On the Hardness of Approximating Spanners

Guy Kortsarz *

June 1, 1999

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 the distance in G by no more than a factor of k. This paper concerns the hardness of finding spanners with a number of edges close to the optimum. It is proved that for every fixed k, approximating the spanner problem is at least as hard as approximating the set cover problem

We also consider a weighted version of the spanner problem, and prove an essential difference between the approximability of the case $k = 2$, and the case $k \geq 5$.

*Department of Computer Science, The Open University, 16 Klauzner st., Ramat Aviv, Israel, guyk@shaked.openu.ac.il.

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 [ADDJ-90, C-94, CK-94, C-86, DFS-87, DJ-89, LR-93, LS-93, PS-89, PU-89, RL-95]. Consider a simple connected 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 [C-86, DFS-87, DJ-89, LL-89, ADM$^+$95]. The first paper to consider spanners was [C-86] dealing with the construction of spanners for sets of points in the Euclidean plane. The [C-86] paper proved that any set of points in the plane has a $\sqrt{10}-$spanner with an edge density $|E|/|V|$ of less than 3. This result has implications for the problem of motion planning in the plane. Spanners for general graphs were first introduced in [PU-89], 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 [PU-89] to construct a new type of synchronizer for an asynchronous network. Spanners are also used to construct efficient routing tables [PU-88]. Each node of such a network is a processor with some local memory. The processors in the network communicate, thus an efficient routing scheme should be able to use a short path to send messages through the network while keeping the routing information stored in the processors as succinct as possible. In [PU-88] a technique based on spanners is developed for this goal. In [RL-95] it is suggested that spanners may be used as network topologies. If one has an expensive desired topology, often a sparse or low-weight (and therefore less expensive) spanner can be substituted, retaining a similar structure with only a slight increase in communication cost. Finally, in [ABP-92] spanners are used to model broadcast operations. Sparse spanners can be used for efficient broadcast with respect to the following parameters: the maximum number of messages sent, the total cost of the messages and the worst-case delay incurred for any destination.

For this, and other applications, it is desirable that the spanners be as *sparse* as possible, i.e., have as few edges as possible. 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 [PS-89] 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 we are unlikely to find an exact solution for the sparsest $k-$spanner problem even when $k = 2$. In [C-94] this result is extended to hold for any integer $k \geq 3$, even when restricted to bipartite graphs.

Recently, in [VRM$^+$97] it was shown that the $k-$spanner problem is hard even in other restricted cases. More specifically, it was shown there that the problem is $NP-$complete even when restricted to chordal graphs, for any $k \geq 2$ (a graph is chordal if any cycle of length at least 4 has a chord, namely, has an edge connecting two non-adjacent vertices in the cycle). They also show that the problem is $NP-$complete on split graphs, for $k = 2$ (a split graph contains a clique C and an independent set I, with arbitrary connections between the vertices of C and I). The paper also gives relatively simple proofs that the problem is $NP-$complete on bipartite graphs, as well as on bounded-degree graphs (graphs where the degree is bounded by some fixed constant). This somewhat simplifies the proofs in [C-94, CK-94]. Finally, in [BH-97] the problem is proven $NP-$complete on planar graphs for any $k \geq 5$.

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 [PS-89] it is shown that every $n-$vertex graph G has a polynomial time constructible $(4k+1)-$spanner with $O(n^{1+1/k})$ 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 [PS-89] in the sense that there are graphs with $S_{4k+1}(G) = \Omega(n^{1+\Omega(1/k)})$.

The results of [PS-89] were improved and generalized in [ADDJ-90, CDNS-92] to the weighted case, in which there are non-negative weights associated with the edges, and the distance between two vertices is the weighted distance. Specifically, it is shown in [ADDJ-90] that given an $n-$vertex graph and an integer $k \geq 1$, there is a polynomially constructible $(2k+1)-$spanner G' such that $|E(G')| < n \cdot \lceil n^{\frac{1}{k}} \rceil$. They also show that the weight (sum of weights of the edges) of the constructed spanner is close to the weight of the minimum spanning tree. More specifically, $w(G') \leq O(n^{O(1/k)}) \cdot w(MST)$ where $w(MST)$ is the weight of a minimum spanning tree.

The algorithms of [ADDJ-90, PS-89] provide us with *global* upper bounds for sparse $k-$spanners, i.e., general bounds that hold true *for every graph*. However, it may be that considerably sparser spanners exist for specific graphs. 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* which yield near-optimal *local* bounds applying to the specific graph at hand, by exploiting its individual properties.

The only logarithmic ratio approximation algorithm known for constructing sparse spanners exists for the 2−spanner problem. Specifically, in [KP-92] an $O(\log(|E|/|V|))$ approximation algorithm is given for the 2−spanner problem. That is, given a graph $G = (V, E)$, the algorithm generates a 2−spanner $G' = (V, E')$ with $|E'| = O\left(S_2(G) \cdot \log \frac{|E|}{|V|}\right)$ edges. No *small ratio* (for example, polylogarithmic ratio) approximation algorithm is known, even for the 3−spanner problem. However, the results in [ADDJ-90] indicate that any graph admits a 3−spanner with girth (minimum length cycle) 5. Now, every graph of girth 5 has $O(n^{3/2})$ edges. This "global" result can be considered an $O(\sqrt{n})$ "approximation" algorithm for the k−spanner problem, for $k \geq 3$. Note that this bound cannot, in general, be improved. Consider a projective plane of order q (cf. [B-86]). A projective plane of order q is a $(q+1)$-regular bipartite graph with $n = q^2 + q + 1$ vertices on each side, with the additional property that every two vertices on the same side share *exactly* one neighbor. Such a structure is known to exist, for example, for prime q. Clearly, the girth of this graph is 6. Therefore, the only 3− (and 4−) spanner for the graph is the graph itself. Furthermore, this graph contains $\theta(n^{3/2})$ edges.

In this paper we first prove that the (unweighted) 2−spanner problem is NP−hard to approximate even when restricted to 3−colorable graphs, within $c \log n$−ratio for some constant $c < 1$. This matches the approximation ratio of $O(\log n)$ of [KP-92]. Hence the algorithm in [KP-92] is the best possible algorithm for approximating the 2−spanner problem, up to constants.

We also show that the (unweighted) k−spanner problem is hard *to approximate* within small ratio, even when restricted to bipartite graphs. Specifically, we prove that for every fixed integer $k \geq 3$, there exists a constant $c < 1$ such that it is NP−hard to approximate the k−spanner problem on bipartite graphs within ratio $c \log n$ (the constant c depends on the constant k.) This improves the NP−hardness result from [C-94] for fixed values of k. It also improves, for fixed k, the NP−hardness result for general graphs.

Remark: In fact we prove that for any $k = o(\log n)$ (not necessarily fixed) there exists a constant $c < 1$ such that the k−spanner has no $c \cdot \log n/k$−ratio approximation, unless $NP \subseteq DTIME(n^{O(k)})$. Indeed, for $k = \log n$, the $\log n$−spanner problem *can* be approximated within ratio $O(1)$ since, as mentioned earlier, there is always a $\log n$−spanner with $O(n)$ edges. Thus, it is mainly interesting to prove hardness results for $k = o(\log n)$.

Finally, we define a new weighted version of the spanner problem which we believe to be natural. In this version, called the edge-weighted k−spanner problem, each edge $e \in E$ has a positive length $\ell(e)$ but also a non-negative weight $w(e)$. The goal is to find a k−spanner G' with a low weight. Thus, in the first place, the graph G' should have stretch factor

k, namely, the $\ell-$distance of every pair of vertices u and v in G' should increase by no more than a factor of k. Second, the sum of weights $w(e)$ of the edges in G' should be as small as possible. For example, in the unweighted case, $\ell(e) = w(e) = 1$ for every edge e. Also, in the standard weighted case, considered in [ADDJ-90, CDNS-92], for every edge e, $w(e) = \ell(e)$. The more general version of the problem is useful in the following case. Given a desired network topology as described above, for a given link $e = (v, u)$ the function $\ell(e)$ may represent the delay in sending messages over e, while the $w(e)$ function may represent the cost incurred in establishing and maintaining a connection between v and u. In the generalized version of the problem it is possible to deal with the case that ℓ and w are not necessarily identical. The result for the weighted case may give some insight for proving a possible similar result for the unweighted case.

For the edge-weighted $k-$spanner problem we have the following results. We consider the case where $\ell(e) = 1$ for every edge and w is arbitrary. For $k = 2$, this version of the problem admits an $O(\log n)-$ratio approximation. However, for every $k \geq 5$, the problem has no $2^{\log^{1-\epsilon} n}-$ ratio approximation, unless $NP \subseteq DTIME(n^{\text{polylogn}})$, for any $\epsilon > 0$. This later result follows by a reduction from one-round two-prover interactive proof system.

We note that ours are the first results on the hardness of approximating the spanner problem.

2 Preliminaries

First, recall the following alternative definition of spanners:

Lemma 2.1 [PS-89] *The subgraph $G' = (V, E')$ is a $k-spanner$ of the graph $G = (V, E)$ iff $dist(u, v, G') \leq k$ for every $(v, u) \in E$.* ∎

Thus the (unweighted) sparsest $k-$spanner problem can be restated as follows: We look for a minimum subset of edges $E' \subset E$ such that every edge e which does not belong to E' lies on a cycle of length $k+1$ or less, with edges which do belong to E'. In this case we say that e is *spanned* in E' (by the remaining edges of the cycle).

In the following we say that two (independent) sets C and D are *cliqued* if every vertex in C is connected to every vertex in D; thus C and D induce a complete bipartite graph. We say that C and D are matched if $|C| = |D|$ (i.e., C and D are of the same size) and every vertex in C has a unique neighbor in D (that is, the two sets induce a perfect matching).

The set-cover problem: For our purpose it is convenient to state the set-cover problem as follows: The input for the set-cover problem consists of a bipartite graph $G(V_1, V_2, E)$ in which the edges cross from V_1 to V_2 (that is, V_1 and V_2 contain no internal edges) with n vertices on each side. The goal is to find the smallest possible subset $S \subseteq V_1$ that *covers* V_2, namely, the smallest S such that every vertex in V_2 has a neighbor in S. We assume throughout that the entire set V_1 covers V_2, otherwise the problem has no solution.

The following result is known [RS-97]. This result followed two results by [LY-93] and [F-96]. These two results were proven under a weaker assumption, i.e., that $NP \not\subseteq DTIME(n^{O(\log \log n)})$.

Theorem 2.2 [RS-97] *There exists a constant $c < 1$ such that it is $NP-$hard to approximate the set-cover problem within ratio $c \ln n$.* ▮

We require the following lemma regarding a restrictive case of the set-cover problem. Consider the $n^\rho-$degree-bounded-set-cover ($n^\rho - DBSC$) problem, which is the set-cover problem in which Δ, the maximum degree of any vertex in $V_1 \cup V_2$, is bounded by n^ρ for some (fixed) specified $0 < \rho < 1$. The usual greedy algorithm ([J-74, L-75]) gives a $\rho \cdot \ln\ n-$ratio approximation algorithm for the $n^\rho - DBSC$ problem. On the other hand, we have the following:

Lemma 2.3 *It is $NP-$hard to approximate the $n^\rho - DBSC$ problem within $c \cdot \rho \cdot \ln n$ ratio, where c is the same constant as in Theorem 2.2.*

Proof: Let G be an instance of the set-cover problem. Denote $\mu = \frac{1}{\rho} - 1$. Now, construct a new instance of the $n^\rho - DBSC$ problem by taking n^μ copies of G. Call this instance of the problem $\tilde{G}$. The number of vertices in each side is now $\tilde{n} = n^{1/\rho}$. Thus, the maximum degree in $\tilde{G}$ is bounded by $\tilde{n}^\rho$, hence the new cover problem is indeed an $n^\rho - DBSC$ instance. A cover in $\tilde{G}$ would consist of n^μ covers, one for each copy of G. Thus the optimum size of the cover in $\tilde{G}$ is exactly $n^\mu \cdot s^*$, where s^* is the size of the optimum cover of G.

Recall that the set-cover problem cannot be approximated within ratio $c \ln n$. If the $n^\rho - DBSC$ problem can be approximated within a ratio better than $c \cdot \rho \cdot \ln n$, then the algorithm will produce a collection of n^μ covers of G, whose collective size is bounded by:

$$c \cdot \rho \cdot \ln \tilde{n} \cdot n^\mu \cdot s^* = c \cdot \ln n \cdot n^\mu \cdot s^*$$

By an averaging argument, one of these covers contains at most $c \cdot \ln n \cdot s^*$ vertices, contradicting Theorem 2.2. ▮

In the sequel we estimate the probability of the deviation of some random variables from their expectation, using the Chernoff bound [C-52]. The Chernoff bound will be used later in order to bound the expected size of some collection $\mathcal{C}_c \cup \mathcal{D}_c$ of "undesirable" cycles from above .

In the next lemma we use $\exp(\delta)$ for e^{δ}. The following holds true for any $\delta > 0$.

Lemma 2.4 [C-52] *Let $X_1, X_2, \ldots, X_m$ be independent Bernoulli trials with $\mathbb{P}(X_i = 1) = p_i$. Let $X = \sum_{i=1}^{m} X_i$ and $\mu = \sum_{i=1}^{m} p_i$. Then*

$$\mathbb{P}(X > (1+\delta)\mu) < \left[\frac{exp(\delta)}{(1+\delta)^{(1+\delta)}}\right]^{\mu}. \quad \blacksquare$$

The following is derived from the Chernoff bound (c.f. page 72 in [MR-95]).

Suppose that $\mu = \Omega(n^{\rho})$ for $0 < \rho < 1$. Then for any constant c, there is a constant c_1 such that $\mathbb{P}(X > \mu + c_1\sqrt{\mu \cdot \log n}) < 1/n^c$.

For example, we see that $\mathbb{P}(X > 2 \cdot \mu) \leq 1/n^3$, for a large enough n.

Corollary 2.5 *Suppose that $\mu = n^{\rho}$ as above. Then for any (fixed) $c > 0$ for a large enough n, $\mathbb{P}(X > 2 \cdot \mu) < 1/n^c$.* $\blacksquare$

3 A hardness result for any $k \geq 5$ in the unweighted case

In this section we prove our hardness result for the unweighted case. Due to technical reasons, the proof is divided into three parts. First we prove the result for $k \geq 5$. Then we prove the result for $k = 3$ and $k = 4$. These results show hardness of approximation on bipartite graphs. Finally, we prove the result for $k = 2$ (this hardness result is for $3-$colorable graphs).

We consider the hardness of approximating the $k-$spanner problem for constant odd k, $k = 2t + 1$ and $t \geq 2$. The constructed graph is bipartite, thus it contains no odd cycles. It follows that any $(2t + 2)-$spanner in such a graph is a $(2t + 1)-$spanner as well, since the graph has no $(2t + 3)-$cycles. Hence, the lower bound on the approximability for $k = 2t + 1$ on bipartite graphs will automatically imply a lower bound for $k = 2t + 2$.

3.1 Intuition

Before describing the construction, we discuss some intuition. Consider the graph $G(V_1, V_2, E)$ of the set-cover problem. Suppose we form a complete bipartite graph Bip by introducing a new set A of n vertices and connecting all the vertices in A to all the vertices in V_1; hence the graph Bip denotes this bipartite clique with A and V_1 in the two sides. Now, add two new vertices h_1 and h_2. Connect h_1 to all the vertices in A and h_2 to all the vertices in V_1, and add an edge connecting h_1 and h_2. Consider now the structure of a sparse spanner. In any spanner containing "few" edges, one of the goals is to span the edges of Bip. Note, however, that it is now possible to add the $O(n)$ edges touching h_1 and h_2 to the spanner, hence spanning the edges of Bip and of h_1 and h_2 with a maximum stretch 3.

Moreover, suppose that each vertex in the set A is connected by a collection $\mathcal{P}$ of appropriate paths to every vertex of V_2. Each such path would have length at most $2t$. The vertices in these paths all have degree 2, except for two vertices in the middle of each path. (The middle vertices in these paths will have high degrees in order to ensure, that every vertex of A is connected by a path in $\mathcal{P}$ to every vertex in V_2.) The paths in $\mathcal{P}$ contain no vertices of V_1 or h_1 or h_2.

Next, suppose we can prove that in any spanner close to the optimal, most of the paths in $\mathcal{P}$ have a missing edge.

Thus, in order to find an alternative path for each such missing edge, with a length $2t+1$ or less, one must connect every vertex of A by a path to every vertex of V_2 via the vertices of V_1 (closing a cycle of length at most $2t+2$ with the missing edge). Namely, each vertex of A must be connected to each vertex of V_2 by a path of length 2, which goes through V_1. Given a vertex $a \in A$, we see that the collection S_a of neighbors of a in V_1, is a cover of V_2. In other words, each vertex in V_2 has at least one neighbor in S_a. This holds true for every vertex $a \in A$. Therefore the number of edges needed in the spanner will be roughly $n \cdot \bar{s}$, where $\bar{s}$ is the average size of all the sets S_a.

It is therefore convenient for the algorithm to find a small cover S and connect each vertex in A to S.

The following is a serious technical problem: Consider the case $k = 3$. Can we simply choose the collection of paths $\mathcal{P}$ to be all the edges connecting a vertex in A to a vertex in V_2? This is the case where the collection of paths $\mathcal{P}$ is simply a bipartite clique connecting A and V_2. Note, however, that this construction does not give a hardness result for approximating spanners because a complete bipartite graph can "span itself", i.e., span all its bipartite edges without the assistance of the edges of G. In other words, one can add any edge (a, v_2), $a \in A$ and $v_2 \in V_2$ to the spanner, and then add all the edges connecting a to the vertices in V_2 to the spanner, and all the edges connecting v_2 to the vertices in A (this would give an alternative path with a length 3 for every edge in the bipartite clique). Clearly, we can thus span all the edges of the bipartite clique by choosing a subset of those edges with size $O(n)$.

Thus the edges of a bipartite clique can "span themselves" without the assistance of the graph G induced by $V_1 \cup V_2$. Hence, a better collection $\mathcal{P}$ of paths is required such that in any spanner close to the optimum size, most paths in $\mathcal{P}$ would have a missing edge, such that the only efficient way of spanning all the missing edges of the paths of $\mathcal{P}$ would be to use the edges of G.

3.2 The construction for $k \geq 5$

In this subsection we describe the construction for the $k-$spanner problem in the case of *constant* odd $k \geq 5$. Let $k = 2t+1, \quad t \geq 2$.

Let $\epsilon > 0$ be a *constant* satisfying:

$$1 - \frac{1}{2t+1} < \epsilon < 1 - \frac{1}{2t+2} \tag{1}$$

Let $\delta = (2t+1)(1-\epsilon)$. We note that by definition of ϵ:

$$\delta = (2t+1) \cdot (1-\epsilon) < \frac{2t+1}{2t+1} = 1,$$

and

$$\delta = (2t+1) \cdot (1-\epsilon) > \frac{2t+1}{2t+2} = 1 - \frac{1}{2t+2} > \epsilon,$$

hence, $\epsilon < \delta < 1$. In addition, let δ_1 be a constant satisfying:

$$\max\{\delta, 1-\epsilon/3\} < \delta_1 < 1 \tag{2}$$

We begin the construction with an instance $G(V_1, V_2, E)$ of the $n^{1-\delta_1} - DBSC$ problem. That is, the maximum degree in G is bounded by $n^{1-\delta_1}$. The construction is composed of two main ingredients: the *fixed part* and the *gadgets part*. The fixed part contains the graph G and the set A. We clique A and V_1 as explained above (we connect each vertex $a \in A$ to each vertex $v_1 \in V_1$). Furthermore, we have two special vertices: $h_1, \ h_2$. The vertices h_1 and h_2 are joined by an edge. Then, h_1 is connected to each vertex of A, and h_2 is connected to each vertex of V_1. The role of h_1, h_2 is to span the edges connecting the vertices of A to the vertices of V_1, with stretch factor 3.

Secondly, we describe the "gadgets part" of the construction. This part of the construction is intended, as described above, to connect each vertex in A to each vertex in V_2 by a path of length $2t$.

The construction of the gadget involves randomization. We note that the construction can easily be derandomized. The gadget is a union of $4 \cdot \ln n \cdot n^{\epsilon}$ different gadgets.

For $1 \leq i \leq 4 \ln n \cdot n^{\epsilon}$ take the following steps:

- Define sets $A_1^i, A_2^i, \ldots, A_t^i$, each of cardinality n. The sets corresponding to different i are disjoint.

  For each i, the set A is matched (connected in a perfect matching) to the set A_1^i. The set A_1^i is matched to the set A_2^i, and in general, the set A_j^i is matched to the set A_{j+1}^i, for every $1 \leq j \leq t-1$.

- Define sets $V_{2,1}^i, V_{2,2}^i, \ldots, V_{2,t-1}^i$, each the size of n. The sets corresponding to different i are disjoint.

  The sets V_2 and $V_{2,1}^i$ are matched. In addition, the sets $V_{2,j}^i$ and $V_{2,j+1}^i$ are matched, for each $1 \leq j \leq t-2$. Call all edges of the perfect matchings (as well as those above which match A_j^i with A_{j+1}^i) "matching edges." These are the edges marked M in Figure 1.

- Finally, for every vertex $a_t^i \in A_t^i$ and every vertex $v_{2,t-1}^i \in V_{2,t-1}^i$, put an edge between a_t^i and $v_{2,t-1}^i$ randomly and independently, with probability $1/n^{\epsilon}$. Let R_i denote the collection of random edges resulting among the two sets $V_{2,t-1}^i$ and A_t^i.

See Figure 1 for an example in the case $k = 5$. We note that for each vertex $a \in A$ and gadget i, there is a unique vertex $a_t^i \in A_t^i$ which is connected to the vertex a via a path which goes entirely through the matching edges. For this reason, we throughout the paper call a_t^i a *matched copy* of a. Similarly, every vertex $v_2 \in V_2$ has a unique matched copy $v_{2,t-1}^i \in V_{2,t-1}^i$ in any gadget i.

It is easy to verify that the constructed graph is bipartite.

3.3 Cycles containing R_i edges

In this section we move towards proving that there exist spanners with a number of edges close to the minimum possible which contain none of the edges of R_i. We are particularly interested in the cycles with lengths $2t + 2$ or less and containing edges of R_i, because these cycles can help span the edges of R_i. In other words, given a short enough cycle containing an edge $e \in R_i$, we can choose all the other edges in the cycle, except e and span e (in the sense that we now have an alternative path with a length of $2t + 1$ or less for e).

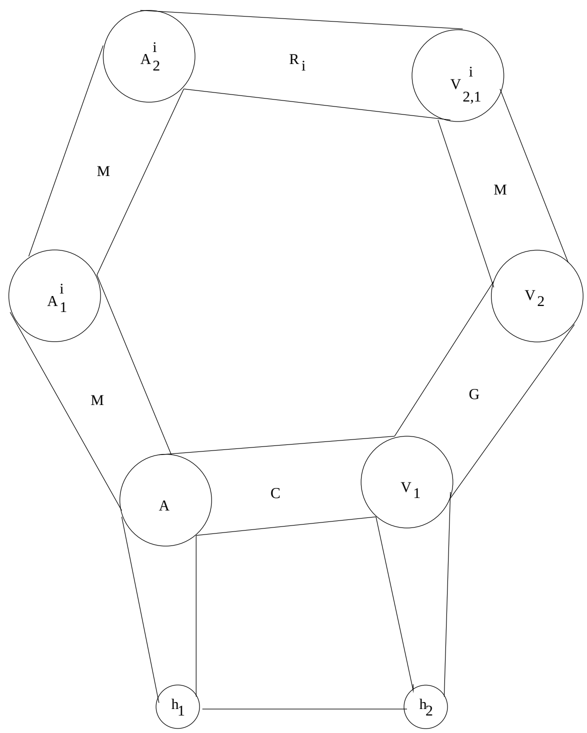

Figure 1: The fixed part and the ith part of the gadget for $k = 5$. M indicates a matching. C indicates a bipartite clique. R_i indicates the collection of edges included by the randomized choice.

One point we should prove, for example, is that the edges of R_i cannot span themselves efficiently. Unlike a bipartite clique, there is no efficient way of choosing a small subset of the R_i edges so that these chosen edges help span "many" of the other unchosen R_i edges. We prove this simply by showing that the number of cycles containing only edges of R_i is small. Similarly, we count the number of other "undesirable" cycles.

It would follow that if the edges of R_i cannot span themselves, each edge of R_i should be spanned using a path from A_t^i to $V_{2,t-1}^i$, which passes through A, V_1 and V_2. Denote by $\mathcal{B}$ the collection of all such paths. More specifically, a path $P \in \mathcal{B}$ starts at a matched copy $a_t^i \in A_t^i$ of a vertex $a \in A$. Then P goes to the sets $A_{t-1}^i, A_{t-2}^i, \ldots, A_1^i$ via the matching edges. P then goes to a, then to a vertex $v_1 \in G$, and then to a neighbor (in G and $\bar{G}$) $v_2 \in V_2$ of v_1. Finally, the path continues to the matched copy $v_{2,t-1}^i$ of v_2, via the matching edges. (Note that the path does not pass through h_1 and h_2). Note that the edge $(a_t^i, v_{2,t-1}^i)$ (if present) closes a cycle with length exactly $2t+2$ with P. Let $\mathcal{B}_c$ denote the collection of such cycles. Call each path $P \in \mathcal{B}$ for short (respectively, each cycle $C \in \mathcal{B}_c$) a $\mathcal{B}-$path (respectively, a $\mathcal{B}_c-$cycle.) In addition to the $\mathcal{B}-$cycles, we show that the other cycles are either too long and cannot play a part in a spanner, or, among the other short cycles, there are only few such cycles and these cycles can have only a limited impact on the number of edges in a sparse spanner.

Let us first make some observations regarding short cycles containing edges of R_i in the construction. In addition to the $\mathcal{B}_c-$cycles we have the following:

1. First, one may try to construct a cycle which contains R_i edges, and remains in the "left side" of the construction. More specifically, this is a cycle which begins at a vertex $v_{2,t-1}^i \in V_{2,t-1}^i$, continues to two matched copies a_t^i and b_t^i, of $a, b \in A$, then goes in parallel via the matching edges to a and b, and closes at a mutual neighbor $v_1 \in V_1$ of a and b.

   Note, however, that the length of the path leading each vertex in $V_{2,t-1}^i$ to V_1 through A, is $t+2$. Hence, any cycle containing two edges of R_i, two vertices of A, and a vertex of V_1 has length $2t+4$. Such a cycle cannot serve as an alternative path for an edge in R_i.

2. Secondly, consider a cycle containing edges from R_i and R_l, for $l \neq i$ (i.e., via two different gadgets.) Such a cycle would contain one or two edges from R_i, and the same number of edges from R_l. The minimum length of such a cycle is exactly $4t$. For example, one can construct such a cycle in a parallel route using two different matchings, from R_i to V_2 and then continue in parallel to R_l. Note, however, that

$4t > 2t+2$. This follows since $t > 1$ (this is precisely the place where the construction fails for $k = 3$ and $t = 1$).

We now see that the only appropriate cycles (having a length bounded by $2t+2$) containing an edge of R_i are the following:

(B) the $\mathcal{B}_c-$cycles;

(C) cycles containing only edges of R_i. Name the collection of such cycles (in all the gadgets) $\mathcal{C}_c$ and call such cycles $\mathcal{C}_c-$cycles;

(D) cycles containing edges of R_i and G, and none of the vertices of A. More specifically, one can choose a vertex $v_1 \in V_1$. Choose two neighbors v_2, u_2 of v_1 in V_2, move in a parallel route using the matching edges to the two matched copies $v^i_{2,t-1}$ of v_2 and $u^i_{2,t-1}$ of u_2 in $V^i_{2,t-1}$, and finally close the cycle using a mutual neighbor $a^i_t \in A^i_t$, of $v^i_{2,t-1}$ and $u^i_{2,t-1}$. Name the collection of such cycles (in all the gadgets) $\mathcal{D}_c$ and call such cycles $\mathcal{D}_c-$cycles.

In the following lemmas we bound the (expected) size of $\mathcal{C}_c$ and $\mathcal{D}_c$.

We first deal with $\mathcal{C}_c$. Call the graph induced by the R_i edges G_{R_i}.

Lemma 3.1 *The expected size of $\mathcal{C}_c$ is bounded by $\tilde{O}(n^{1+\delta})$.*

Proof: Any cycle in R_i, with a length of $j \leq 2t+2$ or less includes j vertices of G_{R_i}, $j/2$ on each side. In addition, any choice of subset of $j/2$ vertices in each side of G_{R_i}, for $j \leq 2t+2$, may lead to a cycle of the appropriate length. (We note that when choosing, one must take the order of the vertices into account). The number of choices (counting the order) of $j/2$ vertices in a given side is less than $n^{j/2}$.

The probability of a cycle with a length of j "surviving" the random choice (i.e., the probability that all its edges are included by the random choice) is $1/n^{j\cdot\epsilon}$. Recall that, by definition, $\delta = (2t+1)(1-\epsilon)$. By defining an appropriate indicator variable for any candidate cycle, we find that the expected size of $\mathcal{C}_c$ is clearly bounded by:

$$4 \ln n \cdot n^{\epsilon} \cdot \sum_{j=4}^{2t+2} n^{j(1-\epsilon)} \tag{3}$$

The largest term in the sum (3) is with $j = 2t+2$. The number of terms in the sum is less than $2 \cdot t$. Hence we get that the expected size of $\mathcal{C}_c$ is less than:

$$4 \ln n \cdot 2 \cdot t \cdot n^{\epsilon} \cdot n^{(2t+2)(1-\epsilon)} = 4 \ln n \cdot 2 \cdot t \cdot n \cdot n^{(2t+1)(1-\epsilon)}$$

The required claim follows by the definition of δ. ▮

Next we must bound the expected size of $\mathcal{D}_c$. First, however, we need the following claim derived directly from the Chernoff Bound [C-52].

For every two vertices $v_{2,t-1}^i$ and $u_{2,t-1}^i$ in $V_{2,t-1}^i$, let $m(v_{2,t-1}^i, u_{2,t-1}^i)$ be the number of mutual neighbors $v_{2,t-1}^i$ and $u_{2,t-1}^i$ have in A_t^i. Let m denote the maximum of $m(v_{2,t-1}^i, u_{2,t-1}^i)$, taken over all values of i, and pairs of vertices in $V_{2,t-1}^i$. We bound m as follows:

Claim 3.2 *For a large enough n, with probability of at least $1 - 1/n^2$, $m \leq 2 \cdot n^{1-2\epsilon}$.*

Proof: For every two vertices $v_{2,t-1}^i$ and $u_{2,t-1}^i$, let $Neig(v_{2,t-1}^i, u_{2,t-1}^i)$ be the random variable which counts the number of mutual neighbors in A_t^i.

The random variable $Neig(v_{2,t-1}^i, u_{2,t-1}^i)$ has a binomial distribution with n trials, and a probability of success $1/n^{2\epsilon}$. This is because of the following reasons.

There are n candidate mutual neighbors of $v_{2,t-1}^i$ and $u_{2,t-1}^i$ in A_t^i. Let $N_{a_t^i}$ denote the event that a_t^i is a mutual neighbor of the two vertices $v_{2,t-1}^i$ and $u_{2,t-1}^i$. For each such candidate $a_t^i \in A_t^i$, the two corresponding edges (which belong to R_i with probability $1/n^\epsilon$ each, independently of each other), should appear in R_i. Moreover, the pairs of edges corresponding to different a_t^i and b_t^i are disjoint. This tells us that the two events $N_{a_t^i}$ and $N_{b_t^i}$ are *independent.*

In summary, the expectation of $Neig(v_{2,t-1}^i, u_{2,t-1}^i)$, $\mathbb{E}(Neig(v_{2,t-1}^i, u_{2,t-1}^i))$, is $n^{1-2\epsilon}$. Thus according to Corollary 2.5, for every n large enough, $\mathbb{P}(Neig(v_{2,t-1}^i, u_{2,t-1}^i) > 2 \cdot n^{1-2\epsilon}) < 1/n^5$.

It follows that m is bounded by $2 \cdot n^{1-2\epsilon}$ with a probability of at least $1 - 1/n^2$. This is because the number of pairs $v_{2,t-1}^i$ and $u_{2,t-1}^i$ in any gadget is less than n^3, as the number of gadgets is $o(n)$ and there are n^2 pairs of vertices $(v_{2,t-1}^i, u_{2,t-1}^i)$ in each gadget. ▮

We now bound the size of $\mathcal{D}_c$. Using the above description (see (D)), we count the number of these cycles as follows: Choose a vertex v_1 in V_1 (n options). Select two neighbors v_2 and u_2 of v_1. Since the maximum degree in G is bounded by $n^{1-\delta_1}$, the number of possible v_2 and u_2 pairs is bounded by $n^{2-2\cdot\delta_1}$. Choose a gadget ($\tilde{O}(n^\epsilon)$ options) and go using the matching edges to the two matched copies $v_{2,t-1}^i$ and $u_{2,t-1}^i$ of v_2 and u_2. Select a mutual neighbor a_t^i of $v_{2,t-1}^i$ and $u_{2,t-1}^i$. According to the above corollary, there is a high probability that the number of such a_t^i neighbors is bounded by $O(n^{1-2\epsilon})$.

Hence, with high probability, the size of $\mathcal{D}_c$ is bounded by

$$n \cdot n^{2-2\cdot\delta_1} \tilde{O}(n^\epsilon) O(n^{1-2\epsilon}) \quad = \quad \tilde{O}(n^{4-\epsilon-2\delta_1}) \tag{4}$$

$$= \quad o(n^{1+\delta_1})$$

The last inequality holds since $1 - \epsilon/3 < \delta_1$ by definition.

We summarize (using the fact that by definition $n^{\delta} = o(n^{\delta_1})$).

Corollary 3.3 *The expected size of $\mathcal{C}_c \cup \mathcal{D}_c$ is bounded by $o(n^{1+\delta_1})$.* ∎

3.4 The lower bound

In this section we prove the lower bound using Corollary 3.3. First we need the following technical lemma, which states that all the vertices of A are connected to all the vertices of V_2 via a path which goes entirely through the matching edges and R_i edges. For the sequel, call this type of path a "proper" path.

Lemma 3.4 *With probability at least $1 - 1/n^2$ for every pair of vertices $a \in A$ and $v_2 \in V_2$, there exists a gadget i such that the edge $(a_t^i, v_{2,t-1}^i)$ was included by the random choice. Hence with probability of at least $1 - 1/n^2$, each (a, v_2) pair is connected via a proper path.*

Proof: Consider any matched copy a_t^i of a and a matched copy $v_{2,t-1}^i$ of v_2. An edge is present between these two vertices with probability $1/n^{\epsilon}$. Therefore the probability that for any i, all these edges are missing, is bounded by:

$$(1 - \frac{1}{n^{\epsilon}})^{4 \cdot \ln n \cdot n^{\epsilon}} \leq \frac{1}{n^4} \tag{5}$$

Thus, for a given pair a and v_2, the probability that a and v_2 are not connected by a proper path is less than $\frac{1}{n^4}$. By summing up these probabilities, the probability that any of these two vertices will not be connected by a proper path is bounded by $1/n^2$ as required. This follows since there are n^2 pairs of vertices (a, v_2) with $a \in A$, $v_2 \in V_2$. ∎

In the sequel we assume that the total size of $\mathcal{C}_c$ and $\mathcal{D}_c$ is indeed bounded by $o(n^{1+\delta_1})$ as stated in Corollary 3.3, and that the implied part in Lemma 3.4 holds firm. On the one hand, the required properties can be guaranteed with fixed probability using the Markov inequality (cf., [MR-95]). This inequality is needed primarily in order to bound the size of $\mathcal{C}_c$. Thus, we may find that, say, with a probability of at least 9/10, the quantity $|\mathcal{C}_c| + |\mathcal{D}_c|$ is bounded by $o(n^{1+\delta_1})$, while Lemma 3.4 holds as well. In turn, one can derandomize the construction using the method of conditional expectation [S-87] in time $n^{O(k)}$. In other words, it is possible to deterministically construct a structure with the desired properties in polynomial time, for fixed k.

Using these assumptions, we can state our two main claims. Let s^* be the size of an optimum cover in G. Note that since G is an instance of the $(1-\delta_1)-$set-cover problem, $s^* \geq n^{\delta_1}$.

Lemma 3.5 *For a large enough n, the instance $\bar{G}$ of the $(2t+1)-$spanner problem admits a $(2t+1)-$spanner with no more than $2 \cdot s^* \cdot n$ edges.*

Proof: Introduce the edges touching h_1, h_2 and the edges of G into the spanner. The number of edges added so far is $O(n^{2-\delta_1})$. We have thus spanned the edges of the fixed part of the construction. The edges touching h_1 and h_2 and the edges of G are spanned with stretch factor 1, while the edges connecting A and V_1 are spanned with stretch factor 3.

Bring all the matching edges into the spanner. The number of edges added here is $\tilde{O}(t \cdot n^{1+\epsilon}) = \tilde{O}(n^{1+\epsilon})$ (the last equality is valid for fixed t or even for $t = O(\log\ n)$ as in our case).

It only remains to span the edges of R_i. Choose a cover $S^* \subseteq V_1$ of V_2 of size s^*. Connect all the vertices of A to all the vertices of S^*. It is easy to check that all the edges of R_i are spanned by an alternative path $P \in \mathcal{B}$ with length exactly $2t+1$. Note that the number of edges in this spanner is $s^* \cdot n + \tilde{O}(n^{1+\epsilon}) + O(n^{2-\delta_1})$. Now, since $n^{\epsilon} = o(n^{\delta_1})$, $\delta_1 > 1/2$ and $s^* \geq n^{\delta_1}$, the claim follows for large enough n. ∎

Lemma 3.6 *For all $l > 0$ and large enough n, given a $(2t+1)-$spanner $H(V, E')$ in $\bar{G}$ with no more than $l \cdot n$ edges, there exists a (polynomially constructible) cover S of V_2 of size $2l$ or less.*

Proof: We show how to modify H so that all the edges of R_i are spanned by a $\mathcal{B}-$path. In this modification, we somewhat increase the number of edges in H; however we prove that the increase is slight. Once all the edges of R_i are spanned by a $\mathcal{B}-$path, we show how to deduce a small cover of V_2 from this modified construction.

Starting with H, change to a new $(2t+1)-$spanner H' as follows: First, add all the edges touching h_1 and all the edges touching h_2 (if they are not already there) to H. Similarly, add all the matching edges and the edges of G.

Now consider the edges in $R_i \cap E'$, i.e., the edges of R_i, which are in the spanner. Remove each such edge $(a_t^i, v_{2,t-1}^i)$ joining a matched copy a_t^i of a to a matched copy $v_{2,t-1}^i$ of v_2. In the present situation, several of the R_i edges may be unspanned. Those are edges in R_i which belong to a $\mathcal{C}_c-$cycle or a $\mathcal{D}_c-$cycle. Recall, however, that we have already shown that there are only few such cycles. This immediately implies that only few of the edges in R_i are not spanned now. Add an arbitrary alternative $\mathcal{B}-$path $P \in \mathcal{B}$ for each such edge.

Note that such a path exists since we assume that V_1 covers V_2. The resulting graph H' is still a legal $(2t+1)-$spanner.

Note that (aside from the matching edges) for every cycle in $\mathcal{C}_c \cup \mathcal{D}_c$ we may have added two additional edges to H', one joining a vertex $a \in A$ and a vertex $v_1 \in V_1$, and the other joining the vertex v_1 to a vertex $v_2 \in V_2$. These are the two relevant edges from the $\mathcal{B}-$path P. Let num denote the number of edges in the new resulting spanner H'. According to Corollary 3.3, num is bounded above by

$$num \leq l \cdot n + \tilde{\theta}(t \cdot n^{1+\epsilon}) + o(n^{1+\delta_1}) + O(n^{2-\delta_1}). \tag{6}$$

Note that the only way now to span the R_i edges is by using a $\mathcal{B}-$path.

Recall that, according to Lemma 3.4, for every vertex $a \in A$ and $v_2 \in V_2$ there are matched copies $v^i_{2,t-1}$ and a^i_t of v_2 and a, which are neighbors in R_i. Since the edge $e = (v^i_{2,t-1}, a^i_t)$ is missing from H', we must span this edge via a $\mathcal{B}-$path P. It follows that a is connected to a neighbor $v_1 \in V_1$ of v_2. In other words, the set S_a of neighbors of a in V_1 in the spanner H' is a *cover* of V_2 in G.

On the one hand, note that the number of edges num in H' is bounded below by

$$num \geq \sum_{a \in A} |S_a| \geq n \cdot s^* \geq n^{1+\delta_1}. \tag{7}$$

which implies that $num \geq n^{1+\delta_1}$. Now, since $\epsilon < \delta_1$ and $\delta_1 > 1/2$, we combine Equations 6 and 7 to deduce that

$$n^{1+\delta_1} \leq l \cdot n + o(n^{1+\delta_1}). \tag{8}$$

From Equation 8, we deduce that $2l \geq n^{\delta_1}$ for a large enough n.

On the other hand:

$$num \geq \sum_{a \in A} |S_a|. \tag{9}$$

By averaging, we find that there is a cover S_a of the size num/n or less. Therefore the size of this set S_a is bounded by $l + \tilde{O}(t \cdot n^{\epsilon}) + o(n^{\delta_1}) + O(n^{1-\delta_1}) \leq 2l$ (the last inequality, again, follows for large enough n since $l \geq n^{\delta_1}/2$).

Thus we may choose S_a as the required cover. ∎

The main theorem is now derived easily. For this theorem let c be a constant such that it is $NP-$hard to approximate set-cover within ratio $c \ln n$.

Theorem 3.7 *The* $k-$*spanner problem for* $k \geq 5$ *cannot be approximated within ratio*

$$\frac{c(1-\delta_1)}{8} \cdot \ln n$$

unless $P = NP$.

Proof: Again, it is only necessary to prove this result for odd values, $k = 2t+1, \quad t \geq 2$, of k and n large enough. Assume an algorithm $\mathcal{A}$ that has the stated ratio. Let G be an instance of the $n^{(1-\delta_1)} - DBSC$ as described above. Let s^* be the size of the minimum cover of V_2 in G. Construct an instance $\bar{G}$ of the $(2t+1)$-spanner as described. According to Lemma 3.5, the graph $\bar{G}$ admits a spanner with $2 \cdot s^* \cdot n$ edges. The number of vertices, $\bar{n}$, in $\bar{G}$ is $\tilde{\theta}(n^{1+\epsilon})$. Thus, $\ln \bar{n} < 2 \ln n$ for n large enough. The theorem therefore assumes that the algorithm $\mathcal{A}$ would produce a spanner of size less than $(c(1-\delta_1)/8) \cdot 2 \cdot \ln n \cdot 2 \cdot s^* \cdot n = (c(1-\delta_1)/2) \cdot \ln n \cdot s^* \cdot n$.

Let $l = (c(1-\delta_1)/2) \cdot \ln n \cdot s^*$. According to Lemma 3.6, one derives a cover with a size at most $c(1-\delta_1) \cdot \ln n \cdot s^*$ from this construction (in polynomial time). This contradicts Lemma 2.3. ∎

4 The case $k = 3$ and $k = 4$, and the case $k = 2$

4.1 The case $k = 3$ and $k = 4$

We begin with the case $k = 3$ and $k = 4$. Again we have a fixed part and a gadget part in the construction. Let ϵ, δ and δ_1 be defined by Inequalities (1) and (2), with $t = 1$. Here we need the additional assumption that $(1+\epsilon)/2 < \delta_1 < 1$. Again, begin with a $n^{(1-\delta_1)} - BDSC$ instance, $G(V_1, V_2, E)$. The fixed part has a copy of V_1 and the set A of n vertices. These two sets are cliqued. As before, we introduce two vertices h_1 and h_2 that take care of the bipartite clique edges.

Let $q = \tilde{O}(n^\epsilon)$. We have now q copies of V_2, $V_2^1, \ldots, V_2^q$, i.e., V_2 belongs to the gadget part in this construction. Form q copies of G by appropriately connecting V_1 and V_2^i for each i.

In addition, add q sets A^i of size n. Each set A^i is matched with A. Finally, draw an edge between each vertex $a_i \in A^i$ and $v_2^s \in V_2^s$, $1 \leq s \leq q$, with probability $1/n^\epsilon$.

Again, there is a high probability that for each $v_2 \in V_2$ and $a \in A$ there is a copy A^i of A and a copy V_2^j of V_2, such that the two vertices v_2^j and a_i corresponding to v_2 and a share an edge. In a proof similar to the one in Lemma 3.6 it is possible to show that there exist a spanner close to the optimal, containing no R_i-edges. Hence, the hardness of approximation follows here in a way similar to the case of $k \geq 5$. The only difference is that we add $\tilde{O}(n^\epsilon)$ copies of G into the spanner; i.e., we add $n^{2-\delta_1+\epsilon}$ edges to the spanner. However, we only have to verify that $2 - \delta_1 + \epsilon < 1 + \delta_1$, which holds by the additional assumption.

Corollary 4.1 *For any fixed k, $k \geq 3$, there exists a constant c such that the $k-$spanner problem restricted to bipartite graphs is $NP-$hard to approximate within ratio $c \log n$.* ∎

4.2 The case $k = 2$

In order to prove a hardness result in the case $k = 2$, we simply take the fixed part of the construction for $k \geq 5$ without the vertex h_2. We also connect h_1 to all the vertices in V_1 (note that h_1 is now connected to all the vertices of $A \cup V_1$.) Finally, we clique A and V_2. As before, a sparse spanner is derived by choosing the edges touching h_1, the edges of G and the edges connecting A to a minimum cover S^* of V_2. Furthermore, the construction of a small cover given a sparse spanner is obtained by inspecting the way the edges connecting A and V_2 are spanned. Therefore, a hardness result similar to the above follows easily, except that the graph is only $3-$partite, i.e., $3-$colorable (in a bipartite graph the only $2-$spanner for the graph is the graph itself, hence the $2-$spanner problem is trivial on bipartite graphs). Combined with the result of [KP-92] we arrive at:

Corollary 4.2 *Unless $P = NP$, the best polynomial approximation algorithm for the $2-$spanner problem has a ratio of $\theta(\log n)$ even when restricted to $3-$partite graphs.* ∎

5 The weighted case

In this section we deal with the following weighted version of the spanner problem. We are given a graph G with a weight function $w(e)$ on the edges. We assume the *length* of each edge to be 1. That is, once again, in every $k-$spanner, a missing edge should be replaced by a cycle containing k edges or less. Here, however, we measure the quality of the spanner by its weight, namely the sum of the weights of its edges. We look for a $k-$spanner with minimum weight. In this section we prove an essential difference between the approximability of cases $k = 2$, and $k \geq 5$, i.e., we prove that for $k = 2$, this version of the problem admits an $O(\log n)$ ratio approximation. However, for every $k \geq 5$, the problem has no $2^{\log^{1-\epsilon} n}-$ ratio approximation, unless $NP \subseteq DTIME(n^{\text{polylogn}})$. This, for example, indicates that it is unlikely that there would be any polylogarithmic ratio approximation for the edge-weighted $k-$spanner problem, for $k \geq 5$.

5.1 The case $k = 2$

In the weighted $2-$spanner problem, one looks for a minimum weight $2-$spanner, namely, a low weight subgraph G' where every missing edge closes a triangle with two edges that do belong to G'. We find out that a method similar to the one employed in [KP-92] for approximating the unweighted case is suitable for approximating even this more general case.

We sketch the variant of the method of [KP-92] required here.

We say that a vertex v $2-$helps an edge $e = (w, z)$ in G' if the two edges (v, w) and (v, z) are included in G', i.e., in G', there is an alternative path of length 2 for e that goes through v.

The idea is to find a vertex v that $2-$helps many edges of E, using low weight. Consider each vertex $v \in G$. Let $N(v, G)$ be the graph induced in G by the neighbors $N(v)$ of v. For every neighbor z of v, put weight $w(e)$ on z in $N(v, G)$, where $e = (z, v)$. For any subset of the vertices $V' \subseteq N(v)$, let $e(V')$ denote the number of edges inside V', and let $w_v(V')$ denote the sum of weights of the vertices of V', in $N(v, G)$. We look for a vertex v and a subset $V' \subseteq N(v)$ that achieves the following minimum:

$$\min_{v} \left\{ \min_{V' \subseteq N(v)} \left\{ \frac{w_v(V')}{e(V')} \right\} \right\}.$$

It is important to note that the pair v, V' achieving this minimum can be found in polynomial time using flow techniques (cf. [GGT-89]). Given v and V', one adds the edges connecting v and V' to the spanner. Note that in this way we $2-$help (or span) all the edges internal to V', using low weight. This is done in iterations until the edges are exhausted.

It follows from a proof similar to that in [KP-92] that this greedy algorithm is an $O(\log(|V|))-$ratio approximation algorithm for the edge-weighted $2-$spanner problem. Details are therefore omitted.

Theorem 5.1 *The edge-weighted $2-$spanner problem, in the case $\ell(e) = 1$, for every edge e, admits an $O(\log(|V|))-$ratio approximation algorithm.* ▮

5.2 The case $k \geq 5$

In this subsection we consider the edge-weighted $k-$spanner problem, for $k \geq 5$, in the special case where $\ell(e) = 1$ for every edge e. We essentially prove hardness by giving a reduction from a one-round two-prover, interactive proof system. For simplicity, however, we abstract away the relation to the interactive proof and describe the problem from which we reduce in the following simpler manner. There are two versions of the problem, a maximization version and a minimization version.

We are given a bipartite graph $G(V_1, V_2, E)$. The sets V_1 and V_2 are split into a disjoint union of k sets: $V_1 = \bigcup_{i=1}^{k} A_i$ and $V_2 =_{j=1}^{k} \bigcup B_j$. The sets A_i and B_j all have size N.

The bipartite graph and the partition of V_1 and V_2 induce a super-graph $\mathcal{H}$ in the following way: The vertices in $\mathcal{H}$ are the sets A_i and B_j. Two sets A_i and B_j are connected by a (super) edge in $\mathcal{H}$ iff there exist $a_i \in A_i$ and $b_j \in B_j$ which are adjacent in G. For our purposes, it is convenient (and possible) to assume that graph $\mathcal{H}$ is regular. Say that every vertex in $\mathcal{H}$ has degree d, and hence, the number of super-edges is $h = k \cdot d$.

In the maximization version, which we call Max-rep, we must select a single "representative" vertex $a_i \in A_i$ from each subset A_i, and a single "representative" vertex $b_j \in B_j$ from each B_j. We say that a super-edge (A_i, B_j) is covered if the two corresponding representatives are neighbors in G, i.e., $(a_i, b_j) \in E$. The goal is to select a single representative from each set and maximize the number of super-edges covered.

Let us now recall the satisfiability (SAT) problem. A CNF boolean formula I is given, and the question is whether there is an assignment satisfying all the clauses. The following result follows from [FL-92]. It can also be deduced from [R-95].

Theorem 5.2 *Let I be an instance of SAT. For any $0 < \epsilon < 1$, there exists a reduction of each instance of the satisfiability problem, to an instance G of Max-rep of size n, such that if I is satisfiable, there is a set of unique representatives which cover all $h = k \cdot d$ super-edges, and if the formula is not satisfiable, in the best choice of representatives, it is possible to cover no more than $h/2^{\log^{1-\epsilon} n}$ of the super-edges.* ∎

In the above reduction, n is polylogarithmic in the size of the SAT formula. The following easily follows from Theorem 5.2.

Theorem 5.3 *Unless $NP \subseteq DTIME(n^{\text{polylogn}})$, Max-rep admits no $2^{\log^{1-\epsilon} n}$−ratio approximation, for any $\epsilon > 0$.* ∎

We need a slight minimization variant of Max-rep, which we call Min-rep. In this case the goal is to choose a minimum size subset $\mathcal{C} \subseteq V_1 \cup V_2$. Unlike the maximization version of the problem, in the minimization version of the problem, one may choose to include many vertices of each set A_i and B_j in $\mathcal{C}$. In Min-rep one must cover *every* super-edge, i.e., for each super-edge (A_i, B_j) there is a pair $a_i \in A_i$ and $b_j \in B_j$, both belonging to C, such that $(a_i, b_j) \in E$.

A limitation on the approximability of Min-rep, similar to that of Max-rep, follows easily from Theorem 5.2. The reduction here is rather standard. It is also implicit in [LY-93]. However, for the sake of completeness we describe the reduction.

The hardness of Min-rep follows from the following observation. Say that you have a solution $\mathcal{C}$ for Min-rep with t representatives. On average, there are $t/2k$ representatives in each A_i and B_j. Therefore, there are no more than $k/2$ sets A_i or B_j containing more than $2t/k$ representatives of $\mathcal{C}$. In removing these sets from $\mathcal{H}$, one deletes no more than $h/2 = d \cdot k/2$ super-edges. Let $\mathcal{H}'$ denote the resulting super-graph. Thus, at least $h/2$ super-edges are internal to $\mathcal{H}'$.

Draw a single representative uniformly at random for each A_i (resp., B_j) in $\mathcal{H}'$. The expected number of super-edges covered is (at least) $h \cdot k^2/8t^2$. The randomization in the choice can be removed using the method of conditional expectation.

Now, assume there exists an $l-$ratio approximation algorithm for Min-rep. Consider the reduction from I to G described in Theorem 5.2. If I is a "yes" instance of SAT, then there exists in G a proper system of representatives $\mathcal{C}$ for Min-rep of size $2k$ (a single vertex can be chosen from each set). The assumed algorithm will produce a solution for Min-rep of a size no larger than $2 \cdot k \cdot l$.

On the other hand, it follows from the above discussion that if I is a "no" instance for SAT, the solution for Min-rep produced by *any* algorithm is of size at least

$$t \geq \frac{k \cdot 2^{1/2 \cdot \log^{1-\epsilon} n}}{2\sqrt{2}},$$

since otherwise, we get a solution to Max-rep covering $hk^2/(8t^2) > h/2^{\log^{1-\epsilon} n}$ super-edges.

Therefore,

Theorem 5.4 *Unless $NP \subseteq DTIME(n^{\mathrm{polylogn}})$, Min-rep admits no $2^{\log^{1-\epsilon} n}-$ratio approximation algorithm, for any $\epsilon > 0$.* ▮

We now give a reduction from Min-rep to the edge-weighted $5-$spanner problem. The reduction for $k > 5$ is similar. Let $G = (V, E)$ be an instance of Min-rep with each A_i, B_j of size N. We build an instance $\bar{G}$ of the edge-weighted 5-spanner as follows: Add G into $\bar{G}$. Give the edges of G weight 0. Match each set A_i to a new set S_i and give these edges weight 1. Match each set B_j to a new set T_j. These edges are also given weight 1. Then, introduce a new vertex v_i (resp., a new vertex u_i) that is adjacent with edges of weight 0, to all vertices A_i (resp., all the vertices of S_i). The vertices v_i and u_i are joined by an edge of weight 0. Finally, for each B_j we introduce a vertex z_j (resp., w_j) joined to all the vertices of B_j (resp., of T_j). Again, z_j and w_j are joined by an edge, and all the edges touching w_j

and z_j have weight 0.

Now, for every super-edge (A_i, B_j) clique the set S_i to a new set X_{ij} of size N, and clique each set T_j to a new set Y_{ij} of size N with edges of weight 0. Finally, clique each pair X_{ij} and Y_{ij} with edges of weight $(N \cdot k)^2$, each.

It easily follows that no edges connecting X_{ij} and Y_{ij} are to be included in a good spanner. Hence, the only way to span an edge connecting X_{ij} and Y_{ij} is via a path of length 5 that goes through a vertex $a_i \in A_i$ and a vertex $b_j \in B_j$. Given a sparse spanner, vertices a_i and b_j which participate in such a path can be chosen into $\mathcal{C}$. In this way one gets a set of representatives $\mathcal{C}$ which is a solution for Min-rep. It is clear that $\mathcal{C}$ covers all super-edges. In addition, the weight of the spanner is exactly $|\mathcal{C}|$, since we have extra weight of 1 corresponding to the matching edge for each vertex in $\mathcal{C}$.

In the other direction, suppose we are given a small subset $\mathcal{C}$ of representatives which covers all the super-edges. We get a sparse spanner as follows: let (A_i, B_j) be a super-edge. Choose a pair $a_i \in A_i \cap \mathcal{C}$, $b_j \in B_j \cap C$, $(a_i, b_j) \in E$. Add an edge from a_i to the corresponding vertex $s_i \in S_i$. Add all the edges from s_i to X_{ij}. Add an edge from b_j to the corresponding vertex $t_j \in T_j$. Add all the edges from t_j to Y_{ij}. Finally, add all the edges touching v_i, u_i, z_j, w_j. Clearly, this establishes a $5-$spanner of weight $|\mathcal{C}|$.

Since the constructed graph is bipartite, the following corollary easily follows from Theorem 5.4.

Corollary 5.5 *Unless $NP \subseteq DTIME(n^{\mathrm{polylogn}})$, the edge-weighted $k-$spanner problem, for $k \geq 5$, admits no $2^{\log^{1-\epsilon} n}-$ratio approximation, even when restricted to bipartite graphs.* ∎

In conclusion, the main open question is whether a similar "gap" in the approximability of $k = 2$, and $k \geq 5$ is valid in the unweighted case as well. The goal is to either prove evidence for such a gap, or give logarithmic ratio approximation algorithms for fixed values of k. The cases $k = 3$ and $k = 4$ also deserve attention.

Acknowledgment

The author thanks Uri Feige for many helpful discussions.

References

[ABP-92] B. Awerbuch and A. Baratz and D. Peleg, Efficient Broadcast and Light-Weight

Spanners, *Technical Report CS92-22, The Weizmann Institute of Science*, Rehovot, Israel, 1992

[ADDJ-90] I. Althöfer and G. Das and D. Dobkin and D. Joseph, Generating sparse spanners for weighted graphs, *Discrete Compu. Geometry*, 9, 1993, 81-100

[ADM+95] S. Arya and G. Das and D.M. Mount and J.S. Salowe and M. Smid, Euclidean Spanners: Short, Thin, and Lanky, *Proc. 27th ACM Symp. on Theory of Computing*, 1995, 489-498

[B-86] B. Bollobás, Combinatorics, *Cambridge University Press*, 1986

[BH-97] U. Brandes and D. Handke, NP-Completeness Results for Minimum Planar Spanners, *Proceedings of the 23th workshop on Graph-Theoretic concepts in Computer Science (WG97)*, Springer Lecture-Notes in Computer Science 1335, 1997, 85-99

[C-52] H. Chernoff, A Measure of Asymptotic Efficiency for Tests of Hypothesis Based on the Sum of Observations, *Ann. Math. Stat.*, 23, 1952, 493-507

[C-86] L.P. Chew, There is a Planar Graph Almost as Good as the Complete Graph, *J. Comput. Syst. Sci.*, 1989, 205-219

[C-94] L. Cai, NP-completeness of minimum spanner problems, *Discrete Applied Math*, 9, 1993, 81-100

[CDNS-92] B. Chandra and G. Das and G. Narasimhan and J. Soares, New sparseness results for graph spanners, *Internat. J. Comput. Geom. Appl.*, 5, 1995, 125-144

[CK-94] L. Cai and M. Keil, Spanners in graphs of bounded degree, *Networks*, 24, 1994, 187-194

[DFS-87] D.P. Dobkin and S.J. Friedman and K.J. Supowit, Delaunay Graphs are Almost as Good as Complete Graphs, *Discrete Compu. Geom.*, 1990, 399-407

[DJ-89] G. Das and D. Joseph, Which Triangulation Approximates the Complete Graph?, *Int. Symp. on Optimal Algorithms*, 1989, 168-192

[F-96] U. Feige, A threshold of $\ln n$ for approximating set cover, *Proc. 28th ACM Symp. on Theory of Computing*, 1996, 314-318

[FL-92] U. Feige L. Lova'sz, Two-prover one-round proof systems: Their power and their problems, *Proc. 24th ACM Symp. on Theory of Computing*, 733-741, 1992

[GGT-89] G. Gallo and M.D. Grigoriadis and R.E. Tarjan, A fast Parametric maximum flow algorithm and applications, *SIAM J. on Comput*, 18, 1989, 30-55

[J-74] D.S. Johnson, Approximation Algorithms for Combinatorial Problems, *J. of computer and system sciences*, 9, 1974, 256-278

[KP-92] G. Kortsarz and D. Peleg, Generating Sparse 2-spanners, *J. of Algorithms*, 17, 1994, 222-236

[L-75] L. Lova'sz, On the ratio of Integral and Fractional Covers, *Discrete Mathematics*, 13, 1975, 383-390

[LL-89] C. Levcopoulos and A. Lingas, There is a Planar Graph Almost as Good as the Complete graph and as Short as Minimum Spanning Trees, *Algorithmica*, 8, 1992, 251-256

[LR-93] A.L. Liestman and T.C. Shermer, Grid Spanners, *Networks*, 23, 123-133, 1993

[LS-93] A. L. Liestman and T. C. Shermer, Additive graph Spanners, *Networks*, 23, 1993, 343-364

[LY-93] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, *Proc 25'th STOC*, 1993, 286-293

[MR-95] R. Motwani and P. Raghavan, Randomized Algorithms, *Cambridge University Press*, 1995

[PS-89] D. Peleg and A. Schäffer, Graph Spanners, *J. of Graph Theory*, 13, 1989, 99-116

[PU-88] D. Peleg and E. Upfal, A Tradeoff between space and efficiency for routing tables, *J. of the ACM*, 1989, 510-530

[PU-89] D. Peleg and J.D. Ullman, An Optimal Synchronizer for the Hypercube, *Siam J. on Comput.*, 18, 1989, 740-747

[R-95] R. Raz, A parallel repetition theorem, *Proc. 27th ACM Symposium on the Theory of Computing*, 1995, 447-456

[RL-95] D. Richards and A.L. Liestman, Degree-Constraint Pyramid Spanners, *J. of Parallel and Distributed Computing*, 25, 1995, 740-747

[RS-97] R. Raz and S. Safra, A sub constant error probability low degree test, and a sub constant error probability PCP characterization of NP, STOC, 1997, 475-484

[S-87] J. Spencer, Ten Lectures on the Probabilistic Method, *SIAM*, 1987

[S-92] J. Soares, Approximating Euclidean distances by small degree graphs, *University of Chicago*, No. 92-05, 1992

[VRM+97] G. Venkatesan and U. Rotics and M.S. Madanlal and J.A. Makowsy and C. Pandu Rangan, Restrictions of Minimum Spanner Problems, 1997, Manuscript