Construction of Multidimensional Spanner Graphs, with Applications to Minimum Spanning Trees

J. S. Salowe

Computer Science Report No. TR-90-28
September 20, 1990

Construction of Multidimensional Spanner Graphs, with Applications to Minimum Spanning Trees

by

J.S. Salowe[1]

Department of Computer Science
University of Virginia
Charlottesville, Virginia 22903

ABSTRACT

Given a connected graph $G = (V,E)$ with positive edge weights, define the distance $d_G(u,v)$ between vertices u and v to be the length of a shortest path from u to v in G. A subgraph G' of G is said to be a t-spanner for G if, for every pair of vertices u and v, $d_{G'}(u,v) \le t \cdot d_G(u,v)$. We show how to construct a $(1+\varepsilon)$-spanner for a complete Euclidean graph in $O(n \log n + \frac{n}{\varepsilon^d})$ time; this algorithm works in any L_p metric. This spanner is used to construct approximate minimum spanning trees, obtaining a result similar to Vaidya [13].

1. Introduction

Let $G = (V,E)$ be a connected graph with positive edge weights. A subgraph $G' = (V,E')$ of G is a t-spanner if

$$\max_{u,v \in V} \frac{d_G'(u,v)}{d_G(u,v)} \le t,$$

where $d_G(u,v)$ is the length of a shortest path from u to v in G. There are several sparse graphs which are known to be t-spanners for the 2-dimensional complete Euclidean graph; by sparse, we mean the number of edges is linear in the number of vertices. These graphs include Delaunay triangulations [3, 6, 10] and the fixed angle theta graph of Keil [9]. Related results are found in Levcopolous and Lingas [11] and Das and Joseph [5]. Althofer, Das, Dobkin, and Joseph [2] prove that there exists a linear-sized graph which is a t-spanner for any d-dimensional complete Euclidean graph, but they do not give an efficient construction. In this paper, we construct a $1 + \frac{1}{2^{m-2}-1}$-spanner, $m \ge 3$, for complete d-dimensional Euclidean graphs, where distance is measure in the L_p metric. Construction of the spanner takes $O((cd)^d n \log n + (cd)^d n\, 2^{md})$ time, and the spanner contains $O((cd)^d d\, n\, 2^{md})$ edges.

We use this spanner to construct approximate minimum spanning trees to obtain a result similar to Vaidya [13]. Given a set of n points and an L_p metric, Vaidya constructs a spanning tree with length at most $1+\varepsilon$ times the length of an actual minimum spanning tree in $O(\frac{n}{\varepsilon^d} \log n)$ time. We can do a bit better, obtaining a spanning tree

[1]Supported in part by DIMACS.

with the same upper bound on length but in time $O(n \log n + \frac{n}{\varepsilon^d} \log \beta(\frac{n}{\varepsilon^d}, n))$, where $\beta(m,n) = \min \{ i : \log^{(i)} n \le m/n \}$.

An outline of the paper is the following. In Section 2, we show how to construct a t-spanner for complete Euclidean graphs. Section 3 presents the result on approximate minimum spanning trees, and Section 4 summarizes the results and states some open problems.

2. Spanner Construction

Throughout this section, let $G = (V,E)$ be a complete Euclidean graph, where intervertex distance is measured in an L_p metric. Let u and v be vertices of G; we will abbreviate the distance from u to v in G by $d(u,v)$.

Lemma 1: Let $t > 1$ be a real number. Let $G' = (V,E')$ be a connected subgraph of G such that for all $u,v \in V$, there is a pair $x,y \in V$ satisfying

1. $(x,y) \in E'$,
2. $d(u,x) + d(y,v) \le \gamma d(x,y)$, $0 < \gamma < \frac{1}{2}$, and
3. $\frac{t\gamma + 1}{1 - \gamma} \le t$.

Then G' is a t-spanner for G.

Proof: We prove the lemma by induction on the rank of the interdistance $d(u,v)$. As a basis, if u and v are a closest pair, then $x = u$ and $y = v$, and $d_{G'}(u,v) = d(u,v)$. To prove this, note that if (2) is true,

$$d(u,x) \le \gamma d(x,y).$$

By the triangle inequality,

$$d(u,x) + d(u,v) + d(y,v) \ge d(x,y).$$

These two imply that

$$\gamma d(x,y) + d(u,v) + d(y,v) \ge d(x,y)$$

$$d(u,v) + d(y,v) \ge (1-\gamma) d(x,y).$$

Again applying (2),

$$\gamma d(x,y) \ge d(u,v) + d(y,v) \ge (1-\gamma) d(x,y).$$

This implies that

$$\gamma \ge 1 - \gamma$$

and

$$\gamma \geq \frac{1}{2}.$$

This contradicts the assumption $0 < \gamma < \frac{1}{2}$.

Now suppose $d(u,v)$ has rank greater than one. We first claim that $d(u,x) < d(u,v)$, so the rank of $d(u,x)$ is less than the rank of $d(u,v)$. By the triangle inequality,

$$d(u,v) + d(u,x) + d(v,y) \geq d(x,y)$$

and (2) implies that

$$d(u,v) + d(u,x) + d(v,y) \geq \frac{1}{\gamma}\,(d(u,x) + d(v,y))$$

so

$$d(u,v) \geq (\frac{1}{\gamma} - 1)(d(u,x) + d(v,y)).$$

Since $0 < \gamma < \frac{1}{2}$, $\frac{1}{\gamma} > 2$, and $d(u,v) > d(u,x)$.

By the inductive hypothesis, $d_{G'}(u,x) \leq t \cdot d(u,x)$. Similarly, $d_{G'}(v,y) \leq t \cdot d(v,y)$, and we have

$$\begin{aligned} d_{G'}(u,v) &\leq d_{G'}(u,x) + d_{G'}(v,y) + d(x,y) \\ &\leq t\,(d(u,x) + d(v,y)) + d(x,y) \\ &\leq (t\,\gamma + 1)d(x,y) \end{aligned}$$

Again using the triangle inequality,

$$\begin{aligned} d(x,y) &\leq d(u,x) + d(v,y) + d(u,v) \\ &\leq \gamma\, d(x,y) + d(u,v) \end{aligned}$$

so

$$d(x,y) \leq \frac{1}{1-\gamma}\, d(u,v).$$

This implies

$$d_{G'}(u,v) \leq \frac{t\,\gamma + 1}{1-\gamma} d(u,v).$$

Finally, condition (3) implies that G' is a t-spanner. □

Construction of the t-spanner is based on Vaidya's algorithm for the all-nearest-neighbor problem [. vaidya all-nearest-neighbor .]. Roughly speaking, Vaidya's algorithm is a process which, over a series of $O(n)$ stages, recursively divides the input set P into a "tree of boxes." That is, a d-dimensional hypercube (a box, say b_0) with smallest possible side length is placed around point set P. In the first stage, this box b_0 is divided into 2^d smaller boxes $\overline{b}_0^1, \ldots, \overline{b}_0^{2^d}$ by d orthogonal hyperplanes passing through the center of b_0. These boxes $\overline{b}_0^1, \ldots, \overline{b}_0^{2^d}$ make up the set *immediate-successors*(b_0). Discarding the empty immediate successors and shrinking the remaining boxes as small as possible, one arrives at a second set of boxes, $b_0^1, \ldots, b_0^j$, making up the set *successors*(b_0). In the tree of boxes, *successors*(b_0) are the children of b_0.

For each box b, Vaidya associates three additional quantities, a real number *estimate*(b), a set of boxes *neighbors*(b), and another set of boxes *attractors*(b). Given boxes b and b', define

$$d_{\max}(b) = \max \{d(u,v) : u,v \in b\}$$

$$d_{\min}(b,b') = \min \{d(u,v) : u \in b, v \in b'\}$$

$$d_{\max}(b,b') = \max \{d(u,v) : u \in b, v \in b'\}$$

The number *estimate*(b) is defined as

$$estimate(b) = \begin{cases} d_{\max}(b), |b \cap P| \geq 2 \\ \min_{b' \in neighbors(b)} \{d_{\max}(b,b')\}, |b \cap P| = 1 \end{cases}$$

Vaidya's algorithm subdivides the box with the largest value of $d_{\max}(b)$. During a particular stage, the set *neighbors*(b) consists of currently existing boxes within *estimate*(b) of b, and the set *attractors*(b) is defined as

$$attractors(b) = \{b' \mid b \in neighbors(b')\}.$$

We refer the reader to Vaidya [14] for details about the algorithm.

Vaidya proves that the maximum size of *neighbors*(b) and the maximum size of *attractors*(b) are constants for a fixed dimension. The actual boxes contained in these sets vary during a box-subdivision step. Since each box-subdivision step affects only a constant number of boxes and there are $O(n)$ box-subdivision steps, the total size of all the distinct *neighbor* and *attractor* sets is $O(n)$. With a minor change to Vaidya's algorithm, the following restriction can be enforced without affecting Vaidya's results.

Invariant: When b' is removed from *neighbors*(b), $d_{\min}(b,b') \geq d_{\max}(b)$.

We are now ready to present the algorithm to construct t-spanners. Let $p(b)$ be the parent of box b, $p^2(b) = p(p(b))$ be the grandparent, and so on. Let $c_1(b)$ be the leftmost child of b, $c_2(b)$ be the second child from the left, $c_1^2(b)$ be the leftmost grandchild, and so on. For any box existing after the final stage, let *Fneighbors*(b) be the ultimate neighbor set of b; by the correctness of Vaidya's algorithm, this is the nearest neighbor to the single point in b under the L_p metric.

Algorithm Construct-spanner (G, t)

1. $m = \lfloor lg \frac{t}{t-1} \rfloor + 2$

2. Run Vaidya's algorithm in the appropriate metric. For each box b, store the set *deleted* (b) of boxes ever deleted from *neighbors*(b) during step 4 of Modify-set-estimates. These appear as pairs (b,b') and are placed in the *deleted pairs* list. Also store *representative*(b), an arbitrary point in $b \cap V$.

3. For each box b
   if $|b \cap V| \geq 2$ then
      For each $b' \in$ *deleted*(b) do
         For each choice of $\bar{b} = c_j^m(b)$ or leaf $\bar{b} = c_j^i(b)$, $0 \leq i \leq m-1$
         and for each choice of $\bar{b}' = c_k^m(b')$ or leaf $\bar{b}' = c_k^i(b')$, $0 \leq i \leq m-1$
            Place (*representative*$(\bar{b})$,*representative*$(\bar{b}')$) into E'.

   if $|b \cap V| = 1$ then
      For each box $b' \in$ *Fneighbors*$(b) \cup$ *deleted*(b) do
         For each choice of $\bar{b}' = c_k^m(b')$ or leaf $\bar{b}' = c_k^i(b')$, $0 \leq i \leq m-1$
            Place (*representative*(b),*representative*$(\bar{b}')$) into E'.

First we prove correctness and then argue the time complexity:

Lemma 2: G' is connected.

Proof: By induction on the number of vertices in a component of G. As a basis, any single vertex is trivially connected. Now consider a box b_0 containing a set of n vertices in V. Box b_0 is split into $b_0^1, \ldots, b_0^j$. By induction, the components in b_0^1 through b_0^j are connected. We show that there is always an edge from b_0^s to b_0^t, $s \neq t$. Let $u \in b_0^s$ and $v \in b_0^t$. Consider the family of boxes containing u: these boxes form a path P_u in the tree of boxes; the path P_v can be similarly defined. If $(\bar{b}^t, \bar{b}^s)$ appears on the deleted pairs list, where $\bar{b}^t \in P_u$ and $\bar{b}^s \in P_v$, then step 3 implies there is an edge from some vertex in $\bar{b}^t$ to some vertex in $\bar{b}^s$. This implies that b_0^s and b_0^t are connected by an edge.

If no pair $(\bar{b}^t, \bar{b}^s)$ appears on the deleted pairs list, then for every box $\bar{b}^t \in P_u$, some box on P_v is in every neighbor set *neighbors*$(\bar{b}^t)$. This can only happen if v is a closest vertex to u, so in step 3, u and v are connected by an edge. □

Lemma 3: For all $u,v \in V$, there is a pair $x,y \in V$ such that $(x,y) \in E'$ and $d(u,x) + d(v,y) \leq \gamma d(x,y)$, where $0 < \gamma \leq \frac{1}{2^{m-1}}$, $m \geq 3$.

Proof: Again define the paths P_u and P_v. Note that there is at most one pair of the form (b,b') in the deleted pairs list, where $b \in P_u$ and $b' \in P_v$. To prove this, suppose there are two pairs (b,b') and $(\bar{b},\bar{b}')$. Suppose without loss of generality that $\bar{b}$ is a (not necessarily proper) descendent of b. Then the time b was present during the course of the algorithm preceded the time $\bar{b}$ was present. This implies that $\bar{b}'$ must be a (not necessarily proper) descendent of b'. As b' was deleted from *neighbors*(b) and the construction of Vaidya's algorithm implies that no descendent of b can contain a proper descendent of b' in its neighbors list, $(\bar{b},\bar{b}')$ cannot have been placed in the deleted pairs list.

This observation implies that there are at most two pairs involving boxes in P_u and boxes in P_v, one of the form (b,b') and the other of the form $(\bar{b}',\bar{b})$, $b,\bar{b} \in P_u$, $b',\bar{b}' \in P_v$. Consider the pair which was added to the deleted pairs list in a later stage, say (b,b'). (If they were both added during the same stage, then $b = \bar{b}$ and $b' = \bar{b}'$.). Note that b must be a descendent of $\bar{b}$ and b' must be a descendent of $\bar{b}'$ (not necessarily proper). We call (b,b') a *good pair*.

The important property of a good pair (b,b') is

$$d_{\min}(b,b') \geq \max(d_{\max}(b),d_{\max}(b')) \geq \frac{1}{2}(d_{\max}(b) + d_{\max}(b')).$$

To prove this, recall that the invariant states that when b' is removed from *neighbors*(b),

$$d_{\min}(b,b') \geq d_{\max}(b)$$

and when $\bar{b}$ is removed from *neighbors*$(\bar{b}')$

$$d_{\min}(\bar{b}',\bar{b}) \geq d_{\max}(\bar{b}').$$

Since b' is a descendent of $\bar{b}'$ and b is a descendent of $\bar{b}$,

$$d_{\min}(b,b') \geq d_{\min}(\bar{b}',\bar{b})$$

and

$$d_{\max}(\bar{b}') \geq d_{\max}(b'),$$

implying the property above.

Since box subdivisions are at the physical center, $d_{\max}(c(b)) \leq \frac{1}{2} d_{\max}(b)$. In general, we have

$$d_{\max}(c^m(b)) \leq \frac{1}{2^m} d_{\max}(b).$$

Let x either be *representative*$(c_j^m(b))$, $u \in c_j^m(b)$, or u if no such box $c_j^m(b)$ exists, and let y either be *representative*$(c_k^m(b'))$, $v \in c_k^m(b')$, or v if no such box $c_k^m(b')$ exists. Then

$$d(u,x) \leq \frac{1}{2^m} d_{\max}(b)$$

$$d(v,y) \leq \frac{1}{2^m} d_{\max}(b').$$

Therefore,

$$d(x,y) \geq d_{\min}(b,b') \geq 2^{m-1}(d(u,x) + d(v,y)).$$

It is evident that $\gamma \leq \frac{1}{2^{m-1}}$. □

Theorem 1: Let G be a complete Euclidean graph, where intervertex distance is measured in an L_p metric. The algorithm above constructs G', a t-spanner for G of size $O((cd)^d n 2^{md})$, where $t = 1 + \frac{1}{2^{m-2} - 1}$. This algorithm takes $O((cd)^d n \log n + (cd)^d n 2^{md})$ time.

Proof: Lemma 2 implies that G' is connected. For G' to be a spanner, it is sufficient to prove that the three conditions in Lemma 1 are satisfied. Lemma 3 implies that (1) and (2) are satisfied with $0 < \gamma \le \frac{1}{2^{m-1}}$, $m \ge 3$. For (3), let $t = \frac{1}{1-2\gamma}$. Then

$$\frac{t\gamma + 1}{1-\gamma} = \frac{1}{1-2\gamma}\,\frac{\gamma + 1 - 2\gamma}{1-\gamma} = \frac{1}{1-2\gamma} = t.$$

By Lemma 3, $\gamma \le \frac{1}{2^{m-1}}$, so $t < 1 + \frac{1}{2^{m-2}-1}$. Vaidya's algorithm takes $O((cd)^d n \log n)$ time, and the entire size of the neighbor sets is $O((cd)^d n)$ [14]. For each deleted neighbor, at most 2^{md} work is done, since the maximum degree of the tree is 2^d. □

3. Approximate Minimum Spanning Trees

A t-spanner can be used in the construction of approximate minimum spanning trees for complete d-dimensional L_p distance graphs. For $d = 2$, Shamos and Hoey [12] gave an $O(n \log n)$ time algorithm for Euclidean minimum spanning trees in the L_2 metric. Yao [15] obtained $o(N^2)$ time algorithms for the metrics L_1, L_2, and L_∞ in $d \ge 3$. in the L_1 and L_∞ metrics, Gabow, Bentley, and Tarjan [7] presented $O(n \log^r n \log\log n)$ time algorithms, where for L_∞, $r = d-2$ when $d \ge 3$, and for L_1, $r = 1, 2, 4$ when $d = 3, 4, 5$ and $r = d$ when $d > 5$. Agarwal, Edelsbrunner, Schwarzkopf, and Welzl indicate how to reduce the time complexity to $O(n \log^d n)$ [1]. In addition, Agarwal, Edelsbrunner, Schwarzkopf, and Welzl gave a randomized $O(n^{4/3} \log^{4/3} n)$ expected time algorithm for Euclidean minimum spanning trees in 3 dimensions.

Regarding approximate minimum spanning tree algorithms, important papers are by Clarkson [4] and Vaidya [13]. The most general result is Vaidya's, where in $O(\varepsilon^{-d} n \log n)$ time he extract a graph of $O(\varepsilon^{-d} n)$ edges that is guaranteed to contain a spanning tree of length at most $(1 + \varepsilon)$ times the length of a minimum spanning tree. In comparison, our algorithm extracts a $(1 + \varepsilon)$-spanner of size $O(\frac{n}{\varepsilon^d})$ in $O(n \log n + \frac{n}{\varepsilon^d})$ time. Theorem 3 below relates the length (the sum of the edge weights) of a minimum spanning tree of G' to the length of a minimum spanning tree for a complete Euclidean graph G.

Let $MST(G)$ be the length of a minimum spanning tree for G, and let $length(G)$ be the length of a graph G.

Theorem 2: Let G' be a t-spanner for G. Then $MST(G') \le t \cdot MST(G)$.

Proof: Let $T = (V, E'')$ be a minimum spanning tree for G. We argue that there is a connected subgraph $\overline{G}$ of G' for which $length(\overline{G}) \le t \cdot MST(G)$. For each edge $e = (u,v) \in E''$, there is a path p from u to v in G' of length at most $t \cdot d(u,v)$. Include all the edges in p in $\overline{G}$. Note that $\overline{G}$ is connected since T is connected, it is a subgraph of G', and $length(\overline{G}) \le t \cdot MST(G)$. Therefore

$$MST(G') \leq length(\overline{G}) \leq t \cdot MST(G).$$

□.

After constructing G', we can run the minimum spanning tree algorithm of Gabow, Galil, Spencer, and Tarjan [8] to get an approximate minimum spanning tree. Their algorithm takes $O(m \log \beta(m,n))$, where m is the number of edges, n is the number of vertices, and $\beta(m,n) = \min\{ i : \log^{(i)} n \leq m/n \}$. This approximation is within $(1+\varepsilon)$ times the length of an actual minimum spanning tree.

Theorem 3: Let G be a complete Euclidean graph, where intervertex distance is measured in an L_p metric. For any $\varepsilon > 0$, there is a tree with length at most $(1+\varepsilon) MST(G)$ which can be built in $O(n \log n + \frac{n}{\varepsilon^d} \log \beta(\frac{n}{\varepsilon^d}, n))$ time.

4. Final Remarks and Open Problems

We present a method to construct sparse spanners for complete Euclidean graphs in any dimension. The spanning factor of this graph can be made arbitrarily small at the cost of larger size and more expensive construction. It would be interesting to find a sparse t-spanner for some constant t where each vertex has constant degree. Such a spanner could be used in an algorithm to select the n^{th} smallest interdistance determined by n points in $\Re^d$.

5. References

1. P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf and E. Welzl, Euclidean Minimum Spanning Trees and Bichromatic Closest Pairs, *Sixth Symposium on Computational Geometry*, 1990, pp. 203-210.
2. I. Althofer, G. Das, D. Dobkin and D. Joseph, Generating Sparse Spanners for Weighted Graphs, *Second Scandinavian Workshop on Algorithm Theory*, 1990, pp. 26-37.
3. P. Chew, There is a Planar Graph Almost as Good as the Complete Graph, *Second ACM Symposium on Computational Geometry*, 1986, pp. 169-177.
4. K. Clarkson, Fast Expected-Time and Approximate Algorithms for Geometric Minimum Spanning Trees, *Sixteenth ACM Symp. Theory Comput.*, 1984, pp. 342-348.
5. G. Das and D. Joseph, Which Triangulations Approximate the Complete Graph?, *Symposium on Optimal Algorithms*, 1989, pp. 168-192.
6. D. P. Dobkin, S. J. Friedman and K. J. Supowit, Delaunay Graphs are Almost as Good as Complete Graphs, *Disc. Comput. Geom.*, **5**, 1990, pp. 399-407.
7. H. N. Gabow, J. L. Bentley and R. E. Tarjan, Scaling and Related Techniques for Geometry Problems, *Sixteenth ACM Symp. Theory Comput.*, 1984, pp. 135-143.

8. H. N. Gabow, Z. Galil, T. H. Spencer and R. E. Tarjan, Efficient Algorithms for Minimum Spanning Trees on Directed and Undirected Graphs, *Combinatorica*, **6**, 1986, pp. 109-122.

9. J. M. Keil, Approximating the Complete Eulcidean Graph, *First Scandinavian Workshop on Algorithm Theory*, 1988, pp. 208-213.

10. J. M. Keil and C. A. Gutwin, The Delaunay Triangulation Closely Approximates the Complete Euclidean Graph, *First Workshop on Algorithms and Data Structures*, 1989, pp. 47-56.

11. C. Levcopoulos and A. Lingas, There are Planar Graphs Almost as Good as the Complete Graphs and as Short as Minimum Spanning Trees, *Symposium on Optimal Algorithms*, 1989, pp. 9-13.

12. M. I. Shamos and D. Hoey, Closest-Point Problems, *Sixteenth IEEE Symposium Found. Comput. Sci.*, 1975, pp. 151-162.

13. P. M. Vaidya, Minimum Spanning Trees in k-Dimensional Space, *Siam J. on Computing*, **17**(3), 1988, pp. 572-582.

14. P. M. Vaidya, An $O(n \log n)$ Algorithm for the All-Nearest-Neighbors Problem, *Discrete Comput. Geom.*, **4**, 1989, pp. 101-115.

15. A. C. Yao, On Constructing Minimum Spanning Trees in k-Dimensional Spaces and Related Problems, *Siam J. on Computing*, **11**, 1982, pp. 721-736.