Experimental Study of Geometric t-Spanners

Mohammad Farshi[1,⋆] and Joachim Gudmundsson[2]

[1] Department of Mathematics and Computing Science, TU Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
m.farshi@tue.nl
[2] NICTA[⋆⋆], Sydney, Australia
joachim.gudmundsson@nicta.com.au

Abstract. The construction of t-spanners of a given point set has received a lot of attention, especially from a theoretical perspective. In this paper we perform the first extensive experimental study of the properties of t-spanners. The main aim is to examine the quality of the produced spanners in the plane. We implemented the most common t-spanner algorithms and tested them on a number of different point sets. The experiments are discussed and compared to the theoretical results and in several cases we suggest modifications that are implemented and evaluated. The quality measurements that we consider are the number of edges, the weight, the maximum degree, the diameter and the number of crossings.

1 Introduction

Consider a set V of n points in the plane. A network on V can be modeled as an undirected graph G with vertex set V of size n and an edge set E of size m where every edge $e = (u, v)$ has a weight $wt(e)$. A geometric (Euclidean) network is a network where the weight of the edge $e = (u, v)$ is the Euclidean distance $d(u, v)$ between its endpoints u and v. Let $t > 1$ be a real number. We say that G is a *t-spanner* for V, if for each pair of points $u, v \in V$, there exists a path in G of weight at most t times the Euclidean distance between u and v. We call this path a t-path between u and v. The minimum t such that G is a t-spanner for V is called the stretch factor, or dilation, of G. Finally, a subgraph G' of a given graph G is a t-spanner for G if for each pair of points $u, v \in V$, there exists a path in G' of weight at most t times the weight of the shortest path between u and v in G.

Complete graphs represent ideal communication networks, but they are expensive to build; sparse spanners represent low-cost alternatives. The weight of the spanner is a measure of its sparseness; other sparseness measures include the number of edges, the maximum degree, and the number of crossings. Spanners

⋆ Supported by Ministry of Science, Research and Technology of I. R. Iran.
⋆⋆ National ICT Australia is funded through the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

G.S. Brodal and S. Leonardi (Eds.): ESA 2005, LNCS 3669, pp. 556–567, 2005.

for complete Euclidean graphs as well as for arbitrary weighted graphs find applications in robotics, network topology design, distributed systems, design of parallel machines, and many other areas and have been a subject of considerable research. Recently low-weight spanners found interesting practical applications in areas such as metric space searching [15,16] and broadcasting in communication networks [8,13]. Several well-known theoretical results also use the construction of t-spanners as a building block, for example, Rao and Smith [18] made a breakthrough by showing an optimal $O(n \log n)$-time approximation scheme for the well-known Euclidean *traveling salesperson problem*, using t-spanners (or banyans). Similarly, Czumaj and Lingas [6] showed approximation schemes for minimum-cost multi-connectivity problems in geometric networks. The problem of constructing spanners has received considerable attention from a theoretical perspective, see the survey [7], but almost no attention from a practical, or experimental perspective [15,20].

In this paper we consider the most well-known algorithms for the construction of t-spanners in the plane: greedy spanners, Θ-graphs, ordered Θ-graphs and spanners constructed from the well-separated pair decomposition (WSPD). The quality measurements used in the literature is the number of edges, the weight, the maximum degree, the diameter and the number of crossings. We study each of the algorithms independently, but also in combination with each other. To the best of the authors knowledge, this is the first time an extensive experimental study has been performed on the construction of t-spanners. Navarro and Paredes [15] presented four heuristics for point sets in high-dimensional metric space ($d = 20$) and showed by empirical methods that the running time was $\mathcal{O}(n^{2.24})$ and the number of edges in the produced graphs was $\mathcal{O}(n^{1.13})$. In [20] Sigurd and Zachariasen considered the problem of constructing a minimum weight t-spanner of a given graph, but they only considered sparse graphs of small size, i.e., graphs with at most 64 vertices and with average vertex degree 4 or 8.

The paper is organized as follows. Next we briefly go through the different properties that we considered in the experiments, and present a first algorithm that computes the t-diameter of a t-spanner. In Section 3 we give a short description of each of the algorithms that we implemented, together with their theoretical bounds. Then, in Sections 4–5 we discuss the implementations, the data we used and the experimental results, followed by Section 6 where we discuss the results and propose improvements.

2 Spanner Properties

As input we are given a set V of n points in the plane and a positive real value $t > 1$. The aim is to compute a t-spanner for V with some good properties where the quality measurements that we will consider in this paper are as follows:

Size: Defined to be the number of edges in the graph. This is the most important property of the constructed networks and all the implemented algorithms produce spanners with only $\mathcal{O}(n)$ edges. This key feature has made the construction of spanners one of the fundamental tools in the development of fast approximation algorithms for geometrical problems.

Degree: The maximum number of edges incident to a vertex. This property has been shown to be useful in the development of approximation algorithms [10] and for the construction of ad hoc networks where small degree is essential in trying to develop fast localized algorithms [13].

Weight: The weight of a Euclidean network is the sum of the edge weights. The best that can be achieved is a constant times the weight of the minimum spanning tree of input point set, denoted $wt(MST)$.

Diameter: Defined as the smallest integer d such that for any pair of vertices u and v in V, there is a t-path in the graph between u and v containing at most d edges, i.e., a path of weight at most $t \cdot |uv|$. Note that there is a trade-off between the degree and the diameter and between the diameter and the size [2]. For wireless ad hoc networks it is often desirable to have small diameter since it determines the maximum number of times a message has to be transmitted in a network.

Crossings: A pair of edges with a non-empty intersection is said to *cross*. The total number of pairs of edges in a graph that cross is the number of crossings. One wants to minimize the number of crossings since it decreases the complexity and the readability of a graph [17].

2.1 Diameter

As mentioned before, the diameter of a t-spanner is the smallest integer k such that there is a t-path between each pair of points that contains at most k edges. As far as we know there is no known algorithm on how to compute the diameter of a t-spanner, below we present a dynamic programming approach for the problem which we will use in Section 4 to calculate the diameter of the constructed t-spanners.

Assume that $L[p, q, k]$ is the shortest path length between p and q with at most k edges. If there is no such path, we set $L[p, q, k]$ to ∞. With this definition, the diameter of a t-spanner is the smallest integer k such that $L[p, q, k] \leq t \cdot |pq|$, for all points p and q.

Lemma 1. *Let $G = (V, E)$ be a graph and p and q are two vertices of G. For each integer $k \geq 2$,*

$$L[p, q, k] = \min \left\{ L[p, q, k-1], \min_{r \in V \setminus \{p,q\}} \{L[p, r, k-1] + L[r, q, 1]\} \right\}.$$

Note that, based on Lemma 1, $L[-, -, 1]$ and $L[-, -, k-1]$ is sufficient for computing $L[-, -, k]$.

Now for computing the diameter, we start with $k = 1$. Obviously $L[p, q, 1] = |pq|$ for all edges (p, q) in the graph and $L[p, q, 1] = \infty$ otherwise. For $k \geq 2$, we can compute $L[-, -, k]$ using $L[-, -, k-1]$ and $L[-, -, 1]$. We continue construction until we received a k such that $L[p,q,k] \leq t \cdot |pq|$, for all points p and q.

Corollary 1. *The diameter of a t-spanner G can be computed in $\mathcal{O}(dn^3)$ time using $\mathcal{O}(n^2)$ space, where d is the diameter of G.*

3 The Spanner Construction Algorithms

In this section we give a short description of each of the algorithms implemented together with their theoretical bounds.

3.1 The Greedy Algorithm

The greedy algorithm was discovered independently by Bern in 1989 and Althöfer et al. [1]. The graph constructed using the greedy algorithm will be called a greedy graph.

The greedy algorithm maintains a partial t-spanner G' while processing all point pairs in order of increasing length. Processing a pair u, v entails a shortest-path query in G' between u and v. If there is a t-path between u and v in G' then the edge (u, v) is discarded, otherwise it is added to G'.

The time complexity of the greedy algorithm is $\mathcal{O}(n^3 \log n)$ and it uses $\mathcal{O}(n^2)$ space. There exists an $\mathcal{O}(n \log n)$ time greedy algorithm [9] that only uses linear space but, unfortunately, it is very complicated and therefore we decided to implement the simple version. The following theorem states the theoretical upper bounds.

Theorem 1. *The greedy graph is a t-spanner of V with $\mathcal{O}(n)$ edges, maximum degree $\mathcal{O}(1)$ and weight $\mathcal{O}(wt(MST(V)))$, and can be computed in time $\mathcal{O}(n \log n)$.*

Note also that a trivial $\Omega(n)$ lower bound on the diameter of a greedy graph is obtained by placing n points on a line.

The greedy approach can also be used to prune a given t-spanner $G = (V, E)$, that is, instead of considering all point pairs, the algorithm only considers the endpoints of the edges in E. In this paper we also perform experiments using the pruning tool in combination with the other algorithms, see Section 5. The time complexity of the implemented greedy pruning is $\mathcal{O}(mn \log(n)$, where m is the number of edges in the input graph.

Improvement. As mentioned above the running time of the implemented algorithm is $\mathcal{O}(n^3 \log n)$, which is too slow when performing experiments with up to 13.000 points. We use a speed-up strategy that turned out to decrease the running time considerably in practice. Originally the algorithm computes the shortest path for each pair of points to check if there is a t-path between the two points or not. But there are two simple observations. First, we only need to know "Is there a t-path between the points?" and second, the algorithm only adds $\mathcal{O}(n)$ edge to the graph in total, so the graph does not change very often during the execution.

Therefore, we use a matrix to save the shortest path between each two points and update it only when we need to, thus it is not always up to date. Instead of computing a shortest path for each pair, the main change is that we only check if there is a t-path or not.

With these changes, the space complexity of the greedy algorithm increases, but the gain in running time is considerable. We counted the number of shortest path queries that this algorithm performs in the experiments and surprisingly it seems that $\mathcal{O}(n)$ shortest path queries is sufficient.

Conjecture 1. The modified greedy algorithm performs $\mathcal{O}(n)$ shortest path queries.

3.2 The Θ-Graph Construction

The Θ-graph was discovered independently by Clarkson [5] and Keil [11]. Keil only considered the graph in two dimensions while Clarkson extended his construction to also include three dimensions. Later Rupert and Seidel [19] and Althöfer et al. [1] defined the Θ-graph for higher dimensions.

The Θ-graph algorithm processes each point u as follows. Consider k non-overlapping cones with apex u and with angle $\theta = \frac{2\pi}{k}$ (k is related to t). For each non-empty cone C add an edge between u and v to the graph, where v is the point within C whose orthogonal projection onto the bisector of C is closest to u.

Theorem 2. *The Θ-graphs is a t-spanner of V for $t = \frac{1}{\cos\theta - \sin\theta}$ with $\mathcal{O}(kn)$ edges and can be computed in $\mathcal{O}(kn\log n)$ time.*

Note the even though the “out-degree” of each vertex is bounded by k the “in-degree” could be linear and the weight of the Θ-graph can be $\Omega(n\cdot wt(MST(V)))$. Finally, by placing n points on a line it follows that the diameter of the Θ-graph is $\Omega(n)$.

3.3 Ordered Θ-Graph

A simple variant of the Θ-graph that has been shown to have good theoretical performance is the *Ordered Θ-graph* by Bose et al. [3]. An ordered Θ-graph of V is obtained by inserting the points of V in some order. When a point p is inserted, we draw the cones as in the Θ-graph algorithm around p and connect p to its closest previously-inserted point in each cone.

Theorem 3. *The Ordered Θ-graphs is a t-spanner of V for $t = \frac{1}{\cos\theta - \sin\theta}$ with $\mathcal{O}(kn)$ edges and $\mathcal{O}(k\log n)$ degree, and can be computed in $\mathcal{O}(kn\log n)$ time.*

Remark 1. If the points are processed in random order then the diameter will be bounded by $\mathcal{O}(\log n)$ with high probability [3], but then the degree bound does not hold anymore.

3.4 The WSPD-Graph

The well-separated pair decomposition (WSPD) was developed by Callahan and Kosaraju [4]. Callahan and Kosaraju show that a WSPD of size $m = \mathcal{O}(s^d n)$ can be computed in $\mathcal{O}(s^d n \log n)$ time.

Constructing a t-spanner using the WSPD is surprisingly easy. It is sufficient to compute a WSPD of V w.r.t. $s = \frac{4(t+1)}{t-1}$ and then add an each between each well-separated pair in the WSPD.

Theorem 4. *The WSPD-graph is a t-spanner for $V \subset \mathbb{R}^2$ with $\mathcal{O}((\frac{t}{t-1})^2 \cdot n)$ edges, and can be constructed in time $\mathcal{O}((\frac{t}{t-1})^2 n \log n)$.*

An $\Omega(n)$ lower bound on the degree and the diameter of a WSPD-graph can be shown by placing n points on a line with exponentially decreasing inter point distance from left to right. However we have not been able to find any non-trivial lower or upper bound on the weight of a WSPD-graph.

4 Experimental Results

In this section we discuss the results in more detail by considering each of the five properties. The experiments were done on point sets ranging from 100 to 13.000 points with five different distributions:

- uniform distribution,
- normal distribution with mean 500 and deviation 100,
- gamma distribution with shape parameter 0.75,
- $\frac{n}{100}$ uniformly distributed unit squares with 100 uniformly distributed points, and
- $\sqrt{n}$ uniformly distributed unit squares with $\sqrt{n}$ uniformly distributed points.

In the discussion that follows we will call the point sets produced with the two latter distributions the clustered sets, and the other point sets will be called the non-clustered sets. To avoid the effect of specific instances, we ran the algorithms on many different instances and took the average of the results.

We produced t-spanners using values of t between 1.05 and 2. For larger values of t one can use the Delaunay triangulation which is known to have dilation ≈ 2.42 [12]. The algorithms were implemented in C++ using the LEDA 5.0 library [14].

4.1 Size

The number of edges in the produced graphs were all linear with respect to the number of points and, as expected, the greedy graph had the smallest number of edges. For $t = 2$, $t = 1.1$ and $t = 1.05$ the number of edges in the greedy graph is approximately $2n$, $4n$ and $6n$ respectively, which is surprisingly small. For comparison it is interesting to note that the Delaunay triangulation has approximately $3n$ edges and dilation bounded by 2.42 [12]. A short summary of the results is that the size of the WSPD-graph is roughly a factor 7 to 13 times greater than the size of the (ordered) Θ-graph which in turn is roughly a factor 5 to 10 times greater than the size of the greedy graph. For the uniform distribution and for $t = 2$ the results can be seen in Fig. 1a. The WSPD algorithm was expected to perform slightly better for clustered sets since it uses a clustering approach, but the improvement was greater than predicted. On clustered sets the WSPD algorithm produced graphs where the number of edges is comparable, or even smaller, than the number of edges in the (ordered) Θ-graph. Especially for small values of t the algorithm performed better.

4.2 Degree

As above, the greedy algorithm again outperformed the other approaches. It is the only algorithm that has a theoretical constant upper bound on its degree, see Theorem 1, and the experiments supports the theory. In the tests the greedy algorithm produced graphs with degree about 5, 7 and 23 for $t = 2$, $t = 1.5$ and $t = 1.05$ respectively and the bounds are roughly the same for all the test sets.

For non-clustered sets the degree of the (ordered) Θ-graphs increases very slowly with respect to the number of points, for example for the uniform distribution the ordered Θ-graph has a degree of 24 for 100 points and the degree then slowly increases to 31 for 10.000 points. The ordered Θ-graph generally performs slightly better than the Θ-graph. However, for clustered sets the results change unexpectedly. The degree of the Θ-graph deteriorate rapidly and the degree varies highly between different instances. The ordered Θ-graph on the other hand performs slightly better than for the other distributions, as shown in Fig. 1b. Again it seems that the experiments supports the theory stated in Theorem 3 since the ordered Θ-graph has $\mathcal{O}(k \log n)$ degree.

As expected the WSPD-algorithm generated the graphs with the highest degree. For small values of t it almost shows a linear behavior for sets with up to 13.000 points. Although for larger values of t it seems to converge slowly, but to be able to draw any distinct conclusions more experiments has to be performed on much larger point sets. But as observed in the previous section, the WSPD-algorithm performs much better on clustered sets and seems to converge to a constant for large point sets. For example for $t = 1.05$ and $t = 1.1$ the degree is bounded by 350 and 290 respectively, and it does not seem to increase.

Finally, we tried to improve the WSPD-graph by considering a modification of the WSPD-algorithm. Instead of adding an arbitrary edge between a well-separated pair, we add an edge between the two endpoints in the pairs with smallest degree. This does not improve the theoretical upper bound but we were hoping to see some improvements in the experimental bounds. There is a small improvement, unfortunately this improvement only shows up for graphs with $t > 1.5$, for smaller values the difference is negligible. For $t = 1.5$ the improvement is roughly a factor 1.5 and increases to about 3 for $t = 4$. Note also that the $\Omega(n)$ lower bound stated in Section 3 does not hold for the modified WSPD-algorithm.

4.3 Weight

Recall that theoretically the weight of the greedy graph is $\mathcal{O}(wt(MST))$ while the weight of the (ordered) Θ-graph and the WSPD-graph is only bounded by $\mathcal{O}(n \cdot wt(MST))$ so the fact that the weight of the greedy graphs is much less than the weight of the other graphs is hardly surprising. For $t = 2$ the weight of the greedy graph is approximately 2 times the $wt(MST)$ and for $t = 1.1$ and $t = 1.05$ the factors are 10 and 18 respectively, as can be seen in Fig. 1c. For the clustered sets the bounds are even slightly better.

For the non-clustered sets the weight of the Θ-graph was unexpectedly small and the ratio between its weight and the $wt(MST)$ increased very slowly. For example for $t = 1.1$ it went from 133 for 100 points to 330 for 13.000 points. An interesting question that we have not been able to answer neither through the experiments nor in theory is if the expected weight of the Θ-graph for uniform point sets is bounded by a constant times the $wt(MST)$. For clustered sets the weight of the Θ-graph is almost linear with respect to the number of clusters and its weight is highly dependent on the different instances.

One would expect a similar behavior from the ordered Θ-graph but the weight of the ordered Θ-graph is much higher than both the greedy graph and the Θ-graph. The ratio between the weight of the ordered Θ-graph and the $wt(MST)$ is almost a linear function with respect to the number of points up to 10.000 points before it starts to level out. Moreover the behavior of the weight of the ordered Θ-graph is unpredictable and seems to be highly dependent on the specific instances.

The weight of the WSPD-graph has the same behavior as the degree of the WSPD-graph. For a small stretch factor it shows a linear behavior compared to the $wt(MST)$ and for larger values of t it seems to converge slowly. Just as for the degree, the WSPD-algorithm performs very well on clustered sets and seems to converge to a large constant times the $wt(MST)$ for large sets. For example for $t = 1.05$ the ratio was bounded by 900 and for, $t = 2$ it was bounded by 230.

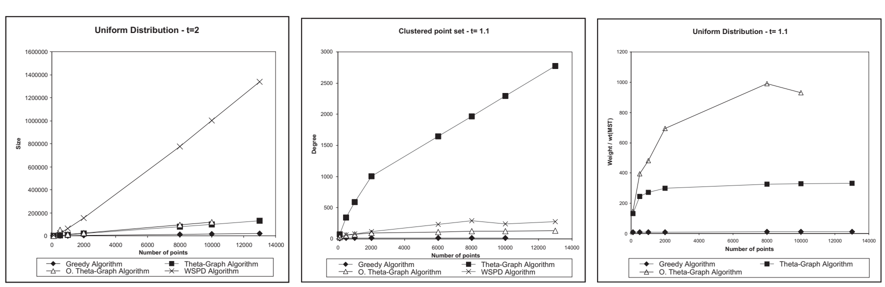

Fig. 1. (a) Size for uniform distribution sets. (b) Degree for clustered sets. (c) Weight/$wt(MST)$ for uniform distribution sets.

4.4 Diameter

Due to the high complexity of computing the spanner diameter of a graph we could only compute the diameter for graphs with up to 2000 points.

Note that all of the algorithms that we implemented may produce graphs whose diameter is $\Theta(n)$. The only known algorithm that produces graphs with smaller diameter is the modified ordered Θ-graphs which was shown to have $\mathcal{O}(\log n)$ diameter with high probability [3].

As expected the greedy graphs had the highest diameter. This follows from the fact that the greedy graph has fewer edges than the other graphs and the greedy approach favors short edges and avoids adding long edges. The diameter of a 2-spanner generated by the greedy algorithm (normal distribution) is about 16 for a set with 100 points and it reaches 58 for a set with 2000 points. The diameter seems to increase linearly with the size of the input set. Also the diameter changes slightly depending on the different distributions, for $t = 1.1$ and $n = 1000$ the diameter varies between 22 for normal distribution to 36 for the clustered distribution.

The diameter of the (ordered) Θ-graph is much smaller than the greedy graph. A 2-spanner generated by the (ordered) Θ-graph has diameter approximately 5 for a set with 100 points and it increases to 15 for a set with 2000 points. The ordered Θ-graph has generally a slightly lower diameter compared to the Θ-graph. The diameter of the (ordered) Θ-graphs is almost the same for all the distributions.

Finally, the WSPD-graph has very low diameter and it converges fast. The diameter of a 2-spanner on a set with 100 points (normal distribution) is 3.4 and it increases to 5 for a point set with 2.000 points. In the clustered sets, the diameter is slightly higher, between 4 and 7, probably because the number of edges in these graphs are smaller compared to the graphs for the non-clustered sets. An oddity is that the modified WSPD algorithm which add an edge between the two points of well separated pair with smallest degree has slightly lower diameter compared with the WSPD-graphs.

4.5 Crossings

The last property we discuss is the number of crossings which obviously is highly dependent on the number of edges, therefore it is not surprising that the greedy graph is superior to the other graphs and it is the only graph with a reasonable number of crossings. Just as for the diameter the experiments could only be done on sets with up to 2.000 points since the number of crossings only is bounded by $\mathcal{O}(m^2)$ where m is the number of edges in the graph. For $t = 1.5$ and $n = 2.000$ the number of crossings in the greedy graph is on average 94. Also the number of crossings seems to increase linearly with respect to the number of points which is surprisingly low. For $t = 1.1$ the number of crossings for $n = 100$, 500 and 2.000 is on average 1.727, 23.851 and 50.210.

We did some initial experiments with the (ordered) Θ-graphs but the number of crossings were so high that we found the results uninteresting, for example for $t = 1.1$ and $n = 100$ the Θ-graph has 2.437 edges and 300K crossings! Thus if the number of crossings is a priority to the user then the only option is to use the greedy algorithm.

5 Hybrid Algorithms

As you can see in Section 4, the greedy algorithm produced graphs whose size, weight, degree and number of crossings are superior to the graphs produced from the other approaches. However the running time of the greedy algorithm is $\mathcal{O}(n^3 \log n)$. A way to improve the running time while, hopefully, still obtaining the high-quality graphs is to first compute a t^α-spanner $(0 < \alpha < 1)$ G of the input set which contains linear number of edges and then compute a $(t^{1-\alpha})$-spanner of G using the greedy pruning algorithm. The dilation of the resulting graph is bounded by $t^\alpha \cdot t^{1-\alpha} = t$. The t^α-spanner can be constructed using (ordered) Θ-graphs or WSPD-graphs, ensuring that the number of edges is $\mathcal{O}(n)$, and consequently the total running time would decrease to $\mathcal{O}(n^2 \log n)$.

A second reason why we consider hybrid algorithms is the fact that the (ordered) Θ-graphs and the WSPD-graph actually have much smaller dilation than the specified t-value. For example for $t = 2$ the greedy graph has dilation close to 2 while the ordered Θ-graph and the WSPD-graph has dilation 1.4 and the Θ-graph has dilation 1.2. For $t = 1.1$ the Θ-graph has dilation 1.02, the ordered Θ-graph has dilation 1.06 and the WSPD-graph has dilation 1.04. By first producing the (ordered) Θ-graph or the WSPD-graph we use the fact that they can be constructed fast and the number of edges remaining is linear. Since their dilation in practice is very small it leaves a lot of freedom for the greedy algorithm to produce a t-spanner with good properties.

This approach has another advantage which is that the parameter α can be adjusted to fit the application. If α is chosen to be close to zero then the resulting graph is very similar to the greedy graph but the gain in running time is small. If α is chosen close to 1 then the algorithm is faster but the quality of the graph is worse.

We performed the same experiments using different algorithmic combinations and compared the properties of the generated graphs with the greedy graphs. We implemented the combination of (ordered) Θ-graph and WSPD-graph plus greedy pruning.

The experiments showed the following interesting observations. The graphs generated by the three hybrid algorithms has more or less the same properties, and even though the number of edges, degree and weight is higher their behavior is very similar to the greedy algorithm. This probably follows from the fact that the actual dilation of the (ordered) Θ-graphs and the WSPD-graph is much smaller in reality than the given parameter t^α, thus as expected the resulting graphs are mainly decided by the greedy pruning. For the uniform distribution with $\alpha = 0.9$ and $t = 1.1$ the number of edges and degree of the graphs generated by the hybrid algorithms are roughly a factor 3 greater than the number of edges and degree of the greedy graph. Also the weight is approximately six times greater and the diameter is less than a half that of the greedy graph. Finally, if the value of α is very small, say 0.1, then the resulting graphs have more or less the same properties as the greedy graphs.

6 Conclusion and Future Work

In short the conclusions from the experiments are as follows:

- The greedy graph has surprisingly good quality when it comes to the number of edges, the weight, the degree and the number of crossings. The diameter of the greedy graph is considerably higher than the diameter of the other graphs.
- The greedy improvement worked out much better than expected and it would be very interesting if one could prove that the improved algorithm has a running time of $\mathcal{O}(n^2 \log n)$ (expected).
- The experiments shows that the weight of the Θ-graph is very small for non-clustered sets. Proving, or disproving, that the expected weight of the Θ-graph for uniform distributions is an interesting and challenging open question.
- The Θ-graph has unexpectedly high degree when the sets are clustered and its degree varies greatly between different instances. Also the degree of the ordered Θ-graph is much smaller, for clustered sets, than the Θ-graph. Surprisingly the same results can not be seen in the weight of the (ordered) Θ-graphs.
- The WSPD-algorithm produces graphs with unexpectedly poor quality for non-clustered sets. For clustered sets the results are much better; the weight and degree of the WSPD-graph are considerably smaller than the weight and degree of the (ordered) Θ-graphs. The weight and the degree of the WSPD-graphs seem to converge for very large data sets. However, to answer this conjecture we need to perform tests on much larger sets.

Future work includes more extensive tests with larger sets to be able to experimentally answer some of the remaining open questions. However the main objective will be to experimentally compare the time complexity of the algorithms.

References

1. I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.
2. S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. 27th ACM Symposium on Theory of Computing*, pages 489–498, 1995.
3. P. Bose, J. Gudmundsson, and P. Morin. Ordered theta graphs. *Computational Geometry: Theory and Applications*, 28:11–18, 2004.
4. 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.

5. K. L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th ACM Symposium on Computational Geometry*, pages 56–65, 1987.
6. A. Czumaj and A. Lingas. Fast approximation schemes for Euclidean multi-connectivity problems. In *Proc. 27th International Colloquium on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 856–868. Springer-Verlag, 2000.
7. D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier Science Publishers, Amsterdam, 2000.
8. A. M. Farley, A. Proskurowski, D. Zappala, and K. J. Windisch. Spanners and message distribution in networks. *Discrete Applied Mathematics*, 137(2):159–171, 2004.
9. J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Improved greedy algorithms for constructing sparse geometric spanners. *SIAM Journal of Computing*, 31(5):1479–1500, 2002.
10. J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles for geometric graph. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms*, 2002.
11. J. M. Keil. Approximating the complete Euclidean graph. In *Proc. 1st Scandinavian Workshop on Algorithmic Theory*, pages 208–213, 1988.
12. J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete and Computational Geometry*, 7:13–28, 1992.
13. X.-Y. Li. Applications of computational geomety in wireless ad hoc networks. In X.-Z. Cheng, X. Huang, and D.-Z. Du, editors, *Ad Hoc Wireless Networking*. Kluwer, 2003.
14. K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, UK, 2000.
15. G. Navarro and R. Paredes. Practical construction of metric t-spanners. In *Proc. 5th Workshop on Algorithm Engineering and Experiments*, pages 69–81. SIAM Press, 2003.
16. G. Navarro, R. Paredes, and E. Chvez. t-spanners as a data structure for metric space searching. In *Proc. 9th International Symposium on String Processing and Information Retrieval*, volume 2476 of *Lecture Notes in Computer Science*, pages 298–309. Springer-Verlag, 2002.
17. Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In *Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 1997.
18. S. Rao and W. D. Smith. Approximating geometrical graphs via spanners and banyans. In *Proc. 30th ACM Symposium on the Theory of Computing*, pages 540–550. ACM, 1998.
19. J. Ruppert and R. Seidel. Approximating the d-dimensional complete Euclidean graph. In *Proc. 3rd Canadian Conference on Computational Geometry*, pages 207–210, 1991.
20. M. Sigurd and M. Zachariasen. Construction of minimum-weight spanners. In *Proc. 12th European Symposium on Algorithms*, number 3221 in Lecture Notes in Computer Science. Springer-Verlag, 2004.