)^4 n \log n),
\end{aligned}$$

and its total construction time is $O(t \cdot l \cdot (\beta/\alpha)^2 n \log n) = O((\beta/\alpha)^4 n \log n)$. In the compact representation of $\mathrm{PD}_{\alpha,\beta}$, we keep only one copy of each set A of the SSPD (instead of $t \cdot l$ copies). ■

2.2 $\mathrm{PD}^{\mathrm{w}}_{\alpha,\beta}$ of P

Lay a regular grid over P, such that the side length of a grid cell is $\alpha/2$, and set $P_\sigma = P \cap \sigma$, for each grid cell σ. Next, for each pair of non-empty grid cells σ, σ' ($\sigma \neq \sigma'$), such that the distance between the boundaries of their enclosing disks is at most β (where the distance is 0 if the disks intersect), add the pair $\{P_\sigma, P_{\sigma'}\}$ to $\mathcal{W}$. Clearly, the collection of pairs $\mathcal{W}$ satisfies properties (i) and (ii') above, and the size and weight of $\mathcal{W}$ is only $O((\beta/\alpha)^2 n)$. It is also easy to see that $\mathcal{W}$ can be computed in $O((\beta/\alpha)^2 n \log n)$ time, using orthogonal range searching. We thus have

[1] This follows from the fact that $\arcsin(x) > x$, for any $x > 0$.

[2] Notice that the size of the produced $\mathrm{PD}_{\alpha,\beta}$ can not exceed the weight of the SSPD. Thus, the size is $O((\beta/\alpha)^2 n \cdot \min\{\log n, (\beta/\alpha)^2\})$.

Theorem 2.3. *Given a set P of n points in the plane and two real numbers $\beta > \alpha > 0$, a $\mathrm{PD}^{\mathrm{W}}_{\alpha,\beta}$ of P of size and weight $O(c^2 n)$ can be computed in $O(c^2 n \log n)$ time, where $c = \beta/\alpha$.*

The advantage of the weak version of an (α,β)-pair decomposition is obvious; it is simpler, the power of the factor β/α in the size and weight bounds is only 2 rather than 4, and moreover the weight bound is $O(n)$ rather than $O(n \log n)$. However, in some cases we do need the strong version, as in the construction of a hop-sensitive spanner in Section 3 below.

Remark. Notice that a $\mathrm{PD}^{\mathrm{W}}_{\alpha,\beta}$ (that is constructed according to the description above) actually satisfies a slightly stronger property (ii'); this will become useful in Section 6.

(ii") For any two distinct points $p, q \in P$ such that $|pq| \le \beta$, exactly one of the following two conditions holds. There exists a single subset P_σ such that both p and q belong to P_σ, or there exists a single pair $\{P_\sigma, P_{\sigma'}\} \in \mathcal{W}$ such that $p \in P_\sigma$ and $q \in P'_\sigma$, or vice versa. Moreover, if $\alpha < |pq| \le \beta$, then the latter condition holds.

3 Hop-Sensitive Spanner

In this section we demonstrate the power of an (α,β)-pair decomposition by using it to construct a spanner of a unit disk graph with some desirable properties. Let P be a set of n points in the plane and let $E = \{\{p,q\} \,|\, p,q \in P,\ 0 < |pq| \le 1\}$. The graph (P,E) is called the *unit disk graph* of P and is denoted $\textsc{udg}(P)$. A Euclidean graph G over P is a *strong t-spanner* of $\textsc{udg}(P)$, for a given constant $t > 1$, if for any two points $p, q \in P$ there exists a path $\Pi_G(p,q)$ in G between p and q, such that (i) $\Pi_G(p,q)$ is a *t-spanning path*, i.e., its length is bounded by t times the length of the shortest path between p and q in $\textsc{udg}(P)$, and (ii) each edge in $\Pi_G(p,q)$ is of length at most $|pq|$. Finally, let $0 < \lambda < 1$ be a constant. A path $\Pi_G(p,q)$ in G between p and q is a *λ-hop-sensitive path*, if the number of edges in $\Pi_G(p,q)$ of length greater than λ is at most the number of such edges in the shortest path between p and q in $\textsc{udg}(P)$.

Our goal in this section is to construct a strong λ-hop-sensitive t-spanner of $\textsc{udg}(P)$. That is, given constants $t > 1$ and $0 < \lambda < 1$, we wish to construct a Euclidean graph G over P, such that for any two points $p, q \in P$, there exists a path $\Pi_G(p,q)$ in G between p and q, such that (i) $\Pi_G(p,q)$ is a t-spanning path, (ii) each edge in $\Pi_G(p,q)$ is of length at most $|pq|$, and (iii) $\Pi_G(p,q)$ is a λ-hop-sensitive path. Moreover, we require that the number of edges in G is only $O(n)$. Such a spanner is useful when, e.g., the cost of an edge is proportional to its length and the delay of an edge is negligible if its length is at most λ but substantial otherwise. For the rest of this section, we assume that $t \le 2$, since otherwise, we simply set $t = 2$ in our construction.

We construct such a spanner from an appropriate (α,β)-pair decomposition of P. Recall that in the construction of an (α,β)-pair decomposition (Section 2.1), we set the separation (of the underlying SSPD) to $s = \frac{\beta}{\alpha}$ and the angle (of the cones) to $\theta = 2\arcsin\left(\frac{\alpha}{2(\beta+\gamma_A)}\right)$. However, when t is small (i.e., $t \to 1$), these values might not be suitable (i.e., not big enough, in the case of s, or not small enough, in the case of θ) to obtain a t-spanner. We thus first update the values of s and θ to accommodate t as well; we denote the new values by s^* and θ^*.

Given P, constants $t > 1$ and $0 < \lambda < 1$, set $\alpha = \frac{\lambda}{2}$ and $\beta = 1$. Moreover, set $s' = \frac{4}{\sqrt{t}-1}$ and $s^* = \max\{s, s'\}$. Let ψ be such that $\cos\psi - \sin\psi = \frac{1}{\sqrt{t}}$, that is, let $\psi = \arccos(\frac{1}{\sqrt{2t}}) - \frac{\pi}{4}$, and set $\theta' = \psi/2$ and $\theta^* = \min\{\theta, \theta'\}$. We construct an (α, β)-pair decomposition $\mathcal{W} = \{(A_1, B_1), \ldots, (A_m, B_m)\}$ of P, by constructing a SSPD of P with respect to separation s^* (instead of s) and using the angle θ^* (instead of θ), as described in Section 2.1. Next, we construct G from $\mathcal{W}$. Initially, G's edge set E is empty. We add at most m edges to E as follows. For each pair $(A_i, B_i) \in \mathcal{W}$, we find the closest pair of points (p_i, q_i) in $A_i \times B_i$ and add the edge $\{p_i, q_i\}$ to E if its length is at most 1. We now claim that G has the desired properties.

Lemma 3.1. *G has properties (i)–(iii).*

Proof. It is enough to show that for each edge $\{p, q\}$ of UDG(P), there exists a path $\Pi_G(p, q)$ in G between p and q, such that (i) its length is bounded by $t|pq|$, (ii) each edge in $\Pi_G(p, q)$ is of length at most $|pq|$, and (iii) the number of edges in $\Pi_G(p, q)$ of length greater than λ is at most 1, if $|pq| > \lambda$, and 0, otherwise. Given an edge $\{p, q\}$ of UDG(P), Algorithm 1 below constructs a path $\Pi_G(p, q)$ in G between p and q.

Algorithm 1 $\Pi_G(p, q)$

```
$p_0 \leftarrow p$
$i \leftarrow 0$
while $p_i \neq q$ do
    let $(A_i, B_i)$ be the pair in $\mathcal{W}$, such that $p_i \in A_i$ and $q \in B_i$ (or vice versa)
    let $(a_i, b_i)$ be the closest pair of points in $A_i \times B_i$
    $\pi_i \leftarrow \Pi_G(p_i, a_i)$ (i.e., the path returned when calling Algorithm 1 with points $p_i$ and $a_i$)
    $p_{i+1} \leftarrow b_i$
    $i \leftarrow i + 1$
$\Pi_G(p, q) \leftarrow p_0, \pi_0, p_1, \pi_1, \ldots, p_i$
return $\Pi_G(p, q)$
```

Consider the path $\Pi_G(p, q) = (p = p_0, \ldots, a_0, p_1, \ldots, a_1, \ \ldots \ , p_{m-1}, \ldots, a_{m-1}, p_m)$ obtained by Algorithm 1. First notice that if $\{p, q\} \in E$, then $\Pi_G(p, q) = (p, q)$. Moreover, since s^* is sufficiently large ($s^* \geq s \geq \frac{4}{\sqrt{t}-1} > 8$), $|p_i a_i| < |pq|$, and the angle between $|p_i q|$ and $|p_i p_{i+1}|$ is at most $2\theta^* \leq 2\theta' = \psi$, for $0 \leq i \leq m-1$, and therefore

$$|p_{i+1}q| \leq |p_i q| - (\cos\psi - \sin\psi)|p_i p_{i+1}| < |p_i q| \qquad (*)$$

(see [26], Lemma 4.1.4). The latter inequalities imply that the algorithm terminates and that indeed $p_m = q$ (since, as i increases the distance between p_i and q decreases).

Before we proceed to prove that $\Pi_G(p, q)$ has the desired properties, consider the following "virtual" path

$$\Pi_V(p, q) = (p = p_0, p_1, \ldots, p_m = q).$$

We call it virtual since it is not necessarily in G. We bound the length of $\Pi_V(p, q)$ as in [26] (where it is called a Θ-walk or a ψ-walk in our notation). From $(*)$, we have

$$|p_ip_{i+1}| \leq \frac{|p_iq| - |p_{i+1}q|}{\cos\psi - \sin\psi} = \sqrt{t}(|p_iq| - |p_{i+1}q|)\,,$$

for $0 \leq i \leq m-1$. Therefore, the length of $\Pi_V(p,q)$ is at most

$$\sum_{i=0}^{m-1} |p_ip_{i+1}| \leq \sqrt{t}\sum_{i=0}^{m-1}(|p_iq| - |p_{i+1}q|) \leq \sqrt{t}(|p_0q| - |p_mq|) \leq \sqrt{t}|pq|\,.$$

We conclude that the length of $\Pi_V(p,q)$ is at most $\sqrt{t}|pq|$.

We now prove that $\Pi_G(p,q)$ satisfies properties (i)–(iii). The proof is by induction on the rank of the edges of $\textsc{udg}(P)$ sorted by length.

Base: $\{p,q\}$ is the shortest edge of $\textsc{udg}(P)$. In this case, $\{p,q\} \in E$, and therefore, by the observation above, $\Pi_G(p,q) = (p,q)$ and satisfies properties (i)–(iii).

Induction hypothesis: For any edge $\{u,v\}$ of $\textsc{udg}(P)$, such that $|uv| < |pq|$, the path $\Pi_G(u,v)$ between u and v in G satisfies properties (i)–(iii).

Inductive step: Since $\mathcal{W}$ is a refinement of a SSPD, we have the following two inequalities for any $0 \leq i \leq m-1$ (see [26]):

(a) $|p_ia_i| \leq \frac{2}{s^*}|a_ip_{i+1}| \leq \frac{2}{s^*}|pq|$, and

(b) $|p_ip_{i+1}| \leq (1+\frac{2}{s^*})|a_ip_{i+1}|$.

We first prove that $\Pi_G(p,q)$ has properties (ii) and (iii). We cut $\Pi_G(p,q)$ into 3 paths, the path between p and a_0 (i.e., $\pi_0 = \Pi_G(p,a_0)$), the single edge (a_0,p_1), and the path between p_1 and q (which is $\Pi_G(p_1,q)$, as can be easily verified). Since $|p_0a_0| < |pq|$ (by inequality (a)) and since $|p_1q| < |pq|$ (by $(*)$), we can apply the induction hypothesis to $\{p_0,a_0\}$ and to $\{p_1,q\}$. By applying the induction hypothesis to the first pair, we get that π_0 has property (ii), with respect to $|p_0a_0| \leq 2\alpha \leq \lambda < 1$, and therefore it does not contain edges of length greater than λ. Similarly, by applying the induction hypothesis to second pair, we get that $\Pi_G(p_1,q)$ has property (ii), with respect to $|p_1q| \leq 2\alpha \leq \lambda < 1$, and therefore it also does not contain edges of length greater than λ. Finally, since $|p_0p_1| \leq |pq|$, we conclude that $\Pi_G(p,q)$ has properties (ii) and (iii).

We now prove that $\Pi_G(p,q)$ is also a t-spanning path. By inequality (a) and by the induction hypothesis we have that $w(\pi_i) \leq t|p_ia_i|$ (where $w(\pi_i)$ is the length of π_i), and therefore, by inequality (a) again, $w(\pi_i) \leq \frac{2t}{s^*}|a_ip_{i+1}|$. The length of $\Pi_G(p,q)$ is at most

$$\begin{aligned}\sum_{i=0}^{m-1}(w(\pi_i) + |a_ip_{i+1}|) &\leq \sum_{i=0}^{m-1}(\frac{2t}{s^*}|a_ip_{i+1}| + |a_ip_{i+1}|)\\ &= (\frac{2t}{s^*}+1)\sum_{i=0}^{m-1}|a_ip_{i+1}| \leq (\frac{2t}{s^*}+1)\sum_{i=0}^{m-1}|p_ip_{i+1}|\\ &= (\frac{2t}{s^*}+1)w(\Pi_V(p,q)) \leq (\frac{2t}{s^*}+1)\sqrt{t}|pq|\\ &\leq (\frac{4}{4/(\sqrt{t}-1)}+1)\sqrt{t}|pq| = t|pq|.\end{aligned}$$

■

The following theorem summarizes the result of this section.

Theorem 3.2. *Let P be a set of n points in the plane, and let $t > 1$ and $0 < \lambda < 1$ be two constants. Then, one can construct a strong λ-hop-sensitive t-spanner of* UDG(P) *with $O(s^{*4}n)$ edges in $O(s^{*4}n \log^2 n)$ time.*

Remark. Our spanner G actually satisfies a stronger version of property (iii): For any two points $p, q \in P$ and for any path ρ in UDG(P) between p and q, there exists a path in G, such that the number of edges in it of length greater than λ is at most the number of such edges in ρ.

4 Euclidean Bottleneck Steiner Paths

Let P be a set of n terminals in the plane. For two terminals $p, q \in P$, let $\delta(p, q)$ denote a path in P (i.e., in the complete Euclidean graph over P) between p and q, and let $bn(\delta(p, q))$ be the *bottleneck* of $\delta(p, q)$, i.e., $bn(\delta(p, q))$ is the length of a longest edge in $\delta(p, q)$. A *bottleneck path* between p and q in P, denoted $\delta^*(p, q)$, is a path $\delta(p, q)$ with smallest bottleneck.

Given a set P of n terminals in the plane, two terminals $s, t \in P$ and an integer k, the goal in the Euclidean Bottleneck Steiner Path (EBSP) problem is to place at most k new terminals, called Steiner points, so as to minimize the bottleneck of a bottleneck path between s and t. (We refer to such a path as a *bottleneck Steiner path.*)

Problem 1 (EBSP). Given a set P of n terminals in the plane, two terminals $s, t \in P$ and a positive integer k, find a bottleneck Steiner path between s and t with at most k Steiner points.

The dual version of the EBSP problem is defined as follows:

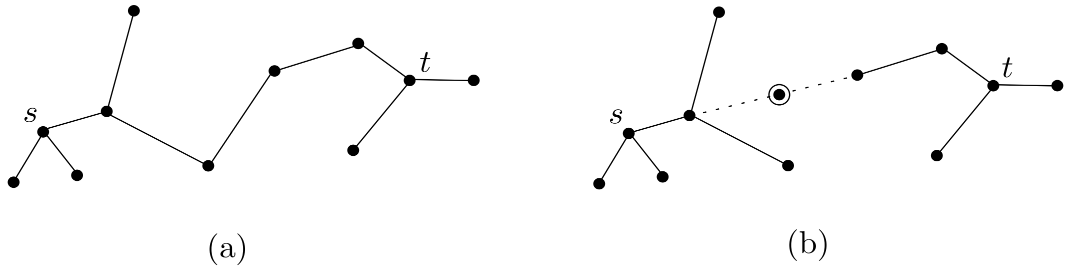
Problem 2 (Dual-EBSP). Given a set P of n terminals in the plane, two terminals $s, t \in P$ and a constant $\lambda > 0$, find a Steiner path between s and t with bottleneck at most λ that uses as few Steiner points as possible.

The decision version of both problems is:

Problem 3 (Dec-EBSP). Given a set P of n terminals in the plane, two terminals $s, t \in P$, a positive integer k and a constant $\lambda > 0$. Does there exist a Steiner path between s and t with at most k Steiner points and bottleneck at most λ?

Problems 1 and 2 are dual in the following sense. Assume we have computed (perhaps implicitly) a finite set of values, such that the optimum bottleneck (resp., the optimum number of Steiner points) is in the set (i.e., we have restricted the search space to some finite set), then we can solve Problem 1 (resp., Problem 2) by applying binary search to this set of values, using an efficient algorithm for Problem 2 (resp., Problem 1) as the decision algorithm.

We first observe that we may assume that the Steiner points must lie on edges of the complete graph over P. (Consider an optimal Steiner path and two consecutive original terminals x and y in the path. If the subpath between x and y passes through l Steiner points, then one can replace them with l evenly-spaced Steiner points on the edge between x and y, without increasing the bottleneck of the path.) Notice, however, that it is not enough to consider only the edges of a minimum spanning tree of P, as shown in Figure 2. By the observation above, there exists a trivial



(a) (b)

Figure 2: (a) A minimum spanning tree of P. (b) An optimal location for a single Steiner point.

search space for Problem 1 of size $O(kn^2)$: For each edge e in the complete graph over P, add the $k+1$ values $|e|, \frac{|e|}{2}, \ldots, \frac{|e|}{k+1}$ to the search space.

In Section 3.1, we present a simple algorithm for solving Dec-EBSP. We then use this algorithm, in Section 4.2, after computing a (non-trivial) linear-size search space, to solve EBSP. In Section 3.3, we study the dual version of EBSP.

4.1 Solving Dec-EBSP

Given a set P of n terminals in the plane, two terminals $s, t \in P$, a positive integer k and a constant $\lambda > 0$, the following simple algorithm decides whether there exists a Steiner path between s and t in P with at most k Steiner points and bottleneck at most λ. Assume for now that $G = (P, E)$ is the complete graph over P.

Algorithm 2 Dec-EBSP($G = (P, E), s, t, k, \lambda$)

```
for each $e \in E$ do
    $w(e) \leftarrow \lceil |e|/\lambda \rceil - 1$
$\delta(s,t) \leftarrow$ the shortest path between $s$ and $t$ in $G$ under $w$
if $w(\delta(s,t)) \leq k$ then
    return $TRUE$
else
    return $FALSE$
```

For each edge $e \in E$, we assign a weight $w(e)$ equal to the minimum number of Steiner points needed to traverse e by steps of size at most λ; see Algorithm 1. We now apply Dijkstra's algorithm [14] to find a shortest path $\delta(s,t)$ between s and t in G, with respect to the weight function w. It is easy to see that $w(\delta(s,t))$ Steiner points are both necessary and sufficient to achieve bottleneck at most λ.

Lemma 4.1. *Given a set P of n terminals in the plane, two terminals $s, t \in P$ and a constant $\lambda > 0$, one can decide in $O(|E| + n \log n)$ time whether there exists a Steiner path between s and t in G with at most k Steiner points and bottleneck at most λ.*

Note that an $O(n^2)$-time algorithm for Dual-EBSP can be derived immediately from Algorithm 2. Moreover, in view of the remark preceding Section 4.1, an $O(kn^2 + n^2 \log n)$-time algorithm for EBSP can be derived from Algorithm 2. (Actually, the latter bound can be improved by observing that the search space can be represented implicitly as a sorted matrix; see the complexity analysis at the end of Section 4.2.)

4.2 Solving EBSP

In this section we assume that $k = o(n)$. We develop an $O(n \log^2 n)$-time algorithm for EBSP, if k is a constant. More precisely, the running time of our algorithm is $O(k^2 n \log n + n \log^2 n)$.

Let $G = (P, E)$ be the complete graph over P. Let $\delta^*_G(s, t, k)$ be a bottleneck Steiner path between s and t in G with at most k Steiner points and let λ^* denote its bottleneck. We first compute a 2-approximation of λ^*.

Lemma 4.2. *Let $t' > 1$ be a constant and let G' be a t'-spanner of G. Then, $bn(\delta^*_{G'}(s, t, k)) \leq t' \cdot \lambda^*$.*

Proof: Let $\delta_G(s, t)$ be the path between s and t in G that is obtained from $\delta^*_G(s, t, k)$ by removing the Steiner points. Consider any edge $e_i = (p, q)$ in $\delta_G(s, t)$ and let k_i be the number of Steiner points located on e_i in $\delta^*_G(s, t, k)$. (Recall that we showed that we may assume that the Steiner points in $\delta^*_G(s, t, k)$ lie on edges of G.) Since G' is a t'-spanner of G, there exists a path $\delta_{G'}(p, q)$ in G' of length at most $t' \cdot |e_i|$. Thus, by locating k_i evenly-spaced Steiner points on $\delta_{G'}(p, q)$, we obtain a Steiner path between p and q in G' of bottleneck at most $\frac{|\delta_{G'}(p,q)|}{k_i+1} \leq t' \cdot \frac{|e_i|}{k_i+1} \leq t' \cdot \lambda^*$, where $|\delta_{G'}(p, q)| = \sum_{e \in \delta_{G'}(p,q)} |e|$. ■

Let $G' = (P, E')$ be a t'-spanner of G. Let $\delta^*_{G'}(s, t, k)$ be a bottleneck Steiner path between s and t in G' with at most k Steiner points and let λ' be its bottleneck. Finally, let $S = \left\{ \frac{|e|}{i} : e \in E', i \in \{1, \ldots, k+1\} \right\}$ and notice that $\lambda' \in S$. It is well known that G' has $O(n)$ edges and can be constructed in $O(n \log n)$ time; see [26]. Thus, we can compute λ' by performing a binary search in S, and using Algorithm 2 to decide whether, for a given value $\lambda \in S$, there exists a path from s to t in G' with at most k Steiner points and bottleneck at most λ.

Corollary 4.3. *For any constant $t' > 1$, a Steiner path between s and t in P with at most k Steiner points and bottleneck at most $t' \cdot \lambda^*$, can be computed in $O(n \log^2 n)$ time.*

Proof. Compute a t'-spanner G' of G in $O(n \log n)$ time, and then compute a bottleneck Steiner path between s and t in G' with at most k Steiner points, as described above. Notice that S can be represented implicitly by a $(k + 1) \times |E'|$ sorted matrix, and selection in S can be performed in $O(k \log n)$ time [15] (see the analysis above Theorem 4.6). Therefore, the total running time is $O((k \log n + n \log n) \log n) = O(n \log^2 n)$. ■

We now set $t' = 2$ and apply Corollary 4.3 to obtain, in $O(n \log^2 n)$ time, a value λ', such that $\lambda' \leq 2\lambda^*$. More precisely, $\frac{\lambda'}{2} \leq \lambda^* \leq \lambda'$.

Our next goal is to characterize the edges of E on which we may want to place Steiner points in the construction of a bottleneck Steiner path. Notice that it does not make sense to locate a

Steiner point on an edge $e \in E$ with $|e| \leq \frac{\lambda'}{2}$ or $|e| > (k+1)\lambda'$. Thus, Steiner points will be located only on edges that belong to the set $E'' = \{e \in E : \frac{\lambda'}{2} < |e| \leq (k+1)\lambda'\}$.

Let T be a minimum spanning tree of G. Let $T_1, T_2, \ldots, T_m$ be the forest $\mathcal{T}$ obtained from T by removing the edges of length greater than $\frac{\lambda'}{2}$; see Figure 3. Let $V_i \subset P$ be the vertex set of T_i, for $i = 1, \ldots, m$. For $1 \leq i, j \leq m$, $i \neq j$, let $e_{i,j}$ be the shortest edge between a terminal in V_i and a terminal in V_j, that is, $|e_{i,j}| = d(V_i, V_j)$. Finally, let E^* be the set consisting of all edges $e_{i,j}$, for which $|e_{i,j}| \leq (k+1)\lambda'$. Notice that all edges of E^* are of length greater than $\frac{\lambda'}{2}$, and, therefore, $E^* \subseteq E''$. In the following lemmas, we show that it is sufficient to consider the edges of E^* (together with the edges of $\mathcal{T}$) in the construction of a bottleneck Steiner path, that $|E^*| = O(k^2 n)$, and that E^* can be computed efficiently.

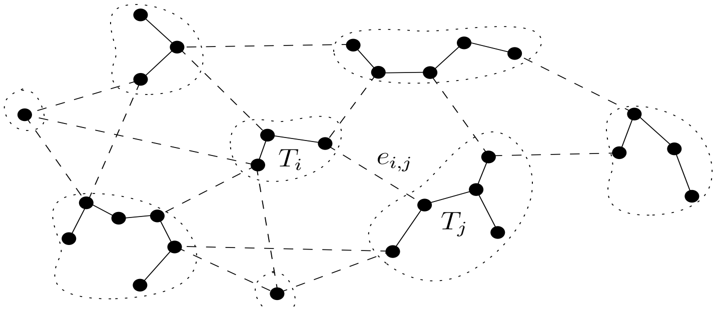


Figure 3: The edges of $\mathcal{T}$ (solid lines) and E^* (dashed lines).

Lemma 4.4. *There exists a bottleneck Steiner path $\delta^*_G(s,t,k)$, such that, for any two consecutive (original) terminals p and q in $\delta^*_G(s,t,k)$,*

(i) if $|pq| \leq \frac{\lambda'}{2}$, then $(p,q) \in \mathcal{T}$, and

(ii) if $|pq| > \frac{\lambda'}{2}$, then $(p,q) \in E^*$.

Proof: Let $\delta_G(s,t,k)$ be a bottleneck Steiner path between s and t. Consider any two consecutive terminals p_1 and q_1 in $\delta_G(s,t,k)$. If $|p_1q_1| \leq \frac{\lambda'}{2}$ but $(p_1,q_1) \notin \mathcal{T}$, then p_1 and q_1 are in the same subset V_i, for some $1 \leq i \leq m$, and there exists a path $\delta_{T_i}(p_1,q_1)$ between p_1 and q_1 in T_i, such that each edge in $\delta_{T_i}(p_1,q_1)$ is of length at most $|p_1q_1|$. We replace the edge (p_1,q_1) in $\delta_G(s,t,k)$ by the path $\delta_{T_i}(p_1,q_1)$ to obtain a new bottleneck Steiner path between s and t that satisfies condition (i) (and (ii)) (for the subpath between p_1 and q_1).

If $|p_1q_1| > \frac{\lambda'}{2}$ but $(p_1,q_1) \notin E^*$, then we distinguish between two cases:

Case 1. $p_1, q_1 \in V_i$, for some $1 \leq i \leq m$. Then, there exists a path $\delta_{T_i}(p_1,q_1)$ between p_1 and q_1 in T_i, such that each edge in $\delta_{T_i}(p_1,q_1)$ is of length at most $|p_1q_1|$. As above, we replace the edge (p_1,q_1) in $\delta_G(s,t,k)$ by the path $\delta_{T_i}(p_1,q_1)$ to obtain a new bottleneck Steiner path between s and t that satisfies condition (i) (and (ii)) (for the subpath between p_1 and q_1).

Case 2. $p_1 \in V_i$ and $q_1 \in V_j$, for $i \neq j$. By the optimality of $\delta_G(s,t,k)$, $|p_1q_1| \leq (k+1)\lambda'$. Hence, $d(V_i,V_j) \leq (k+1)\lambda'$ and the edge $e_{i,j} = (p,q)$, whose length is $d(V_i,V_j)$, belongs to the set E^*.

Let $l \geq 0$ be the number of Steiner points between p_1 and q_1 in $\delta_G(s,t,k)$. We replace the subpath of $\delta_G(s,t,k)$ between p_1 and q_1 by the path that is formed by concatenating the paths $\delta_{T_i}(p_1,p)$, the edge $e_{i,j} = (p,q)$ with l evenly-spaced Steiner points on it, and the path $\delta_{T_j}(q,q_1)$. We have obtained a new bottleneck Steiner path between s and t that satisfies conditions (i) and (ii) (for the subpath between p_1 and q_1).

We repeat this operation for the remaining pairs of consecutive terminals that violate one of the two conditions. Eventually, we obtain a bottleneck Steiner path $\delta^*_G(s,t,k)$ as required. ■

Lemma 4.5. *E^* is of size $O(k^2 n)$ and can be computed in $O(k^2 n \log n + n \log^2 n)$ time.*

Proof: Let G' be a 2-spanner of G and let $\delta^*_{G'}(s,t,k)$ be a bottleneck Steiner path between s and t in G' with at most k Steiner points and bottleneck λ'. By Corollary 4.3, λ' can be computed in $O(n \log^2 n)$ time. Put $\alpha = \frac{\lambda'}{2}$ and $\beta = (k+1)\lambda'$. We compute, according to Theorem 2.3, a weak (α,β)-pair decomposition $\mathcal{W} = \{(A_1,B_1),\ldots,(A_m,B_m)\}$ of P in $O((\beta/\alpha)^2 n \log n) = O(k^2 n \log n)$ time. For each pair $(A,B) \in \mathcal{W}$, let (a,b) be a shortest edge connecting between a point $a \in A$ and a point $b \in B$, and let E^{**} be the set of these m edges. For a pair $(A,B) \in \mathcal{W}$, (a,b) can be computed in $O((||A||+||B||)\log(||A||))$ time, by computing the Voronoi diagram of A together with a corresponding point location data structure and performing point location queries in the diagram with the points in B. Thus, since $m = O(k^2 n)$ and $\sum_{i=1}^{m}(||A_i|| + ||B_i||) = O(k^2 n)$, $|E^{**}| = O(k^2 n)$ and it can be computed in time

$$\sum_{i=1}^{m}(||A_i|| + ||B_i||)\log(||A_i||) \leq \log n \cdot \sum_{i=1}^{m}(||A_i|| + ||B_i||) = O(k^2 n \log n)\,.$$

We now show that $E^* \subseteq E^{**}$. Recall that, for any two points $p,q \in P$, with $\lambda'/2 = \alpha < |pq| \leq \beta = (k+1)\lambda'$, there exists a single pair $(A_i,B_i) \in \mathcal{W}$, such that $p \in A_i$ and $q \in B_i$, or vice versa. Let V_i, V_j, $i \neq j$, be any two subsets for which $|e_{i,j}| = |(p,q)| \leq (k+1)\lambda'$. (Recall that $e_{i,j}$ is the shortest edge between V_i and V_j and that $|e_{i,j}| > \lambda'/2$.) We need to show that $e_{i,j} \in E^{**}$. There exists a single pair $(A_l,B_l) \in \mathcal{W}$, such that $p \in A_l$ and $q \in B_l$. Since A_l (resp. B_l) is contained in a grid cell of diagonal $\lambda'/(2\sqrt{2})$, the distance between any two points in A_l (resp. in B_l) is less than $\frac{\lambda'}{2}$, and, therefore, $A_l \subseteq V_i$ and $B_l \subseteq V_j$. This implies that $e_{i,j}$ is also the shortest edge between A_l and B_l, and therefore $e_{i,j} \in E^{**}$.

Finally, we obtain E^* from E^{**} as follows. We first compute the sets $V_1, V_2, \ldots$, and, for each $p \in P$, we indicate the set V_i to which it belongs. Then, for each edge $(a,b) \in E^{**}$, such that a and b are in different sets V_i, V_j, we proceed as follows. If there already exists an edge in E^* between V_i and V_j, then we replace it by (a,b) if and only if (a,b) is shorter. Otherwise, we add the edge (a,b) to E^*. This stage can be performed in $O(k^2 n + n \log n)$ time. ■

We are now ready to present our algorithm. Given a set P of n terminals, two terminals $s,t \in P$ and a positive integer k, Algorithm 3 finds a Steiner path between s and t with at most k Steiner points and with bottleneck λ^*, where λ^* is the optimum bottleneck. It first constructs the set $E^* \cup \mathcal{T}$, and then generates a search space of $(k+1)|E^*|$ potential values that necessarily includes λ^*. Finally, it finds the value λ^*, by performing a binary search in the search space, using Algorithm 2 to decide whether, for a given value λ, there exists a path from s to t with at most k

Steiner points and bottleneck at most λ. Once λ^* is found, it is easy to produce the desired Steiner path.

Algorithm 3 $EBSP(P, s, t, k)$

```
compute $E^* \cup \mathcal{T}$
$S \leftarrow \left\{ \frac{|e|}{i} : e \in E^*, i \in \{1, \dots, k+1\} \right\}$
find $\lambda^*$ by performing a binary search in $S$, resolving comparisons of the form $\lambda \le \lambda^*$, for $\lambda \in S$,
by calling $Dec\text{-}EBSP(G = (P, E^* \cup \mathcal{T}), s, t, k, \lambda)$
for each $e \in E^* \cup \mathcal{T}$ do
    $w[e] \leftarrow \lceil |e|/\lambda^* \rceil - 1$
$\delta(s,t) \leftarrow$ the shortest path between $s$ and $t$ in $(P, E^* \cup \mathcal{T})$ under $w$
for each $e \in \delta(s,t)$ do
    place $w[e]$ evenly-spaced Steiner points on $e$
```

Correctness. The correctness of the algorithm follows from the lemmas and text preceding it.

Complexity. A minimum spanning tree T can be computed in $O(n \log n)$ time, and, by Lemma 4.5, E^* is of size $O(k^2 n)$ and can be computed in $O(k^2 n \log n + n \log^2 n)$ time. Thus, the number of values in the search space S is $O(k^3 n)$. Using Dijkstra's algorithm [14], a shortest path between s and t in $G = (P, E^* \cup \mathcal{T})$ can be found in $O(k^2 n + n \log n)$ time, and, since Dijkstra's algorithm is applied $O(\log n)$ times, the total running time of Algorithm 3 is $O(k^3 n + k^2 n \log n + n \log^2 n)$. Actually, a more careful analysis allows us to drop the term $k^3 n$ from the running time of Algorithm 3. Observe that S can be represented *implicitly* by the $(k+1) \times |E^*|$ sorted matrix $S = \{s_{i,j}\}$, where $s_{i,j} = \frac{e_j}{k-i+2}$ and e_j is the j-th shortest edge in E^*, and therefore selection in S can be performed in $O(k \log n)$ time [15].

The following theorem summarizes our main result.

Theorem 4.6. *Given a set P of n terminals in the plane, two terminals $s, t \in P$ and a positive integer k, a bottleneck Steiner path between s and t with at most k Steiner points, can be computed in $O(k^2 n \log n + n \log^2 n)$ time.*

Remarks. (i) Note that Algorithm 3 runs in $O(n \log^2 n)$ time for any constant $k \ge 0$, and in subquadratic time for $k = o(n^{1/2})$. (ii) If we forbid edges between Steiner points in our path, then the problem can be solved in $O(n \log^2 n)$ time, for any $k \ge 0$.

4.3 Solving Dual-EBSP

In the dual problem, we are given a set P of n terminals in the plane, two terminals $s, t \in P$ and a constant $\lambda > 0$, and we must find a Steiner path between s and t with bottleneck at most λ, using as few Steiner points as possible. Let k^* denotes the minimum number of Steiner points needed to achieve bottleneck at most λ. We discuss three approaches for computing k^*.

As mentioned in Section 4.1, an $O(|E|) = O(n^2)$-time algorithm for Dual-EBSP can be derived immediately from Algorithm 1. The second approach is based on Algorithm 3. We search for k^* in the sequence $1, 2, \ldots, 2^{\lceil \log k^* \rceil}$, resolving comparisons by calls to Algorithm 3. This approach leads to an algorithm with running time $O(\log k^* \cdot ({k^*}^2 n \log n + n \log^2 n))$.

The third approach is based on geometric floor spanners (or, more accurately, on spanners for the weight function $w(p, q) = \lceil |pq| \rceil - 1)$, introduced in the next section (see also remark at the end of next section). For any $\varepsilon > 0$, the $(1 + \varepsilon)$-spanner of P (w.r.t. w) contains a path $\delta(s, t)$ between s and t, such that $\sum_{e \in \delta(s,t)} (\lceil |e|/\lambda \rceil - 1) \leq (1 + \varepsilon) \cdot k^*$. Moreover, the spanner can be computed in time $O((1/\varepsilon^2) n \log^2 n)$. Therefore, by taking $\varepsilon < 1/k^*$, we can obtain an optimal path between s and t.

A value smaller than $1/k^*$ is found by bootstrapping, as follows. We first construct in $O(n \log^2 n)$ time a $(3/2)$-spanner G' of G. Let $\delta_{G'}(s, t)$ be the shortest path between s and t in G' and let $k' = \sum_{e \in \delta_{G'}(s,t)} (\lceil |e|/\lambda \rceil - 1)$. Clearly, $k^* \leq k' \leq 3k^*/2$. Next, we take $\varepsilon = 1/k' \leq 1/k^*$ and construct in $O({k'}^2 n \log^2 n) = O({k^*}^2 n \log^2 n)$ time a $(1 + 1/k')$-spanner G'', of size $O({k'}^2 n) = O({k^*}^2 n)$, that contains an optimal path between s and t. Finally, we apply the first approach above to compute k^* in $O({k^*}^2 n + n \log n)$ time. The following theorem summarizes our result.

Theorem 4.7. *Given a set P of n terminals in the plane, two terminals $s, t \in P$ and a constant λ, a Steiner path between s and t with bottleneck at most λ, using k^* Steiner points, can be computed in $O({k^*}^2 n \log^2 n)$ time.*

5 Floor Spanner

In this section we consider the problem of constructing a *floor* $(1+\varepsilon)$*-spanner* of a set P of n points in the plane. Let $G = (P, E)$ be the complete graph over P with weight function $w'(p, q) = \lfloor |pq| \rfloor$. For two points $p, q \in P$, let $\delta_G(p, q)$ denote the shortest path between p and q in G (under w'), where the weight of a path is, as usual, the sum of the weights of the edges along the path. Notice that under w', the shortest path between two points is not necessarily the edge between them.

Definition 5.1. Let P be a set of n points in the plane and let $t > 1$ be a constant. A graph $G' = (P, E')$ is a *floor t-spanner* of P, if for each pair $p, q \in P$, $w'(\delta_{G'}(p, q)) \leq t \cdot w'(\delta_G(p, q))$. [3]

We now describe how to construct, in $O(n \log^2 n)$ time, a floor t-spanner of P, for any constant $t > 1$. Put $\alpha = 1$ and $\beta = \frac{t}{(\gamma t - 1) \cos \frac{\theta}{2}}$, where $\gamma = (\cos \theta - \sin \theta)$ and θ is an appropriate constant to be specified later. We first compute a weak (α, β)-pair decomposition, $\mathcal{W}$, of P. Next, we compute from $\mathcal{W}$ the set of edges E^{**}, as in the proof of Lemma 4.5. Let E_1 to be the set of edges containing E^{**} and the edges of a minimum spanning tree of P, $\mathcal{MST}(P)$. We now construct a second set of edges, E_2.

The construction of E_2 is similar to that of a Θ-graph [13, 20]. For each point $p \in P$, partition the plane into $k = \lceil 2\pi/\theta \rceil$ cones $C_{p,1}, C_{p,2}, \ldots, C_{p,k}$, with apex at p and angle at most θ. Consider a cone $C_{p,i}$. Let $l_{p,i}$ be the ray emanating from p and dividing the angle of $C_{p,i}$ into two halves, and let

[3] Notice that the latter condition is equivalent to $w'(\delta_{G'}(p, q)) \leq \lfloor t \cdot w'(\delta_G(p, q)) \rfloor$, since $w'(\delta_{G'}(p, q))$ is an integer.

$P_{p,i}$ be the set of points in $P \cap C_{p,i}$, for which the distance between their orthogonal projection onto $l_{p,i}$ and p is greater than $\frac{t}{\gamma t-1}$. For each cone $C_{p,i}$, we add to E_2 an edge between p and the point q in $P_{p,i}$ whose orthogonal projection onto $l_{p,i}$ is closest to p. We now show that $G' = (P, E_1 \cup E_2)$ is a floor t-spanner of P.

Lemma 5.2. *If θ is chosen so that $\gamma = \cos\theta - \sin\theta > \frac{1}{t}$, then G' is a floor t-spanner of P.*

Proof: To prove the lemma it suffices to show that, for any two points $p, q \in P$, $w'(\delta_{G'}(p,q)) \leq t \cdot \lfloor |pq| \rfloor$. We show this by induction on the rank of the distance $|pq|$ in the sorted sequence of distances in P.

Base case: If p, q are the closest pair in P, then (p,q) is an edge of the minimum spanning tree of P, and therefore also of E_1.

Induction step: If p, q are not the closest pair, then we distinguish between two cases. Assume first that $|pq| \leq \beta$. If p, q lie in the same grid cell, then all edges of the path in $\mathcal{MST}(P)$ between p and q are of length at most $|pq| \leq \alpha/\sqrt{2} < 1$, and therefore, since $\mathcal{MST}(P) \subseteq E_1$, $w'(\delta_{G'}(p,q)) = 0$. If p, q lie in different grid cells, then there is a pair $(A, B) \in \mathcal{W}$ such that $p \in A$ and $q \in B$. If $(p,q) \in E_1$, then we are done. Otherwise, there is an edge $(a,b) \in E_1$ such that $a \in A$, $b \in B$ and $|ab| \leq |pq|$. Since, $|pa|, |bq| \leq \alpha/\sqrt{2} < 1$, $w'(\delta_{G'}(p,a)) = w'(\delta_{G'}(b,q)) = 0$, and therefore

$$w'(\delta_{G'}(p,q)) \leq w'(\delta_{G'}(p,a)) + \lfloor |ab| \rfloor + w'(\delta_{G'}(b,q)) \leq \lfloor |pq| \rfloor \ .$$

Assume now that $|pq| > \beta = \frac{t}{(\gamma t-1)\cos\frac{\theta}{2}}$, and assume that our claim is true for all distances of rank less than that of $|pq|$. Let $C_{p,i}$ be the cone (with apex at p) containing q. Notice that, since $|pq| > \frac{t}{(\gamma t-1)\cos\frac{\theta}{2}}$, the distance between p and the orthogonal projection of q onto $l_{p,i}$, denoted d_q, is greater than $\frac{t}{\gamma t-1}$. Hence, the point q was considered during the construction of E_2. If $(p,q) \in E_2$, then we are done, otherwise, there is a point q' in $C_{p,i}$ such that $(p,q') \in E_2$ and $\frac{t}{\gamma t-1} < d_{q'} \leq d_q$. Therefore,

$$w'(\delta_{G'}(p,q)) \leq \lfloor |pq'| \rfloor + w'(\delta_{G'}(q',q)) \,.$$

By Lemma 4.1.4 in [26], we have $|q'q| \leq |pq| - (\cos\theta - \sin\theta) \cdot |pq'| = |pq| - \gamma \cdot |pq'|$. Since $\gamma > 0$, we conclude that $|q'q| < |pq|$ and, therefore, we may apply the induction hypothesis to $|q'q|$, that is,

$$\begin{aligned} w'(\delta_{G'}(p,q)) &\leq \lfloor |pq'| \rfloor + t \cdot \lfloor |q'q| \rfloor \\ &\leq \lfloor |pq'| \rfloor + t \cdot \lfloor |pq| - \gamma \cdot |pq'| \rfloor \\ &\leq |pq'| + t \cdot |pq| - t\gamma \cdot |pq'| \\ &\leq t \cdot |pq| - (t\gamma - 1) \cdot |pq'| \,. \end{aligned}$$

Finally, since $\gamma > \frac{1}{t}$ and $|pq'| > \frac{t}{\gamma t-1}$, we obtain

$$\begin{aligned} w'(\delta_{G'}(p,q)) &\leq t \cdot |pq| - (t\gamma - 1) \cdot \frac{t}{\gamma t - 1} \\ &\leq t \cdot (|pq| - 1) \leq t \cdot \lfloor |pq| \rfloor \ . \end{aligned}$$

This completes the proof of the lemma. ■

Lemma 5.3. *Let $t = 1 + \varepsilon$, where $0 < \varepsilon \le 1/2$. Then, for $\theta = \varepsilon/4$, G' is a floor t-spanner of P with $O(n/\varepsilon^2)$ edges. Moreover, G' can be constructed in $O((1/\varepsilon^2)n \log^2 n)$ time.*

Proof: Since $0 < \theta \le 1/8$, we have $\sin\theta < \theta$ and $1 - \cos\theta < \theta$. Hence,

$$\gamma = \cos\theta - \sin\theta > 1 - 2\theta = 1 - \frac{\varepsilon}{2} > \frac{1}{t},$$

and, by Lemma 5.2, G' is a floor t-spanner of P. We now bound the size of G'.

The size of E_2 is $O(n/\theta) = O(n/\varepsilon)$. As for E_1, $\mathcal{MST}(P)$ is of size $O(n)$ and E^{**} is of size $O((\beta/\alpha)^2 n)$. Hence, E_1 is of size $O((\beta/\alpha)^2 n)$. Since $\alpha = 1$ and

$$\beta = \frac{t}{(\gamma t - 1)\cos\frac{\theta}{2}} < \frac{2}{((1-\frac{\varepsilon}{2})(1+\varepsilon) - 1)\cos\frac{\varepsilon}{8}} < \frac{4}{\cos\frac{1}{16}} \cdot \frac{1}{\varepsilon(1-\varepsilon)} < \frac{8}{\cos\frac{1}{16}} \cdot \frac{1}{\varepsilon},$$

the size of E_1 is $O(n/\varepsilon^2)$.

Finally, according to the proof of Lemma 4.5, E^{**} can be constructed in $O((\beta/\alpha)^2 n \log n) = O((1/\varepsilon^2)n \log n)$ time. Moreover, a minimum spanning tree of P can be constructed in $O(n \log n)$ time [5]. The set E_2 can be constructed in $O((1/\varepsilon)n \log^2 n)$ time, by a slight modification of the $O((1/\theta)n \log n)$-time algorithm for computing the Θ-graph (see [26]). Therefore, G' can be constructed in $O((1/\varepsilon^2)n \log^2 n)$ time, as claimed. ■

By combining Lemma 5.2 and Lemma 5.3, we obtain the following theorem.

Theorem 5.4. *For any set P of n points in the plane and for any $\varepsilon > 0$, a floor $(1+\varepsilon)$-spanner of P with $O(n/\varepsilon^2)$ edges can be constructed in $O((1/\varepsilon^2)n \log^2 n)$ time.*

Remark. Notice that sparse ceiling $(1+\varepsilon)$-spanners do not exist. For example, take n points in a disk of diameter 1. Then, for any $0 < \varepsilon < 1$, any $(1+\varepsilon)$-spanner will have $\Omega(n^2)$ edges. However, for the function w, defined as $w(p,q) = \lceil |pq| \rceil - 1$, one can adapt our proof above to show that there exists a sparse $(1+\varepsilon)$-spanner of P, even though $w(p,q) \ne w'(p,q)$.

6 Distance Selection

In this section we consider the distance selection problem. Given a set P of n points and an integer parameter k, $1 \le k \le \binom{n}{2}$, we wish to compute the k-th smallest Euclidean distance among the $\binom{n}{2}$ distances determined by pairs of points in P. Let $d_1 \le d_2 \le \ldots \le d_{\binom{n}{2}}$ be the distances determined by the points in P. The best deterministic algorithm to date for finding d_k runs in time $O(n^{4/3} \log^{2+\varepsilon} n)$ (see Katz and Sharir [19]), and the currently best deterministic algorithm for computing a one-sided approximation of d_k (i.e., a value x such that $d_k \le x \le (1+\varepsilon)d_k$) runs in time $O(n \log^3 n + n/\varepsilon^2)$ (see Bespamyatnikh and Segal [6]). The latter authors also presented an algorithm for computing a two-sided approximation of d_k (i.e., a value x such that $(1-\varepsilon)d_k \le x \le (1+\varepsilon)d_k$) that runs in $O(n \log n + n/\varepsilon^2)$ time. In what follows we show how to improve the running time of the algorithm for computing a one-sided approximation of d_k, using a weak (α, β)-pair decomposition.

We proceed as follows. First, we compute a weak (α, β)-pair decomposition $\mathcal{W} = \{(A_1, B_1), \ldots, (A_l, B_l)\}$ of P, for appropriate values of α and β (see below). Then, for each pair $(A_i, B_i) \in \mathcal{W}$, we compute i) the maximal distance M_i defined by a point from A_i and a point from B_i, ii) the minimal distance m_i defined by a point from A_i and a point from B_i, and iii) the number of distances δ_i defined by the pair (A_i, B_i), which is $\delta_i = |A_i| \cdot |B_i|$. Observe that the values m_i and M_i, $i = 1, \ldots, l$, can be computed in total time $O(n \log n)$, using the nearest and farthest Voronoi diagrams of A_i, as described in Lemma 4.5. In order to guarantee that the distances defined by a pair (A_i, B_i) are roughly the same, i.e., up to a factor of $1 + \varepsilon$ of each other, we need to set the values of α and β such that $\frac{\sqrt{2}\alpha+\beta}{\beta} \leq 1+\varepsilon$. We also want to ensure that all the distances defined by the points within a single grid cell of the decomposition are strictly smaller than d_k, so we require that $\alpha/\sqrt{2} < d_k$. Ideally, we would set $\beta = d_k$, however d_k is not known. Instead, let x be a two-sided approximation of d_k, obtained in $O(n \log n + n\varepsilon^2)$ time, by applying the algorithm mentioned above. We set $\beta = \frac{x}{1-\varepsilon} \geq d_k$ and $\alpha = \frac{\varepsilon}{\sqrt{2}} \cdot \frac{x}{1-\varepsilon}$. Notice that $M_i/m_i \leq \frac{\sqrt{2}\alpha+\beta}{\beta} = 1 + \varepsilon$, and that $\alpha/\sqrt{2} < \frac{x}{1+\varepsilon} \leq d_k$, assuming $\varepsilon \leq 1/2$. Moreover, $\beta/\alpha = \sqrt{2}/\varepsilon$.

Let r be the number of distances defined by pairs of points residing in the same grid cell, over all cells of the decomposition. Our choice of α ensures that $k > r$. Let us assume, without loss of generality, that $m_1 \leq m_2 \leq \cdots \leq m_l$. We find the smallest integer j such that $r + \Sigma_{i=1}^{j}\delta_i \geq k$, and set $M = \max_{i=1}^{j} M_i$. We claim that M is a one-sided approximation of d_k, i.e., $d_k \leq M \leq (1+\varepsilon)d_k$. Clearly $d_k \leq M$, since M is the maximum distance among the distances defined by the pairs $(A_1, B_1), \ldots, (A_j, B_j)$ and each of the additional r distances is strictly less than d_k, and the total number of all these distances is at least k. On the other hand, let δ_{l+1} be the number of distances defined by pairs of points residing in different grid cells that are not represented in $\mathcal{W}$. Then, by definition, all these distances are greater than β and therefore strictly greater than d_k. Consider the sum of δ_{l+1} and the number of distances defined by the pairs $(A_j, B_j), \ldots, (A_l, B_l)$. This sum is equal to $\binom{n}{2} - \Sigma_{i=1}^{j-1}\delta_i - r$, and therefore is strictly greater than $\binom{n}{2} - k$. We now establish that $m_j \leq d_k$. Clearly, m_j is the minimum distance among the distances defined by the pairs $(A_j, B_j), \ldots, (A_l, B_l)$. Also, $m_j < m_{l+1}$, where m_{l+1} is the minimum distance among the distances counted by δ_{l+1}, since otherwise there are more than $\binom{n}{2} - k$ distances that are strictly greater than d_k, which is impossible. Now, since m_j is the minimum distance among a set of more than $\binom{n}{2} - k$ distances, it follows that $m_j \leq d_k$. To conclude, let t, $1 \leq t \leq j$, be an index such that $M = M_t$, then, by our choice of α and β, we get that $M = M_t \leq (1 + \varepsilon)m_t \leq (1 + \varepsilon)m_j \leq (1 + \varepsilon)d_k$. The following theorem summarizes the result of this section.

Theorem 6.1. *For any set P of n points in the plane, for any integer parameter k, $1 \leq k \leq \binom{n}{2}$, and for any $0 < \varepsilon \leq 1/2$, one can compute a one-sided approximation of d_k (i.e., of the k'th smallest Euclidean distance among the distances determined by pairs of points in P) in $O((1/\varepsilon^2)n \log n)$ time.*

7 Concluding Remarks

In this paper, we studied the Euclidean bottleneck Steiner path problem and its dual version, and presented efficient solutions for both problems. Our solutions are based on two new geometric structures, namely, a (weak) (α, β)-pair decomposition of a planar point set P and a floor $(1+\varepsilon)$-spanner of P (where the former is used in the construction of the latter). We also used an (α, β)-pair decomposition of P to obtain a strong hop-sensitive spanner and to improve the running time of the algorithm of Bespamyatnikh and Segal [6] for computing a one-sided approximation for planar distance selection. It would be interesting to see additional applications of (α, β)-pair decomposition.

Acknowledgment. We wish to thank an anonymous reviewer who pointed out that the weak version of an (α, β)-pair decomposition of P is sufficient for some of our applications.

References

[1] M.A. Abam, M. de Berg, M. Farshi, and J. Gudmundsson. Region-fault tolerant geometric spanners. *Discrete and Computational Geometry*, 41(4):556–582, 2009.

[2] M.A. Abam and S. Har-Peled. New constructions of SSPDs and their applications. *Computational Geometry: Theory and Applications*, 45(5-6):200–214, 2012.

[3] F. Araújo and L. Rodrigues. Fast localized Delaunay triangulation. In *Proceedings of the 8th International Conference on Principles of Distributed Systems*, pages 81–93, 2004.

[4] S.W. Bae, C. Lee, and S. Choi. On exact solutions to the euclidean bottleneck Steiner tree problem. *Information Processing Letters*, 110:672–678, 2010.

[5] M. de Berg, O. Cheong, M. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.

[6] S. Bespamyatnikh and M. Segal. Fast algorithms for approximating distances. *Algorithmica*, 33: 263–269, 2002.

[7] B. Bollobás and A. Scott. On separating systems. *European Journal on Combinatorics*, 28(4):1068–1071, 2007.

[8] P. Bose, P. Carmi, M. Smid, and D. Xu. Communication-efficient construction of the plane localized Delaunay graph. In *Proceedings LATIN*, pages 282–293, 2010.

[9] P.B. Callahan and S.R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 42:67–90, 1995.

[10] T. Chan. On enumerating and selecting distances. *International Journal of Computational Geometry and Applications*, 11(3):291–304, 2001.

[11] D. Chen, D.-Z. Du, X. Hu, G Lin, L. Wang, and G. Xue. Approximations for Steiner trees with minimum number of Steiner points. *Journal of Global Optimization*, 18:17–33, 2000.

[12] X. Cheng, D.-Z. Du, L. Wang, and B. Xu. Relay sensor placement in wireless sensor networks. *Wireless Networks*, 14:347–355, 2008.

[13] K.L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC '87)*, pages 56–65, 1987.

[14] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms, 3rd edition.* MIT Press, 2009.

[15] G.N. Frederickson and D.B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM Journal of Computing*, 13:14–30, 1984.

[16] S. Har-Peled and B.A. Raichel. Net and prune: A linear time algorithm for Euclidean distance problems. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 605–614, 2013.

[17] Y.-T. Hou, C.-M. Chen, and B. Jeng. An optimal new-node placement to enhance the coverage of wireless sensor networks. *Wireless Networks*, 16:1033–1043, 2010.

[18] H. Huang, A.W. Richa, and M. Segal. Dynamic coverage in ad-hoc sensor networks. *Mobile Networks and Applications*, 10(1):9–17, 2005.

[19] M.J. Katz and M. Sharir, An expander-based approach to geometric optimization. *SIAM Journal on Computing*, 26(5), 1384–1408, 1997.

[20] J.M. Keil. Approximating the complete Euclidean graph. In *Proceedings of the 1st Scandinavian Workshop on Algorithm Theory (SWAT '88), volume 318 of LNCS*, pages 208–213, 1988.

[21] X.-Y. Li. *Wireless Ad Hoc and Sensor Networks: Theory and Applications.* Cambridge University Press, 2008.

[22] 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.

[23] Z.-M. Li, D.-M. Zhu, and S.-H. Ma. Approximation algorithm for bottleneck Steiner tree problem in the Euclidean plane. *Journal of Computer Science and Technology*, 19(6):791–794, 2004.

[24] G.H. Lin and G. Xue. Steiner tree problem with minimum number of Steiner points and bounded edge-length. *Information Processing Letters*, 69:53–57, 1999.

[25] S. Megerian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing*, 4(1):84–92, 2005.

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

[27] L. Roditty and M. Segal. On bounded leg shortest path problems. *Algorithmica*, 59:583–600, 2011.

[28] K.R. Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '98)*, pages 320–331, 1998.

[29] L. Wang and D.-Z. Du. Approximations for a bottleneck Steiner tree problem. *Algorithmica*, 32:554–561, 2002.

[30] L. Wang and Z.-M. Li. Approximation algorithm for a bottleneck k-Steiner tree problem in the Euclidean plane. *Information Processing Letters*, 81:151–156, 2002.

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