

Constructing Degree-3 Spanners with Other Sparseness Properties

Gautam Das*
Paul J. Heffernan

Department of Mathematical Sciences, Memphis State University
Memphis, TN 38152

Abstract

Let V be any set of n points in k -dimensional Euclidean space. A subgraph of the complete Euclidean graph is a t -spanner if for any u and v in V , the length of the shortest path from u to v in the spanner is at most t times $d(u, v)$. We show that for any $\delta > 1$, there exists a polynomial-time constructible t -spanner (where t is a constant that depends only on δ and k) with the following properties. Its maximum degree is 3, it has at most $n \cdot \delta$ edges, and its total edge weight is comparable to the minimum spanning tree of V (for $k \leq 3$ its weight is $O(1) \cdot wt(MST)$, and for $k > 3$ its weight is $O(\log n) \cdot wt(MST)$).

1 Introduction

Let V be any set of n points in k -dimensional Euclidean space. A subgraph of the complete Euclidean graph is a t -spanner if for any u and v in V , the length of the shortest path from u to v in the spanner is at most t times $d(u, v)$. The value of t is called the *stretch factor* of G . We consider the problem of constructing sparse spanners, with the objective of keeping the stretch factor a constant (that is, independent of n). The motivation is, complete graphs represent ideal communication networks but are expensive to build in practice, and sparse spanners represent low cost alternatives. Sparseness of spanners is usually measured by various criteria such as small degree, few edges, and/or small edge weight. This paper is significant because we construct a single spanner which is extremely sparse by *all the above criteria*. Some of the sparseness results are optimal in a very strong sense.

Spanners for complete Euclidean graphs as well as for general graphs find applications in robotics, network topology design, distributed systems, design of parallel machines, and have also been investigated by graph theorists. Recent surveys on spanners may be found in [2, 3, 12].

The minimum spanning tree of V , MST , is obviously the sparsest possible connected spanner both in number of edges and weight (although not in degree), but its

*Supported in part by NSF grant CCR-9306822

stretch factor can be as bad as $n - 1$ [1]. The sparseness of a spanner is frequently judged by comparing it to the number of edges and weight of the *MST*. In this paper we prove the following theorem.

Theorem 1.1 *Let V be any set of n points in k -dimensional Euclidean space. For any $\delta > 1$, there exists a polynomial-time constructible t -spanner (where t depends only on δ and k) with the following properties:*

1. *Its maximum degree is 3.*
2. *It has at most $n \cdot \delta$ edges.*
3. *For $k \leq 3$ its weight is $O(1) \cdot wt(MST)$, and for $k > 3$ its weight is $O(\log n) \cdot wt(MST)$. The constants implicit in the O -notation depend only on δ and k .*

Let us first consider the problem of constructing spanners with a small degree. In [2] an algorithm is described which constructs a spanner in k -dimensions with a constant maximum degree (that is, not dependent on n), however the degree is exponentially dependent on k and hence quite large for higher dimensions. In 2 dimensions, a degree-7 spanner is reported in [6], and a degree-5 spanner in [13]. It is known that the lower bound for the degree is 3 because, given a set of points arranged in a grid, a Hamiltonian path or circuit cannot be a spanner with a constant stretch factor. A breakthrough in this problem appeared in [11] where it is shown that there exists a degree-4 spanner with a constant stretch factor in k -dimensional space. In that paper the author reports that it may be difficult to extend the technique to construct degree-3 spanners, but conjectures that such spanners exist.

In this paper we prove the existence of a degree-3 spanner in k -dimensional space. With this result, the problem of constructing spanners with small degree is satisfactorily solved. The proof employs some of the techniques in [11], but several new and non-trivial ideas are used.

Now consider the problem of generating spanners with few edges. A lower bound on the number of edges is $n - 1$ because a spanning tree is a connected graph with the fewest edges. Spanners with $O(n)$ edges and constant stretch factors have been designed in several papers, but in most of these spanners the constant implicit in the O -notation depends on the dimension [1, 2, 3, 4, 7, 9, 10, 14]. The degree-4 spanner constructed in [11] is an improvement, because it has at most $2 \cdot n$ edges. The next question is, *can the proportionality constant be reduced even further?* We are thus seeking edge sparseness in a very strong sense. This has been answered in the affirmative in 2-dimensions. In [1] it is shown that, given any $\delta > 1$, a spanner with a constant stretch factor (which depends on δ) can be constructed with at most $n \cdot \delta$ edges. However, the techniques employ planarity properties and do not extend to higher dimensions.

In this paper we show that, given any $\delta > 1$, a degree-3 spanner in k -dimensional space can be constructed with a constant stretch factor (which depends on k and δ), and with at most $n \cdot \delta$ edges. Thus the number of edges can be made *arbitrarily small*, with a corresponding increase in the stretch factor. This result satisfactorily solves the general problem of constructing spanners with few edges.

We now turn our attention to the third criterion, that is the weight of the spanner. Let $wt(G)$ be the total edge weight of a graph G . In [5, 8] it is shown that a

2-dimensional spanner exists with weight at most $O(1) \cdot wt(MST)$, which is an asymptotically optimal result. Both papers exploit planarity and the techniques do not extend to higher dimensions. In [2] it is shown that a k -dimensional spanner exists with weight at most $O(\log n) \cdot wt(MST)$. In [4] it is shown that for $k \leq 3$, a spanner exists with weight at most $O(1) \cdot wt(MST)$. Thus asymptotic optimality has been achieved up to the third dimension. But the spanners in [2, 4] do not have the degree and edges bounds presented here.

In addition to having a degree of 3 and the number of edges bounded by $n \cdot \delta$, the spanner in this paper has a weight bound that match the spanners in [2, 4]. However, the general problem of constructing spanners with low weight has not yet been satisfactorily solved. The $O(\log n)$ factor in dimensions higher than 3 is probably suboptimal, and one would like to go beyond asymptotic optimality even for $k \leq 3$. For example, given any $\gamma > 1$, are there spanners with weight at most $wt(MST) \cdot \gamma$?

The paper is organized as follows. In Section 2 we introduce some notations and summarize previous research relevant to our proofs. Section 3 describes the algorithm and shows that the resulting spanner is degree-3. Sections 4 and 5 are devoted to proving upper bounds on the number of edges and the total weight respectively. We conclude with some open problems.

Remark: Throughout the paper various constants are used, but lack of space prohibits us from explicitly presenting their values or their functional dependencies on other parameters such as k and δ . They are available in the complete version of this paper.

2 Preliminaries

In this section we introduce a few basic definitions and also summarize some previous relevant results. The Euclidean distance between two points u and v is $d(u, v)$. The length of the shortest path between two vertices u and v of a graph G is $d_G(u, v)$.

In [2, 4] a greedy algorithm is developed for constructing spanners of any set V of n points in k -dimensional space. Given any $t' > 1$, this algorithm constructs a degree- s t' -spanner, where s depends on k and t' . Such spanners are called *greedy spanners* because the algorithm is greedy. Select a fixed value of t' (such as $t' = 2$). In the rest of this paper, assume that whenever the greedy algorithm is run, it uses this value of t' . Thus the degree of the resulting spanner, s , depends only on k . It is known that for $k \leq 3$ the weight of the greedy spanner is $O(1) \cdot wt(MST)$, and for $k > 3$ the weight is $O(\log n) \cdot wt(MST)$. The constants implicit in the O -notation depend only on k .

The technique in [11] generates a degree-4 spanner whose stretch factor depends only on k . This result uses the concept of *nearest-neighbor graphs*. Since our paper also relies on the same concept, we present it in detail here.

Assume no two interdistances among the points in V are identical. The *nearest-neighbor graph* is a digraph over the vertex set V , such that from each vertex u , there is a directed edge to its nearest neighboring vertex, $n_V(u)$. The following are true. Every simple path in this digraph follows edges of decreasing length. The only

cycles are *short cycles*, containing exactly two vertices. In every simply-connected component of this digraph, exactly one short cycle appears. Thus, each simply-connected component has at least two vertices.

This digraph can be transformed into an undirected forest as follows. Replace directed edges by undirected edges. Since short cycles will be transformed into multi-edges, replace multi-edges by single edges. The result is the undirected *nearest-neighbor forest*. In this forest, each tree contains a special edge created from the unique short cycle of the corresponding simply-connected component of the digraph. Designate one of the two adjacent vertices of this edge the root of the tree. We use the notation $r(T)$ to represent the root of a tree T . Thus the nearest-neighbor forest is a forest of rooted trees. It is noted in [11] that the degree of a nearest-neighbor forest is at most 3^k .

3 Constructing a Degree-3 Spanner

In this section we show how to construct a degree-3 spanner in k -dimensional space, whose stretch factor depends only on k . In [11] an algorithm is described for constructing degree-4 spanners. We borrow two techniques from that paper, namely the use of nearest-neighbor forests, and a method of transforming a fixed degree tree into a degree-3 tree. However, to construct a degree-3 spanner with other sparseness properties requires several new and non-trivial ideas.

Recall that s is the maximum degree of greedy spanners, where s depends only on k . The intuitive idea is to partition V into clusters such that each cluster has a representative vertex u (called the *root*) and several other vertices, of which at least $s - 2$ vertices (say w_1, w_2, \dots, w_{s-2}) are quite close to u as compared to the distance between u and roots of other clusters. Then an individual spanner for each cluster is constructed, such that each root is degree-1, each of w_1, w_2, \dots, w_{s-2} is degree-2, and the remaining vertices in the cluster are at most degree-3. A spanner G for V is created by taking the union of all individual spanners with a greedy spanner constructed for the set of roots of all clusters. The degree of G is reduced to 3 by shifting $s - 2$ of the greedy spanner edges incident to a root u to its nearby vertices w_1, w_2, \dots, w_{s-2} .

The result in [11] also uses a clustering concept. There each cluster may have several vertices, but only one other vertex is guaranteed to be close to the representative vertex. The drawback is that a degree-3 spanner cannot be constructed. The advantage of our clustering technique is two-fold. Apart from constructing degree-3 spanners, we also reduce the total number of edges in the spanner. Our clusters are constructed from a hierarchy of nearest-neighbor forests, whereas the clusters in [11] are constructed from a single level nearest-neighbor forest.

The construction will consist of a series of steps.

Step 1: Let $V^1 = V$. Construct the nearest-neighbor forest of V^1 , which is called the *first-level nearest-neighbor forest* and denoted NNF^1 . Then construct the *fragmented first-level nearest-neighbor forest* (denoted $FNNF^1$) as follows. Select any $c > 1$. Label each edge of NNF^1 as either *short* or *long*. An edge in a tree is labeled short if it is adjacent to the root or if its length is less than

c times the length of its parent edge. Otherwise the edge is labeled long. After the labeling is over, repeat the following step until no tree in the forest has three consecutive short edges on a root to leaf path. Locate a tree which has three consecutive short edges, with the middle short edge being (u, w) , with u the parent of w . Fragment the tree into two by removing the middle short edge, and make w the root of the new tree.

Step 2: Let $l = s - 2$. (A larger l will also work, and in fact, in the next section the algorithm is run with a larger l). Thus l depends only on k . Repeat the following procedure for $i = 2, 3, \dots, l$. Set V^i to be the set of all the roots in $FNNF^{i-1}$. Construct the nearest-neighbor forest of V^i , denoted NNF^i . Then construct $FNNF^i$ by fragmenting NNF^i exactly as in Step 1, using the same value of c .

At this stage we have constructed a hierarchy of l fragmented forests. Several properties hold. The forests are pairwise edge disjoint. Each tree in $FNNF^i$ has at least two vertices. The set V^i is a proper subset of V^{i-1} . The following lemmas describe additional properties.

Lemma 3.1 *Let u be a vertex of tree T^i belonging to $FNNF^i$. Let v be any other vertex of V^i . Then $d_{T^i}(u, r(T^i))$ is at most a constant times $d(u, v)$.*

Proof : Clearly $d(u, n_{V^i}(u)) \leq d(u, v)$. But $(u, n_{V^i}(u))$ is the first edge along the path in T^i from u to the root. The way $FNNF^i$ is constructed, no three consecutive edges on this path are short edges. Thus the length of this path converges to at most a constant times the length of the first edge, $d(u, n_{V^i}(u))$. ■

We next show that each root has a close child compared to the distance between the root and other roots in the same level.

Lemma 3.2 *Let u and v be two roots of $FNNF^i$. Let w be the closest child of u . Then $d(u, v)$ is at least a constant times $d(u, w)$.*

Proof : The vertex u could either be a root of NNF^i , or a root of a tree created by the fragmentation process. In the first case, $w = n_{V^i}(u)$ so the lemma trivially holds. In the second case, u became a root because, both (w, u) and $(u, n_{V^i}(u))$ were short edges in NNF^i and $(u, n_{V^i}(u))$ was removed. Thus $d(w, u) < c \cdot d(u, n_{V^i}(u)) \leq c \cdot d(u, v)$. ■

Step 3: Construct the *composite forest* by taking the union of all $FNNF^i$, for $i = 1, 2, \dots, l$.

Each connected component R in the composite forest is a tree, and is basically the union of a single l^{th} level tree T^l , with all $(l-1)^{\text{th}}$ level trees whose roots are vertices of T^l , with all $(l-2)^{\text{th}}$ level trees whose roots are vertices of these $(l-1)^{\text{th}}$ level trees, and so on. Define the root of R as $r(R) = r(T^l)$. The maximum degree of R depends only on k (it is at most l times the maximum degree of any nearest-neighbor forest). Each R has at least 2^l vertices. Furthermore, $r(R)$ has at least l children because it is the common root to l different level trees.

Lemma 3.3 *Each composite tree R is a spanner over its own vertices, with a stretch factor dependent only on k .*

Proof: Let u and v be two vertices in R . We shall show that $d_R(u, v)$ is at most a constant times $d(u, v)$. Let w be the nearest common ancestor of u and v in R . Let T^1, T^2, \dots, T^l be trees such that u is in T^1 , $r(T^1)$ is in T^2 , $r(T^2)$ is in T^3 , and so on. Let H^1, H^2, \dots, H^l be trees such that v is in H^1 , $r(H^1)$ is in H^2 , $r(H^2)$ is in H^3 , and so on. Let j be the smallest integer such that $T^j = H^j$. Thus w belongs to T^j . The path from u to w in R goes via $r(T^1), r(T^2), \dots, r(T^{j-1})$. The path from v to w in R goes via $r(H^1), r(H^2), \dots, r(H^{j-1})$.

By applying Lemma 3.1, both $d_{T^1}(u, r(T^1))$ and $d_{H^1}(v, r(H^1))$ are at most a constant times $d(u, v)$. This also implies that $d(r(T^1), r(H^1))$ is at most a constant times $d(u, v)$. If we carry this argument to higher levels we can conclude that both $d_{T^i}(r(T^{i-1}), r(T^i))$ and $d_{H^i}(r(H^{i-1}), r(H^i))$ are at most a constant times $d(u, v)$. This also implies that $d(r(T^i), r(H^i))$ is at most a constant times $d(u, v)$. At level j we can conclude that both $d_{T^j}(r(T^{j-1}), w)$ and $d_{H^j}(r(H^{j-1}), w)$ are at most a constant times $d(u, v)$. However, these constants now depend on the level.

Since we have shown that each fragment of the path from u to v via w is at most a constant times $d(u, v)$, and there are $2 \cdot j$ such fragments, the lemma is proved. Of course, the constant depends on l , which in turn depends on k . ■

We next show that in the composite forest each root has l distinct close children compared to the distance between the root and other roots. This fact is crucial in the eventual construction. This is different from Lemma 3.2 where trees at any given level have only one child guaranteed to be close to the root.

Lemma 3.4 *Let u and v be two roots of the composite forest. Let R be the composite tree such that $u = r(R)$. For $i = 1, 2, \dots, l$, let T^i be the i^{th} level tree such that $u = r(T^i)$, and let w_i be the closest child of u in T^i . Then each w_i is a distinct child of u in R , and $d(u, v)$ is at least a constant times $d(w_i, u)$.*

Proof: By applying Lemma 3.2 at all levels. ■

We now introduce a *tree transformation* called f , which is similar to one used in [11]. Let R be any rooted tree with a constant maximum degree. $f(R)$ is another tree defined as follows. The root and the vertex set remain unchanged. For every vertex u in R , order its children in terms of increasing distance from u , say x_1, x_2, \dots, x_m . Remove all edges between u and its children, and reconnect by creating the chain $(u, x_1), (x_1, x_2), \dots, (x_{m-1}, x_m)$. We see that the root of $f(R)$ is degree-1, and the maximum degree of $f(R)$ is 3. If we apply the transformation again and create $f(f(R))$, the degree of the root will remain 1, the maximum degree will remain 3, and additionally the child of the root will have degree at most 2. In general, for the tree $f^i(R)$, the root will form one end of a chain of $i + 1$ vertices, with the rest of the tree connected to the other end of the chain. The maximum degree will remain 3.

Lemma 3.5 *Let R be a tree with a constant degree, such that it is a spanner over its own vertices, with a constant stretch factor. Then $f^i(R)$ is also a spanner with a stretch factor dependent upon i .*

Proof : Let (u, x) be an edge removed from R where u was the parent of x . An alternate path is created from u to x in $f(R)$ consisting of a constant number of edges, the length of each being no more than a constant times $d(u, x)$. Thus $f(R)$ is a spanner. Applying this argument i times proves the lemma. ■

Step 4: Construct a *transformed forest* from the composite forest as follows. Replace each tree R in the composite forest by the transformed tree $F = f^{i+1}(R)$.

Each F is a spanner over its vertices with a stretch factor dependent only on k , due to Lemmas 3.3 and 3.5. In addition, the root is connected to $l+1$ other vertices by a chain. In this chain the root is degree-1 and the l intermediate vertices are each degree-2. The last vertex on this chain is attached to the rest of the tree, and may be degree-3.

We next show that in the transformed forest the chain attached to each root is quite short compared to the distance between the root and other roots.

Lemma 3.6 *Let u and v be two roots of the transformed forest. Let F be the transformed tree such that $u = r(F)$. Then $d(u, v)$ is at least a constant times the path length in F from u to its l^{th} descendent, where this constant depends on k .*

Proof : Let R be the composite tree such that $u = r(R)$. In R , u has l distinct children w_1, w_2, \dots, w_l satisfying Lemma 3.4. The tree F is transformed from R , and thus has a chain whose edges are say, $(u, x_1), (x_1, x_2), \dots, (x_l, x_{l+1})$. The final vertex x_{l+1} is connected to the rest of the tree. Since the vertex sets of R and F are the same, one of w_1, w_2, \dots, w_l has to belong to the subtree of F with root x_l . Suppose this is vertex w_i . We know F is a spanner, so $d_F(w_i, u)$ is at most a constant (which depends on k) times $d(w_i, u)$. But $d_F(w_i, u) \geq d_F(x_l, u)$ because the path from w_i to u has to traverse x_l . So $d_F(x_l, u)$ is at most a constant times $d(w_i, x)$, which in turn is at most a constant times $d(u, v)$ due to Lemma 3.4. ■

Step 5: Let V^{l+1} be the set of roots of the transformed forest. Construct a degree- s greedy spanner of V^{l+1} using the greedy algorithm in [2, 4]. Let G be the graph formed by the union of the greedy spanner and the transformed forest.

Lemma 3.7 *G is a spanner of V with a stretch factor dependent only on k .*

Proof : Let u and v be two vertices in V . We have to construct a short path between them within G . If both belong to the same transformed tree F , there exists a short path between them within F since F is a spanner over its vertices. Otherwise let u belong to F_1 and v belong to F_2 . Let F_1 and F_2 be transformed from the composite trees R_1 and R_2 . Arguments similar to those in Lemma 3.3 can be used to show that both $d_{R_1}(u, r(R_1))$ and $d_{R_2}(v, r(R_2))$ are at most a constant (which depends only on k) times $d(u, v)$. Since F_1 and F_2 are themselves spanners, this implies that both $d_{F_1}(u, r(F_1))$ and $d_{F_2}(v, r(F_2))$ are at most a constant (which depends only on k) times $d(u, v)$. This in turn implies that $d(r(F_1), r(F_2))$ is also at most a constant (which depends only on k) times $d(u, v)$. Since the greedy spanner has a short path between $r(F_1)$ and $r(F_2)$, we conclude that the path from u to $r(F_1)$ within F_1 , then

from $r(F_1)$ to $r(F_2)$ within the greedy spanner, then from $r(F_2)$ to v within F_2 is at most a constant times $d(u, v)$, where the constant depends only on k . ■

The vertices of G are at most degree-3, except for the vertices in V^{l+1} . Such vertices could potentially be degree- $(s+1)$, one edge belonging to the transformed tree, and s edges belonging to the greedy spanner. We now show how to reduce these to degree-3.

Step 6: Perform the following at each vertex u in V^{l+1} . Let F be the transformed tree whose root is u . Suppose the chain in F originating from u is $(u, x_1), (x_1, x_2), \dots, (x_l, x_{l+1})$. Suppose the greedy spanner edges incident at u are e_1, e_2, \dots, e_s . Recall that $l = s - 2$. Shift e_1, e_2, \dots, e_{s-2} away from u so that they are incident to x_1, x_2, \dots, x_{s-2} respectively.

Lemma 3.8 G is a degree-3 spanner of V with a stretch factor dependent only on k .

Proof : It is easy to see that G is a degree-3 graph. Now consider an edge (u, v) that was shifted in Step 6. Let the new endpoints be (u_1, v_1) . The vertex u_1 can be no further away from u than the l^{th} descendent of u in the transformed forest. Similarly the vertex v_1 can be no further away from v than the l^{th} descendent of v in the transformed forest. By applying Lemmas 3.6 and 3.7, there is a path in G from u to v which is at most a constant times $d(u, v)$, where this constant depends only on k . ■

4 Reducing the Number of Edges

In the previous section we showed how to construct a degree-3 spanner G with a stretch factor dependent only on k . However, graphs with maximum degree of 3 may still have as many as $n \cdot (3/2)$ edges. It is of interest to see whether the proportionality factor can be reduced even further. In this section we show that given any $\delta > 1$, there exists a degree-3 spanner with at most $n \cdot \delta$ edges, and with a stretch factor dependent only on k and δ .

Earlier we had chosen $l = s - 2$. A larger l will work, and the resulting degree-3 spanner will have fewer edges. Select the smallest integer l such that $l \geq s - 2$ and $\delta > 1 + \frac{s}{2^{l+1}}$, and then construct the spanner G as in the previous section. The stretch factor now depends on k and δ .

Lemma 4.1 G has at most $n \cdot \delta$ edges.

Proof : There are two types of edges in G : the *forest edges* that belong to the transformed forest, and the *shifted greedy spanner edges* that belonged to the greedy spanner over V^{l+1} but were eventually shifted in Step 6 (see previous section). There are at most $n - 1$ forest edges. Since each transformed tree has at least 2^l vertices, the number of roots in the transformed forest is at most $n/2^l$. Thus the number of shifted greedy spanner edges is at most $n \cdot (\frac{s}{2^{l+1}})$, which implies that the number of edges in G is at most $n \cdot (1 + \frac{s}{2^{l+1}}) - 1$. ■

5 Estimating the Weight of the Spanner

In this section we provide an upper bound on the total edge weight of the spanner G constructed in Section 3. Let $MST(V)$ denote a minimum spanning tree of the point set V . We show that for $k \leq 3$, $wt(G) = O(1) \cdot wt(MST(V))$. This is an asymptotically optimal result. For $k > 3$ we show that $wt(G) = O(\log n) \cdot wt(MST(V))$. The constants implicit in the O -notation depend on k and δ .

Lemma 5.1 *Let U be a subset of V . The following are true.*

1. *The nearest-neighbor forest of V is contained in $MST(V)$.*
2. *$wt(MST(U)) = O(wt(MST(V)))$.*

Proof: Straightforward. ■

The edges of G consist of forest edges and shifted greedy spanner edges. Consider the forest edges first. Let F be a transformed tree which is constructed from a composite tree R . The nature of the transformation makes it easy to see that the weight of F is at most a constant (which depends on k and δ) times the weight of R . Thus the weight of the transformed forest is at most a constant times the weight of the composite forest. But the weight of the composite forest is the sum of the weight of each $FNNF^i$, where $FNNF^i$ is constructed over V^i . But V^i is a subset of V . By applying Lemma 5.1, we can conclude that the transformed forest weighs at most a constant times $MST(V)$, and this constant depends on k and δ .

Now consider the shifted greedy spanner edges. Each edge (u_1, v_1) corresponds to an edge (u, v) of the greedy spanner over V^{l+1} . But $d(u_1, v_1)$ is at most a constant (which depends on k and δ) times $d(u, v)$. Using the known weight bounds on greedy spanners (see Section 2), the fact that V^{l+1} is a subset of V , and Lemma 5.1, we can conclude that the total weight of these edges is $O(1) \cdot wt(MST(V))$ for $k \leq 3$, and $O(\log n) \cdot wt(MST(V))$ for $k > 3$. Thus the weight of the shifted greedy spanner edges dominates the weight of forest edges in an asymptotic sense, and we have proven the following lemma.

Lemma 5.2 *For $k \leq 3$, $wt(G) = O(1) \cdot wt(MST(V))$, and for $k > 3$, $wt(G) = O(\log n) \cdot wt(MST(V))$. The constants implicit in the O -notation depend only on δ and k .*

6 Open Problems

We conclude with some open problems. Most of the problems considered in this paper have their optimization counterparts. For example, given V , design a degree-3 spanner with the minimum stretch factor. Another version is, given a stretch factor, design a spanner with minimum number of edges, or minimum weight. It is likely that most of these are intractable. If so, then good approximation algorithms are necessary.

The weight result has room for improvement. For $k > 3$ the $O(\log n)$ factor needs to be eliminated. Even if that is achieved, we can still ask stronger questions such as, given $\gamma > 1$, are there spanners with weight $wt(MST(V)) \cdot \gamma$?

References

- [1] I. Althöfer, G. Das, D.P. Dobkin, D. Joseph, J. Soares: On Sparse Spanners of Weighted Graphs. *Discrete and Computational Geometry*, 9, 1993, pp. 81-100
- [2] B. Chandra, G. Das, G. Narasimhan, J. Soares: New Sparseness Results on Graph Spanners. *ACM Symposium on Computational Geometry*, 1992, pp. 192-201
- [3] G. Das: Approximation Schemes in Computational Geometry. PhD Thesis, CS Dept, Univ of Wisconsin-Madison, 1990
- [4] G. Das, P. Heffernan, G. Narasimhan: Optimally Sparse Spanners in 3-Dimensional Euclidean Space. *ACM Symposium on Computational Geometry*, 1993, pp. 53-62
- [5] G. Das, D. Joseph: Which Triangulations Approximate the Complete Graph? *International Symposium on Optimal Algorithms, LNCS, Springer-Verlag*, 1989
- [6] D.P. Dobkin, S.J. Friedman, K.J. Supowit: Delaunay Graphs are Almost as Good as Complete Graphs. *Discrete and Computational Geometry*, 5, 1990, pp. 399-407
- [7] J.M. Keil: Approximating the Complete Euclidean Graph. *SWAT, LNCS, Springer-Verlag*, 1989
- [8] C. Levcopoulos, A. Lingas: There are Planar Graphs Almost as Good as the Complete Graphs and as Short as the Minimum Spanning Trees. *Symposium on Optimal Algorithms, LNCS, Springer-Verlag*, 1989, pp. 9-13
- [9] D. Rupert, R. Seidel: Approximating the d-Dimensional Complete Euclidean Graph. *Canadian Conference on Computational Geometry*, 1991, pp. 207-210
- [10] J.S. Salowe: Construction of Multidimensional Spanner Graphs with Applications to Minimum Spanning Trees. *ACM Symposium on Computational Geometry*, 1991, pp. 256-261
- [11] J.S. Salowe: On Euclidean Spanner Graphs with Small Degree. *ACM Symposium on Computational Geometry*, 1992, pp. 186-191
- [12] J. Soares: Graph Spanners. Ph.D Thesis, Univ. of Chicago Technical Report CS 92-14, 1992
- [13] J. Soares: Approximating Complete Euclidean Graphs by Bounded Degree Graphs. Manuscript, 1991
- [14] P.M. Vaidya: A Sparse Graph Almost as Good as the Complete Graph on Points in K Dimensions. *Discrete and Computational Geometry*, 6, 1991, pp. 369-381