Generating Sparse 2−spanners

Guy Kortsarz [*] David Peleg[*] [†]

November 22, 1993

Abstract

A $k-$spanner of a connected graph $G = (V, E)$ is a subgraph G' consisting of all the vertices of V and a subset of the edges, with the additional property that the distance between any two vertices in G' is larger than that distance in G by no more than a factor of k. This note concerns the problem of finding the sparsest 2-spanner in a given graph, and presents an approximation algorithm for this problem with approximation ratio $\log(|E|/|V|)$.

[*]Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel.

[†]Supported in part by a Walter and Elise Haas Career Development Award and by a grant from the Basic Research Foundation.

1 Introduction

The concept of *graph spanners* has been studied in several recent papers, in the context of communication networks, distributed computing, robotics and computational geometry [ADDJ90, Cai91, Che86, DFS87, DJ89, LL89, PS89, PU89]. Consider a connected simple graph $G = (V, E)$, with $|V| = n$ vertices. A subgraph $G' = (V, E')$ of G is a $k-$ *spanner* if for every $u, v \in V$,

$$\frac{dist(u,v,G')}{dist(u,v,G)} \leq k,$$

where $dist(u, v, G')$ denotes the distance from u to v in G', i.e., the minimum number of edges in a path connecting them in G'. We refer to k as the *stretch factor* of G'.

In the Euclidean setting, spanners were studied in [Cai91, DFS87, DJ89, LL89]. Spanners for general graphs were first introduced in [PU89], where it was shown that for every $n-$vertex hypercube there exists a 3-spanner with no more than $7n$ edges. Spanners were used in [PU89] to construct a new type of synchronizer for an asynchronous network. For this, and other applications, it is desirable that the spanners be as *sparse* as possible, namely, have few edges. This leads to the following problem. Let $S_k(G)$ denote the minimum number of edges in a $k-$spanner for the graph G. The *sparsest k-spanner* problem involves constructing a $k-$spanner with $S_k(G)$ edges for a given graph G.

It is shown in [PS89] that the problem of determining, for a given graph $G = (V, E)$ and an integer m, whether $S_2(G) \leq m$ is NP-complete. This indicates that it is unlikely to find an exact solution for the sparsest $k-$spanner problem even in the case $k = 2$. Consequently, two possible remaining courses of action for investigating the problem are establishing global bounds on $S_k(G)$ and devising approximation algorithms for the problem.

In [PS89] it is shown that every $n-$vertex graph G has a polynomial time constructible $(4k + 1)-$spanner with at most $O(n^{1+1/k})$ edges, or in other words, $S_{4k+1}(G) = O(n^{1+1/k})$ for every graph G. Hence in particular, every graph G has an $O(\log n)-$spanner with $O(n)$ edges. These results are close to the best possible in general, as implied by the lower bound given in [PS89]. The construction of [PS89] is based on the concept of *sparse covers* or *partitions* (cf. [AP90]). Consequently, faster algorithms for constructing sparse covers, in either the sequential, parallel or distributed modes [LS91, ABCP91, ABCP92b, ABCP92a], directly translate into faster algorithms for spanner construction as well.

The results of [PS89] were improved and generalized in [ADDJ90] to the weighted case, in which there are positive weights associated with the edges, and the distance between two vertices is the weighted distance. Specifically, it is shown in [ADDJ90] that given an

$n-$vertex graph and an integer $k \geq 1$, there is a polynomially constructible $(2k+1)-$spanner G' such that $|E(G')| < n \cdot \lceil n^{\frac{1}{k}} \rceil$. Again, this result is shown to be the best possible.

The algorithms of [ADDJ90, PS89] provide us with *global* upper bounds for sparse $k-$spanners, i.e., general bounds that hold *for every graph*. However, it may be that for specific graphs, considerably sparser spanners exist. Furthermore, the upper bounds on sparsity given by these algorithms are small (i.e., close to n) only for large values of k. It is therefore interesting to look for *approximation algorithms*, that yield near-optimal *local* bounds applying to the specific graph at hand, by exploiting its individual properties.

In the sequel we concentrate on the sparsest 2-spanner problem. For this case, the best *global* upper bound is $S_2(G) = O(n^2)$. To see why this cannot be improved in general, consider the complete bipartite graph having $n/2$ vertices on each side. It is not hard to see that the only $2-$spanner for this graph is the graph itself. Thus there are cases where any 2-spanner requires $\Omega(n^2)$ edges. This lends additional motivation to our interest in approximating the sparsest $2-$spanner for specific graphs.

The construction of [ADDJ90] can be thought of as an approximation algorithm for the sparsest $k-$spanner problem. However, for the case of $k = 2$ the ratio provided by this algorithm might be as bad as $\Omega(n)$ (which is also the trivial ratio, since every 2-spanner contains at least $n - 1$ edges).

In this paper we present an approximation algorithm for the sparsest $2-$spanner problem with approximation ratio $\log \frac{|E|}{|V|}$. That is, given a graph $G = (V, E)$, our algorithm generates a $2-$spanner $G' = (V, E')$ with $|E'| = O(S_2(G) \cdot \log \frac{|E|}{|V|})$ edges. In the next three sections we give some preliminary definitions, describe the algorithm and analyze its performance. In the last section we show a matching lower bound for our algorithm. In particular, we exhibit a family of graphs G_k with $\Theta(k)$ vertices and $\Omega(k^2)$ edges for which our algorithm may find a 2-spanner with $\Omega(S_2(G) \cdot \log k)$ edges.

2 Preliminaries

We start by introducing some definitions. Let $U \subseteq V$ be a subset of the vertices. The graph induced by U is denoted by $G(U)$. The set of edges in $G(U)$ is denoted by $E(U)$. The *density* of U in G is defined as

$$\rho_G(U) = \frac{|E(U)|}{|U|} .$$

The *maximum density* of the graph G is defined to be

$$\rho(G) = \max_{U \subseteq V} \{\rho_G(U)\}.$$

We call the problem of finding a subgraph of G with density $\rho(G)$ the *maximum density* problem. We recall the following fact, derivable, e.g., from [Law76]; pp. 125-127, or alternatively from [GGT89].

Lemma 2.1 [Law76, GGT89] *The maximum density problem can be solved polynomially using flow techniques.*

The fastest algorithm known for the maximum density problem is given in [GGT89]. This algorithm runs in time $O(mn\log(n^2/m))$.

We make use of an alternative characterization of $k-$spanners, given in the following lemma of [PS89].

Lemma 2.2 [PS89] *The subgraph $G' = (V, E')$ is a $k-spanner$ of the graph $G = (V, E)$ iff $dist(u, v, G') \leq k$ for every $(v, u) \in E$.*

Next, we introduce the definition of a k-spanner of a subset $E' \subset E$ of the edges.

Definition 2.3 *Let E' be a subset of the edges. An* optimal k-spanner for E' *in G is a minimum subset $E'' \subset E$ such that every edge $e \in E' \setminus E''$ lies on a cycle of length $k+1$ or less with the edges of E''.*

Thus the sparsest $2-$spanner problem can be restated as follows: we look for a minimum subset of edges $E' \subset E$ such that every edge e that does not belong to E' lies on a triangle with two edges that do belong to E'. Since a spanning graph of any set E' is also a spanning graph of any subset $E'' \subset E'$, the following fact holds.

Fact 2.4 *Let E_1 be an arbitrary subset of E_2, and let $E_1' \subseteq E$ and $E_2' \subseteq E$ be the edge sets of an optimal $k-$spanner for E_1 and E_2 in G, respectively. Then $|E_1'| \leq |E_2'|$.* ▮

Given a graph G, we denote by $N(v)$ the set of neighbors of v in G, i.e.,

$$N(v) = \{u \mid (u, v) \in E\} .$$

Let E' be an arbitrary set of edges and U an arbitrary subset of vertices. Denote by $\mathcal{R}(E', U)$ the subset of the edges in the induced graph $G(U)$ *restricted* to E', namely,

$$\mathcal{R}(E', U) = E(U) \cap E' .$$

We denote

$$cov_G(E', v) = |\mathcal{R}(E', N(v))|$$

and say that v *covers* the edges of $\mathcal{R}(E', N(v))$ in G. Note that if all the edges adjacent to v are in the spanner, then all the edges of $\mathcal{R}(E', N(v))$ lie on a triangle with these spanner edges, and thus are taken care of. Denote the graph of neighbors of v restricted to E' by

$$N(E', v) = (N(v), \mathcal{R}(E', N(v))) \ .$$

Denote the maximum density of this restricted neighborhood graph by

$$\rho(E', v) = \rho(N(E', v)) \ .$$

3 The approximation algorithm

Let us first explain the idea behind our approximation algorithm for the 2−spanner problem. Throughout the run of the algorithm we maintain a *cover* of the edge set E by three sets of edges, denoted H^s, H^c and H^u. The set H^s contains *spanner edges*, i.e., edges that were already added to the constructed spanner. The set H^c consists of *covered edges*, i.e., edges that are either in the spanner, or lie on a triangle with two edges that are included in H^s. That is, at any given moment, for every edge $e \in H^c \setminus H^s$ there exist two edges $e_1, e_2 \in H^s$ such that e, e_1 and e_2 form a triangle. Finally, H^u consists of *unspanned edges*, i.e., edges that are still neither in the spanner nor covered by spanner edges.

Our algorithm operates by repeatedly performing the following operation. For every vertex v, we consider the graph $N(H^u, v)$, consisting of the set of neighbors of v, with the edge set restricted to the unspanned edges H^u. In this graph we look for a subset U_v of maximum density, relying on Lemma 2.1. Then we choose the most dense such set among all the sets $\{U_v \mid v \in V\}$. Assume that the chosen set is U_w.

After finding U_w, we add the "star" composed of the edges connecting U_w and w, to the edge set of the spanner H^s. In this way we *cover* a "large" set of edges (namely, those in $H^u \cap E(U_w)$), while adding only a "small" number of new edges (specifically, $|U_w|$) to the spanner.

This operation is repeated until all sets U_v are "sufficiently sparse," whence the algorithm halts and $H^s \cup H^u$ is taken to be the edge set of the resulting spanner.

We now state our approximation algorithm more precisely.

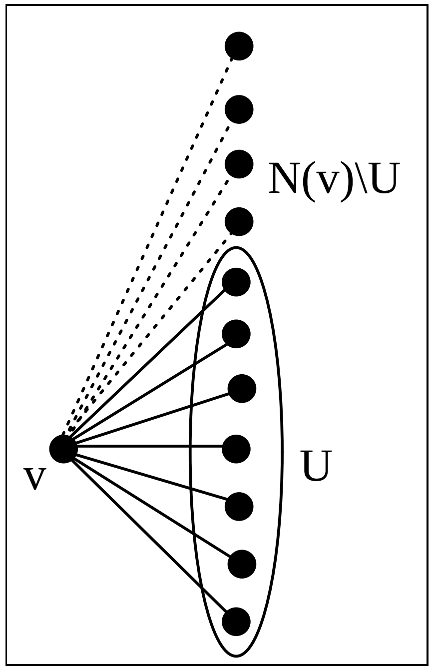

Figure 1: The set U represents a dense subset of $N(v)$. The solid edges are the ones added to the constructed spanner.

Algorithm 3.1 *An approximation algorithm for the* $2-$*spanner problem Input: a graph* $G = (V, E)$.

1. *Set* $H^u \leftarrow E$; $H^c \leftarrow \emptyset$; $H^s \leftarrow \emptyset$;
2. While *there exists some* v *for which* $\rho(H^u, v) \geq 1$ do
   (a) *Choose a vertex* v *for which* $\rho(H^u, v)$ *is maximum.*
   (b) *Let* U_v *be the corresponding dense subset of* $N(v)$.
   $H^s \leftarrow H^s \cup \{(u, v) \mid u \in U_v\}$.
   $H^c \leftarrow (H^c \cup \mathcal{R}(H^u, U_v)) \cup H_s$.
   $H^u \leftarrow H^u \setminus H^c$.

   End-While.
3. *Return*($H^s \cup H^u$)

4 Analysis

4.1 The approximation ratio

Note that the output set of edges indeed forms a $2-$spanner of G, since every edge in H^c lies on a triangle with two edges of H^s. Denote the edge set of an optimal $2-$spanner for G by H^*. Let us now proceed to bound from above the ratio between the sizes of the sets $H^s \cup H^u$ and H^*.

Let us break the execution of the main loop of the algorithm into phases as follows. Denote $r = \frac{|E|}{|V|}$ and $f = \lceil \log r \rceil$. Note that since the set H^u decreases in size at every step, $\rho(H^u, v)$ is monotonically decreasing as well.

Definition 4.1 *We define the first phase to include all the iterations during which for every selected vertex v, $\rho(H^u, v) \geq \frac{r}{2}$. For $2 \leq i \leq f$, the i'th phase consists of the iterations during which every selected vertex v satisfies*

$$\frac{r}{2^{i-1}} > \rho(H^u, v) \geq \frac{r}{2^i} .$$

Let H_i^s (respectively, H_i^c) be the set of new edges added to H^s (resp., H^c) in the i'th phase, and let H_i^u be the set of edges left in H^u at the *end* of the i'th phase. Note that from the above definition of the phases, and the fact that the algorithm always picks the vertex v maximizing $\rho(H^u, v)$, it follows that H_i^u satisfies

$$\rho(H_i^u, v) < \frac{r}{2^i} \tag{1}$$

for every v. Let H_i^* be the edge set of an optimal 2-*spanner* for H_i^u in G. We denote by X_i the set of vertices selected by the algorithm during step (a) of the iterations of the i'th phase (namely, those vertices for which $\rho(H^u, v)$ was maximum in the iterations of the i'th phase).

Note that a vertex v may be picked more than once during a phase, and in more than one phase. Consider a particular phase i. Each time that the vertex v is picked in the i'th phase, a subset $S_v = \{(w, v) | w \in U_v\}$ of its adjacent edges is added to H^s, namely, those edges connecting it to U_v. Also, there is a corresponding set $C_v = \mathcal{R}(H^u, U_v)$ of edges from H^u that lie on a triangle with the edges of S_v and are thus added to H^c. Since $|S_v| = |U_v|$, by definition of ρ, these sets S_v and C_v satisfy

$$\frac{|C_v|}{|S_v|} = \rho(H_u, v).$$

Denote the cardinality of the union of these sets C_v added during the i'th phase by $h_i^c(v)$, and the cardinality of the union of the sets S_v by $h_i^s(v)$, for every vertex $v \in X_i$. Note that by the definition of the i'th phase, it follows from the above that for every $v \in X_i$,

$$h_i^c(v) \geq \frac{r}{2^i} \cdot h_i^s(v) \ . \tag{2}$$

Observe that an edge $e = (v, u)$ may belong to two different sets S_v, S_u, hence

$$|H_i^s| \leq \sum_{v \in X_i} h_i^s(v) \ . \tag{3}$$

On the other hand, edges are included in sets C_v atmost once, hence

$$|H_i^c| \geq \sum_{v \in X_i} h_i^c(v) \ . \tag{4}$$

It follows from (2),(3) and (4) that

$$|H_i^c| \geq \frac{r}{2^i} |H_i^s| \ . \tag{5}$$

We now prove the following claim. Let $G_i^* = (V, H_i^*)$ be an optimal 2-spanner of H_i^u, and for every $v \in V$ let $d_i^*(v)$ be the degree of v in the graph G_i^*. Recall that $cov_{G_i^*}(H_i^u, v)$ is the number of edges of H_i^u covered by v in G_i^*. Denote

$$\rho_i(v) = \frac{cov_{G_i^*}(H_i^u, v)}{d_i^*(v)} \ .$$

Lemma 4.2 *For every $v \in V$,*

$$\rho_i(v) < \frac{r}{2^i} \ .$$

Proof: Let $N^*(v)$ be the set of vertices adjacent to v in G_i^*. Thus $|N^*(v)| = d_i^*(v)$. Also

$$cov_{G_i^*}(H_i^u, v) = |\mathcal{R}(H_i^u, N^*(v))| = |E(N^*(v)) \cap H_i^u| \ .$$

Thus $\rho_i(v)$ is the density of $N^*(v)$ in the restricted neighborhood graph $N(H_i^u, v)$, i.e., $\rho_i(v) = \rho_{N(H_i^u, v)}(N^*(v))$. This density is no larger than the maximum density of the graph $N(H_i^u, v)$, namely, $\rho_i(v) \leq \rho(H_i^u, v)$. Thus the required claim follows directly from inequality (1). ∎

Lemma 4.3 *For every $1 \leq i \leq f$,*

$$\frac{|H_i^u|}{|H^*|} < \frac{r}{2^{i-1}} + 1 \ .$$

Proof: First let us remark that

$$|H_i^u| \le |H_i^*| + \sum_{v \in V} cov_{G_i^*}(H_i^u, v), \tag{6}$$

since every edge $e \in H_i^u$ either belongs to H_i^* or is covered by some vertex in G_i^*. Secondly, by Fact 2.4 we have that

$$|H_i^*| \le |H^*| \ . \tag{7}$$

Thirdly, note that

$$|H_i^*| = \frac{1}{2} \sum_{v \in V} d_i^*(v) \ . \tag{8}$$

Combining Eq. (6), (7) and (8) we conclude that

$$\begin{aligned}
\frac{|H_i^u|}{|H^*|} &\le \frac{|H_i^u|}{|H_i^*|} \le \frac{|H_i^*| + \sum_{v \in V} cov_{G_i^*}(H_i^u, v)}{|H_i^*|} \\
&= 1 + \frac{\sum_{v \in V} cov_{G_i^*}(H_i^u, v)}{\frac{1}{2}\sum_{v \in V} d_i^*(v)} \\
&\le 1 + 2 \cdot \max_{v \in V} \left\{ \frac{cov_{G_i^*}(H_i^u, v)}{d_i^*(v)} \right\} \\
&= 1 + 2 \cdot \max_{v \in V} \{\rho_i(v)\}.
\end{aligned}$$

Thus by Lemma 4.2 we have

$$\frac{|H_i^u|}{|H^*|} < 2 \cdot \frac{r}{2^i} + 1 = \frac{r}{2^{i-1}} + 1 \ . \qquad \blacksquare$$

We now proceed to prove our main lemma.

Lemma 4.4 *For every* $1 \le i \le f$,

$$\frac{|H_i^s|}{|H^*|} < 4 + \frac{2^i}{r} \ .$$

Proof: We first prove the claim for $i = 1$. We may assume w.l.o.g that $n \ge 2$. In this case by Eq. (5) and by the choice of r

$$\frac{|H_1^s|}{|H^*|} \le \frac{\frac{2}{r} \cdot |H_1^c|}{|H^*|} \le \frac{\frac{2 \cdot n}{|E|} \cdot |E|}{|H^*|} \le \frac{2 \cdot n}{n-1} \le 4 \ .$$

We now prove the claim for $i > 1$. By Eq. (5) and by the fact that $H_i^c \subseteq H_{i-1}^u$, we have

$$\frac{|H_i^s|}{|H^*|} \le \frac{\frac{2^i}{r} \cdot |H_i^c|}{|H^*|} \le \frac{2^i}{r} \cdot \frac{|H_{i-1}^u|}{|H^*|} \ .$$

Using Lemma 4.3 we get

$$\frac{|H_i^s|}{|H^*|} < \frac{2^i}{r} \cdot \left(\frac{r}{2^{i-2}} + 1\right) = 4 + \frac{2^i}{r} \ . \qquad \blacksquare$$

Corollary 4.5

$$\frac{|H^s|}{|H^*|} = O(\log r) \ .$$

Proof: By Lemma 4.4 and the choice of f,

$$\begin{aligned} \frac{|H^s|}{|H^*|} &= \frac{\sum_{i=1}^{f} |H_i^s|}{|H^*|} \\ &\leq \sum_{i=1}^{f} \left(4 + \frac{2^i}{r}\right) \\ &= 4 \cdot f + \frac{1}{r} \cdot \sum_{i=1}^{f} 2^i = 4 \cdot f + O(1) = O(\log r). \qquad \blacksquare \end{aligned}$$

Furthermore, by Lemma 4.3 we have

Corollary 4.6 $\frac{|H^u|}{|H^*|} < 3$. $\blacksquare$

From Corollaries 4.5 and 4.6 we conclude our main result.

Theorem 4.7 *Algorithm 3.1 is an $O(\log \frac{|E|}{|V|})$ approximation algorithm for the sparsest 2-spanner problem.* $\blacksquare$

4.2 The time complexity

We now analyze the time complexity of our algorithm. In each iteration of the algorithm, the value of $\rho(H^u, v)$ is computed for each vertex v. This requires solving a maximum density problem. The algorithm given in [GGT89] for the maximum density problem has time complexity $O(m \cdot n \cdot \log(n^2/m))$, hence each iteration of our algorithm requires $O(m \cdot n^2 \cdot \log(n^2/m))$ operations. At each iteration, at least one edge is added to H_c. Since every edge is added to H_c at most once, there are at most m iterations, so the complexity of the algorithm is bounded by $O(m^2 \cdot n^2 \cdot \log(n^2/m))$ which is polynomial in the input size.

Since this complexity is rather high, we suggest the following way to speed up the algorithm, while losing only a *constant* factor in the approximation ratio. Instead of calculating at each stage the maximum density ρ of a subset of $N(v)$ for every v, we rather *approximate*

ρ. That is, we find a subset with density within a constant c from the maximum. It is easy to see that the approximation ratio of the algorithm remains asymptotically unchanged (it only grows by the constant c). The proof of this claim follows exactly as the proof of the previous subsection.

It remains to show how to approximate the maximum density problem. Given a graph G and a number ρ we check if the densest subgraph G' has density ρ or more. Note that every vertex v in the graph with degree $\rho - 1$ or less can not be contained in G', since by eliminating v from G' the density $\rho(G')$ is increased. Thus we apply the following iterative procedure. Let G_1 be a copy of G. Iteratively find in G_1 a vertex v with degree $\rho - 1$ or less (if exists), and eliminate v and its adjacent edges from G_1. If G_1 ends up empty, we conclude that the maximum density $\rho(G)$ is less than ρ. (This is because if the density is ρ or higher, the subgraph G' contains only vertices of degree ρ or more, and therefore must be preserved in G_1 throughout the elimination process.) Else, we found a subgraph G'' of G with minimum degree at least ρ, implying that the density of G'' is $\rho(G'') \geq \rho/2$. Thus by conducting a binary search over the possible values of ρ we obtain a subgraph G'' of density $\rho(G'') \geq \rho(G)/2$. This implies an approximation ratio of 2.

Clearly, the approximation procedure for the maximum density problem is considerably faster than the exact solution. In particular, using appropriate data structures (whose description is omitted from the paper), we get an approximation procedure for the maximum density problem with time complexity $O(m \log n + n \log^2 n)$. This, in turn, yields a logarithmic ratio approximation algorithm for the 2-spanner problem, with time complexity bounded by $O(m^2 \cdot n \log n + m \cdot n^2 \cdot \log^2 n)$.

5 A lower bound

In this section we establish tightness of the analysis in Section 4.1 by presenting a family of graphs for which the greedy algorithm for the 2-spanner problem performs as badly as $\Omega(\log n)$. That is, we exhibit a family of graphs G_k, for infinitely many values of k, with $\Theta(k)$ vertices and $\Omega(k^2)$ edges for which the greedy algorithm outputs a 2-spanner with $\Omega(S_2(G) \cdot \log k)$ edges.

5.1 The graph G_k

Let $k = 2^p$ for an integer p. Denote $k' = k-4$. Let $U = \{u_1, \ldots, u_{k'}\}$ and $W = \{w_1, \ldots, w_{k'}\}$. Break the set U into $p-2$ subsets by successive halving, letting U_1 contain the first $k/2$ of

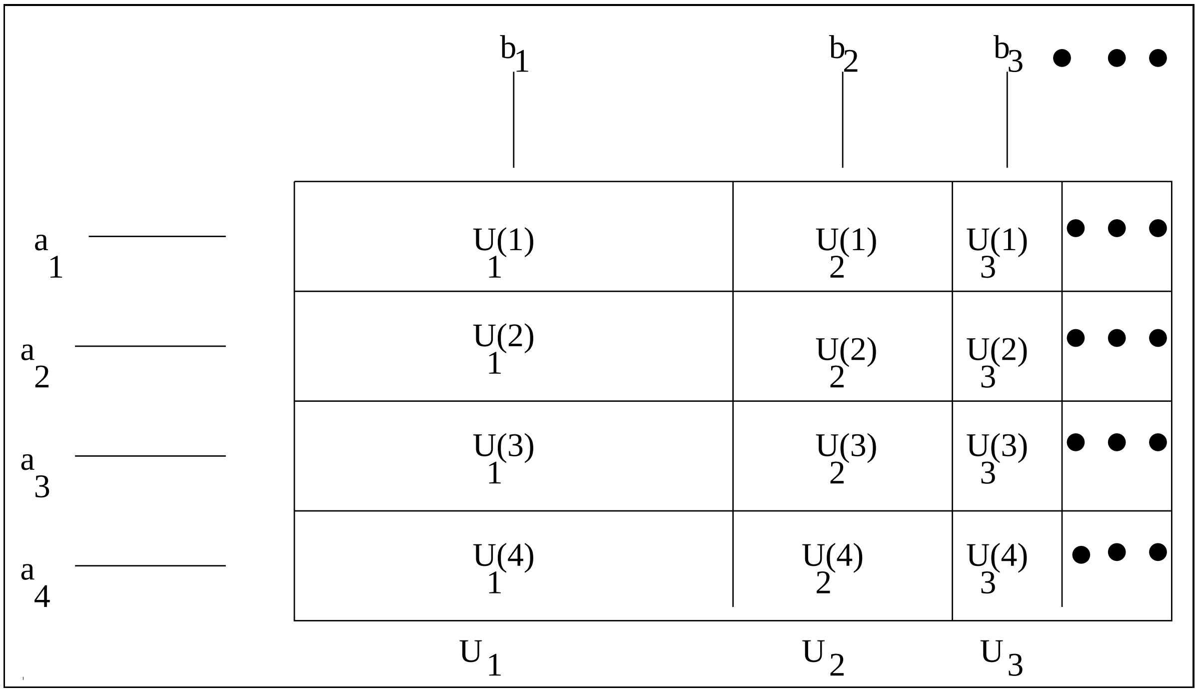

Figure 2: Break-up of the set U into subsets, and connections to the vertices of A and B.

the vertices, U_2 contain the next $k/4$ and so on. I.e.,

$$U_1 = \{u_1, \dots, u_{k/2}\}\ ,\ U_2 = \{u_{k/2+1}, \dots, u_{3k/4}\}, \dots\ ,\ U_{p-2} = \{u_{k-7}, u_{k-6}, u_{k-5}, u_{k-4}\}.$$

Also define two additional sets $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, \dots, b_{p-2}\}$. The vertex set is $V = U \cup W \cup A \cup B$. Note that the number of vertices, n, satisfies $n = \Theta(k)$. We shall further break each set U_i into four equal-sized subsets $U_i(j)$, for $j = 1, 2, 3, 4$ (see Fig. 2). Formally, for a set $P = \{p_1, \dots, p_{4\cdot l}\}$ of $4 \cdot l$ elements denote $P(j) = \{P_{(j-1)l+1}, \dots, P_{jl}\}$. For example, $U_1(1)$ consists of the first quarter of the vertices of U_1, i.e., $U_1(1) = \{u_1, \dots, u_{k/8}\}$ and so on.

We now specify the edge set of G_k.

(E_1) For $1 \le i \le 4$, the vertex a_i is connected to $W \cup \bigcup_j U_j(i)$.

(E_2) For $1 \le i \le p-2$, the vertex b_i is connected to every vertex in $W \cup U_i$.

(E_3) The sets W and U are connected by a complete balanced bipartite graph (that is, W and U are independent sets, and every vertex of W is connected to every vertex in U.)

(E_4) $A \cup B$ forms a clique.

5.2 The approximation ratio of the greedy algorithm on G_k

We now consider the question of a sparse $2-$spanner for the graph G_k defined above. Let us first observe that this graph has a $2-$spanner with $O(n)$ edges. This spanner is obtained by taking the edge subsets (E_1) and (E_4) above, namely, all the edges adjacent to a_i, for $1 \leq i \leq 4$. Hence we have:

Claim 5.1 $S_2(G_k) = O(n)$. ▮

On the other hand, we claim that our algorithm will construct for G_k a $2-$spanner with $\Omega(n \log n)$ edges. (Specifically, it includes the edge subset (E_2) above, namely, all the edges connecting b_i to $W \cup U_i$ for every $1 \leq i \leq p-2$.) In particular, the algorithm will select the vertices $b_1, \ldots, b_{p-2}$ in its iterations. To see this, observe first the following claim.

Claim 5.2 *The first vertex to be selected by our algorithm is b_1.*

Proof: Let us first compute bounds on $\rho(H^u, v)$ for every v in the initial situation. Consider a vertex $v \in A \cup B$, and denote its corresponding densest subgraph by $U_v = \hat{W} \cup \hat{U} \cup Z$, where $\hat{W}$ and $\hat{U}$ are subsets of W and U respectively, and Z is subset of $A \cup B$, $|Z| = l$. The density of the set U_v is bounded above by

$$\rho(U_v) = \frac{|E(U_v)|}{|U_v|} \leq \frac{|\hat{U}||\hat{W}| + |\hat{W}||Z| + |\hat{U}||Z| + |Z|^2}{|\hat{U}| + |\hat{W}| + |Z|} \leq \frac{|\hat{U}||\hat{W}|}{|\hat{U}| + |\hat{W}|} + l \tag{9}$$

For a vertex v in $A \cup B \setminus \{b_1\}$ the maximum size of $\hat{U}$ is $k/4$, and the maximum size of $\hat{W}$ is less than k. It thus follows that for a vertex in $A \cup B \setminus \{b_1\}$ the maximum density is bounded above by

$$\rho(U_v) \leq \frac{k \cdot k/4}{k + k/4} + l = k/5 + l.$$

On the other hand, $N(b_1)$ contains a subgraph of density larger than $k'/3$, namely $W \cup U_1$. It is easy to see that the density of $N(v)$ for vertices $v \in U \cup W$ is smaller. Thus for sufficiently large k (it is enough to choose k such that $2k/15 > p + 3$), b_1 is chosen. ▮

Next, we prove that when b_1 is selected, the densest subgraph (i.e., the subset of $N(b_1)$ selected) is in fact the entire neighborhood, $N(b_1)$.

Note that the maximum density of $N(b_1)$ is bounded by

$$k'/3 < \rho(N(b_1)) < k/3 + p + 1. \tag{10}$$

Let U_{b_1} denote the densest subgraph of $N(H^u, b_1)$. As before, let $\hat{U}$, $\hat{W}$ and Z, $|Z| = l$ be the subsets chosen, namely, $U_{b_1} = \hat{U} \cup \hat{W} \cup Z$, $Z \subseteq A \cup B$, $\hat{U} \subseteq U$, $\hat{W} \subseteq W$. We have the following claim.

Claim 5.3 *For sufficiently large k,*

$$|\hat{U}|, |\hat{W}| \geq 5k'/12$$

Proof: Suppose that the claim does not hold. W.l.o.g assume that $|\hat{U}| < 5k'/12$. Note that $|\hat{W}| \leq k'$ and therefore Eq. (9) would imply that the density of U_{b_1} is bounded above by

$$\rho(U_{b_1}) \leq \frac{|\hat{U}||\hat{W}|}{|\hat{U}| + |\hat{W}|} + l \leq \frac{5k'/12 \cdot k'}{5k'/12 + k'} + p + 1 = 5k'/17 + p + 1 < k'/3,$$

in contradiction with Eq. (10). ∎

From Claim 5.3 we deduce that (for sufficiently large k) $U_{b_1} = N(b_1)$. To see this, note that if this is not the case, it is possible to add to U_{b_1} an outside vertex $v' \in N(b_1) \setminus U_{b_1}$. By Claim 5.3 the number of neighbors of v' in U_{b_1} is at least $5k'/12$ and by Eq. (10), $5k'/12 > k/3 + p + 1 > \rho(U_{b_1})$. Thus the density of $U_{b_1} \cup \{v'\}$ is larger, a contradiction.

Thus in the first iteration of our algorithm, the star composed of all the edges of b_1 is added to the spanner.

At the end of the first iteration the situation becomes somewhat simpler. All the edges connecting vertex pairs in $A \cup B$, $A \cup W$ and $B \cup W$ are already spanned, and it remains only to take care of edges connecting U to $A \cup B \cup W$. Hence starting from the second stage, for every $v \in A \cup B$, the neighborhood of v in the collection of unspanned edges H^u, $N(H^u, v)$, is a bipartite graph, with the vertices of U in one side, and the rest of v's neighbors in the other. It follows by arguments similar to the above that

Claim 5.4 *In the i'th iteration, $2 \leq i \leq p - 2$, the vertex b_i is chosen and all the edges connecting b_i to the vertices of W and U (and no other edges) are added to the spanner.* ∎

It thus follows from Claim 5.4 that the number of edges in the constructed spanner is $\Omega(k \log k) = \Omega(n \log n)$. Combined with Claim 5.1, we conclude:

Lemma 5.5 *On the graph G_k, the approximation ratio provided by our algorithm is $\Omega(\log n)$.* ∎

6 Conclusion and open problems

We have shown that there exists an approximation algorithm for the sparsest $2-$spanner problem with a worst case approximation ratio of $\Theta(\log \frac{|E|}{|V|})$. Note that while the worst case ratio of the algorithm is $O(\log n)$, it performs better for sparse graphs. The next immediate problem is to approximate the sparsest $k-$spanner problem for an arbitrary fixed value of k with a similar ratio. This seems to be a considerably more difficult problem than the one solved in this paper, even for $k = 3$. Another interesting problem is to give an approximation algorithm for the weighted version of this problem.

Acknowledgment

We are grateful to Noga Alon for his helpful comments, and for directing us to Lemma 2.1 in [Law76]. We would like to thank Barun Chandra for helpful comments on previous drafts.

References

[ABCP91] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast constructions of sparse neighborhood covers. Unpublished manuscript, 1991.

[ABCP92a] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast network decomposition. In *Proc. 11th ACM Symp. on Principles of Distributed Computing*, pages 169–177, August 1992.

[ABCP92b] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Low diameter graph decomposition is in nc. volume LNCS-621, pages 83–93. Springer-Verlag, 1992.

[ADDJ90] I. Althöfer, G. Das, D. Dobkin, and D. Joseph. Generating sparse spanners for weighted graphs. In *Proc. 2nd Scandinavian Workshop on Algorithm Theory*, volume LNCS-447, pages 26–37. Springer-Verlag, July 1990.

[AP90] B. Awerbuch and D. Peleg. Sparse partitions. In 31^{st} *IEEE Symp. on Foundations of Computer Science*, pages 503–513, October 1990.

[Cai91] L. Cai. Tree 2-spanners. Technical Report No. 91-4, Simon Fraser University, 1991.

[Che86] L.P. Chew. There is a planar graph almost as good as the complete graph. In *ACM Symposium on Computational Geometry*, pages 169–177, 1986.

[DFS87] D.P. Dobkin, S.J. Friedman, and K.J. Supowit. Delaunay graphs are almost as good as complete graphs. In *Proc. 31'st IEEE Symp. on Foundations of Computer Science*, pages 20–26, 1987.

[DJ89] G. Das and D. Joseph. Which triangulation approximates the complete graph? In *International Symposium on Optimal Algorithms*, volume LNCS-401, pages 168–192. Springer-Verlag, 1989.

[GGT89] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. on Comput.*, 18:30–55, 1989.

[Law76] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.

[LL89] C. Levcopoulos and A. Lingas. There are planar graphs almost as good as the complete graph and as short as minimum spanning trees. In *International*

Symposium on Optimal Algorithms, volume LNCS-401, pages 9–13. Springer-Verlag, 1989.

[LS91] A.L. Liestman and T. Shermer. Grid and hypercube spanners. Technical Report No. 91-1, Simon Fraser University, 1991.

[PS89] D. Peleg and A. Schäffer. Graph spanners. *J. of Graph Theory*, 13:99–116, 1989.

[PU89] D. Peleg and J.D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. on Comput.*, 18:740–747, August 1989.