Computing a $(1+\epsilon)$-Approximate Geometric Minimum-Diameter Spanning Tree

Michael J. Spriggs[*], J. Mark Keil[†]
Sergei Bespamyatnikh[‡], Michael Segal[§], Jack Snoeyink[¶]

July 26, 2003

Abstract

Given a set P of points in the plane, a geometric minimum-diameter spanning tree (GMDST) of P is a spanning tree of P such that the longest path through the tree is minimized. For several years, the best upper bound on the time to compute a GMDST was cubic with respect to the number of points in the input set. Recently, Timothy Chan introduced a sub-cubic time algorithm. In this paper, we present an algorithm that generates a tree whose diameter is no more than $(1+\epsilon)$ times that of a GMDST, for any $\epsilon > 0$. Our algorithm reduces the problem to several grid-aligned versions of the problem and runs within time $O(\epsilon^{-3}+n)$ and space $O(n)$.

1 Introduction

Given a set P of points in the plane, define the weight of an edge between two points of P as the Euclidean (L_2) distance between the points. Compute a spanning tree of P such that the longest path through the tree between any two points is minimized. This tree is known as a *geometric minimum diameter spanning tree* (GMDST) of the point set.

GMDSTs are relevant to interconnect optimization in VLSI physical design [4, 5]. However, in this application the total length of the spanning tree is also important.

Ho et al. [8] describe an algorithm that uses $O(n^3)$ time to generate a GMDST of n points in the plane. In a generalized version of the problem, the input is a graph that consists of n vertices and m weighted edges. We must compute a *minimum diameter spanning tree* (MDST) of the graph. The graph version of the problem is relevant to the design of communication networks. Camerini, Galbiati and Maffioli [2] and Hassin and Tamir [7] show that the problem of generating a MDST of a weighted graph is reducible to the *absolute 1-center* problem on a graph. This problem is

[*]School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1, mjspriggs@uwaterloo.ca. Supported in part by NSERC.

[†]Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, S7N 5A9, keil@cs.usask.ca, 306-966-4894, FAX 306-966-4884. Supported by NSERC.

[‡]Department of Computer Science, Duke University, Box 90129, Durham, NC 27708, U.S.A., besp@cs.duke.edu

[§]Communication Systems Engineering Department, Ben-Gurion University of the Negev, Beer-Sheva, 84105 Israel, segal@cse.bgu.ac.il.

[¶]Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175 U.S.A., snoeyink@cs.unc.edu

computable within $O(mn + n^2 \log n)$ time [10]. Note that the time bound degenerates to $O(n^3)$ when the input graph is the complete graph induced by n points in the plane.

For several years, no improvement was made to the cubic time bound on computing a GMDST. While our paper was being refereed, Timothy Chan [3] improved this bound. He describes an algorithm that computes a GMDST of n points in the plane within $\tilde{O}(n^{\frac{17}{6}})$ time, where the $\tilde{O}$-notation hides terms of the form $o(n^{\alpha})$, for any fixed $\alpha > 0$. Chan's approach can be applied to point sets in any d-dimensional space. As the dimension increases, the time bound of the algorithm quickly approaches cubic.

The (almost) cubic time bound on GMDST-generation may be too large for some applications. We are interested in finding an efficent approximation algorithm that generates a spanning tree whose diameter is no more than $(1 + \epsilon)$ times the length of a GMDST, for any $\epsilon > 0$. The efficiency of such an algorithm is measured in terms of n and ϵ. One of our goals is to reduce the interdependancy of ϵ and n as much as possible. In this paper, we describe an algorithm that performs the required task in $O(n + \frac{1}{\epsilon^3})$ time and $O(n)$ space.

Remark. While our paper was being refereed, Gudmundsson et al. [6] also considered the problem of approximating a geometric minimum-diameter spanning tree. Using such heavy machinery as well-separated pair decomposition (WSPD) – first introduced by Callahan and Kosaraju [1] – they were able to obtain an $O(\frac{n}{\epsilon^3} + \frac{n \log n}{\epsilon})$ time algorithm that uses $O(\frac{n}{\epsilon^2} + n \log n)$ space. The authors also describe a grid-based approximation algorithm that takes $O{*}(\epsilon^{-17/3} + n)$ time and $O{*}(\epsilon^{-2} + n)$ space, where the $O{*}$-notation hides terms of the form $O(\log^{O(1)} \epsilon^{-1})$. Combining these two approaches, the authors devise an approximation algorithm that runs in $O{*}(\epsilon^{-5} + n)$ time and $O(\epsilon^{-4} + n)$ space. Note that our grid-based algorithm has better performance in both characteristics than any of these algorithms.

In the next section we present several known properties of GMDSTs. In sections 3 and 4, we solve restricted versions of the problem where the points are grid-aligned and the GMDST is restricted to also be aligned with the grid. Section 5 presents our approximation algorithm where we show how to reduce the problem to several of the restricted versions we solved in section 4. We conclude with section 6.

2 Preliminaries

A *geometric graph* is a graph that is composed of a set of *points*, and a set of *edges*. Each edge can be represented by a line segment between two points. The *weight* of an edge is the Euclidean, straight-line distance between its endpoints. The *diameter* of a graph is the sum of the edge-weights of the longest path through the graph.

Given a point set P, a k-star, $1 \leq k \leq n$ for P is a spanning tree of the complete Euclidean graph on P which has k internal nodes. Ho et al. [8] show that every point set admits a GMDST that is either a 1-star or a 2-star, by a nice geometric application of the triangle inequality.

For a given spanning tree of a point set P, $\mathcal{T}(P)$, let $|\mathcal{T}(P)|$ denote the diameter of $\mathcal{T}(P)$. For a path $p_1 p_2 \ldots p_k$ through a tree, let $|p_1 p_2 \ldots p_k|$ denote the length of the path.

It is tempting to conjecture that one may form a minimum diameter 2-star with internal vertices p_1 and p_2 by simply joining every other point to the closer of p_1 or p_2. Figure 1 shows that this is not sufficient, but illustrates a condition of Jones [9] for how the remaining points in P can be connected to the given interior vertices of a 2-star.

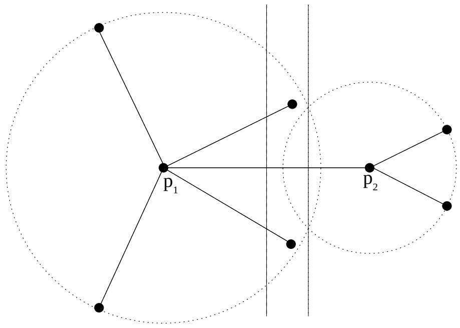

Figure 1: A 2-star GMDST for which connecting each point to the closer internal vertex would produce a spanning tree of greater diameter. This tree satisfies the stability condition, since neither circle contains all points.

Lemma 2.1 *[9] Suppose that points $p_1, p_2 \in P$ form a horizontal line $\overline{p_1 p_2}$ from left to right. There exists a vertical line L such that connecting points left of L to p_1 and right of L to p_2 produces a minimum diameter spanning tree among all spanning trees with interior vertices p_1 and p_2.*

Ho et al. [8] also establish a *stability condition* for 2-stars, which can also be illustrated in Figure 1. Let $P_i \subset P \setminus \{p_1, p_2\}$ denote the points joined to interior vertex p_i.

Lemma 2.2 *[8] The diameter of a 2-star is determined by a three-edge path if, for $i \in \{1, 2\}$, some point not joined to p_i is farther than all points joined to p_i. That is,*

$$\max_{q \in P_i} |p_i q| < \max_{q' \in P_{3-i}} |p_i q'|.$$

We define a $(1+\epsilon)$-*approximation* of an optimal GMDST as a spanning tree whose diameter is no more than $(1+\epsilon)$ times the diameter of a GMDST. We call an algorithm that produces such a tree, a $(1+\epsilon)$-*approximation algorithm.*

A *uniform grid* is a grid composed of an infinite number of horizontal and vertical lines in the plane, such that adjacent lines are placed at uniform intervals. The grid breaks up the plane into square regions that we call *grid-squares.* The *center* of a grid-square is the point in the middle of the square that lies equidistant from all four corners of the grid-square. Define a *grid-aligned point set* as a set of points in the plane, such that points lie only at the centers of grid-squares, with respect to some underlying grid.

To eliminate special cases concerning point sets containing fewer than four points, we shall assume throughout this paper that the input set P is such that $|P| \geq 4$.

Note that our approximation algorithm relies upon a real-RAM model of computation, in which the sum of two real numbers can be computed exactly within constant time.

3 A Simpler Problem

Instead of a general point set, we first consider a grid-aligned point set P contained in an m row and m column bounding box. We also restrict the spanning tree to be either a 1-star or a 2-star such that both interior vertices lie in a single row of the grid. We call this special version of the GMDST a restricted geometric minimum diameter spanning tree (RGMDST).

Suppose further that we have been given, for each row j of the m rows, two candidate points $p_{j1}, p_{j2} \in P$ such that (a) if the RGMDST is a 1-star, then the interior vertex is p_{ja} for some $1 \leq j \leq m$ and $a \in \{1, 2\}$, and (b) if the RGMDST of P is a 2-star then the interior vertices are p_{j1} and p_{j2} for some $1 \leq j \leq m$. We now show how we can efficiently exactly solve the RGMDST problem when we are given the candidate interior vertices. To do this we first find the optimal 1-star, then find the optimal 2-star and take the minimum of the two.

Lemma 3.1 *Let P be a grid-aligned set of n points contained in an $m \times m$ bounding box. Given two candidate interior vertices $p_{j1}p_{j2} \in P$ in each row j, $1 \leq j \leq m$, such that the optimal interior vertex is among the candidates, then the minimum diameter 1-star of P can be found in $O(m \log m + n)$ time and $O(m + n)$ space.*

Proof: The optimal interior vertex candidate will minimize the sum of the distance to its furthest neighbor and its second furthest neighbor. The potential furthest and second furthest neighbors are the top and bottom two points of P in each column. Ho et al. [8] show how the second order furthest-neighbor Voronoi diagram of these $4m$ points can be computed in $O(m \log m)$ time. Using point location in this diagram, each of the $2m$ candidate monopoles can be evaluated in $O(\log m)$ time. $\square$

We now turn to 2-stars.

Lemma 3.2 *Let P be a grid-aligned set of n points contained in an m row and m column bounding box. Given candidate interior vertices $p_{j1}p_{j2} \in P$ in each row j, $1 \leq j \leq m$, such that the optimal horizontally restricted 2-star has interior vertices among the candidate pairs, then the optimal 2-star RGMDST of P can be found in $O(m^2 + n)$ time and $O(m + n)$ space.*

Proof: Assume that p_{j1} is left of p_{j2} for all grid-rows j. By lemma 2.1 we know that a minimum diameter 2-star with interior vertices p_{j1} and p_{j2} can be found by joining the points left of some vertical line to p_{j1} and the remaining points to p_{j2}. Since we do not know in advance which vertical line is appropriate, we use a sweepline to examine all possibilities.

To evaluate a particular position of the sweepline we need to know the furthest neighbor of p_{j1} left of the sweepline and the furthest neighbor of p_{j2} right of the sweepline. The potential furthest neighbors are the top and bottom points in each column. These extreme points can be precomputed in $O(m + n)$ time and space.

Let us now fix our attention to a particular row j. The sweep begins with the vertical sweepline left of the grid-aligned bounding box B. We note that there is no furthest point from p_{j1} left of the sweepline. We then move the sweep line to the right one grid-column at a time. After each move we compute the furthest neighbor of p_{j1} that is left of the sweepline. This furthest neighbor will either be the highest or lowest point of $P \backslash \{p_{j1}, p_{j2}\}$ in the column just left of the sweepline, or it will be the previous furthest neighbor from before the move. We record for each column k the distance from the furthest left neighbor, $l_1(k)$, which is the distance from the element of $P \backslash \{p_{j1}, p_{j2}\}$ left of or in column k that is furthest from p_{j1}. If there is no such furthest left neighbor $l_1(k)$ is zero. Each $l_1(k)$ is computed in constant time. We can likewise sweep a vertical line from the right and compute $r_1(k)$, the distance from the element of $P \backslash \{p_{j1}, p_{j2}\}$ right of or in column k that is furthest from p_{j1}. Finally, we also compute $l_2(k)$ and $r_2(k)$, as the corresponding distances to pivot p_{j2}.

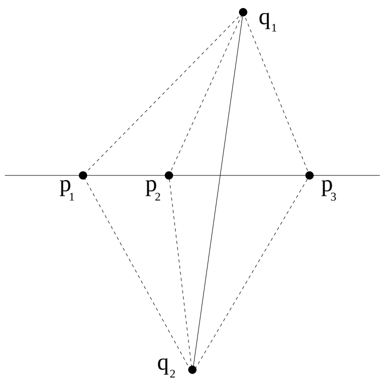

Figure 2: The diameter of a 1-star is non-decreasing about the optimum.

After this is done we can compute the diameter of the 2-star with interior vertices p_{j1} and p_{j2} as

$$D_j = \min_k \{l_1(k) + d(p_{j1}, p_{j2}) + r_2(k+1)\} \tag{1}$$

where the minimum is taken over all columns for which the stability conditions $l_1(k) < r_1(k+1)$ and $r_2(k+1) < l_2(k)$ both hold. If one of the two conditions does not hold then the computation of equation 1 underestimates the actual diameter determined by the two furthest points from one of the two interior vertices. Since there are m columns, altogether $O(m)$ time and space is expended in computing and storing the $l_i(k)$ and $r_i(k)$ values for the row j. Testing stability conditions and computing the result, D_j, then requires a further $O(m)$ time using equation 1.

We repeat this process for each of the m rows. Altogether $O(m^2)$ time and $O(m)$ space is expended in computing all the D_j. The diameter of the optimal 2-star with interior vertices p_{j1} and p_{j2}, for all grid-rows, can then be computed in a further $O(m)$ time. □

4 Solving the RGMDST problem

In this section we show how to compute candidate interior vertices so that the RGMDST problem of a grid-aligned point set P can be reduced to the simpler problem of section 3.

4.1 Analysis

Imagine a 1-star of a planar point set. Without changing the structure of the graph, move the interior vertex to the left and to the right along a horizontal line. How does moving the interior vertex in this manner affect the diameter of the 1-star? As we show in the following lemma, there is an optimum region where the diameter of the tree is minimum. As we move away from this region, the diameter of the tree is non-decreasing, i.e. there are no points of "local optimum" along the line.

Lemma 4.1 *Given a point set P, and three collinear points p_1, p_2, p_3, such that p_2 lies between p_1 and p_3, the diameter of the 1-star of P in which p_2 is the interior vertex cannot be larger than both the diameter of the 1-star with interior vertex p_1, and the diameter of the 1-star with interior vertex p_3.*

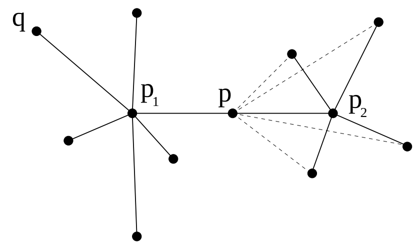

Figure 3: Improving a 2-star spanning tree.

Proof: Let points $q_1, q_2 \in P$ be the furthest two points from p_2 in P. The longest path through the tree in which p_2 is the interior vertex is $q_1p_2q_2$. The longest path through the 1-star in which p_1 is the interior vertex must be at least as large as the path $q_1p_1q_2$. Likewise the longest path through the 1-star in which p_3 is the interior vertex must be at least as large as the path $q_1p_3q_2$.

Suppose, for the moment, that q_1 and q_2 at opposite sides of the line through p_1, p_2 and p_3, as is shown in Figure 2. The shortest path with endpoints q_1 and q_2 is the straight line segment between q_1 and q_2. The more a path deviates from the straight line, the longer it is. Clearly, $q_1p_2q_2$ deviates less than at least one of $q_1p_1q_2$, or $q_1p_3q_2$. Hence, the diameter of the 1-star with interior vertex p_2 cannot be larger than both the diameter of the 1-star with interior vertex p_1, and the diameter of the 1-star with interior vertex p_3.

A similar argument can be made for the situation in which q_1 and q_2 lie on the same side of the line through p_1, p_2 and p_3. Finally, it is not difficult to show that the lemma holds in the degenerate case in which one or both of q_1, q_2 lie collinear with p_1, p_2 and p_3. We leave the proof up to the reader. $\square$

For a given grid-row r and a point set P, we define a *grid-aligned Steiner monopole* of r and P to be a grid-aligned point s that lies in r, such that the diameter of the 1-star of $P \cup \{s\}$ with interior vertex s is minimum among all such 1-stars with interior, grid-aligned Steiner vertices in r. Note that s is not necessarily an element of P. Also, we do not exclude the possibility of the existence of multiple grid-aligned Steiner monopoles in r. However, by Lemma 4.1, these will form a continuous set of grid-aligned points in r.

The following lemma describes a useful relationship between the optimal grid-aligned Steiner monopoles in a given grid-row, and the optimal pair of interior vertices of a 2-star that is restricted to lie in that grid-row.

Lemma 4.2 *Let P be a grid-aligned point set, and let r be a grid-row containing points of P. Let $\Delta_r(P)$ be a minimum diameter spanning tree of P which is restricted to be either a 1-star with interior vertex in row r, or a 2-star with both interior vertices in row r. If there exists no 1-star $\Delta_r(P)$ then there exists a 2-star $\Delta_r(P)$ of P in which (1) no point of P lies on the horizontal line segment between the two interior vertices of $\Delta_r(P)$, p_1 and p_2, and (2) every grid-aligned Steiner monopole s in grid-row r lies between p_1 and p_2.*

Proof: Assume that there exists no 1-star $\Delta_r(P)$. Then $\Delta_r(P)$ is 2-star with interior vertices p_1 and p_2, where p_1 is to the left of p_2.

Suppose that there exists a point $p \in P$ on the horizontal line segment between p_1 and p_2. Let P_i be the points of $P \backslash \{p_1, p_2, p\}$ connected to p_i in $\Delta_r(P)$, $i = 1, 2$. If the furthest point from p in

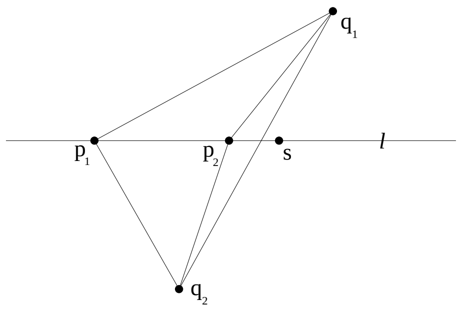

Figure 4: The Steiner monopole s on line l cannot lie to the right of the optimal interior vertices p_1 and p_2.

P lies in P_1, then generate a new 2-star with interior vertices p_1 and p. In Figure 3, the furthest point from p is labeled q. In the new tree, add an edge between p_1 to p, add edges between p_1 and every other point (besides itself and p) in P_1, and add edges between p and every other point in P_2 (these edges are shown dashed in Figure 3).

Since the furthest point from p lies in P_1, the longest path through the new tree will either run (1) from a point in P_1, to p_1, and to another point in P_1, or else (2) from a point in P_1, to p_1, to p, and finally to some point in P_2. In either case, it is easy to see that by the triangle inequality, the diameter of the new tree cannot be larger than the diameter of $\Delta_r(P)$.

Therefore, if $\Delta_r(P)$ must be a 2-star there exists a $\Delta_r(P)$ such that no point of P lies on the horizontal line segment between the interior vertices.

We must now show that, if there exists only a 2-star $\Delta_r(P)$, then all grid-aligned Steiner monopoles in grid-row r lie between the interior vertices. Assume to the contrary that a Steiner monopole s of grid-row r lies to the right of p_2. The case that it lies left of p_1 is symmetric.

Let q_1 and $q_2 \in P$ be the two furthest points from p_2. For the moment, let us suppose that q_1 and q_2 lie on opposite sides of the horizontal line through the interior vertices. This situation is depicted in Figure 4.

The line through q_1 and q_2 must intersect the horizontal line through interior vertices p_1 and p_2 to the right of p_2. If this were not the case, then notice that the 1-star with interior vertex at p_2 would have a diameter smaller than that of the monopolar tree with interior vertex s. The diameter of the 1-star with interior vertex p_2 is $|q_1 p_2 q_2|$. A 2-star with interior vertices p_1 and p_2 must have at least this diameter, contradicting our initial assumption that $\Delta_r(P)$ cannot be a 1-star.

The case that q_1 and q_2 reside on the same side of the horizontal line through the dipoles is similar. If a 2-star $\Delta_r(P)$ exists in which a Steiner monopole in the same grid-row as the interior vertices does not lie between the interior vertices, then there exists a 1-star $\Delta_r(P)$. $\square$

4.2 Computing candidate interior vertices

We now show how to compute, for each grid-row in the bounding box of P, a grid-aligned Steiner monopole of that row. From this information, we can compute a set of candidate interior vertices. These are the two points of each row, that are nearest the grid-aligned Steiner monopole, such that one lies left of the Steiner monopole, and the other lies to the right, by Lemma 4.2.

As in the proof of lemma 3.1, the second order furthest point Voronoi Diagram of P can be computed in $O(m \log m)$ time. Once the second order furthest-neighbor diagram is computed, the furthest two points from any point p are those two points corresponding with the Voronoi cell that contains p. For each row of the grid, we can compute the furthest two points from every grid-aligned point in that row in an additional $O(m)$ time by traversing Voronoi cells, from a cell to an adjacent cell, along the line that passes through the points in the row. Once we know the furthest two points from a point p, in constant time we can compute the diameter of the 1-star with interior vertex p. There are m rows, and so it will take a total of $O(m^2)$ time to process all rows in this manner. This gives us a best possible grid-aligned Steiner monopole and its cost in $O(m^2)$ time. The candidate interior vertices of each row can be computed in an additional $O(m)$ time per row.

We now present the main result of this section.

Theorem 4.1 *Given a set P of n points aligned with a grid $\mathcal{G}$ and contained within a bounding box of $m \times m$ grid-squares, there is an $O(m^2)$-time algorithm that generates a RGMDST of P.*

Proof: Once we determine the best candidate interior vertices in each grid-row, in $O(m)$ time per row, this reduces the problem to the simpler problem of section 3. Lemmas 3.1 and 3.2 complete the proof. □

5 The approximation algorithm

In order to approximate an optimum GMDST of a general point set, we transform the problem to several instances of the RGMDST problem. Let P be an arbitrary set of points in the plane with a GMDST $\Delta(P)$.

Suppose that $\Delta(P)$ is a 2-star. We seek to overlay the plane with a grid $\mathcal{G}$, with grid-square edges of length ϕ, so that that the two interior vertices of $\Delta(P)$, p_1 and p_2, lie in a single grid-row. Intuitively, the angle between the line that passes through the two interior vertices of $\Delta(P)$ and the orientation of the "horizontal" grid lines of $\mathcal{G}$ should be small. Let $\mathcal{D}$ denote the distance between the furthest two points of P. Therefore, $|p_1p_2| \leq \mathcal{D}$. Let θ denote the smaller angle between the line through p_1 and p_2, and the horizontal grid-lines of $\mathcal{G}$.

Using trigonometry, we find that if

$$\sin\theta < \frac{\phi}{\mathcal{D}} \tag{2}$$

then p_1 and p_2 can reside in a single grid-row of a grid oriented like $\mathcal{G}$. By insisting that

$$\sin\theta < \frac{\phi/2}{\mathcal{D}} = \frac{\phi}{2\mathcal{D}} \tag{3}$$

we need consider only two such grids, offset from one another by a vertical distance of $\frac{\phi}{2}$.

Figure 5 shows that if θ is small enough the segment joining the interior vertices. will lie in one of the two grids whose horizontal grid lines are offset by $\frac{\phi}{2}$

In order to account for all possible orientations of the line through p_1 and p_2, several orientations of grids are used.

Lemma 5.1 *The number of orientations of grids can be bounded by,*

$$\frac{\pi}{\arcsin(\phi/2\mathcal{D})} = O\left(\frac{\mathcal{D}}{\phi}\right) \tag{4}$$

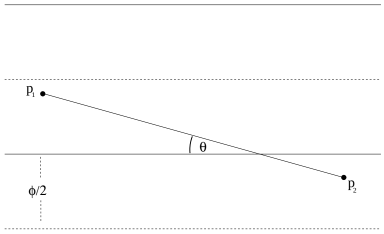

Figure 5: The line through p_1 and p_2, closely aligned with the horizontal grid-lines of two offset grids.

5.1 Grid transformation

One of the $O(\frac{\mathcal{D}}{\phi})$ oriented grids will contain the two interior vertices of an optimal 2-star of P in a single grid-row. For each such grid, we need to generate a grid-aligned point set P' from P and analyze how the gridding changes the GMDST. If any point $p \in P$ lies on the boundary of two or more grid-squares, the point is moved a negligible distance in some direction until it no longer lies on a boundary. We generate the grid-aligned point set P' as follows. For each grid-square in $\mathcal{G}$, if the grid-square contains a single point of P, then add a single point in the center of the grid-square to P'. If two or more points of P reside in the grid-square, then add exactly two points to P', such that both points lie at the center of the grid-square. Notice that set P' is aligned with grid $\mathcal{G}$. Below, we show how a GMDST of P' can be converted to a spanning tree of P, such that the diameter of this spanning tree is close to optimum.

Let $\Delta(P')$ be a GMDST of P'. Generate a spanning tree of P, $\mathcal{T}(P)$, as follows. If $\Delta(P')$ is a 1-star or a 2-star with both interior vertices in the same grid-square, then make $\mathcal{T}(P)$ a 1-star such that the interior vertex of $\mathcal{T}(P)$ is any point $p \in P$ that lies in the same grid-square as does the interior vertex of $\Delta(P')$.

Otherwise $\Delta(P')$ is a 2-star with interior vertices p'_1 and p'_2 in different grid-squares. Choose two interior vertices $p_1, p_2 \in P$ such that p_1 lies in the same grid-square as p'_1 and p_2 lies in the same grid-square as p'_2. In $\mathcal{T}(P)$, generate an edge between p_1 and p_2. For every point $p \in P$, where $p \neq p_i$ for $i \in \{1, 2\}$, such that p resides in the same grid-square as p_i, add edge pp_i to $\mathcal{T}(P)$.

For any grid-square that contains at least one point of P', if one or both of these points are linked by an edge to vertex p'_1 in $\Delta(P')$, then for each point $p \in P$ that resides in the grid-square add edge p_1p to $\mathcal{T}(P)$. Otherwise, for each point $p \in P$ that resides in the grid-square, add edge p_2p to $\mathcal{T}(P)$.

We call the above procedure of converting P to P', and then using a GMDST of P' to generate a spanning tree of P, the *grid transformation*.

Lemma 5.2 *Given a set P of points in the plane with GMDST $\Delta(P)$, and some value $\phi > 0$, the grid transformation generates a spanning tree of P, $\mathcal{T}(P)$, such that*

$$|\mathcal{T}(P)| \leq |\Delta(P)| + 6\sqrt{2}\phi \tag{5}$$

Proof: Each point of P' lies at the center of a grid-square, while each point of P may lie anywhere in the plane. Given points $p \in P$ and $p' \in P'$ such that both points lie in the same grid-square, the distance between the points will not exceed $\sqrt{2}\phi/2$, i.e. half of the distance between two furthest corners of the grid-square.

Let $\Delta(P)$ be a GMDST of P. The longest path from one point of P to another through $\Delta(P)$ will consist of at most three edges. Let $\mathcal{G}$ be the underlying grid of P'. Suppose that we move each point of P to the center of the grid-square of $\mathcal{G}$ in which it is contained. As the points are moved, each edge of $\Delta(P)$ is stretched by at most length $\sqrt{2}\phi$. Since the longest path through $\Delta(P)$ consists of three edges, the diameter of the stretched version of $\Delta(P)$ is no longer than $|\Delta(P)| + 3\sqrt{2}\phi$.

Let $\Delta(P')$ be a GMDST of P'. A spanning tree of P' can be constructed from the above stretched version of $\Delta(P)$ by removing all but at most two of the points (and those edges connected to these points) from each grid-square of $\mathcal{G}$. Notice that this operation does not increase the diameter of the spanning tree. Hence,

$$|\Delta(P')| \leq |\Delta(P)| + 3\sqrt{2}\phi \tag{6}$$

Given $\Delta(P')$, the grid transformation describes how to convert this spanning tree to a spanning tree of P, $\mathcal{T}(P)$. Consider the longest path between two points of P through $\mathcal{T}(P)$. Imagine that all the points of P are moved into the centers of the grid-squares of $\mathcal{G}$ without altering the edges of $\mathcal{T}(P)$. Since the longest path through $\mathcal{T}(P)$ consists of at most three edges, this path is no longer than the longest path through the stretched version, plus $3\sqrt{2}\phi$. To convert the stretched version of $\mathcal{T}(P)$ back to P', we must remove some of the points (and attached edges) such that there are only two points in each grid-square of $\mathcal{G}$. This is done by removing leaves. Since there remain up to two points in each grid-square, this removal of points does not shorten the diameter of the stretched tree.

Therefore,

$$|\mathcal{T}(P)| \leq |\Delta(P')| + 3\sqrt{2}\phi \tag{7}$$

Combining Equation 6 with Equation 7, it follows that,

$$|\mathcal{T}(P)| \leq |\Delta(P)| + 6\sqrt{2}\phi \tag{8}$$

□

5.2 Putting it all together

In this subsection, we combine results of the preceding subsections to form an approximation algorithm for the problem of GMDST generation. The following is a useful lower bound on the diameter of a GMDST.

Lemma 5.3 *Given a set P of points in the plane such that $\mathcal{D}$ is the largest distance between any two points of P, any GMDST of P must be of size $\mathcal{D}$ or larger.*

We now present our main result.

Theorem 5.1 *Given a set P of n points in the plane, there exists an algorithm such that, for any $0 < \epsilon$, the algorithm generates a $(1+\epsilon)$-approximate GMDST of P within time $O(\epsilon^{-3} + n)$ and space $O(n)$.*

Proof: If $\epsilon < \frac{1}{n}$ then we use the exact algorithm of Ho et al. [8] which runs in $O(n^3) \subseteq O(\epsilon^{-3})$ time and $O(n)$ space. Otherwise we proceed using the grid transformation. If n is larger than the number of nonempty grid squares then it will be too costly to repeatedly place all n points on each of the oriented grids. Instead we apply an initial grid transformation to reduce the number of points and then repeatedly apply the grid transformation to these initially gridded points to move them to the variously oriented grids. Using the grid transformation twice in this way will double the additive error of lemma 5.2.

Recall that $\mathcal{D}$ is the furthest distance between two points in the input set P. For each grid transformation let ϕ be the edge-length of the grid-squares. Set ϕ such that,

$$\phi = \frac{\mathcal{D}\epsilon}{12\sqrt{2}} \tag{9}$$

Let $\Delta(P)$ be a GMDST of P. By applying the grid transformation twice the problem of generating a spanning tree that is no larger than $|\Delta(P)| + 12\sqrt{2}\phi$ can be transformed to several instances of the GMDST on a grid-aligned point set. For at least one of these instances a RGMDST will serve as an approximation. For the correct orientation of grid with our chosen value of ϕ,

$$\frac{|\Delta(P)| + 12\sqrt{2}\phi}{|\Delta(P)|} \leq 1 + \frac{12\sqrt{2}}{|\Delta(P)|} \cdot \frac{\mathcal{D}\epsilon}{12\sqrt{2}} \leq (1+\epsilon) \tag{10}$$

since, by Lemma 5.3, $\mathcal{D} \leq |\Delta(P)|$.

Therefore, our algorithm computes a $(1+\epsilon)$-approximate GMDST of P. We now examine the complexity of this algorithm.

Consider the conversion of P to each grid-aligned set P'. Before this conversion can occur, we must compute the size of the grid-squares, ϕ, using ϵ and $\mathcal{D}$. We know ϵ, but must compute $\mathcal{D}$ from P. We can approximate the defined value of $\mathcal{D}$ by setting $\mathcal{D}$ equal to the maximum of (1) the largest vertical distance between two points of P, and (2) the largest horizontal distance between two points of P. We get a value for $\mathcal{D}$ such that $\mathcal{D} \leq |\Delta(P)|$. Further, this value can be computed in linear time with respect to the number of points in P.

Since no two points of P lie vertically or horizontally further apart than distance $\mathcal{D}$, set P will fit entirely within a grid-aligned, $\mathcal{D} \times \mathcal{D}$ bounding box, B. These dimensions can be rewritten in terms of the number of grid-squares on each side of B as,

$$\frac{\mathcal{D} \times \mathcal{D}}{\phi^2} = \frac{\mathcal{D} \times \mathcal{D}}{(\mathcal{D}\epsilon)^2/(12\sqrt{2})^2} = \frac{288}{\epsilon^2} = O(\epsilon^{-1}) \times O(\epsilon^{-1}) \tag{11}$$

Recall that set P' is generated by adding up to two points to the center of each grid-square using the two grid transformations. The first transformation can be accomplished within time (and space) $O(\epsilon^{-1} + n)$ by using radix sort and is done only once. The second transformation can be performed in $O(\epsilon^{-2})$ time and $O(n)$ space. Once an optimum RGMDST spanning tree of P' is generated the size is recorded. Only for the grid orientation that allows the minimum sized RGMDST is the tree converted to a spanning tree of P within an additional $O(\epsilon^{-2} + n)$ time. Hence, the grid transformation runs in a total of $O(\epsilon^{-2})$ time per orientation plus $O(\epsilon^{-2}+n)$ initial and final costs.

By Lemma 5.1, the number of grid orientations considered for the grid transformation can be bounded by $O(\mathcal{D}/\phi)$, where $\mathcal{D}$ is the largest distance between any two points in P. Therefore, the number of orientations can be bounded by,

$$O\left(\frac{\mathcal{D}}{\phi}\right) = O\left(\frac{\mathcal{D}}{\mathcal{D}\epsilon}\right) = O(\epsilon^{-1}) \tag{12}$$

For each of these $O(\epsilon^{-1})$ orientations we expend $O(\epsilon^{-2})$ time for the second grid transformation plus an additional $O(\epsilon^{-2})$ time to solve the RGMDST problem on the gridded point set. Overall, our approximation algorithm runs in $O(\epsilon^{-3} + n)$ time and $O(n)$ space. □

6 Conclusions

The approximation algorithm that we present in this paper is linear with respect to the size of the input set. For applications in which an approximate GMDST will suffice, this is a significant improvement over the existing cubic time algorithms. It remains open as to whether or not the cubic time bound can be improved upon by an algorithm that is guaranteed to find an optimum solution.

Using the algorithm in the paper we obtain a spanning tree whose diameter d is no more than $(1+\epsilon)$ times the diameter d^* of GMDST, i.e. $d^* \leq d \leq (1+\epsilon)d^*$. We may ask another question: Given a value δ, find a spanning tree whose diameter d is no more than the optimal diameter $d^* + \delta$. In other words, $d^* \leq d \leq d^* + \delta$. To achieve this with our algorithm we need $\epsilon d^* \leq \delta$ and therefore $\epsilon \leq \frac{\delta}{d^*}$ However, we don't know d^*. To determine the needed ϵ we first run the algorithm with $\epsilon = 1$ and obtain d', such that $d^* \leq d' \leq 2d^*$, then choose $\epsilon = \frac{\delta}{d'} \leq \frac{\delta}{d^*}$ and run the algorithm with this ϵ. The running time will be $O(\epsilon^{-3} + n) = O(\frac{d'^3}{\delta^3} + n) = O(\frac{d^{*3}}{\delta^3} + n)$. Notice that if all pairwise distances between points are integers, then by taking $\delta = \frac{1}{2}$ we obtain an exact output sensitive solution in $O(d^{*3} + n)$ time.

We expect that the approach used in this paper can be used to compute approximate GMDSTs for point sets in higher dimensions. As we move into d-dimensional space, the uniform grids will be composed of d-dimensional hypercubes. However, as d increases the exponent on ϵ^{-1} in the complexity bounds will grow fairly quickly, likely making the approach impractical for dimensions above three or four.

References

[1] P. Callahan and R. Kosaraju, A decomposition of multidimensional point sets with applications to k-nearest neighbors and n-body potential fields, *Journal of the ACM*, 42(1) (1995), 67–90.

[2] P.M. Camerini, G. Galbiati, and F. Maffioli, Complexity of spanning tree problems I, *European J. Oper. Res.*, 5 (1980), 346–352.

[3] T. Chan, Semi-online maintenance of geometric optima and measures, *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2002), 474–483.

[4] J. Cong, L. He, C. Koh, and P. Madden, Performance optimization of VLSI interconnection layout, *Integration: the VLSI Journal*, 21 (1996), 1–94.

[5] D. Eppstein, Spanning trees and spanners, in *Handbook of Computational Geometry* (Sack and Urrutia, eds.), North Holland, 2000, pp. 425–462.

[6] J. Gudmundsson, H. Haverkort, S.-M. Park, C.-S. Shin, and A. Wolff, Facility location and the geometric minimum-diameter spanning tree, in *Proc. 5th International Workshop on Approx. Algorithms for Comb. Opt. (APPROX)* (2002), 146–160.

[7] R. Hassin and A. Tamir, On the minimum diameter spanning tree problem, *Information Processing Letters* 53 (1995), 109–111.

[8] J. Ho, D. Lee, C. Chang, and C. Wong, Minimum diameter spanning trees and related problems, *SIAM Journal on Computing*, 20 (1991), 987–997.

[9] W. D. Jones, Euclidean Communication Spanning Trees, M.Sc. Thesis, Univ. of Saskatchewan, 1994.

[10] O. Kariv and S. L. Hakimi, An algorithmic approach to network location problems I: the p-centers, *SIAM J. Applied Mathematics*, 37 (1979), 513–537.