Approximate a Minimum-Diameter Spanning Tree with Bounded Degree*

Hee-Kap Ahn† Yo-Sub Han‡ Chan-Su Shin§

1 Introduction

Given a set S of n points in the Euclidean space, the geometric minimum-diameter spanning tree (MDST) of S is a tree that spans S and minimizes its diameter. The diameter of an MDST is the maximum distance over all pairs of points of S, where the distance between points p and q is the sum of edge lengths along the path connecting p and q in the tree. An MDST of S is a variant of the minimum spanning tree (MST) that considers the length of the longest path in the tree. An MDST is also deeply related with a t-spanner of S. A t-spanner for some real number $t > 1$ is a graph G with vertex set S such that any two vertices p and q are connected by a path in G whose length is at most $t \cdot |pq|$, where $|pq|$ denotes the Euclidean distance between p and q. A t-spanner tries to minimize the distance for every pair of points while an MDST focuses on the distance for one specific pair, called diametral pair. Ho et al. [4] presented an $O(n^3)$ time algorithm to compute an MDST and Chan [3] recently proposed an improved algorithm running in $O(n^{3-1/6+\delta})$ time for any $\delta > 0$, where n is the number of points. The algorithm [4] is based on the observation that there always exists a geometric minimum-diameter spanning tree of S with simple topology. Namely, a tree is either monopolar or dipolar, where a *pole* of a tree is a non-leaf point, which has degree of at least two. Thus, the maximum degree for the pole is $n-1$ in a monopolar tree for n points whereas there always exists an MST for a point set in the plane with degree of at most five [6]. It attracts us to consider a problem to construct a bounded degree spanning tree for S in the plane with a constant approximation to an MDST in diameter.

In this paper, we present a simple yet good constant approximation algorithm that bounds the maximum degree of a point in MDST. Actually, we can obtain such a spanning tree by using a t-spanner of S in $O(n \log n)$ time as follows: take a plane t-spanner with $O(n)$ edges in $O(n \log n)$ time [2], and compute a shortest path tree of the spanner from any source point by a conventional Dijkstra's algorithm. The resulting tree has no edge crossings, degree of at most 27, and diameter no more than $2t$ times that of MDST. Here $t \approx 9.24$, so the diameter does not exceed 18.48 times the optimal one. The spanning tree presented in this paper has better performances over all aspects; degree of $2m+1$ for any integer $m > 1$, and diameter of $2(1+\pi/(m-1))$ times the optimal one (less than 2.7 when keeping the degree 27 for the comparison). Furthermore, the tree has an additional interesting property, called *monotonicity.* A tree is said to be *monotone to its root* p such that for any path $P = \langle p, p_1, p_2, \dots, p_t \rangle$ to p_t in the tree $|pp_i| \leqslant |pp_{i+1}|$. This monotone property would be useful to visualize the tree in the plane as an interconnection network.

2 Building a bounded degree tree

Let x, y be a *diametral pair* of points that gives the maximum Euclidean distance between any two points in S. Without loss of generality, we assume that $\overline{xy}$ is vertical and x is above y. Let $C_{p,q}$ denote a disk centered at p with radius $|pq|$. The line passing through x and orthogonal to $\overline{xy}$ separates $C_{x,y}$ into two half-disks. Let C be the lower half-disk. Since x, y are diametral points, C contains all the points in S.

We first explain the algorithm when the root of the spanning tree is one of two diametral points, x here. Similarly, we can also construct

*This research was partially supported by BK21 Program of MOE(Ahn), RGC/CER Grant HKUST6197/01E(Han) and grant No. R05-2002-000-00780-0 from the Basic Research Program of KOSEF(Shin).

†Dept. Computer Science, Korea Advanced Inst. of Science & Tech., Email: heekap@jupiter.kaist.ac.kr

‡Dept. Computer Science, Hong Kong Univ. of Science & Technology, Email: emmous@cs.ust.hk

§School of Electronics and Information Engineering, HUFS, South Korea. Email: cssin@hufs.ac.kr

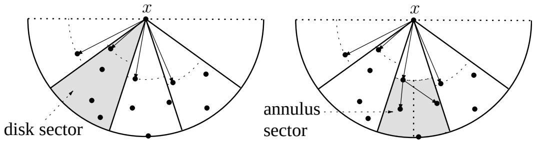

Figure 1: First two stages when $m = 2$.

a spanning tree with an arbitrary point of S as the root, which increases only the diameter a bit.

Let m be a positive integer > 1 to bound the degree. We construct a spanning tree $\mathcal{T}$ for S with degree of $2m{+}1$. Initially, $\mathcal{T}$ consists of the root x only. First, we divide C into $2m{+}1$ equal *disk sectors* as illustrated in Figure 1. For each disk sector K, we choose the closet point p in K from x. This can be answered by solving the three-sided orthogonal range searching problem. The data structure for this range searching must support the deletion. Using the dynamic priority search tree, we can find the closest point p in K from x in $O(\log n)$ time [1]. We connect p to x to be a child of x and the root of the subtree that spans points of S in the *annulus sector* $\overline{K} = K \setminus C_{x,p}$.

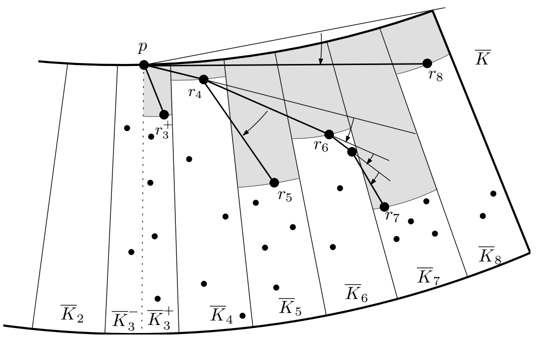

Figure 2: Recursive upper convex hulls in $\overline{K}$.

Now we divide each annulus sector $\overline{K}$ into m equal annulus subsectors and partition the subsection containing the root p into two subsectors to make $m{+}1$ subsectors in total. Then we compute a set S_c of closest points in each subsector from p. For example, $S_c = \{\cdots, r_3^+, r_4, r_5, r_6, r_7, r_8\}$ in Figure 2.

We compute the upper convex hull $\mathrm{H}^+(\overline{K})$ of points in $\overline{K}$ to the right of p. (Of course, we compute $\mathrm{H}^-(\overline{K})$ for the points to the left of p, but we omit it here.) Note that p is on the boundary of $\mathrm{H}^+(\overline{K})$. Then, we connect all points on the boundary of $\mathrm{H}^+(\overline{K})$ into $\mathcal{T}$ and remove them from $\overline{K}$ and recompute $\mathrm{H}^+(\overline{K})$. We repeat this process until all points in S_c are connected into $\mathcal{T}$. Once all points in S_c are connected in $\mathcal{T}$, we take each point from S_c to be a new root and repeat the preceding process while the corresponding annulus sector of a new root is not empty. We need to maintain an upper convex hull $\mathrm{H}^+(\overline{K})$ dynamically. But we need only deletion and split operations on $\mathrm{H}^+(\overline{K})$, therefore we can use the semi-dynamic data structure proposed by Hershberger and Suri [5]. The deletion and split can be done both in $O(\log n)$ time. Thus the entire process takes $O(n \log n)$ time to compute $\mathcal{T}$ for S. We can prove the resulting tree $\mathcal{T}$ has the following four properties: (i) $\mathcal{T}$ has degree of at most $2m{+}1$, (ii) $\mathcal{T}$ has diameter no more than $2(1{+}\pi/(2(m{-}1)))$ times the diameter of MDST, (iii) $\mathcal{T}$ is monotone to the tree root, and (iv) $\mathcal{T}$ has no edge crossing.

The algorithm described so far can be applied when we pick an arbitrary point of S as a root of $\mathcal{T}$. The only difference is that we begin to divide a disk containing S instead of a half-disk. This affects the diameter only.

Theorem 1 *For any integer $m > 1$, we can construct, in $O(n \log n)$ time, a spanning tree $\mathcal{T}$ of n points with an arbitrary point as its root satisfying four properties:* (i) *degree is at most* $2m{+}1$, (ii) *diameter is at most* $2(1{+}\pi/(m{-}1))$ *times the diameter of an MDST,* (iii) *it is monotone to the root, and* (iv) *there is no edge crossing.*

References

[1] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer, Berlin, 1997.

[2] P. Bose, J. Gudmundsson, M. Smid. Constructing Plane Spanners of Bounded Degree and Low Weight. In *ESA*, pp. 234-246, 2002.

[3] T. M. Chan. Semi-online maintenance of geometric optima and measures. In *SODA*, pp. 474–483, 2002.

[4] J.-M. Ho, D. T. Lee, C.-H. Chang, and C. K. Wong. Minimum diameter spanning trees and related problems. *SIAM J. Comput.*, 20:987–997, 1991.

[5] J. Hershberger and S. Suri. Applications of a Semi-Dynamic Convex Hull Algorithm. *BIT*, 32(2):249–267, 1992.

[6] C. Monma and S. Suri. Transitions in geometric minimum spanning trees. *SOCG*, pp. 239–249, 1991.