

GENERATING LOW-DEGREE 2-SPANNERS*

GUY KORTSARZ[†] AND DAVID PELEG[‡]

Abstract. A k -spanner of a connected (undirected unweighted) 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 paper is concerned with approximating the problem of finding a 2-spanner in a given graph, with minimum maximum degree. We first show that the problem is at least as hard to approximate as set cover. Then a randomized approximation algorithm is provided for this problem, with approximation ratio of $\tilde{O}(\Delta^{1/4})$. We then present a probabilistic algorithm that is more efficient for sparse graphs. Our algorithms are converted into deterministic ones using derandomization.

Key words. graph spanners, NP-hardness, approximation, randomized rounding

AMS subject classifications. 05C05, 05C12, 05C85, 68Q25, 68R10, 90C35

PII. S0097539794268753

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, LR94, CDNS92]. Consider a connected simple (unweighted) graph $G = (V, E)$, with $|V| = n$ vertices. A subgraph $G' = (V, E')$ of G is a k -spanner if for every $u, v \in V$,

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

where $\text{dist}(u, v, G')$ 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, Soa92]. 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 and then studied further in [PS89, LR94, ADDJ90, CDNS92]. Spanners were used in [PU89] to construct a new type of synchronizer for an asynchronous network.

The usual criteria for the quality of the spanner are its *stretch* and its *sparsity*. Namely, a good spanner is one with low stretch and as few edges as possible. For the problem of finding a 2-spanner which is as sparse as possible, a logarithmic-ratio approximation is given in [KP94].

However, another parameter of significance when selecting a good spanner is the maximum degree of the spanner. In terms of applications, a high degree might mean a high *local load* on a single vertex, increasing the cost of its local management. For instance, in the application of using a spanner for implementing a δ -synchronizer in a distributed network [PU89], or when using a spanner for efficient broadcast [ABP91],

*Received by the editors June 3, 1994; accepted for publication (in revised form) August 9, 1996; published electronically May 19, 1998.

<http://www.siam.org/journals/sicomp/27-5/26875.html>

[†]Department of Computer Science, The Open University, Ramat-Aviv, Israel (guyk@tavor.openu.ac.il). Part of this work was done while the author was at the Weizmann Institute.

[‡]Department of Applied Mathematics and Computer Science, The Weizmann Institute, Rehovot 76100, Israel (peleg@wisdom.weizmann.ac.il). This research was supported in part by a Walter and Elise Haas Career Development Award and by a grant from the Israel Science Foundation.

the degree of each node in the spanner directly translates into memory requirements at the node, and high local degrees mean higher workload on the involved nodes.

It is clear that focusing on optimizing the sparsity measure alone may result in a spanner with high vertex degrees. For example, if there is a vertex v in the graph that is connected to all the rest of the vertices, then its edges form a 2-spanner and, therefore, a k -spanner for any $k \geq 2$. However, in such a choice the local load of v might be too high to handle, while the local load of any other vertex w may be much less than what w can handle. In fact, the algorithm proposed in [KP94] will pick the vertex v and is therefore unsuitable for selecting “balanced load” spanners. It is therefore natural to try to design an algorithm that will perform a more “balanced” selection of the edges. In particular, letting $\Delta(G')$ denote the maximum vertex degree in a spanner G' , we consider the question of choosing a k -spanner G' with minimum $\Delta(G')$ for some parameter k . We call this “low degree” variant of the problem LD- k SP. The problem of designing low degree spanners is addressed in [LR94, LS93b, LS93a] for some special graph classes such as pyramids and grids. The problem of designing small degree spanners for Euclidean and geometric graphs is studied in [CDNS92, Soa92]. The distance is measured therein by the appropriate *norm* defined in the vector space.

This paper treats LD-2SP in general graphs. We first show that the problem is at least as hard to approximate as set cover (up to constants). This implies the following results. There is no $\ln n/5$ -ratio approximation algorithm for LD-2SP unless $NP \subset DTIME(n^{\text{polylog}(n)})$. Also, no approximation algorithm with constant ratio exists for the problem unless $P = NP$. We next give a probabilistic algorithm that outputs a 2-spanner G' such that with high probability $\Delta(G')$ is no more than $\tilde{O}(\Delta^{1/4})$ times the optimum. In other words, our algorithm has an approximation ratio of $\tilde{O}(\Delta^{1/4})$ with high probability. (\tilde{O} is a relaxed variant of the usual O notation that ignores polylogarithmic factors.) The algorithm is then turned into a deterministic one using derandomization.

The technique used in [KP94] to approximate the sparsest 2-spanner problem is the “greedy” method that constructs the spanner gradually, attempting to 2-span a large number of edges in every iteration. (An edge $e = (u, v)$ is *2-spanned* once either itself or two other edges lying on a triangle with it, say (u, x) and (x, v) , are added to the spanner.) The LD-2SP problem seems to be harder to approximate. In particular, the greedy approach seems to fail (i.e., be inefficient) for it. Hence a different (and more involved) approach is required. The technique used in this paper for the LD-2SP problem is a variant of the “randomized rounding” technique of [RT87].

Our algorithm is composed of two different procedures. The first procedure is designed to 2-span edges lying on “many” triangles. The second procedure deals with the yet unspanned edges, i.e., edges that lie on a “small” number of triangles. We describe the “2-spanning” problem for these edges as a linear program, solve it in the fractional setting, and randomly round the fractional solutions. We note that in the rounding process, we use only a subset of the variables. We also note that every variable is rounded with probability considerably exceeding its fractional value. These higher rounding probabilities seem to be needed in order to overcome some “quadratic” behavior of the linear program.

We also present an additional probabilistic algorithm that is efficient for *sparse* graphs. This algorithm can also be transformed into a deterministic one using derandomization.

Finally we deal with the problem of 2-spanning only the edges adjacent to a

small subset $V_k, |V_k| = k$ of the vertices. We give an $O(k \cdot \log n)$ -ratio approximation algorithm for this problem. For fixed k , our hardness result implies that unless $NP \subset DTIME(n^{\log \log n})$ this is the best ratio possible (asymptotically).

2. Preliminaries. We start by introducing some definitions. In the sequel, let $G = (V, E)$ be the underlying n -vertex graph. We sometimes use E also to denote the *size* of the set E , i.e., the number of edges. 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 $E(U)$. For a vertex v we denote by $E(v)$ the set of edges adjacent to v in G . Similarly, we denote by $N(v)$ the set of neighbors of v in G , i.e.,

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

We denote the degree of a vertex v by $\deg(v) = |N(v)|$. The maximum degree in a subgraph $G' = (V', E')$, where $V' \subset V$ and $E' \subseteq E$, is denoted by $\Delta(G')$. We denote by $\Delta(E')$ the largest degree in the subgraph (V, E') . We sometimes write Δ for $\Delta(G)$.

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

LEMMA 2.1 (see [PS89]). *The subgraph $G' = (V, E')$ is a k -spanner of the graph $G = (V, E)$ iff $\text{dist}(u, v, G') \leq k$ for every $(u, v) \in E$. \square*

Thus the LD-2SP problem can be restated as follows: we look for a 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' and such that $\Delta(E')$ is minimum.

Given an edge $e \in E$, let $\text{Tri}(e)$ denote the set of triangles e lies on in the graph G . Namely,

$$\text{Tri}(e) = \{\{e, e_1, e_2\} \mid e, e_1 \text{ and } e_2 \text{ form a triangle in } G\}.$$

Let $D(e)$ be the set of vertices that lie on a triangle with e but do not touch e . (Note that $|D(e)| = |\text{Tri}(e)|$, as each vertex in $D(e)$ corresponds to exactly one triangle in $\text{Tri}(e)$.) We say that a vertex $v \in D(e)$ (sharing a triangle T with e) *2-helps* e in the spanner H if the two edges incident to v on T are chosen into H .

In the sequel we estimate the probability of the deviation of some random variables from their expectation, using the Chernoff bound [Che52].

LEMMA 2.2 (see [Che52]). *Let X_1, X_2, \dots, 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*

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

In the sequel we assume that $\Delta(G) \geq \Omega(\log^2 n)$. If this is not the case, then taking the entire graph as our spanner results in a polylogarithmic-ratio approximation.

Unless stated otherwise, all logarithms in this paper are taken to the base 2.

3. Basic properties.

3.1. Low degree spanners for general graphs and special graph families.

The problem of designing low degree spanners is addressed in [LR94] for the special case where the underlying graph is the *pyramid*. In particular, it is proven therein that this graph enjoys a 2-spanner (respectively, 3, 7) with maximum degree 6 (respectively, 4, 3). The problem of designing small degree spanners for Euclidean and geometric

graphs is studied in [CDNS92, Soa92]. There, however, the distance is measured by the appropriate *norm* defined in the vector space.

We now establish some basic properties concerning the degrees of 2-spanners. The next lemma indicates that for a graph with large Δ , the minimum degree in a 2-spanner must also be large. We prove this by showing that even for the sake of 2-spanning the edges of a *single* vertex v with degree Δ , it is necessary to have a vertex in the spanner with degree at least $\sqrt{\Delta}$.

LEMMA 3.1. *Let v be a vertex of degree d in G . Let $H = (V, E(H))$ be a 2-spanner of G . Then in H either v or some vertex in $N(v)$ has degree at least \sqrt{d} .*

Proof. Let t denote the maximum degree of any vertex from $N(v)$ in H . Then the number of vertices reachable from v in two steps over H edges is at most t^2 . Since all d edges incident to v must be 2-spanned in H , necessarily $t^2 \geq d$. \square

As an immediate conclusion we have the following.

LEMMA 3.2. *Let $H^* = (V, E^*)$ be a 2-spanner for G with minimum maximum degree. Then*

$$\Delta(H^*) \geq \sqrt{\Delta(G)}. \quad \square$$

Let us remark that a similar result holds for k -spanners H^* of minimum maximum degree for any $k \geq 2$, namely, $\Delta(H^*)$ is $\Omega(\Delta(G)^{1/k})$ (the proof is also similar).

Note that there are graphs G for which $\Delta(H) = \Delta(G)$ for any 2-spanner H of G . One particular such graph is the star of $n - 1$ vertices. However, there are dense graphs, where the lower bound $\sqrt{\Delta}$ can be achieved (up to constants). The clique (complete graph) K_n of n vertices admits low degree 2-spanners. In order to prove this, we use the notion of a *projective plane of order q* for prime q . The existence of projective planes of order q for every prime q is well known. A projective plane $\mathcal{P} = (P, L)$ of order q is composed of a collection $P = \{p_1, \dots, p_m\}$ of points and a collection $L = \{l_1, \dots, l_m\}$ of lines where $m = q^2 + q + 1$. Every line l_i is a subset of P containing exactly $d = q + 1$ points and every point is contained in exactly d lines. Every two lines intersect in exactly one point and every two points share exactly one line.

Consider now $K_n = (V, V \times V)$ where $V = \{v_1, \dots, v_n\}$. Let q be a prime number such that $\lfloor \sqrt{n} \rfloor \leq q \leq 2\lfloor \sqrt{n} \rfloor$. (Such a prime exists by Bertrand's postulate; cf. [HW56].) Thus, $n < q^2 + q + 1 < 5n$. Let $\mathcal{P} = (P, L)$ be a projective plane of order q . Define the following spanning subgraph $H = (V, E')$ of K_n . Add the edge (v_i, v_j) to E' iff there exist t and r that satisfy $t \equiv i \pmod{n}$ and $r \equiv j \pmod{n}$ such that either $p_t \in l_r$ or $p_r \in l_t$.

We now proceed to prove that H is a low degree 2-spanner for K_n . First we note the following claim.

CLAIM 3.3. *The subgraph H is a 2-spanner for K_n .*

Proof. Let $e = (v_i, v_j)$ be an arbitrary edge of K_n . The lines l_i and l_j share some point $p_s \in l_i \cap l_j$. Let f be the integer satisfying $1 \leq f \leq n$, $f \equiv s \pmod{n}$. If $f = i$ or $f = j$ (i.e., the case is, for example, that $p_j \in l_i \cap l_j$) then by definition $(v_i, v_j) \in E'$. Otherwise, again by definition, both $(v_i, v_f) \in E'$ and $(v_j, v_f) \in E'$, and the edge e is spanned. \square

We now estimate the degree of the vertices in H' . Note that the degree of a vertex $v_i \in H$ is only increased due to vertices in the set $S_i = \{l_j \mid j \equiv i \pmod{n}\} \cup \{p_j \mid j \equiv i \pmod{n}\}$, and $|S_i| \leq 10$. Each vertex in S_i increases the degree of v_i by at most $d = q + 1$, and thus the degree of v_i in H is bounded by $O(d) = O(\sqrt{n})$.

In conclusion, we have established the following claim.

LEMMA 3.4. *The complete graph K_n admits a 2-spanner H with $\Delta(H) = O(\sqrt{\Delta(K_n)})$. \square*

4. A hardness result for approximating LD-2SP. In this section we establish that the LD-2SP problem is (up to a constant factor) at least as hard to approximate as the *set cover* problem. Formally, the set cover problem is defined as follows. Given a bipartite graph $G(V_1, V_2, E)$ with $|V_1| = |V_2| = n$, find a minimum cardinality subset S of V_1 , that *covers* V_2 , i.e., such that every vertex in V_2 has a neighbor in S .

It is known that this problem is hard to approximate. In particular, the following theorem is proved in [LY93, Fei96].

THEOREM 4.1 (see [Fei96]). *The set cover problem cannot be approximated with ratio $\ln n - \epsilon$, for any fixed $\epsilon > 0$, unless $NP \subset DTIME(n^{\log \log n})$. \square*

Also, the following theorem is proven in [BGLR93].

THEOREM 4.2 (see [BGLR93]). *The set cover problem cannot be approximated with any constant ratio c , unless $P = NP$. \square*

For our purpose, we need a slightly different version of the set cover problem. Define the \sqrt{n} -set cover problem as a variant of the set cover problem in which $d(v) \leq \sqrt{n}$ for each vertex $v \in V_1 \cup V_2$. The usual greedy algorithm approximates this problem with ratio $\ln \sqrt{n} + 1 = \ln n/2 + 1$ [Joh74, Lov75]. On the other hand, a simple observation gives the following fact.

FACT 4.3. *The \sqrt{n} -set cover problem cannot be approximated with ratio better than $\ln n/2$, unless $NP \subset DTIME(n^{\log \log n})$.*

Proof. Assume the existence of an approximation algorithm A for the \sqrt{n} -set cover problem, with ratio $\ln n/2$ or better. Let $G(V_1, V_2, E)$ be an instance of the set cover problem. Let \tilde{G} be a graph consisting of n (separate) copies of G . The graph \tilde{G} contains n^2 vertices on each side, and the maximum degree in \tilde{G} is bounded by n . Thus, \tilde{G} is amenable to approximation by algorithm A , and, consequently, the set cover instance represented by \tilde{G} can be approximated with ratio better than $\ln(n^2)/2 = \ln n$. Since any cover in \tilde{G} is composed of n separate covers of V_2 , we get by a straightforward averaging argument that one of these covers approximates the optimum cover of V_2 by a ratio better than $\ln n$. By Theorem 4.1, this implies $NP \subset DTIME(n^{\log \log n})$. \square

In the remainder of this section, we consider the \sqrt{n} -set cover problem, with $|V_1| = |V_2| = n$. We show that the LD-2SP problem is at least as hard to approximate as this problem. Throughout, we denote by t^* the size of the optimum cover of V_2 in G . Let $\ell = \lceil \sqrt{n} \rceil$. Note that

$$(1) \quad t^* \geq \ell.$$

4.1. The construction. We use an auxiliary graph \tilde{G} constructed from G as follows. The vertices of \tilde{G} are

$$V_1 \cup V_2 \cup \{s\} \cup \{c(v_1) \mid v_1 \in V_1\} \cup \{u_1, \dots, u_\ell\}.$$

Divide the vertices of V_1 arbitrarily into ℓ disjoint sets V_1^i , where each V_1^i contains no more than \sqrt{n} vertices. The edge set of \tilde{G} is given by defining a number of edge

classes as follows. Let

$$\begin{aligned}\mathcal{E}_1 &= E(G), \\ \mathcal{E}_2 &= \{(s, v_1) \mid v_1 \in V_1\}, \\ \mathcal{E}_3 &= \{(s, v_2) \mid v_2 \in V_2\}, \\ \mathcal{E}_4 &= \{(c(v_1), v_1) \mid v_1 \in V_1\} \cup \{(c(v_1), v_2) \mid (v_1, v_2) \in E(G)\}, \\ \mathcal{E}_5 &= \{(u_i, v_1) \mid v_1 \in V_1^i, 1 \leq i \leq \ell\}, \\ \mathcal{E}_6 &= \{(s, u_i) \mid 1 \leq i \leq \ell\},\end{aligned}$$

and set

$$E(\bar{G}) = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3 \cup \mathcal{E}_4 \cup \mathcal{E}_5 \cup \mathcal{E}_6.$$

Let us make a few remarks on this construction. As the maximum degree in \bar{G} is more than $2n$, the best 2-spanner H of G has maximum degree at least $\sqrt{2n}$. We note that for the sake of choosing a good spanner, one can “afford” the edges of classes \mathcal{E}_1 , \mathcal{E}_4 , \mathcal{E}_5 , and \mathcal{E}_6 , since it easily follows from the construction that these edges induce a subgraph with maximum degree bounded by ℓ . This observation relies on the fact that the degrees of the vertices of V_1 and V_2 in G are bounded by \sqrt{n} .

Consequently, the edges from which a “good” spanner may need to omit are the edges of classes \mathcal{E}_2 and \mathcal{E}_3 . These edges give s its degree in \bar{G} , which is higher than $2n$. The heart of our proof lies in 2-spanning the edges of \mathcal{E}_3 . Such an edge e must either be included in the spanner or be 2-helped by some vertex $v_1 \in D(e)$. In order to keep the degree of s low, one has to choose a small subset of V_1 that covers V_2 .

4.2. The main claim. We prove the main result of this section using the following two lemmas.

LEMMA 4.4. *If G contains a cover of size t^* then \bar{G} contains a 2-spanner with maximum degree bounded by $2t^*$.*

Proof. Assume a graph $G(V_1, V_2, E)$ with a t^* -cover $C = \{v_1, \dots, v_{t^*}\} \subseteq V_1$ of V_2 . For every vertex $v_2 \in V_2$ choose a vertex $R(v_2) \in C$ connected to v_2 . Create a spanner H consisting of the following two edge sets:

$$\begin{aligned}\mathcal{H}_1 &= \{(v_2, R(v_2)) \mid v_2 \in V_2\} \cup \{(s, v_1) \mid v_1 \in C\}, \\ \mathcal{H}_2 &= \mathcal{E}_4 \cup \mathcal{E}_5 \cup \mathcal{E}_6,\end{aligned}$$

and set $E(H) = \mathcal{H}_1 \cup \mathcal{H}_2$.

(*Remark.* The above construction can be slightly improved. For example, it is not necessary to put in H edges $(c(v_1), u)$ if v_1 is the vertex chosen to cover u . We note, nevertheless, that the maximum degree is not decreased by these improvements.)

We need the following observations.

CLAIM 4.5. *The maximum degree in H is bounded by $2t^*$.*

Proof. As explained before, the edges of classes \mathcal{E}_4 , \mathcal{E}_5 , and \mathcal{E}_6 induce a graph with maximum degree bounded by ℓ . Specifically, s , the $c(v_i)$ vertices and the vertices of V_2 have degree bounded by ℓ and the vertices of V_1 have degree bounded by 2 (i.e., a vertex v_1 is connected to its u_i and to $c(v_1)$.)

Now consider the edges of \mathcal{H}_1 . The degree of each vertex in C is increased by no more than $\sqrt{n} + 1$ due to these edges (these edges connect a vertex v_1 to its neighbors in V_2 and to s). Note, however, that t^* new edges are added to s (connecting s to the vertices of C), thus making its degree no more than $t^* + \ell \leq 2t^*$ by inequality (1). Hence the maximum degree is as stated. \square

Next we establish the following claim.

CLAIM 4.6. *The graph H is a 2-spanner of \bar{G} .*

Proof. The edges of classes \mathcal{E}_4 , \mathcal{E}_5 , and \mathcal{E}_6 are in the spanner. We must show that the edges of classes \mathcal{E}_1 , \mathcal{E}_2 , and \mathcal{E}_3 are 2-spanned by H .

An edge $(v_1, v_2) \in \mathcal{E}_1$ is 2-spanned in the spanner by the edges $(c(v_1), v_1)$ and $(c(v_1), v_2)$, since both are in \mathcal{E}_4 .

Next consider an edge $(v_1, s) \in \mathcal{E}_2$. The node v_1 must belong to some set V_1^i . Therefore this edge is 2-spanned in H by the edges (u_i, v_1) and (u_i, s) , which are in \mathcal{E}_5 and \mathcal{E}_6 , respectively.

Finally, an edge $(v_2, s) \in \mathcal{E}_3$ is 2-spanned in H by the edges $(v_2, R(v_2))$ and $(R(v_2), s)$, which are in \mathcal{H}_1 . \square

The second central lemma in this section is the following.

LEMMA 4.7. *Given a 2-spanner H of \bar{G} , it is possible to find a cover of V_2 in G with cardinality bounded by $\Delta(H)$.*

Proof. Consider a 2-spanner H . Partition the edges of class \mathcal{E}_3 into two disjoint sets

$$\mathcal{E}_3^A = \mathcal{E}_3 \cap E(H) \quad \text{and} \quad \mathcal{E}_3^B = \mathcal{E}_3 \setminus E(H).$$

Note that each of the edges of \mathcal{E}_3^B is 2-spanned in H by a 2-helping vertex.

Construct a cover of V_2 in G as follows. First, for every edge $(v_2, s) \in \mathcal{E}_3^A$, choose an arbitrary vertex v_1 connected to v_2 in G , and let D_1 be the set of selected vertices. Second, for every edge $(s, v_2) \in \mathcal{E}_3^B$, choose an arbitrary vertex $v_1 \in V_1$ such that both (s, v_1) and (v_2, v_1) are in $E(H)$. (A single vertex of V_1 may be chosen several times, as it may cover many edges in \mathcal{E}_3^B .) Let D_2 be the set of vertices chosen by this process.

Clearly, the set $D_1 \cup D_2$ forms a cover of V_2 in G . It remains to bound its cardinality. First note that $|D_1| \leq |\mathcal{E}_3^A|$ (at worst, a *different* vertex v_1 is chosen to D_1 for every vertex of \mathcal{E}_3^A). Now, every edge in \mathcal{E}_3^A adds 1 to the degree of s in H . Also, every vertex in D_2 is connected to s in H and thus adds 1 to its degree. Hence $|D_1| + |D_2| \leq \Delta(H)$. This proves the claim. \square

The following corollary is now immediate.

COROLLARY 4.8. *The LD-2SP problem cannot be approximated with ratio better than $\ln n/5$, unless $NP \subset DTIME(n^{\log \log n})$.*

Proof. Assume the existence of an approximation algorithm A for the LD-2SP problem, with ratio better than $\ln n/5$. Take an input G of the \sqrt{n} -set cover problem. Let t^* be the size of a minimum cover of V_2 in G . Construct \bar{G} as explained before (this can clearly be done in polynomial time). By Lemma 4.4 the graph \bar{G} contains a 2-spanner with maximum degree bounded by $2t^*$. Note that the number of vertices in \bar{G} is less than $4n$. By the assumption, it is possible to use algorithm A and find a 2-spanner with maximum degree bounded by

$$2t^*(\ln |\bar{G}|)/5 \leq 2t^*(\ln n + 4)/5 < t^*(\ln n/2 - 1)$$

(the last inequality holds for sufficiently large n). By Lemma 4.7 this implies that it is possible to find in polynomial time a cover of V_2 in G with cardinality bounded by $t^*(\ln n/2 - 1)$. By Fact 4.3, this implies $NP \subset DTIME(n^{\log \log n})$. \square

The following corollary also follows easily.

COROLLARY 4.9. *The LD-2SP problem cannot be approximated with ratio c , for any constant c , unless $P = NP$.* \square

5. The approximation algorithm for LD-2SP. Let us first explain the idea behind our approximation algorithm for the LD-2SP problem. We separate the edge set of our graph into two disjoint classes. The class E^- is the class of edges that lies on a small number of triangles, and the class E^+ contains the rest of the edges, namely,

$$E^- = \{e \in E \mid \text{Tri}(e) < \sqrt{\Delta}\}, \quad E^+ = \{e \in E \mid \text{Tri}(e) \geq \sqrt{\Delta}\}.$$

(Recall that $\text{Tri}(e)$ is the set $\{e, e_1, e_2\}$ of triangles containing e .) We 2-span these two classes of edges, using two separate procedures. Our general approach for handling E^- is to use the “randomized rounding” scheme of Raghavan and Thompson [RT87]. This scheme is based on the following idea. Let Δ^* be the maximum degree in the best 2-spanner. (We shall soon see that it is possible, without loss of generality, to assume that Δ^* is known.) We then formulate the problem as the integer linear program (P1) below.

The program (P1). Given are some subset $E_u \subseteq E$ of “unspanned” edges and a (possibly empty) set E_r of edges that have already been added to the spanner. Create for every edge $e_k \in E_u$ and vertex $v_i \in D(e_k)$ a variable $\hat{y}_{i,k}$. (Recall that $D(e)$ is the set of vertices that lies on a triangle with e but does not touch it.) For every two vertices $v_i, v_j \in V, i < j$, such that $(v_i, v_j) \in E$, create a variable $\hat{x}_{i,j}$. (We shall freely use both $\hat{x}_{i,j}$ and $\hat{x}_{j,i}$ to denote this unique variable.) The program is composed of the following sets of inequalities:

- (2) $\sum_{v_j \in N(v_i), (v_i, v_j) \notin E_r} \hat{x}_{i,j} \leq \Delta^* \quad \text{for all } v_i \in V,$
- (3) $\hat{x}_{l,t} + \sum_{v_i \in D(e_k)} \hat{y}_{i,k} \geq 1 \quad \text{for all } e_k = (v_l, v_t) \in E_u,$
- (4) $\hat{y}_{i,k} \leq \hat{x}_{i,l}, \hat{x}_{i,t} \quad \text{for all } e_k = (v_l, v_t) \in E_u \text{ and } v_i \in D(e_k),$
- (5) $\hat{x}_{ij} = 1 \quad \text{for all } e = (v_i, v_j) \in E_r,$
- (6) $\hat{x}_{i,j}, \hat{y}_{i,k} \in \{0, 1\} \quad \text{for all } i, j, k.$

The intuitive meaning of the program is as follows. Every $\hat{x}_{i,j}$ variable indicates if the edge (v_i, v_j) is in the chosen spanner. Thus constraint (2) says that every vertex v_i has no more than Δ^* new spanner edges. It is important to note that here we do not count the edges of E_r (those edges are counted separately in the analysis). The variable $\hat{y}_{i,k}$, associated with a vertex v_i and an edge $e_k = (v_l, v_t)$, indicates if v_i 2-helps e_k in the chosen spanner. This is enforced by constraint (4), which says that v_i 2-helps e_k only if both the edges (v_i, v_l) and (v_i, v_t) are included in the spanner. Constraint (3) says that in a feasible 2-spanner, every edge is either in the spanner or is 2-helped by some vertex.

After writing the program, we solve the fractional relaxation of (P1) using the well-known polynomial-time algorithms of [Kha80, Kar84]. Having the fractional values of the variables, we round each variable to be 1 with probability proportional to its fractional value.

When using this program for 2-spanning E^- , we get a good result; i.e., the randomized process gives a 2-spanner whose maximum degree is “close” to the “fractional degree” of the fractional solution. However, using this method we are not expected to 2-span all the edges of E^+ . To see this, consider an edge e lying in $\Omega(n)$ triangles. The fractional program may give all (the variables of) the edges in these triangles a

value in $\Theta(1/n)$. In this way constraints (2) and (3) are easily satisfied. However, the probability of a triangle to “survive the randomized rounding” (i.e., to have both its edges set to 1) is $\Theta(1/n^2)$. Since there are only $\Theta(n)$ triangles, the edge is not expected to be 2-spanned. (This phenomenon captures the unfortunate “quadratic” behavior of our linear program.)

We therefore 2-span the edges of E^+ using a different procedure. We draw every edge $e \in E$ randomly to the spanner, with some fixed (small) probability. We then show that the edges of E^+ , i.e., edges that belong to sufficiently many triangles, are likely to be 2-spanned in the resulting subgraph (namely are likely to lie on a triangle whose two other edges were selected by the randomized choice). We also show that with high probability the degree that is added to each vertex in this procedure is “small.”

In the remainder of this section we present our approximation algorithm, and in the next section we give its analysis. Throughout the algorithm, we denote by E_u the set of edges yet to be 2-spanned.

ALGORITHM 5.1.

Input: A graph $G = (V, E)$ of maximum degree Δ .

1. Let

$$p = \frac{2 \cdot \sqrt{\log n}}{\Delta^{1/4}}, \quad M = 2 \cdot \Delta^{1/4} \cdot \sqrt{\log n}$$

and set $E_r^1, E_r^2 \leftarrow \emptyset$.

2. For every edge $e \in E$, draw e randomly and independently to be in the spanner with probability p . Let E_r^1 denote the set of edges selected into the spanner by the randomized process.
3. Set $E_u = \{e \in E \mid e \notin E_r^1 \text{ and no two edges } e_1, e_2 \in E_r^1 \text{ form a triangle with } e\}$.
4. Solve the fractional relaxation of the program (P1) corresponding to E_u , E_r^1 , and Δ^* .
5. Let $\{x_{i,j}, y_{i,k}\}$ be the optimal (fractional) solutions corresponding to (P1). For every variable $\hat{x}_{i,j}$ create a respective random variable $\bar{x}_{i,j}$.
6. Randomly and uniformly set $\bar{x}_{i,j}$ to be 1 with probability $\min\{1, M \cdot x_{i,j}\}$.
7. If $\bar{x}_{i,j}$ is set to 1, then add the edge (v_i, v_j) to E_r^2 .
8. Let E_u be the set of edges that are still unspanned by $E_r^1 \cup E_r^2$. Set $E_r = E_r^1 \cup E_r^2 \cup E_u$.
9. Output E_r .

6. Analysis. First we explain how to overcome the assumption that Δ^* is known. Let Δ_f^* be the smallest value for which (P1) has a feasible (fractional) solution. We call Δ_f^* the smallest *fractional* degree of the best *fractional* 2-spanner. Indeed, we only have to know (and run (P1) with) Δ_f^* for our scheme to work. Clearly, $\Delta^* \geq \Delta_f^*$. This follows from the following simple claim.

LEMMA 6.1. *If we run the program (P1) with L replacing Δ^* , and (P1) has no fractional feasible solution, then $\Delta^* > L$. \square*

The value Δ_f^* is found through binary search, by running (P1) with values taken from the (discrete) interval $[\lceil \sqrt{\Delta} \rceil, \Delta]$. The search ends with some specific L such that the program succeed with $L+1$ but fails with L . By Lemma 6.1, $\Delta^* \geq \Delta_f^* > L$. On the other hand, we have a fractional feasible solution for $L+1$. Thus, we found the best fractional Δ_f^* (up to a difference of 1). For proving the desired approximation ratio, we show how to construct an (integer) spanner with maximum degree “close” to $L+1$ and therefore close to $\Delta^*(>L)$.

We now observe that the output is indeed a 2-spanner: the set of edges E_r forms a 2-spanner of G , since every edge not in E_r lies on a triangle with two edges of E_r .

Denote the optimum low-degree 2-spanner for G by H^* . Let us now proceed to bound from above the ratio between $\Delta(E_r)$ and $\Delta(H^*)$.

Throughout the subsequent analysis, we set $p = 2\sqrt{\log n}/\Delta^{1/4}$ and $M = 4\Delta^{1/4} \cdot \sqrt{\log n}$.

6.1. Handling E^+ . Our first aim is to show that edges in E^+ , i.e., edges with large $|Tri(e)|$, are likely to be 2-spanned in step 2 of our algorithm.

LEMMA 6.2. *With probability at least $1 - 1/n^2$, every edge $e \in E^+$ is 2-spanned in step 2 of Algorithm 5.1.*

Proof. Denote $m = |Tri(e)|$ and assume that $m \geq \sqrt{\Delta}$. Let $Tri(e) = \{T_1, T_2, \dots, T_m\}$ with $T_i = \{e_1^i, e_2^i, e\}$; i.e., the three edges of T_i form a triangle in G . The probability that a triangle T_i does not 2-span e , namely, that neither e_1^i nor e_2^i are selected into the spanner in step 2, is $1 - p^2$. The probability that neither of the triangles 2-span e is

$$(1 - p^2)^m \leq (1 - p^2)^{\sqrt{\Delta}} = \left((1 - p^2)^{1/p^2}\right)^{4 \log n} < \frac{1}{n^4}.$$

(The last inequality follows from the fact that $(1 - x)^{1/x} \leq 1/e$ for $x \leq 1$.) Therefore, the probability that there exists one such an edge which is not 2-spanned is bounded by

$$\frac{|E|}{n^4} \leq \frac{1}{n^2}. \quad \square$$

Next we estimate the maximum degree $\Delta(E_r^1)$ in the graph induced by E_r^1 by proving the following lemma.

LEMMA 6.3. *With probability at least $1 - 1/n^2$, $\Delta(E_r^1) \leq 4 \cdot \Delta^{3/4} \cdot \sqrt{\log n}$.*

Proof. For vertices v such that $\deg(v) < \Delta^{3/4} \cdot \sqrt{\log n}$, the claim follows vacuously. Hence we need to prove a degree bound only for vertices v such that $\deg(v) > \Delta^{3/4} \cdot \sqrt{\log n}$. Let $sp_1(e)$ be the random variable indicating if e was drawn to be in E_r^1 ; namely,

$$sp_1(e) = \begin{cases} 1, & e \in E_r^1, \\ 0 & \text{otherwise.} \end{cases}$$

Let $d_r^1(v)$ denote the random variable that equals the degree of v in the graph induced by E_r^1 . Thus,

$$d_r^1(v) = \sum_{e \in E(v)} sp_1(e).$$

Clearly, $\mathbb{E}(sp_1(e)) = p$ and therefore

$$\mathbb{E}(d_r^1(v)) = \sum_{e \in E(v)} \mathbb{E}(sp_1(e)) = p \cdot \deg(v).$$

(Recall the assumption that $d(v) \geq \sqrt{\log n} \Delta^{3/4}$.) By the definition of p we get that the expected degree of v , $\mu(v)$ is bounded below by $2 \log n \sqrt{\Delta}$. Since $d_r^1(v)$ is a sum

of independent Bernoulli trials, we can apply Lemma 2.2 to it with $\delta = 1$ and deduce that

$$\mathbb{P}(d_r^1(v) > 2 \cdot \mu(v)) \leq \left(\frac{e}{4}\right)^{\mu(v)} < \frac{1}{n^3}.$$

(For the last inequality, recall that one may assume that $\Delta = \Omega(\log^2 n)$. Indeed, here we only need Δ to be bounded below by some constant.)

Therefore, with probability at least $1 - 1/n^3$,

$$d_r^1(v) \leq 4 \cdot \Delta^{3/4} \cdot \sqrt{\log n}.$$

By summing up the probabilities over all the vertices, the lemma follows. \square

(We note that as in [Rag88] slightly better results are attainable; i.e., it is possible to show a degree bound of $2 \cdot \sqrt{\log n} \Delta^{3/4} + o(\Delta^{3/4})$. However, we give the simpler bound here, since in our case we already have an approximation ratio of $\Delta^{1/4}$ and therefore the improvement can only (slightly) affect the constants. A similar situation holds, later, with regards to Lemma 6.4.)

6.2. Handling E^- . We would next like to show that the maximum degree in the subgraph selected by our linear program algorithm is small. Define the variable $d_r^2(v)$ to be the degree of v in the random choices made in steps 6 and 7 of Algorithm 5.1.

LEMMA 6.4. *With probability at least $1 - 1/n^2$, $d_r^2(v) < 2M \cdot \Delta^*$ for every $v \in V$.*

Proof. In order to establish an upper bound on the maximum degree, we must prove that in the random choices of steps 6 and 7 of Algorithm 5.1, the expected number of edges chosen for every vertex is “small.” Let $v_i \in V$ be an arbitrary vertex, and let u_1, \dots, u_m be its neighbors. Let $e_j = (v_i, u_j)$, $j = 1, \dots, m$. We denote by $sp_2(e_j)$ the random indicator variable for the inclusion of e_j in the spanner, in steps 6 and 7; namely,

$$sp_2(e_j) = \begin{cases} 1, & e_j \text{ is chosen to be in the spanner in steps 6 and 7,} \\ 0 & \text{otherwise} \end{cases}$$

and thus

$$d_r^2(v_i) = \sum_{j=1}^m sp_2(e_j).$$

The expected value of $d_r^2(v_i)$ satisfies

$$\mathbb{E}(d_r^2(v_i)) = \sum_{j=1}^m \mathbb{E}(sp_2(e_j)) = \sum_{j=1}^m \min\{1, Mx_{ij}\} \leq M \cdot \sum_{j=1}^m x_{i,j}.$$

By (2) we have

$$\mathbb{E}(d_r^2(v_i)) \leq M \cdot \Delta^*.$$

Since $d_r^2(v_i)$ is the sum of independent Bernoulli trials, it follows from Lemma 2.2 with $\delta = 1$ that

$$\mathbb{P}(d_r^2(v_i) > 2 \cdot M \cdot \Delta^*) < \left(\frac{e}{4}\right)^{M\Delta^*} \leq \frac{1}{n^3}.$$

(The last inequality follows from the assumption that $\Delta \geq \Omega(\log^2 n)$ and from the definition of M .) Summing up the probabilities for all the vertices, the claim follows. \square

In order to bound the number of edges added to each vertex v in step 8 of the algorithm (i.e., when E_u is added to E_r), we have to estimate how many edges are 2-spanned in steps 6 and 7. We show that with high probability, E_u is empty after step 7, and therefore E_u does not change $\Delta(E_r)$.

LEMMA 6.5. *The probability that an edge $e_k = (v_l, v_t)$ of E^- is 2-spanned in steps 6 and 7 is at least $1 - 1/n^4$.*

Proof. For the sake of proving the lemma we need the following technical lemma (cf. Chapter 8 of [AS92]).

LEMMA 6.6. *Let $\{A_i\}_{i=1}^m$ be m independent events and let $\mathbb{P}(A_i) = p_i$ and $\sum_{i=1}^m p_i \geq d$. Then $\mathbb{P}(\bigcup_{i=1}^m A_i) \geq 1 - 1/e^d$. \square*

Let $e_k = (v_l, v_t) \in E^-$ and let $D(e) = \{v_1, \dots, v_d\}$, $d = |Tri(e_k)| \leq \sqrt{\Delta}$. The inequality of the program (P1) corresponding to e_k is

$$x_{l,t} + \sum_{v_i \in D(e_k)} y_{i,k} \geq 1.$$

Let A_i be the event that v_i 2-helps e_k in the chosen spanner. Let I_k be the event that e_k is included in the spanner. We now estimate the probability that A_i occurs. This probability equals the probability that both (v_i, v_l) and (v_i, v_t) are selected to the spanner. Clearly, we have only to consider the case that $\min\{M \cdot x_{i,t}, M \cdot x_{i,l}\} < 1$, for if this is not the case, $\mathbb{P}(A_i) = 1$. In the case both $M \cdot x_{i,t} < 1$ and $M \cdot x_{i,l} < 1$, $\mathbb{P}(A_i) = M^2 \cdot x_{i,t} \cdot x_{i,l}$. In the case that, say $M \cdot x_{i,t} \geq 1$ and $M \cdot x_{i,l} < 1$, we have $\mathbb{P}(A_i) = M \cdot x_{i,l} > M^2 \cdot x_{i,t}^2$. In either case, by (4) of the linear program we have

$$(7) \quad \mathbb{P}(A_i) \geq M^2 \cdot y_{i,k}^2.$$

The event $Cov(e_k) = "e_k \text{ is 2-spanned}"$ is the union $Cov(e_k) = \bigcup_i A_i \cup I_k$; namely, e_k is 2-spanned iff it is in the spanner or is 2-helped by some vertex. We now estimate the sum of probabilities of the A_i and I_k .

CLAIM 6.7. $\mathbb{P}(I_k) + \sum_{i=1}^d \mathbb{P}(A_i) \geq 3 \cdot \log n$.

Proof. We may assume that $x_{l,t} < 1/M$ since otherwise the edge e_k is taken into the spanner with probability 1 and hence is 2-spanned. We therefore have $\sum_{v_i \in D(e_k)} y_{i,k} > 1 - 1/M$.

We now have by (7)

$$\sum_{i=1}^d \mathbb{P}(A_i) \geq \sum_{i=1}^d M^2 y_{i,k}^2 = M^2 \sum_{i=1}^d y_{i,k}^2.$$

By the Cauchy-Schwartz inequality (cf. [Fla85]) we have

$$\begin{aligned} \sum_{i=1}^d \mathbb{P}(A_i) &\geq M^2 \cdot \frac{(\sum_{i=1}^d y_{i,k})^2}{d} \geq M^2 \frac{(\sum_{i=1}^d y_{i,k})^2}{\sqrt{\Delta}} = 4 \log n \left(\sum_{i=1}^d y_{i,k} \right)^2 \\ &\geq 4 \log n (1 - 1/M)^2 > 3 \log n. \end{aligned}$$

(The last inequality holds for $n \geq 2$, in which case $M \geq 4$.) Therefore, this proves our claim. \square

Note that the events I_k, A_i are all *independent*, since I_k and A_i all *concern different edges*. Thus we may apply Lemma 6.6 and get

$$\mathbb{P}(\text{Cov}(e_k)) \geq 1 - \frac{1}{e^{3 \log n}} > 1 - \frac{1}{n^4},$$

proving the lemma. \square

Thus, with probability at least $1 - 1/n^2$, all the edges are 2-spanned in step 7. Therefore, with probability at least $1 - 1/n^2$, $E_u = \emptyset$, in which case E_u does not increase the maximum degree. In summary, Lemmas 6.2, 6.3, 6.4 and 3.2, combined with the above discussion, yield the following theorem.

THEOREM 6.8. *With probability at least $1 - 1/n$, the algorithm produces a 2-spanner with maximum degree bounded by $O(\sqrt{\log n} \Delta^{1/4} \Delta^*)$.* \square

COROLLARY 6.9. *Algorithm 5.1 produces a 2-spanner that with probability at least $1 - 1/n$ is an $O(\sqrt{\log n} \cdot \Delta^{1/4})$ approximation for the LD-2SP problem.* \square

Note that the error probability can be reduced to $1/n^c$ for any (constant) c , losing only constants in the approximation ratio.

7. Derandomization. In this section, we show how to transform our randomized algorithm into a deterministic one. We use the well-known “method of conditional probabilities” (cf. [Spe87]) and its generalization, the method of “pessimistic estimators” [Rag88].

7.1. The method of pessimistic estimators. Let us first describe the method of pessimistic estimators in a form which is convenient for our purpose. Let q_1, \dots, q_l be random Boolean variables, set to 0 or 1 with some probabilities, and consider the probability space $\mathcal{Q} = \{(q_1, \dots, q_l) \mid q_i \in \{0, 1\}, 1 \leq i \leq l\}$ of 2^l points. Let X_1, \dots, X_s be a collection of “bad” events over \mathcal{Q} , and suppose that $\mathbb{P}(X_i) = p_i$ and that $\sum_{i=1}^s p_i < 1$. Thus the event $\bigcap_i \bar{X}_i$ has positive probability. We therefore have a point $(\hat{q}_1, \dots, \hat{q}_l)$ in the probability space \mathcal{Q} for which $\bigcap_i \bar{X}_i$ holds. Suppose that for each event X_i and for each $0 \leq j \leq l$ we have a function $f_j^i(q_1, \dots, q_j)$ for which the following holds.

- (1) $\sum_{i=1}^s f_{j-1}^i(q_1, \dots, q_{j-1}) \geq \min\{\sum_{i=1}^s f_j^i(q_1, \dots, q_{j-1}, 0), \sum_{i=1}^s f_j^i(q_1, \dots, q_{j-1}, 1)\}$ for all $1 \leq i \leq s, 0 \leq j \leq l$.
- (2) $f_j^i(q_1, \dots, q_j) \geq \mathbb{P}(X_i \mid q_1, \dots, q_j)$.
- (3) $\sum_{i=1}^s f_0^i < 1$.
- (4) The function $f_j^i(q_1, \dots, q_j)$ can be computed in polynomial time in l for every i and j and $(q_1, \dots, q_j) \in \{0, 1\}^j$. Also, the number of events, s , is polynomial in l .

In this case one can transform the probabilistic existence proof into a polynomial algorithm (in terms of l). This is done by fixing the value of q_i to be 0 or 1 iteratively, one by one. In the j th step, having determined the values of q_1, \dots, q_{j-1} , we decide upon the value q_j (setting it either to 0 or to 1) so as to minimize the sum $\sum_{i=1}^s f_j^i(q_1, \dots, q_j)$. It easily follows from the above conditions that the sum $\sum_{i=1}^s f_j^i(q_1, \dots, q_j)$ never increases and, consequently (it follows from properties 2 and 3 that), at the end of the procedure we remain with a point $(\hat{q}_1, \dots, \hat{q}_l)$ in the sample space for which the event $\bigcap_i \bar{X}_i$ holds. Also by property (4) above, this derandomization procedure can be executed in time polynomial in l .

The functions f_j^i are called *pessimistic estimators* for the actual conditional probabilities. The method of conditional probabilities is the special case where $f_j^i(q_1, \dots, q_j) = \mathbb{P}(X_i \mid q_1, \dots, q_j)$ and property 3 holds, i.e., the case where in addition to property 3

the conditional probabilities can be computed efficiently, so no estimators are needed. When this is the case, the remaining properties 1 and 2 follow immediately.

7.2. Derandomizing the 2-spanner algorithm. In the case of the 2-spanner problem we have a two-stage randomized procedure. Let us first focus on the harder task of derandomizing the second stage, where we 2-span the edges of E^- using randomized rounding. We draw the edges with different probabilities (that depend upon the values of the corresponding variables in the linear program). Our q_i variables, therefore, correspond to the edges, where every edge has some probability to be 1. We will identify the edge e_k with its corresponding random variable. (We will therefore say “ e_k was set to 1” meaning that e_k was added to the spanner.)

We now describe the “bad events” in the second stage. The event $\mathcal{D}(v_i)$ is the event where the degree of v_i is greater than $2 \cdot M \cdot \Delta^*$. The event $\mathcal{U}(e_j)$ is the event that an edge $e_j \in E^-$ is unspanned at the end of Algorithm 5.1.

Throughout the sequel, we assume that we have already decided upon the values of e_1, \dots, e_{j-1} , setting them either to 1 or 0, and we want to decide the value of e_j . We denote by $p(e_k)$ the probability by which e_k is drawn in the randomized rounding. (This probability equals M times the value of e_k in the linear program.) We use the following notation (that depends upon previous decisions). The number \tilde{e}_k is defined as

$$\tilde{e}_k = \begin{cases} 1, & e_k \text{ was previously set to 1,} \\ 0, & e_k \text{ was previously set to 0,} \\ p(e_k), & \text{the value of } e_k \text{ was not yet determined.} \end{cases}$$

We next define the pessimistic estimator for the event $\mathcal{U}(e_j)$. Given some edge $e_k = (v_s, v_t)$ say that e_k lies on $t_k = |\text{Tri}(e_k)|$ triangles and denote $\text{Tri}(e_k) = \{(e_1^1, e_1^2, e_k)\}_{i=1}^{t_k}$. We then set the pessimistic estimators for $\mathcal{U}(e_k)$ to be

$$h_j^k(e_1, \dots, e_j) = \prod_{i=1}^{t_k} (1 - \tilde{e}_i^1 \cdot \tilde{e}_i^2).$$

We note that the above expression, *exactly* equals the probability that e_k is unspanned by either of its triangles (given the previous decisions). Note that e_k may be 2-spanned also if e_k itself is drawn into the spanner. This observation proves property 2 for the functions $\mathcal{U}(e_k)$ and $h_j^k(e_1, \dots, e_j)$. Property 4 follows trivially.

We now turn our attention to the event $\mathcal{D}(v_i)$, meaning that the degree of v_i exceeds $2 \cdot M \cdot \Delta^*$. For these events, the conditional probabilities can be calculated in polynomial time using *dynamic programming*. However, this is relatively time consuming, as $O(d(v)\Delta^*)$ time is required in order to calculate each conditional probability associated with every vertex. It is therefore convenient to introduce the following pessimistic estimators, which are a special case of some estimators introduced in [Rag88]. Suppose that the edges of v_i are $e_1^i, e_2^i, \dots, e_{d_i}^i$ (where d_i is the degree of v_i). For $\mathcal{D}(v_i)$ define the following pessimistic estimator. Set

$$g_j^i(e_1, \dots, e_j) = \frac{\prod_{r=1}^{d_i} (\tilde{e}_r^i + 1)}{4^{M \cdot \Delta^*}}.$$

The required property 2 follows in a way similar to the proof of the Chernoff bound, as in [Rag88]. Property 4 also follows trivially.

We now prove property 3. Note that since (as in [Rag88])

$$\begin{aligned} g_0^i &= \frac{\prod_{k=1}^{d_i} (p(e_k) + 1)}{4^{M \cdot \Delta^*}} \leq \frac{\prod_{k=1}^{d_i} e^{p(e_k)}}{4^{M \cdot \Delta^*}} = \frac{e^{\sum_{k=1}^{d_i} p(e_k)}}{4^{M \cdot \Delta^*}} \\ &\leq \left(\frac{e}{4}\right)^{M \cdot \Delta^*} \leq \frac{1}{n^3}, \end{aligned}$$

property 3 follows from Lemmas 6.4 and 6.5. (In fact, the sum of the pessimistic estimators of the bad events is not only smaller than 1 but is also smaller than $1/n$. Nevertheless, the degrees of the vertices can only be reduced by a constant factor.)

Finally, we have to check property 1. We have

$$S = \sum_{i=1}^n g_{j-1}^i + \sum_{k=1}^m h_{j-1}^k = \sum_{i=1}^n \frac{\prod_{r=1}^{d_i} (\tilde{e}_r^i + 1)}{4^{M \cdot \Delta^*}} + \sum_{k=1}^m \prod_{i=1}^{t_k} (1 - \tilde{e}_i^1 \cdot \tilde{e}_i^2).$$

We can write S as the sum $S = S_1(e_j) + S_2(e_j) + S_3$ where

$$S_1(e_j) = (1 + p(e_j)) \cdot \Pi_1, \quad S_2(e_j) = \sum_l (1 - p(e_j) \cdot \tilde{e}_l) \cdot \Pi_2^l,$$

and the expressions Π_1 , Π_2^l , and S_3 do not contain \tilde{e}_j . When setting e_j to 1, the difference between the terms in the first summand is $S_1(1) - S_1(e_j) = (1 - p(e_j)) \cdot \Pi_1$, and the difference in the second summand is $S_2(1) - S_2(e_j) = -\sum_l \tilde{e}_l (1 - p(e_j)) \cdot \Pi_2^l$. So if $\sum_l \tilde{e}_l \cdot \Pi_2^l \geq \Pi_1$ we are done. Otherwise, when setting e_j to 0, the difference in the first summand is $S_1(0) - S_1(e_j) = -p(e_j) \cdot \Pi_1$ and in the second summand, $S_2(0) - S_2(e_j) = \sum_l p(e_j) \cdot \tilde{e}_l \cdot \Pi_2^l < p(e_j) \cdot \Pi_1$. Thus the required property follows.

Now we have to consider the first (and easier to derandomize) stage of the algorithm, where we handle the edges of E^+ . In the first stage, we draw all the edges independently and uniformly with probability $p = 2 \cdot \sqrt{\log n} / \Delta^{1/4}$. The bad event here is similar. We have the event $\mathcal{D}(v_i)$ which is the bad event that the degree of v_i exceeds $4 \cdot \Delta^{3/4} \cdot \sqrt{\log n}$. The event $\mathcal{U}(e_k)$ is the event that an edge $e \in E^+$ is not 2-spanned at the end of Algorithm 5.1.

Thus, the derandomization of this first stage is a special case of the derandomization of the second stage.

We have therefore established the following result.

COROLLARY 7.1. *Algorithm 5.1, together with a derandomization procedure, produces a 2-spanner that is an $O(\sqrt{\log n} \cdot \Delta^{1/4})$ approximation for the LD-2SP problem. \square*

8. An algorithm for sparse graphs. In this section we present a relatively simple algorithm *Sparse*₁ that performs better than Algorithm 5.1 in the case where the underlying graph is sparse. In this case, algorithm *Sparse*₁ yields a $2\sqrt{E}$ additive approximation. (By “ α additive approximation” we mean that the resulting degree is $\Delta^* + \alpha$.) Thus, if the number of edges is up to $n^{3/2}$, we get an additive term of less than $n^{3/4}$ (or, alternatively, a very low multiplicative factor). For the range $n^{3/2} \leq E \leq n^{7/4}$ we have a different algorithm *Sparse*₂ that slightly improves Algorithm 5.1 in the worst case. This algorithm is considerably more complicated and is therefore omitted. The interested reader is referred to [KP93].

8.1. Algorithm *Sparse*₁. Algorithm *Sparse*₁ divides the vertex set into “heavy” and “light” vertices. The set *Heavy* consists of vertices with degrees at least \sqrt{E} , and

the set *Light* consists of vertices whose degrees are *less than* \sqrt{E} . Then the set of all edges with both endpoints in *Heavy*, $E(\text{Heavy})$, and all edges with both endpoints in *Light*, $E(\text{Light})$, are taken into the spanner. The cut edges, with one endpoint in *Heavy* and the other in *Light*, are 2-spanned using a linear programming formulation. The output edge set is denoted E_r . Again we may assume that Δ^* (or, more accurately, Δ_f^*) is known in advance.

ALGORITHM 8.1. *Algorithm Sparse₁*

Input: A graph $G = (V, E)$.

1. Let $\text{Heavy} = \{v \in V \mid \deg(v) \geq \sqrt{E}\}$, $\text{Light} = \{v \in V \mid \deg(v) < \sqrt{E}\}$.
2. Add $E(\text{Heavy}) \cup E(\text{Light})$ into E_r .
3. Let E_u be the set of cut edges having one endpoint in *Heavy* and one in *Light*.
4. Solve the fractional relaxation of the program (P1) corresponding to E_u , E_r , and Δ^* .
5. Let $\{x_{i,j}, y_{i,k}\}$ be the optimum (fractional) solutions corresponding to (P1). For every variable $\hat{x}_{i,j}$ create a respective random variable $\bar{x}_{i,j}$.
6. Randomly and uniformly set $\bar{x}_{i,j}$ to be 1 with probability $\min\{1, 4 \log n \cdot x_{i,j}\}$.
7. If $\bar{x}_{i,j}$ is set to 1, then add the edge (v_i, v_j) to E_r .
8. Add all the remaining non 2-spanned edges to E_r and output E_r .

8.2. Analysis. We now prove that Algorithm 8.1 yields a $2\sqrt{E}$ additive approximation. We first note the following simple fact.

FACT 8.2. *In step 2 of Algorithm 8.1, we add to E_r no more than $2\sqrt{E}$ edges adjacent to any vertex.*

Proof. The claim is clear for vertices v in *Light*, since such a vertex has at most \sqrt{E} adjacent edges. Also note that $|\text{Heavy}| \leq 2 \cdot \sqrt{E}$, and thus for a vertex $v \in \text{Heavy}$ at most $2 \cdot \sqrt{E}$ edges are candidates for addition to E_r in this step. The claim follows. \square

We now note the following simple yet crucial fact.

FACT 8.3. *In every triangle corresponding to a cut edge e in the set E_u defined in step 3 of Algorithm 5.1, exactly one of its edges was already added to E_r in Step 2.*

Proof. Every such triangle contains an edge e' that is not a cut edge. Thus, either both vertices of e' belong to *Heavy* or they both belong to *Light*. In either case e' was added to E_r in step 2. \square

Given some cut edge $e_k = (v_l, v_t)$, let $y_{i,k}$ be a variable corresponding to $v_i \in D(e_k)$ and e_k . Without loss of generality, let (v_i, v_l) be the other cut edge in the triangle. Thus, the probability that v_i 2-helps e_k exactly equals $\min\{1, 4 \log n \cdot x_{i,l}\} \geq \min\{1, 4 \log n \cdot y_{i,k}\}$. Thus a proof along the lines of that in Lemma 6.5 (but simpler, due to the fact that here we can avoid the “squaring” effect) gives the next corollary. The main point is that the sum of the probabilities that the triangles of e_k survive is roughly $4 \log n \sum_i y_{ik} = \Omega(\log n)$. (The squaring effect is avoided, since in any triangle we need only one edge to survive and not two edges together, because the other edge of every triangle was already chosen to the spanner.)

COROLLARY 8.4. *With probability at least $1 - 1/n^2$ all the edges in E_u are 2-spanned by the end of step 7 of Algorithm 8.1.* \square

We note that the expectation of the degree of a vertex (and thus, using Lemma 2.2, the expectation of the maximum degree) is bounded by $2\sqrt{E} + O(\log n \cdot \Delta^*)$. Thus up to logarithmic factors, this approximation is $2\sqrt{E}$ additive. By using derandomization (as explained in the previous section) and Lemma 3.2 we have the following.

COROLLARY 8.5. *There is an $\tilde{O}(\sqrt{E/\Delta})$ approximation algorithm for the LD-2SP problem.* \square

Note that if $E \leq \Delta\sqrt{\Delta}$ then Algorithm 8.1 performs better than Algorithm 5.1. Let us consider the approximation ratio in terms of n . Algorithm 5.1 gives a worst-case ratio of $\tilde{O}(n^{1/4})$. This happens when Δ is very large, i.e., $\Delta = \Theta(n)$. On the other hand, if Δ is very large, we may have $\sqrt{|E|/\Delta}$ small. This implies that the ratio is improved whenever $|E| < n^{3/2}$ by doing the following. Considering a graph with $n^{3/2-\epsilon}$ edges, if $\Delta \leq n^{1-2\epsilon/3}$ then apply Algorithm 5.1, else apply Algorithm 8.1. This gives the following bound.

COROLLARY 8.6. *For every $0 \leq \epsilon \leq 1/2$, there exists an $\tilde{O}(n^{1/4-\epsilon/6})$ approximation algorithm for the LD-2SP problem on graphs G with $E = O(n^{3/2-\epsilon})$.* \square

This result improves the ratio in “absolute terms” (i.e., in terms of n). For example, if $E = O(n)$ then the combined algorithm has an $O(n^{1/6})$ approximation ratio (whereas Algorithm 5.1 would give an $O(n^{1/4})$ ratio in the worst case).

8.3. The case of $n^{3/2} < E \leq n^{7/4}$. In [KP93] we present Algorithm 8.1 which outperforms Algorithm 5.1 in the range $n^{3/2} < E \leq n^{7/4}$. That is, we assume a graph with $O(n^{7/4-\epsilon})$ edges, where $0 \leq \epsilon \leq 1/4$, and show an $\tilde{O}(n^{1/4-\epsilon/11})$ -ratio approximation algorithm, improving over the $n^{1/4}$ -ratio of Algorithm 5.1.

THEOREM 8.7 (see [KP93]). *Given a graph $G = (V, E)$ with $n^{7/4-\epsilon}$ edges, Algorithm 8.1 combined with a derandomization procedure has an $\tilde{O}(n^{1/4-\epsilon/11})$ approximation ratio.* \square

9. Spanning the edges of a single vertex. In this section we consider the weaker problem of spanning the edges adjacent to a single vertex and present an $O(\log n)$ approximation for it. This construction can easily be applied to span the edges adjacent to a small subset of the vertices.

Say that we are given a specific vertex v (presumably with high degree) and we want to 2-span its edges (and do not care about the edges not touching v). Thus our aim is to select some subset $E' \subseteq E$ inducing low degrees such that every missing edge of v is 2-spanned by a triangle. Denote this problem by SLD-2SP.

First we note that the lower bound on approximability for the LD-2SP problem applies to the SLD-2SP problem as well. Given an instance of the set cover problem, we may construct \bar{G} as in section 4, and consider the problem of spanning the edges of s . A similar proof as in section 4 shows that unless $NP \subset DTIME(n^{\log \log n})$, the ratio of any approximation algorithm for the SLD-2SP problem is no better than $\ln n/5$. On the other hand, in this section we match this result, showing a logarithmic-ratio approximation algorithm for SLD-2SP. We use a known greedy approximation for a (slightly) more involved version of the set cover problem. This essentially shows that this weaker problem SLD-2SP is equivalent to set cover, with respect to approximation.

The *bounded-load* set cover problem is a variant of the set cover problem that deals with assigning specific covering vertices to the covered vertices. Namely, along with finding a cover C of V_2 , it is required to provide a function $\varphi : V_2 \rightarrow C$, assigning each vertex v_2 in V_2 a neighbor $\varphi(v_2)$ in C . The load of a vertex $v_1 \in V_1$ is defined as the number of covered vertices it is assigned to, i.e., $L(v_1) = |\{v_2 \in V_2 \mid \varphi(v_2) = v_1\}|$. The problem is now defined as follows. Given a bipartite graph $G(V_1, V_2, E)$ and an integer $L \leq |V_2|$, find a cover $C \subset V_1$ of V_2 and an assignment φ with maximum load bounded by L (i.e., such that no vertex in V_1 is assigned to more than L vertices of V_2).

We recall the following theorem of [Wol82].

THEOREM 9.1 (see [W82]). *The bounded load set cover problem can be approximated with ratio $O(\log |V_2|)$.* \square

Given an instance of the SLD-2SP problem, where our aim is to 2-span the edges of v , we reduce it to an instance of the bounded load set cover problem. We use a reformulation of the SLD-2SP problem as follows. Model the neighbors $N(v)$ of v and the edges $E(v)$ of v in a bipartite graph $Bip = (N(v), E(v), A)$ where a vertex $u \in N(v)$ is connected to an edge $e = (v, w) \in E(v)$, iff $(u, w) \in E$. (Namely, $(u, e) \in A$ iff u belongs to $D(e)$ and can 2-help $e = (v, w) \in E(G)$ in the spanner by the edges (u, w) and (u, v) .)

The aim is to find a *small-sized* cover C of $E(v)$ (in Bip), with *small* maximum load. The merit of this construction is explained by the following observation.

CLAIM 9.2. *Given a small set C covering $E(v)$ with maximum load L , it is possible to construct a 2-spanner of the edges of v with maximum degree bounded by $\max\{|C|, L + 1\}$.*

Proof. Construct the 2-spanner as follows. Define the bipartite graph Bip , and let $C = \{w_1, \dots, w_k\} \subseteq N(v)$. Let $\{e_1^i, \dots, e_{n_i}^i\}$ be the edges incident to v that are covered in Bip by w_i (where $n_i \leq L$). Let $e_j^i = (v, z_j^i)$. Add the edges $\{(v, w_i)\}_{i=1}^k$ to the spanner. Also, add the edges (w_i, z_j^i) . Clearly, we have added $|C|$ edges adjacent to v (since one edge is added to v for each vertex of C). We have also added no more than $L + 1$ edges adjacent to w_i for every i , since w_i covers no more than L edges $e_j = (w_i, z_j^i)$ in the spanner and also the edge (v, w_i) . Thus the SLD-2SP problem is equivalent to finding a cover C and some assignment with load L , minimizing $\max\{|C|, L + 1\}$. \square

It easily follows from Theorem 9.1 that the problem of finding a cover with a load assignment minimizing $\max\{|C|, L + 1\}$ also enjoys a logarithmic approximation. In turn, this gives a logarithmic approximation ratio for the problem of spanning the edges of v .

COROLLARY 9.3. *The problem of spanning the edges of v with low degree has a polynomial time approximation algorithm with ratio $O(\log n)$. Conversely, the problem cannot be approximated with ratio better than $\ln n/5$, unless $NP \subset DTIME(n^{\log \log n})$ holds.* \square

As an additional by-product, if we are required to 2-span *only* the edges of a collection of k of the graph vertices, for small k , we can 2-span the edges of every vertex in the set one by one and get an $O(k \log n)$ approximation for this problem.

COROLLARY 9.4. *The problem of 2-spanning the edges of a subset $V' \subset V$ of k vertices with minimum maximum degree can be approximated within an $O(\log n \cdot k)$ ratio.* \square

REFERENCES

- [ABP91] B. AWERBUCH, A. BARATZ, AND D. PELEG, *Efficient Broadcast and Light-Weight Spanners*, 1991, manuscript.
- [ADDJ90] I. ALTHÖFER, G. DAS, D. DOBKIN, AND D. JOSEPH, *Generating sparse spanners for weighted graphs*, in Proc. 2nd Scandinavian Workshop on Algorithm Theory, Vol. LNCS-447, Springer-Verlag, Berlin, New York, 1990, pp. 26–37.
- [AS92] N. ALON AND J. SPENCER, *The Probabilistic Method*, John Wiley and Sons Inc., New York, 1992.
- [BGLR93] M. BELLARE, S. GOLDWASSER, C. LUND, AND A. RUSSELL, *Efficient probabilistically checkable proofs: Applications to approximation*, in Proc. 25th ACM Symposium on the Theory of Computing, 1993, pp. 294–304.
- [Cai91] L. CAI, *Tree 2-spanners*, Technical Report 91-4, Simon Fraser University, Burnaby, BC, 1991.

- [CDNS92] B. CHANDRA, G. DAS, G. NARASIMHAN, AND J. SOARES, *New sparseness results on graph spanners*, in Proc. 8th ACM Symposium on Computational Geometry, 1992.
- [Che52] H. CHERNOFF, *A measure of asymptotic efficiency for tests of hypothesis based on the sum of observations*, Ann. Math. Statist., 23 (1952), pp. 493–507.
- [Che86] L. P. CHEW, *There is a planar graph almost as good as the complete graph*, in ACM Symposium on Computational Geometry, 1986, pp. 169–177.
- [DFS87] D. P. DOBKIN, S. J. FRIEDMAN, AND K. J. SUPOWIT, *Delaunay graphs are almost as good as complete graphs*, in Proc. 31st IEEE Symp. on Foundations of Computer Science, 1987, pp. 20–26.
- [DJ89] G. DAS AND D. JOSEPH, *Which triangulation approximates the complete graph?* in International Symposium on Optimal Algorithms, Vol. LNCS-401, Springer-Verlag, Berlin, New York, 1989, pp. 168–192.
- [Fei96] U. FEIGE, *A threshold of $\ln n$ for approximating set cover*, in Proc. ACM STOC, 1996.
- [Fla85] H. FLANDERS, *Calculus*, W.H. Freeman & Co., San Francisco, CA, 1985.
- [HW56] G. H. HARDY AND E. M. WRIGHT, *An Introduction to the Theory of Numbers*. Oxford University Press, London, 1956.
- [Joh74] D. S. JOHNSON, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278.
- [Kar84] N. KARMARKAR, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
- [Kha80] L. G. KHACHIAN, *Polynomial algorithms in linear programming*, Zh. Vychisl. Mat. i Mat. Fiz. (1980), pp. 53–72.
- [KP93] G. KORTSARZ AND D. PELEG, *Generating Low-Degree 2-Spanners*, Technical Report CS93-07, The Weizmann Institute, Rehovot, Israel, 1993.
- [KP94] G. KORTSARZ AND D. PELEG, *Generating sparse 2-spanners*, J. Algorithms, 17 (1994), pp. 222–236.
- [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, Vol. LNCS-401, Springer-Verlag, New York, Berlin, 1989, pp. 9–13.
- [Lov75] L. LOVÁSZ, *On the ratio of integral and fractional covers*, Discrete Math., 13 (1975), pp. 383–390.
- [LR94] A. L. LIESTMAN AND D. RICHARDS, *Degree-constrained pyramid spanners*, J. Parallel Distr. Comput., 25 (1995), pp. 1–6.
- [LS93a] A. L. LIESTMAN AND T. C. SHERMER, *Additive graph spanners*, Networks, 23 (1993), pp. 343–364.
- [LS93b] A. L. LIESTMAN AND T. C. SHERMER, *Grid spanners*, Networks, 23 (1993), pp. 123–133.
- [LY93] C. LUND AND M. YANNAKAKIS, *On the hardness of approximating minimization problems*, in Proc. 25th ACM Symposium on the Theory of Computing, 1993, pp. 286–293, 1993.
- [PS89] D. PELEG AND A. SCHÄFFER, *Graph spanners*, J. Graph Theory, 13 (1989), pp. 99–116.
- [PU89] D. PELEG AND J. D. ULLMAN, *An optimal synchronizer for the hypercube*, SIAM J. Comput., 18 (1989), pp. 740–747.
- [Rag88] P. RAGHAVAN, *Probabilistic construction of deterministic algorithms: Approximating packing integer programs*, J. Comput System Sci., 37 (1988), pp. 130–143.
- [RT87] P. RAGHAVAN AND C. D. THOMPSON, *Randomized rounding: A technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.
- [Soa92] J. SOARES, *Approximating Euclidean Distances by Small Degree Graphs*, Technical Report 92-05, University of Chicago, 1992.
- [Spe87] J. SPENCER, *Ten Lectures on the Probabilistic Method*, SIAM, Philadelphia, 1987.
- [Wol82] L. A. WOLSEY, *An analysis of the greedy algorithm for the submodular set covering problem*, Combinatorica, 2 (1982), pp. 385–393.