Approximating k-Spanner Problems for $k > 2$

Michael Elkin * David Peleg †

January 13, 2005

Abstract

Given a graph $G = (V, E)$, a subgraph $G' = (V, H)$, $H \subseteq E$ is a *k-spanner* of G if for any pair of vertices $u, w \in V$ it satisfies $d_H(u, w) \leq k \cdot d_G(u, w)$. The *basic k-spanner* problem is to find a k-spanner of a given graph G with the smallest possible number of edges. This paper considers approximation algorithms for this and some related problems for $k > 2$, known to be $\Omega(2^{\log^{1-\mu} n})$-inapproximable. The basic k-spanner problem over undirected graphs with $k > 2$ has been given a sublinear ratio approximation algorithm (with ratio roughly $O(n^{2/(k+1)})$), but no such algorithms were known for other variants of the problem, including the directed and the client-server variants, as well as for the related k-DSS problem. We present the first approximation algorithms for these problems with sublinear approximation ratio. The second contribution of this paper is in characterizing some wide families of graphs on which the problems do admit a logarithmic and a polylogarithmic approximation ratios. These families are characterized as containing graphs that have optimal or "near-optimal" spanners with certain desirable properties, such as being a tree, having low arboricity or having low girth. All our results generalize to the directed and the client-server variants of the problems. As a simple corollary, we present an algorithm that given a graph G builds a subgraph with $\tilde{O}(n)$ edges and stretch bounded by the *tree-stretch* of G, namely the minimum maximal stretch of a spanning tree for G. The analysis of our algorithms involves the novel notion of *edge-dominating systems* developed in the paper. The technique introduced in the paper reduces the studied algorithmic approximability questions on k-spanners to purely graph-theoretical questions concerning the existence of certain combinatorial objects in families of graphs.

*Department of Computer Science, Yale University, New Haven, CT 06520, USA. E-mail: elkin@sluggo.cs.yale.edu.

†Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, 76100 Israel. E-mail: david.peleg@weizmann.ac.il. Supported in part by grants from the Israel Science Foundation and the Israel Ministry of Science and Art.

1 Introduction

1.1 Motivation

Spanners for general graphs were introduced in [23, 21], and were shown to be the underlying graph structure in a number of constructions in distributed systems and communication networks. Spanners of Euclidean graphs have turned out to be relevant also in the area of computational geometry (cf. [4, 8, 2]).

Given a graph $G = (V, E)$ and a subgraph $G' = (V, E')$, $E' \subseteq E$, let

$$stretch(G') = \max \left\{ \frac{d_{G'}(u, w)}{d_G(u, w)} \mid u, w \in V \right\},$$

where $d_H(u, w)$ denotes the distance between u and w in H. The subgraph G' is a *k-spanner* of G if $stretch(G') \leq k$. The combinatorial problem of finding the sparsest k-spanner of a given graph G (in terms of number of edges) is called the (basic) *k-spanner problem*. We refer to k as the *stretch parameter* of the problem.

The k-spanner problem is NP-hard, which makes it interesting to study its approximability properties. For $k = 2$, the approximation ratio on arbitrary n-vertex graphs is known to be $\Theta(\log n)$, under suitable complexity theoretic assumptions for the lower bound [18, 17]. For $k > 2$, however, the situation is not as good. To begin with, it is known that under the assumption that $NP \not\subseteq DTIME(n^{\text{polylog } n})$, the approximation ratio of any polynomial time algorithm for the k-spanner problem, for any constant $k > 2$ and for any $0 < \mu < 1$, is $\Omega(2^{\log^{1-\mu} n})$ [11]. On the positive side, it is known that every n-vertex graph has a (polynomial-time constructible) k-spanner of size $O(n^{1+\frac{2}{k+1}})$ for odd k and $O(n^{1+\frac{2}{k+2}})$ for even k [21, 16]. Since any spanner must contain at least $n - 1$ edges, the algorithm for constructing such spanners can be thought of as a "universal" approximation algorithm for the problem, with ratio $O(n^{\frac{2}{k+1}})$ for odd k and $O(n^{\frac{2}{k+2}})$ for even k. This performance is poor for small constant values of k, e.g., for $k = 3$ it yields an $O(n^{1/2})$-approximation algorithm.

The situation is even worse for some related edge deletion problems. For instance, the *directed* k-spanner problem for $k > 2$ (namely, the problem on directed graphs) has no known sublinear approximation ratio. In particular, it is known that for certain n-vertex directed graphs, *any* k-spanner requires $\Omega(n^2)$ edges, hence the problem does not enjoy a "universal" sublinear ratio approximation algorithm in the above sense.

The same applies to the *k-diameter spanning subgraph* (or *k-DSS*) problem, defined as follows. Given a graph $G = (V, E)$, a subgraph $G' = (V, H)$, $H \subseteq E$ is a *k-DSS* of G if the distance in H between any two vertices $u, w \in V$ satisfies $d_H(u, w) \leq k$. The k-DSS problem calls for finding the sparsest k-DSS of a given graph G (in terms of number of edges). Again, while the problem is $O(\log n)$-approximable for $k = 2$ [13], the results of [21, 10] indicate that for $k > 2$ no "universal" sublinear ratio approximation algorithm for the (directed or undirected) k-DSS problem exists.

A third example involves the *client-server (CS)* k-spanner problem [12]. This is a generalization of the basic k-spanner problem in which every edge may be either a client, a server, or both a client and a server. The goal is to k-span all the client edges by server edges, using a minimal number of server edges. In the *all-client (AC)* variant of the CS k-spanner problem, all the edges are clients,

and in the *all-server (AS)* variant, all the edges are servers. All of these problems are known to be $O(\log n)$-approximable for $k = 2$ [12] and $\Omega(2^{\log^{1-\mu} n})$-inapproximable for $k > 2$ and any $0 < \mu < 1$ [10]. Moreover, with the exception of the AS k-spanner problem, none of these problems currently enjoys a sublinear ratio approximation algorithm.

1.2 Our results

The current paper is concerned with two main directions. The first involves obtaining sublinear ratio approximation algorithms for a number of the edge deletion problems discussed above. In particular, the paper presents $\tilde{O}(n^{2/3})$-approximation algorithms for the directed 3-spanner, the (directed and undirected) CS 3-spanner and the (directed and undirected) 3-DSS problems. (We use the notation $\tilde{O}(f(n)) = O(f(n)\text{polylog }(n))$.) In fact, our approximation algorithm usually provides a better ratio, and, in particular, we show that for graphs with $O(n^{1+\beta})$ edges, the algorithm has an $\tilde{O}(n^{\frac{\beta+1}{3}})$-approximation ratio.

The second direction aims at developing a better understanding for the causes of the apparent difficulty of the k-spanner problem, by identifying specific parameters which affect its approximability. Specifically, our approach to this problem is based on examining various restricted graph classes with special properties, and attempting to approximate the problem on these classes.

The families of graphs we consider are characterized as containing graphs that have optimal or "near-optimal" spanners with certain properties. Intuitively, we prove that if the given input graph G has a "near-optimal" spanner H of some convenient structure, then it is possible to find a "relatively good" spanner for G (namely, one which is close to H in sparsity).

As a first and most extreme example, we consider the family of graphs that enjoy a *tree k-spanner*, namely, a k-spanner in the form of a tree. Let $S_{TREE}(G)$ denote the minimum k such that G has a tree k-spanner. Finding a tree attaining $S_{TREE}(G)$ is known to be NP-hard even restricted to planar graphs [15]. The problem of computing $S_{TREE}(G)$ for a given graph G is known to be $(1+\sqrt{5})/2$-inapproximable [22], and no nontrivial approximation algorithm for the problem is known. Denote by $SP_k(TREE)$ the family of graphs that admit a tree k-spanner.

The k-spanner problem restricted to the class $SP_k(TREE)$, namely, the problem of finding a tree k-spanner in a graph known to possess one, was shown in [6] to be polynomially solvable for $k = 2$ and NP-complete for $k \geq 4$. In Section 5.2 we present an algorithm providing an $O(\min\{k^2 \log n, \frac{\log^3 n}{(\log\log n)^2}\})$-approximation ratio for the problem on $SP_k(TREE)$ for arbitrary (not necessarily constant) values of k. In particular, for any graph G this algorithm builds a subgraph H of G with $\tilde{O}(n)$ edges and $stretch(H) \leq S_{TREE}(G)$.

We next turn to wider classes of graphs, enjoying a spanner with low arboricity or low girth. For any graph G, a k-spanner H is called a *near-optimal* k-spanner of G if any other k-spanner H' of G satisfies $\frac{|H|}{|H'|} = O(\text{polylog } n)$. Denote by $SP_k(PL)$ the family of graphs that admit a planar k-spanner. Denote by $SP_k(BA)$ (respectively, $SP_k(\log A)$) the family of graphs that admit a near-optimal k-spanner with arboricity (or genus) bounded by a constant (respectively, by polylog n). For any fixed integer g denote by $SP_k(GIRTH(g))$ the family of graphs that admit a near-optimal k-spanner with girth no smaller than g.

In Section 5.1 we present an algorithm providing an $O(\log n \cdot a(H))$-approximation ratio for the 3-spanner problem on general graphs, where $a(G')$ is the arboricity of the graph G' and H

is the optimal 3-spanner of the input graph. It follows that the problem admits a logarithmic approximation ratio when restricted to graphs in the class $SP_3(BA)$ (and in particular $SP_3(PL)$), and a polylogarithmic ratio when restricted to graphs in $SP_3(\log A)$.

All the above results can be easily adapted to the k-DSS problem as well. In particular, define the graph family $DSS_k(TREE)$ (respectively, $DSS_k(PL)$, $DSS_k(BA)$, $DSS_k(\log A)$ or $DSS_k(GIRTH(g))$) analogously, as the set of graphs that admit a k-DSS in the form of a tree (resp., which is planar, with arboricity bounded by a constant, with arboricity bounded by polylog n, or with girth at least g). In Section 6 we present an $O(\min\{k^2 \log n, \frac{\log^3 n}{(\log\log n)^2}\})$-approximation algorithm for the k-DSS problem on $DSS_k(TREE)$, and an $O(\log n \cdot a(H))$-approximation algorithm for the 3-DSS problem on general graphs, where H is the optimal 3-DSS of the input graph, yielding a logarithmic approximation over the class $DSS_3(BA)$ and a polylogarithmic ratio over $DSS_3(\log A)$.

For problems whose definition involves a free parameter (like the parameter k in all the above problems), it is instructive to study their approximability as a function of this parameter. In particular, a set of problems $\{\Pi_p\}$ indexed by a parameter p is said to enjoy the *ratio degradation property* with respect to the parameter if the approximability of the problem Π_p decreases exponentially fast with the inverse of the parameter p. The result of [21] shows that the basic k-spanner problem enjoys the ratio degradation property with respect to the stretch requirement k, whereas the results of [10, 13] show that the CS, the AC and the directed k-spanner problems, as well as the k-DSS problem, do not enjoy it (with respect to k).

We analyze the behavior of the 3-spanner and the 3-DSS problems over the graph classes $SP_3(GIRTH(g))$ and $DSS_3(GIRTH(g))$. Formally, let 3-spanner(g) (resp., 3-DSS(g)) be the 3-spanner (resp., 3-DSS) problem restricted to the family $SP_3(GIRTH(g))$ (resp., $DSS_3(GIRTH(g))$). We show that the problem families $\{\text{3-spanner}(g)\}_{g=1}^{\infty}$ and $\{\text{3-DSS}(g)\}_{g=1}^{\infty}$ enjoy the ratio degradation property with respect to this parameter. All the results mentioned above generalize to the directed and the client-server variants of the problems.

In the final two sections of the paper we derive some additional results. Section 7 concerns bicriteria approximation algorithms. It presents a bicriteria $(O(n^{1/2}), +2)$-approximation algorithm for the AC k-spanner and k-DSS problems. In other words, the algorithm produces an AC $(k+2)$-spanner (respectively, $(k+2)$-DSS subgraph) which is greater by a factor of at most of $O(n^{1/2})$ than an optimal AC k-spanner (k-DSS subgraph). We also present a bicriteria $(O(n^{1/d}), (1+\epsilon, \beta(d,\epsilon)))$-approximation algorithm for the AC k-spanner problem and k-DSS problem for any $\epsilon > 0$ and any positive integer d. In other words, the algorithm produces an AC $((1+\epsilon)\cdot k + \beta(d,\epsilon))$-spanner (resp., $((1+\epsilon)\cdot k + \beta(d,\epsilon))$-DSS subgraph) which is greater by a factor of at most $O(n^{1/d})$ than an optimal AC k-spanner (resp., k-DSS subgraph). The additive term $\beta(d,\epsilon)$ is at most $O(d^{\log\log d - \log\epsilon})$ and thus, it is constant whenever d and ϵ are. Note that the k-DSS and the AC k-spanner problems remain $\Omega(2^{\log^{1-\mu} n})$-inapproximable for any $\mu > 0$, for $k = O(n^{1-\delta})$ for any $\delta > 0$ [13]. For high values of k (e.g., $\theta(n^{\delta})$ for some $\delta > 0$) our algorithm is a bicriteria $(O(n^{1/d}), 1+\epsilon)$-approximation algorithm for the AC k-spanner and k-DSS problems for *any* constant $\epsilon > 0$ and constant positive integer d. To the best of our knowledge these problems constitute the first examples of $\Omega(2^{\log^{1-\mu} n})$-inapproximable problems that admit bicriteria $(O(n^{1+\eta}), 1+\epsilon)$-approximation for *any* constant $\epsilon, \eta > 0$. These results are based on the constructions of sparse $(1+\epsilon, \beta)$-spanners due to [14]. These algorithms can be interpreted also as $O(n^{1/2})$-approximation algorithms for the k-DSS problem

restricted to the graphs of diameter at most $k-2$ and as $O(n^{1/d})$-approximation algorithms for the k-DSS problem restricted to the graphs of diameter at most $\frac{k-\beta(d,\epsilon)}{1+\epsilon}$ for any $\epsilon > 0$ and positive integer d. We also prove an analogous statement for the AC k-spanner problem.

Finally, in Section 8 we consider the k-spanner problem on the class $SP_k(DEG(\Delta))$ of graphs that admit a k-spanner with maximum degree bounded by Δ, and provide an algorithm with $O(\Delta^{k-2} \cdot \log n)$ approximation ratio, for $k > 2$. Note, however, that the k-spanner problem enjoys a trivial $O(\Delta^k)$-approximation algorithm, hence the new algorithm is advantageous only when $\Delta = \Omega(\sqrt{\log n})$.

1.3 Our techniques

Generally speaking, our algorithms generalize the logarithmic approximation algorithm of [18] for the 2-spanner problem. That algorithm is based on decomposing the problem into a finite number of appropriate subproblems and iteratively solving them, where every such subproblem involves performing some *density* computations over a small neighborhood in the graph.

However, as discussed earlier, the k-spanner problem for $k > 2$ is significantly more difficult than the 2-spanner case. Indeed, the k-spanner problem for $k > 2$ is $\Omega(2^{\log^{\epsilon} n})$-inapproximable [11], whereas the approximability of the 2-spanner problem is $\Theta(\log n)$ [18, 17]. (For the k-DSS problem the situation is analogous [13].) Hence a generalization of the algorithm for the 2-spanner problem to the k-spanner one requires the introduction of novel algorithmic and analytic techniques.

The technique introduced here for handling these problems involves a new graph construct called *edge-dominating system*, based on a special type of graph decomposition. Using these systems, we define an algorithmic procedure (for density computation) which is applied to each component of this decomposition, and gives rise to an approximation algorithm for the k-spanner problem. We demonstrate that the approximation ratio of our algorithm equals the *sparsity* of the edge-dominating system used by the algorithm, up to a factor logarithmic in n.

Our approach thus reduces the algorithmic question of the approximability of the k-spanner problem to the pure graph-theoretic question of existence of combinatorial objects with desirable properties (namely, edge-dominating systems with bounded sparsity). In particular, we show that a proof of existence of an edge-dominating system for the 3-spanner problem of sparsity bounded by some power $0 < \delta < 1$ of the arboricity of some near-optimal 3-spanner leads to an approximation algorithm for the 3-spanner problem within an approximation ratio of $\tilde{O}(n^{\frac{3\delta}{2(2+\delta)}})$ (the current state of the art is $O(n^{1/2})$ by the universal construction of [14, 9]; note that the state of the art of the δ-edge-dominating systems is $\delta = 1$ which unfortunately yields a similar $\tilde{O}(n^{1/2})$-approximation ratio). We also show that the lower bound of [11] on the approximability of the 3-spanner problem yields a lower bound on the sparsity of the possible edge-dominating systems.

Consequently, we present some constructions of edge-dominating systems. To illustrate the concept, we start by presenting a construction of constant sparsity for the 2-spanner problem on general graphs. This yields an $O(\log n)$-approximation algorithm for the 2-spanner problem (which is, in fact, simply a more general presentation of the algorithm of [18] and its analysis). We proceed with presenting a construction of constant sparsity (i.e., not depending on n, but depending on k) for the k-spanner problem on $SP_k(TREE)$. This construction yields an $O(k^2 \log n)$-approximation algorithm for the k-spanner problem on $SP_k(TREE)$. Finally, we present a construction for gen-

eral graphs for $k = 3$ (for the 3-spanner problem), but the sparsity of this construction is linear in the arboricity of a near-optimal spanner of the input graph. This construction yields logarithmic and polylogarithmic approximation ratio algorithms for the 3-spanner problem on $SP_3(BA)$ (in particular, on $SP_3(PL)$) and on $SP_3(\log A)$, respectively, and $n^{O(1/g)}$-approximation ratio on $SP_3(GIRTH(g))$.

It is hoped that our techniques may enable in the future to get an improvement for the basic 3-spanner too, and possibly an $o(n^{2/3})$-approximation ratio for the aforementioned problems. Another challenging direction is to generalize our algorithms in such a way that they would provide to a sublinear approximation ratio for these problems for $k > 3$.

2 Density computation

Throughout, we denote the number of vertices in the graph G by n. The *girth* of graph G, denoted $g(G)$, is the length of the shortest cycle in the graph. For a set of nodes W let $E(W)$ be the set of edges with both endpoints in W. The *arboricity* of graph G is defined as $a(G) = \max\left\{\frac{|E(W)|}{|W|-1}\right\}$. We will use Scheinerman's theorem, stating that $a(G) = O(\sqrt{\mu(G)})$ [24], where $\mu(G)$ denotes the *genus* of graph G.

The Nash-Williams theorem (cf. [3]) states that the edge set of a graph G with arboricity t can be decomposed into t edge-disjoint forests, and furthermore, that this decomposition can be computed polynomially. The notion of *graph density*, denoted by $\rho(G)$, is very similar to arboricity, except that it has $|W|$ instead of $|W| - 1$ in the denominator. Note that the relation between the two notions is

$$\rho(G) \leq a(G) \leq 2 \cdot \rho(G), \tag{1}$$

which follows immediately from the fact that for any vertex set W of size greater than 1,

$$\frac{|W|}{2} \leq |W| - 1 \leq |W|.$$

Definition 2.1 *For every graph G and subgraph H for G, the vertex $v \in V$ is said to* (H,k)-dominate *the edge $e = (u, z)$ if H contains a path $P = (u_1, \ldots, u_r, z)$ of $r + 1 \leq k$ edges connecting u and z and going through v, or formally, s.t. $(u, u_1), (u_1, u_2), .., (u_{r-1}, u_r), (u_r, z) \in H$ and $v \in \{u, u_1, u_2, .., u_r, z\}$.*

The algorithm that we present in Section 4 constructs the spanner subgraph denoted H. The algorithm proceeds in steps, in each step increasing the initially empty set H. Through the algorithm H^u denotes the set of edges that are still uncovered.

For any vertex v and any two subsets $H, H^u \subseteq E$ of the edge set E, define $COV_k(H, v, H^u)$ as the subset of H^u edges that are (H, k)-dominated by v.

Define the *ψ_k-density* of a subset of edges H and a vertex v with respect to an edge set H^u as

$$\psi_k(H, v, H^u) \triangleq \frac{|COV_k(H, v, H^u)|}{|H|} .$$

Intuitively, a high-density edge subset H is a good candidate for participating in the spanner, as it covers many edges at a low cost. For a graph $G = (V, E)$, a subset of vertices $W \subseteq V$ and a vertex

$v \in W$, define a *breadth-first search (BFS) tree* rooted at v for the set W to be a tree T rooted at v, whose vertex set coincides with W and such that for any node $u \in W$, $d_T(v,u) = d_G(v,u)$. For a vertex set U and a vertex v, let $T(U,v)$ denote the set of all possible non-empty BFS trees rooted at v and contained in $E(U)$, and let $\hat{T}(U,v)$ denote the set of all possible non-empty trees rooted at v and contained in $E(U)$. Now define the ψ_k^l-density (respectively, $\hat{\psi}_k^l$-density) of a node v with respect to an edge set H^u to be the maximum density achievable by a BFS tree (respectively, arbitrary tree) rooted at v and spanning some part of v's l-neighborhood, i.e.,

$$\psi_k^l(v,H^u) \stackrel{\Delta}{=} \max_{T \in T(\Gamma_l(v),v)} \{\psi_k(T,v,H^u)\} \ , \tag{2}$$

$$\hat{\psi}_k^l(v,H^u) \stackrel{\Delta}{=} \max_{\hat{T} \in \hat{T}(\Gamma_l(v),v)} \left\{\psi_k(\hat{T},v,H^u)\right\} \ . \tag{3}$$

The following lemma shows that in order to approximate $\hat{\psi}_k^l(v,H^u)$ it suffices to compute the value of $\psi_k^l(v,H^u)$.

Lemma 2.2 *For any vertex $v \in V$ and edge set $H^u \subseteq E$ and for any integers $k,l > 1$,*

$$\psi_k^l(v,H^u) \leq \hat{\psi}_k^l(v,H^u) \leq l \cdot \psi_k^l(v,H^u) \ .$$

Proof: The first inequality is obvious, since $T(\Gamma_l(v),v) \subseteq \hat{T}(\Gamma_l(v),v)$. To prove the second, consider a tree T that maximizes the value of $\hat{\psi}_k^l$, i.e., such that

$$\hat{\psi}_k^l(v,H^u) = \psi_k(T,v,H^u) = \frac{cov_k(T,v,H^u)}{|T|} \ .$$

(Recall that *cov* denotes the size of the set COV.) Let T' be a BFS tree rooted at v that spans the vertex set of T. It follows that $|T'| \leq l|T|$, because T' is of depth l and to span any vertex of $V(T)$ it may use at most l new nodes. It remains to show that $cov_k(T,v,H^u) \leq cov_k(T',v,H^u)$, and hence

$$\hat{\psi}_k^l(v,H^u) \leq \frac{cov_k(T',v,H^u)}{|T'|/l} \leq l \cdot \hat{\psi}_k^l(v,H^u) \ .$$

We show this by establishing that $COV_k(T,v,H^u) \subseteq COV_k(T',v,H^u)$. To see this, let $(u,z) \in COV_k(T,v,H^u)$. Then by definition of COV, there exist nodes $u_1,..,u_r$ with $1 \leq r < k$ such that $(u,u_1),(u_1,u_2),\ldots,(u_j,v),(v,u_{j+1}),\ldots,(u_r,z) \in T$. But $d_{T'}(u,v) \leq d_T(u,v)$ and $d_{T'}(v,z) \leq d_T(v,z)$, since T' is a BFS tree, and thus there is a path from u to z of length smaller than or equal to k in T' that passes through the vertex v. Therefore $(u,z) \in COV_k(T',v,H^u)$. ∎

Lemma 2.3 *For any vertex v and subset of edges H^u, and for any integers $k > 1$ and $m \geq \lceil k/2 \rceil$,*

$$\psi_k^{\lceil k/2 \rceil}(v,H^u) = \psi_k^m(v,H^u) \ .$$

Proof: The inequality $\psi_k^{\lceil k/2 \rceil}(v,H^u) \leq \psi_k^m(v,H^u)$ is obvious. For the opposite direction, consider the tree T such that

$$\psi_k^m(v,H^u) = \psi_k(T,v,H^u) \ .$$

Truncate the tree at level $\lceil k/2 \rceil$, i.e., such that all the nodes of this level become leaves and their subtrees are deleted. Denote the obtained tree by T'. Clearly, $|T'| \leq |T|$. We now prove that T'

spans the same set of edges as T. Suppose for contradiction that some edge (u,w) is k-spanned through the vertex v by the tree T but is not k-spanned by T'. Hence either u or w is located at distance greater than $\lceil k/2 \rceil$ from the vertex v. Note that since T is a BFS-tree, the distances in G are equal to the distances in T. Without loss of generality, suppose that $d_T(v,u) > \lceil k/2 \rceil$. Since T spans the edge (u,w), the path from u to w through v is of length smaller than or equal to k, so $d_T(v,u) + d_T(v,w) \leq k$. Hence $d_T(v,w) < \lfloor k/2 \rfloor$. But then G contains a path from v to u through w of length smaller than $\lfloor k/2 \rfloor + 1 \leq \lceil k/2 \rceil$, implying $d_G(v,u) \leq \lceil k/2 \rceil$. This is in contradiction to the assumptions that $d_T(v,u) > \lceil k/2 \rceil$, and T is a BFS-tree rooted at v. ▮

Combining Lemmas 2.2 and 2.3, we have

Lemma 2.4 *For any integer $k > 1$, vertex $v \in V$ and subset of edges $H^u \subseteq E$,*

$$\psi_k^{\lceil k/2 \rceil}(v, H^u) \leq \hat{\psi}_k^{k-1}(v, H^u) \leq (k-1)\psi_k^{\lceil k/2 \rceil}(v, H^u) \ .$$

We denote by T_v the tree that maximizes the value of $\psi_k(\hat{T}, v, H^u)$ and by $COV_k(v, H^u)$ the set $COV_k(T_v, v, H^u)$. Specifically, our algorithm needs to compute the value of $\psi_k^l(v, H^u)$, where $l = \lceil k/2 \rceil$. Observe that an l-deep tree T automatically $2l$-spans all the edges between its vertices.

Furthermore, we next show that the density $\psi_k^l(v, H^u)$ can be approximated with a ratio linear in l in polynomial time on *any graph*.

Hereafter we fix $l = \lceil k/2 \rceil$ and denote the function ψ_k^l by ψ. Also we denote by small letters the sizes of sets denoted by capital letters. We denote $\hat{\Gamma}_l(v) = \{u \mid d_G(u,v) \leq l\}$ and $\Gamma_l(v) = \{u \mid d_G(u,v) = l\}$.

Consider the subgraph $\tilde{G} = (\tilde{V}, \tilde{E})$, where $\tilde{V} = \hat{\Gamma}_l(v)$ and

$$\tilde{E} = \begin{cases} E(\hat{\Gamma}_l(v)) \cap H^u, & k \text{ is even}, \\ E(\hat{\Gamma}_l(v)) \cap (H^u \setminus E(\Gamma_l(v))), & k \text{ odd}. \end{cases} \tag{4}$$

The following close relation holds between $\rho(\tilde{G})$ and $\psi(v, H^u)$.

Lemma 2.5 $\rho(\tilde{G}) \leq \psi(v, H^u) \leq 2 \cdot \rho(\tilde{G})$.

Proof: We start with showing that $\psi(v, H^u) \leq a(\tilde{G})$, which will imply the second inequality of the lemma by (1).

Let T_v be a partial BFS spanning tree of $\hat{\Gamma}_l(v)$ that maximizes the value of $\psi(v, H^u)$. Since T_v is a BFS tree, for every $u, z \in T_v$ we have

$$d_{T_v}(v,u) + d_{T_v}(v,z) = d_G(v,u) + d_G(v,z) \leq k.$$

The last inequality follows by our definition of $\tilde{E}$ and the choice of l. This implies

$$COV_k(T_v, v, H^u) \subseteq \tilde{E} \ .$$

Hence

$$\psi(v, H^u) = \psi_k(T_v, v, H^u) = \frac{cov_k(T_v, v, H^u)}{|T_v|} \leq \frac{|\tilde{E}|}{|V(T_v)| - 1} \leq a(\tilde{G}) \ ,$$

completing this direction.

For the first inequality of the lemma, let $U \subseteq \hat{\Gamma}_l(v)$ be the set that maximizes $\rho(\tilde{G})$, i.e.,

$$\rho(\tilde{G}) \;=\; \frac{|E(U)|}{|U|} \;.$$

For every $i \geq 1$, denote $U_i = \Gamma_i(v) \cap U$.

We build a partial spanning tree of U denoted $T(U)$ in an iterative fashion, by connecting v to all the vertices of U_1, choosing for every vertex $u_2 \in U_2$ one neighbor in U_1, and continuing in this way for $i = 3, \ldots, l$. Note that if $v \in U$ then $|T(U)| = |U| - 1$ and otherwise $|T(U)| = |U|$. In any case, $|T(U)| \leq |U|$. Hence

$$\rho(\tilde{G}) \;=\; \frac{|E(U)|}{|U|} \leq \frac{|E(U)|}{|T(U)|} \;=\; \frac{cov_k(T(U), v, H^u)}{|T(U)|} \;=\; \psi_k(T(U), v, H^u) \leq \psi(v, H_u) \;,$$

completing the proof. ∎

By Lemmas 2.2 and 2.5 we have the following.

Corollary 2.6 $\rho(\tilde{G}) \leq \hat{\psi}(v, H^u) \leq 2l \cdot \rho(\tilde{G})$.

Lemma 2.7 *Given a graph $G = (V, E)$, a vertex v in G and a subset of edges $H^u \subseteq E$, the value of $\psi(v, H^u)$ can be approximated with ratio $2l$ in polynomial time.*

Proof: Computing of the density $\rho(\tilde{G})$ can be done in polynomial time (see [19]). The second direction of the proof of Lemma 2.5 provides a constructive way for building a partial tree $T(U)$, which is a a $2l$-approximation to the tree T that maximizes $\psi_k(T, v, H^u)$. ∎

3 Constructions for dominating systems

3.1 Dominating systems

At this point we introduce the notion of *(H,k)-dominating systems* which will be used in the analysis.

Definition 3.1 *For a graph G and a spanner H of G, an* H-system *for G is a pair (D, S) such that $D \subseteq V$ and $S = \{Sp(v)\}_{v \in D}$, where $Sp(v)$ is an edge set contained in H for every $v \in D$.*

The H-system (D, S) is called an (H,k)-dominating system *for the graph $G = (V, E)$ if for every edge $e \in E$ there exists a vertex $v \in D$ such that v $(Sp(v), k)$-dominates e.*

Our construction for k-spanners makes use of a specific type of (H,k)-dominating system, in which the set $Sp(v)$ is a subtree of the BFS tree rooted at v, of depth at most $k-1$. To define it formally, we need the following terminology.

For a non-root vertex v in a tree T, let $p_T(v)$ denote its parent node in T and $Sub^s_T(v)$ denote the edge set of the s-deep subtree rooted by v. Also let r_T denote the root of the tree T and L_T denote its set of leaves.

In order to build a good spanner, we need a "sparse" (H,k)-dominating system. The sparsity of an (H,k)-dominating system (D, S) is defined as

$$Sparsity_H(D, S) \;\triangleq\; \frac{\sum_{v \in D} sp(v)}{|H|} \;.$$

3.2 $(H,2)$-dominating systems

To illustrate the concept of dominating systems we present a simple construction of an $(H,2)$-dominating system with constant sparsity for arbitrary graphs. This construction can be used to simplify the analysis of the logarithmic ratio approximation algorithm for the 2-spanner problem due to [18].

Construction A

1. $D \leftarrow V$.
2. For every vertex v in D set $Sp(v) = \{(u,v) \in E\}$.
3. $S \leftarrow \{Sp(v)\}_{v \in D}$.

Lemma 3.2 *For any 2-spanner H of a graph G, the H-system (D,S) constructed by Construction A is an $(H,2)$-dominating system with $Sparsity_H(D,S) = 2$.*

Proof: Consider some edge $e = (u,w) \in E$. H is a 2-spanner and so either $e \in H$ or there exists a vertex v such that the edges $(u,v),(v,w) \in H$. In the first case $e \in Sp(u)$ and in the second $e \in Sp(v)$. Finally, note that $sp(v) = deg_H(v)$. Hence

$$Sparsity(D,S) = \frac{\sum_{v \in V} deg_H(v)}{|H|} = 2 \ . \qquad \blacksquare$$

3.3 (H,k)-dominating systems for $SP_k(TREE)$

Now we show that if a graph G admits a tree-spanner H then it has a sparse (H,k)-dominating system, i.e., with $Sparsity_H(D,S) = O(k)$.

Assume that G has a tree spanner H. Consider the following construction of an H-system for G.

Construction B

1. $D \leftarrow V \setminus L_H$.
2. For every vertex v in D, set $Sp(v)$ to be the depth $(k-1)$ subtree of H rooted at v plus the edge from v to its parent (unless v is the root), i.e.,

$$Sp(v) = \begin{cases} \{(p_H(v),v)\} \cup Sub_H^{k-1}(v), & v \neq r_H \\ Sp(v) = Sub_H^{k-1}(v), & v = r_H \end{cases}$$

3. Define S to be the set $\{Sp(v)\}_{v \in D}$.

Lemma 3.3 *For any tree-k-spanner H of a graph G, the H-system (D,S) constructed by Construction B is an (H,k)-dominating system with $Sparsity_H(D,S) \leq k$.*

Proof: To prove the first claim, consider an edge $e = (u_1,u_2) \in E \setminus H$. Since H is a k-spanner for the graph G and e is not in H,

$$1 < d_H(u_1,u_2) \leq k.$$

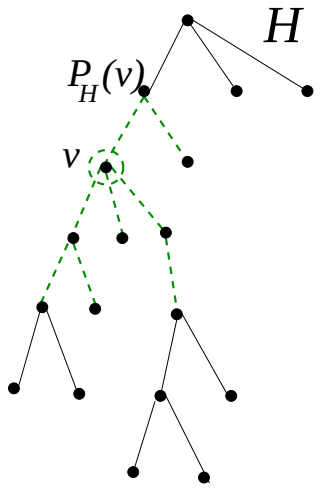

Figure 1: The set $Sp(v)$ defined for v by Construction B; here $k = 3$.

The discussion is now split into two cases: either one of the two nodes is an ancestor of the other, or they both have a least common ancestor u.

In the first case, suppose w.l.o.g. that u_1 is an ancestor of u_2. Then let u be the child of u_1 that is located on the path from u_1 to u_2 in H (since H is a tree, there is only one such path, so u is well-defined). Observe that by the construction, $Sp(u)$ k-spans the edge e, because it contains all the edges of the path between u_1 and u_2 in H.

In the second case, note that both $d_H(u_1, u), d_H(u_2, u) < k$. Hence the whole path from u_1 to u_2 in H is contained in $Sub^k_H(v)$, hence e is spanned by $Sp(u)$. This completes the proof that (D, S) is an (H,k)-dominating system.

To prove that $Sparsity_H(D, S) \leq k$ we consider an edge $e = (u_1, u_2)$ in the tree H. Suppose w.l.o.g. that $u_1 = p_H(u_2)$. Observe that by the construction, the edge e participates only in the subtrees $Sp(v)$ for vertices v which are ancestors of u_2 and which satisfy $d_H(v, u_2) \leq k$. For any node u_2 in the graph there are at most k such vertices. Hence every edge $e \in H$ participates in at most k sets $Sp(v)$, completing the proof. ▮

3.4 $(H, 3)$-dominating systems for $SP_3(BA)$

We now generalize the above construction of the (H,k)-dominating system from tree spanners to spanners with bounded arboricity, albeit only for $k = 3$.

By the Nash-Williams theorem, every graph H with arboricity $a = a(H)$ has a decomposition in to a edge-disjoint forests $F_1, .., F_a$, where for every $i = 1, .., a$, the forest F_i is a collection of vertex-disjoint trees, i.e., $F_i = \{T_i^1, .., T_i^{j_i}\}$. Since the forests are edge-disjoint, every edge participates in exactly one tree. Since the trees in each forest are vertex-disjoint, each vertex may participate in at most one tree per forest. Hence each vertex participates in at most a trees.

Consider the following construction. For every vertex v, denote by $T_i(v)$ the tree in the forest F_i in which v participates. An edge $e = (w, u) \in F_j$ is said to *cross* the node v in F_i, for $i \neq j$, if u or w is a neighbor of v in tree $T_i(v)$, but e does not belong to F_i.

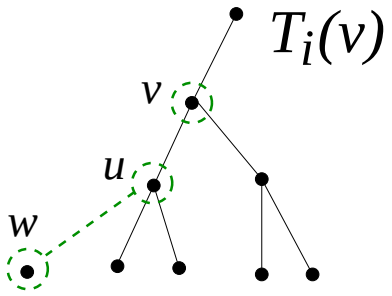

Figure 2: The edge (w, u) crosses the node v in F_i.

Essentially, this construction is based on repeating Construction B for every tree T_i^j individually, and adding the crossing edges. A formal description follows.

Construction C

1. Set $D = V$.
2. For every vertex v set
$$Sp(v) \leftarrow \bigcup_i (\{(p_{T_i(v)}(v), v)\} \cup Sub_{k-1}^{T_i(v)}(v)) \ .$$
(If v is a root of T_i^j then the edge $\{(p_{T_i^j}(v), v)\}$ is not included.)
3. For every vertex v, add to $Sp(v)$ also every edge that crosses v in F_i for some $i = 1, .., a$.
4. Define S to be the set $\{Sp(v)\}_{v \in D}$.

Remark 1: Note that the edges that do not cross to a different tree are taken as well for the children of v but not for its parent. For the parent, we are not interested in taking all its neighboring edges in its own tree, but only the edges that cross between different trees.

Remark 2: For a neighbor w of v the edge (w, z) is said to cross to a different tree if the edge (v, w) is located on the different tree than (w, z). This is well-defined, since each edge belongs to exactly one tree.

Lemma 3.4 *For any graph G with a 3-spanner H, the H-system (D, S) constructed by Construction C satisfies the following two properties:*

1. *(D, S) is a $(H, 3)$-dominating system for the graph G,*
2. *$Sparsity_H(D, S) \leq O(a(H))$,*

Proof: To prove (1) consider some edge $e = (u_1, u_2) \in E \setminus H$. Since H is a 3-spanner for the graph G and e is not in H, we have
$$1 < d_H(u_1, u_2) \leq 3.$$
If there exists a spanning path that consists of edges of the tree T_i^j only, then the system $(H, 3)$-dominates the edge e, by the same argument as in the proof of Lemma 3.3.

So next suppose that there exist two trees T_i^j and $T_{i'}^{j'}$ whose union contains the spanning path P of the edge $e = (u, z)$. If the path P is of length two, then let v be the intermediate vertex (i.e., $P = \{(u, v), (v, z)\}$). It is easy to verify that the vertex v $(H, 3)$-dominates the edge e. So suppose that P is of length 3. Consider the following two cases.

Case 1: The path P contains a subpath P' of length two in one tree, T_i^j, and an edge in another tree, $T_{i'}^{j'}$. Denote by v the intermediate vertex of the subpath P'. I.e., P is of the form $((u, v), (v, w), (w, z))$, where $P' = ((u, v), (v, w))$ is contained in T_i^j, and the edge (w, z) is in $T_{i'}^{j'}$. In this case the vertex v $(H, 3)$-dominates the edge e. Indeed, the edges (u, v) and (v, w) are inserted into $Sp(v)$ on Step 2 of Construction C, and the edge (w, z) is added on Step 3.

Case 2: The path is of the form $P = ((u, v), (v, w), (w, z))$, where the edges (u, v) and (w, z) belong to the first tree T_i^j and the edge (v, w) belongs to the second tree $T_{i'}^{j'}$. In this case too,

v $(H, 3)$-dominates the edge e, by the same considerations as in Case 1.

It is also possible that there are three trees, T_i^j, $T_{i'}^{j'}$ and $T_{i''}^{j''}$, whose union contains the spanning path of the edge $e = (u, z)$. Using the same notation, we assume w.l.o.g. that $(u, v) \in T_i^j$, $(v, w) \in T_{i'}^{j'}$ and $(w, z) \in T_{i''}^{j''}$. Again, by the same considerations as previously, v $(H, 3)$-dominates the edge e.

This completes the proof that the H-system (D, S) is an $(H, 3)$-dominating system.

It remains to prove (2). Denote the arboricity of H by a. As in the proof of Lemma 3.3, when we treat one individual tree and build the edge sets $Sp(v)$, each edge is counted at most three times. Also note that throughout Step 2, each node in the graph is touched by different edge sets $Sp(v)$ at most $4a$ times.

Now we observe that on Step 3 of Construction C, an edge e is taken into some edge set $Sp(v)$ only if one of its endpoints is touched by $Sp(v)$ after Step 2. Each of the endpoints of e is touched at most $4a$ times, hence overall, e could be inserted into at most $8a$ edge sets $Sp(v)$. Therefore overall, each edge of the spanner H could be joined to at most $8a + 3$ edge sets, thus yielding

$$\sum_{v \in D} |Sp(v)| \leq (8a + 3) \cdot |H| \ ,$$

as required. ∎

4 Spanner construction algorithm

This section presents an algorithm for building a k-spanner for a given graph $G = (V, E)$. The algorithm is a modification of the 2-spanner approximation algorithm devised in [18]. Its main part is a loop, repeated while there is at least one vertex with positive ψ-density.

Inside the loop, we pick a vertex v that maximizes the ψ-density, and compute for v the corresponding tree T_v that attains it. The algorithm is described formally next.

Algorithm *Sparse_Spanner*

Input: graph $G = (V, E)$, integer $k \geq 2$.

Output: subset $H \subseteq E$.

1. $H^u \leftarrow E$; $H^c \leftarrow \phi$; $H \leftarrow \phi$; $l \leftarrow \lceil k/2 \rceil$

2. While $\exists v$ s.t. $\hat{\psi}(v, H^u) > 0$ do :

   (a) Choose a vertex v that maximizes $\hat{\psi}(v, H^u)$.

   (b) Approximate $\hat{\psi}(v, H^u)$ with ratio $O(l)$ for this vertex;
   let T_v be the corresponding densest tree of $\hat{\Gamma}_l(v)$.

   (c) $H^c \leftarrow H^c \cup COV_k(T_v, v, H^u)$

   (d) $H \leftarrow H \cup T_v$

   (e) $H^u \leftarrow H^u \setminus H^c$

3. Return(H)

It can be easily seen that if the $\hat{\psi}(v, H^u)$-density is zero for every vertex v in the graph, then H^u is empty. Thus H^c contains all the graph edges. Since an edge is inserted into H^c only when it is 3-spanned by some path contained in of the spanner H, and no edges are removed from H, it follows that the algorithm leaves the loop and returns H only when H is a 3-spanner for G.

The termination of the algorithm follows from the fact that in each iteration, at least one edge is removed from H^u, hence the algorithm terminates after at most $|E|$ iterations.

In what follows, we analyze the algorithm as if it computes the density $\hat{\psi}(v, H^u)$ exactly, rather than approximates it. Later we show that approximating the density by a factor of ρ instead of computing it exactly decreases the approximation ratio of the algorithm by the same factor ρ only.

5 Analysis of the spanner construction algorithm

5.1 Analysis of the 3-spanner case

In this section we prove that Algorithm Sparse_Spanner provides an approximation ratio of $\tilde{O}(a(H^*))$ for the basic 3-spanner problem, where H^* is an optimal 3-spanner for the given input graph G.

For every $j \geq 1$, let H_j be the spanner at the beginning of the jth iteration, let H_j^c and H_j^u be the corresponding sets of covered and uncovered edges, let v_j be the vertex chosen in the jth iteration, and let $T_j = T_{v_j}$ be the corresponding tree of neighbors selected, i.e., the $\hat{\psi}$-densest partial spanning tree of $\Gamma_2(v_j)$.

Observe that since H^u decreases at every step, the value of $\hat{\psi}(v, H^u)$ is monotonically decreasing as well. Let us partition the iterations of the algorithm into phases as follows. Let $r = \frac{|E|}{|V|}$ and $f = \lceil \log r \rceil$. The first phase includes all iterations during which each vertex v_j chosen by the algorithm satisfies

$$\hat{\psi}(v_j, H^u) \geq \frac{r}{2} .$$

For $2 \leq i \leq f$, let the ith phase consist of the iterations during which the chosen v_j satisfies $\frac{r}{2^{i-1}} > \hat{\psi}(v_j, H^u) \geq \frac{r}{2^i}$.

Following [12], denote by P_i the set of indices of iterations in the ith phase. Let $H[i]$ and $H^c[i]$ denote the sets of new edges added to H and H^c, respectively, during the ith phase, and let $H^u[i]$ denote the set of edges left in H^u at the end of the ith phase. Also denote the spanner at the end of the ith phase by $\hat{H}[i] = \bigcup_{j \leq i} H[j]$.

Observe that for every $v \in V$,

$$\hat{\psi}(v, H^u[i]) < \frac{r}{2^i}, \tag{5}$$

because before the first iteration j of the $(i+1)$st phase, $H^u_j = H^u[i]$ and so the vertex v_j chosen at this iteration satisfies $\hat{\psi}(v_j, H[i]) < \frac{r}{2^{i+1-1}} = \frac{r}{2^i}$, and this v_j maximizes $\hat{\psi}$.

Let X_i be the set of vertices chosen during the ith phase. For $v \in X_i$, denote by $P_i(v)$ the subset of P_i containing only those ith phase iterations at which v was chosen, i.e., $P_i(v) = \{j \in P_i \mid v_j = v\}$. Let

$$\begin{aligned} H[i,v] &= \bigcup_{j \in P_i(v)} T_j \ , \\ H^c[i,v] &= \bigcup_{j \in P_i(v)} COV_3(T_j, v, H^u_j) \ . \end{aligned}$$

Since for any integer $2 \leq i \leq f$, vertex $v \in V$ and $j \in P_i(v)$ the set $COV_3(T_j, v, H^u_j)$ includes only edges from H^u and only edges from some set $COV_3(T_j, v, H^u_j)$ are put into H^c and removed from H^u, and since edges taken to T_j are inserted into H we have that

$$\begin{aligned} h[i,v] &\leq \sum_{j \in P_i(v)} |T_j| \ , \\ h^c[i,v] &= \sum_{j \in P_i(v)} cov_3(T_j, v, H^u_j) \ . \end{aligned}$$

Therefore we can state the following lemma.

Lemma 5.1 *For every $v \in X_i$,*

$$h^c[i] \geq h[i] \cdot \frac{r}{2^i}.$$

We omit its proof, since it is analogous to the proof of Lemma 5.1 in [12].

Now let $\bar{H}$ be some 3-spanner for G and $\bar{H}_i$ be (a possibly Steiner) 3-spanner for $H^u[i]$ of size no greater than $\bar{H}$ and satisfying $a(\bar{H}_i) \leq a(\bar{H})$. Specifically, $\bar{H}_i$ is allowed to use all E edges and not only those of $H^u[i]$. This implies the existence of such a spanner, since in particular $\bar{H}$ itself spans $H^u[i]$.

Lemma 5.2

$$h^u[i]/\bar{h} \leq O(a(\bar{H})) \cdot \frac{r}{2^{i-1}} \ .$$

Proof: Let (D, S) be the $(\bar{H}_i, 3)$-dominating system for $H^u[i]$ that satisfies

$$\sum_{v \in D} |Sp(v)| \leq O(a(\bar{H}_i)) \cdot |\bar{H}_i| \ . \tag{6}$$

Such a system exists by Lemma 3.4. By Definition (3.1), for every edge $e \in H^u[i]$ there exists a vertex v that 3-dominates the edge. Hence

$$h^u[i] \leq \sum_{v \in D} cov_3(Sp(v), v, H^u[i]) \ .$$

Since $\bar{h}_i \leq \bar{h}$, it follows that

$$\frac{h^u[i]}{\bar{h}} \leq \frac{h^u[i]}{\bar{h}_i} \leq \frac{\sum_{v \in D} cov_3(Sp(v), v, H^u[i])}{\bar{h}_i} \ .$$

Now by (6) we get

$$\begin{aligned}\frac{h^u[i]}{\bar{h}} &\leq O(a(\bar{H}_i)) \cdot \frac{\sum_{v\in D} cov_3(Sp(v), v, H^u[i])}{\sum_{v\in D} |Sp(v)|} \\ &\leq O(a(\bar{H})) \cdot \max_{v\in D}\left\{\frac{cov_3(Sp(v), v, H^u[i])}{|Sp(v)|}\right\} \\ &= O(a(\bar{H})) \cdot \hat{\psi}_3(Sp(v_0), v_0, H^u[i]) \ ,\end{aligned}$$

for some specific vertex v_0 that maximizes $\hat{\psi}_3(Sp(v), v, H^u[i])$. Thus

$$\frac{h^u[i]}{\bar{h}} \leq O(a(\bar{H})) \cdot \hat{\psi}(v_0, H^u[i]) \leq O(a(\bar{H})) \cdot \frac{r}{2^i} \ ,$$

using (5). ∎

The following Lemma is analogous to Lemma 4.4 of [18].

Lemma 5.3 *For every* $1 \leq i \leq f$, $h[i]/\bar{h} < O(a(\bar{H}))$.

Proof: We first prove the claim for $i = 1$. We may assume w.l.o.g that $n \geq 2$. By Lemma 5.1,

$$\frac{h[1]}{\bar{h}} \leq \frac{\frac{2}{r} \cdot h^c[1]}{\bar{h}} \leq \frac{\frac{2\cdot|V|}{|E|} \cdot |E|}{\bar{h}} \leq \frac{2 \cdot |V|}{|V| - 1} \leq 4 \ .$$

We now prove the claim for $i > 1$. By the fact that $H^c[i] \subseteq H^u[i-1]$, we have

$$\frac{h[i]}{\bar{h}} \leq \frac{\frac{2^i}{r} \cdot h^c[i]}{\bar{h}} \leq \frac{2^i}{r} \cdot \frac{h^u[i-1]}{\bar{h}} \ .$$

Using the previous lemma we get

$$\frac{h[i]}{\bar{h}} < O(a(\bar{H})) \cdot \frac{2^i}{r} \cdot \frac{r}{2^{i-1}} = O(a(\bar{H})) \ . \qquad ∎$$

Lemma 5.4 $h/\bar{h} = O(a(\bar{H}) \cdot \log r)$.

Proof: Now, by the previous lemma and the choice of f,

$$\frac{h}{\bar{h}} = \frac{\sum_{i=1}^{f} O(a(\bar{H}))h[i]}{\bar{h}} \leq O(a(\bar{H})) \cdot f = O(a(\bar{H}) \cdot \log r). \qquad ∎$$

Finally, we observe that exactly as in [18, 12] it suffices to approximate the densities instead of computing them precisely. This is implied by the following lemma.

Lemma 5.5 *Approximating the densities by a factor* $\alpha > 1$ *instead of computing them exactly increases the approximation ratio of Algorithm Sparse_Spanner by the same factor.*

We omit its proof; a very similar one can be found in ([12], Lemma 5.5). Denote by $\mathcal{H}(G)$ the set of all 3-spanners of the graph G. We have proved the following theorem.

Theorem 5.6 *For any graph* G*, running Algorithm Sparse_Spanner with* $k = 3$ *yields a* 3*-spanner* H' *such that*

$$|H'| = (\min_{H\in\mathcal{H}(G)} \{|H| \cdot a(H) \cdot \log r\}) \ .$$

We next generalize Theorem 5.6 to establish a general connection between the approximability of the 3-spanner problem and the sparsity of possible edge-dominating systems.

Theorem 5.7 *Consider a family $\mathcal{F}$ of graphs which admit $(H,3)$-dominating systems (D,S) with $Sparsity_H(D,S) = O(a(H)^\delta)$, for some $0 < \delta < 1$. Then the 3-spanner problem restricted to the family of $\mathcal{F}$-spanned graphs admits an $\tilde{O}(n^{\frac{3\delta}{2(2+\delta)}})$-approximation ratio.*

Proof: First, the previous analysis shows that under the assumption of the theorem our algorithm finds a 3-spanner H' such that $|H'| = O(\min_{H \in \mathcal{H}(G)} \left\{ |H| \cdot a(H)^\delta \cdot \log r \right\})$. This is proved along the same lines as the proof of Theorem 5.6, except that in inequality (6) we now have $\sum_{v \in D} |Sp(v)| \leq O(a(\bar{H}_i)^\delta) \cdot |\bar{H}_i|$, and this change is propagated through the rest of the proof.

Let H^* be an optimal 3-spanner. It follows that our algorithm provides an $\tilde{O}(a(H^*)^\delta)$-approximation ratio. We observe that $|H^*| \geq a(H^*)^2$, so by using the $\tilde{O}(n^{3/2})$-universal construction of [9] for the 3-spanner problem we obtain an $\tilde{O}(\frac{n^{3/2}}{a(H^*)^2})$-approximation ratio.

The minimum of $\max\{\tilde{O}(a(H^*)^\delta), \tilde{O}(\frac{n^{3/2}}{a(H^*)^2})\}$ is attained when $a(H^*) = n^{\frac{3}{2(2+\delta)}}$. This yields the claimed ratio of $\tilde{O}(n^{\frac{3\delta}{2(2+\delta)}})$. ▮

We now prove a lower bound on the sparsity of the possible edge-dominating systems.

Theorem 5.8 *If for every graph G and its optimal 3-spanner H there exists an $(H,3)$-dominating system (D,S) such that $Sparsity_H(D,S) = O(2^{\log^{1-\epsilon} a(H)})$ for some $0 < \epsilon < 1$, then $NP \subseteq DTIME(n^{\text{polylog } n})$.*

Proof: The existence of such an edge-dominating system implies an $O(2^{\log^{1-\epsilon'} n})$-approximation algorithm for the 3-spanner problem for some $0 < \epsilon' < \epsilon$. In view of the hardness result of [11] concerning the 3-spanner problem, this implies that $NP \subseteq DTIME(n^{\text{polylog } n})$. ▮

Denote by $\mathcal{H}_1(G)$ the subfamily of $\mathcal{H}(G)$ of 3-spanners whose size is close to the size of an optimal 3-spanner up to a constant factor and by $\mathcal{H}_2(G)$ the subfamily of $\mathcal{H}(G)$ of spanners whose size is close to the size of an optimal 3-spanner up to a polylogarithmic factor in n. In particular, we get

Corollary 5.9 *Algorithm Sparse_Spanner with $k = 3$ provides an $O(\log r)$-approximation ratio for the 3-spanner problem on the class $SP_3(BA)$, and an $O(\text{polylog } n)$-approximation ratio on the class $SP_3(\log A)$.*

Denote the girth of a graph G by $g(G)$. Recall that $SP_3(GIRTH(g))$ is the family of graphs admitting a near-optimal 3-spanner with girth no smaller than g.

Lemma 5.10 *For any graph G with $g(G) \geq g$ and m edges, $a(G) \leq m/n^{g-2/g}$.*

Proof: Let U be the densest set of G. Denote $l = |U|$. Then by [3], $|E(U)| \leq l^{1+\frac{g-2}{g}}$. Hence $a(G) \leq l^{\frac{2}{g-2}}$ and

$$\frac{a(G)}{m/n} \leq \frac{l^{\frac{2}{g-2}} n}{l^{\frac{2}{g-2}} + n - l} \leq \frac{l^{\frac{2}{g-2}} n}{l^{\frac{g}{g-2}} + n} .$$

Thus

$$\frac{a(G)}{m/n} \leq \max_{1 \leq l \leq n} \left\{ \frac{l^{\frac{2}{g-2}} n}{l^{\frac{g}{g-2}} + n} \right\} .$$

This expression is maximized for $l = n^{\frac{g-2}{g}}$, and so

$$\frac{a(G)}{m/n} \leq n^{2/g} \ ,$$

and we are done. ∎

Theorem 5.11 *Algorithm Sparse_Spanner with $k = 3$ provides an $\tilde{O}(n^{\frac{4g-4}{(g-2)g}})$-approximation for the 3-spanner problem on $SP_3(GIRTH(g))$.*

Proof: The algorithm will find a 3-spanner H' which is an $\tilde{O}(a(H^*))$-approximation to the optimal spanner. Hence

$$\frac{H'}{H^*} = \tilde{O}(a(H^*)) = \tilde{O}(\frac{m}{n^{\frac{g-2}{g}}}) = \tilde{O}(n^{\frac{g}{g-2}-\frac{g-2}{g}}) \ . \qquad \blacksquare$$

Note that the exponent tends to zero as g tends to infinity. Denote by the 3-*spanner*(g) the 3-spanner problem on $SP_3(GIRTH(g))$. Then

Corollary 5.12 *The set $\{3\text{-spanner}(g)\}_{g=1}^{\infty}$ problem enjoys the ratio degradation property in g.*

5.2 Analysis of the k-spanner case

In this section we show that executing Algorithm *Sparse_Spanner* with integer parameter $k \geq 2$ provides a logarithmic approximation ratio for the k-spanner problem on $SP_k(TREE)$. The proof of this fact involves the dominating systems constructed in Section 3.3.

The analysis is analogous to that of the previous section. The notions of $H_c[i,v]$ and $h^c[i,v]$ are changed to

$$\begin{aligned} H^c[i,v] &= \bigcup_{j \in P_i(v)} COV_k(T_j, v, H_j^u) \ , \\ h^c[i,v] &= \sum_{j \in P_i(v)} cov_k(T_j, v, H_j^u) \ . \end{aligned}$$

Lemma 5.1 holds as is. Let $\bar{H}$ be some tree k-spanner for G and $\bar{H}_i$ be (a possibly Steiner) tree k-spanner for $H^u[i]$.

Lemma 5.13

$$h^u[i]/\bar{h} \leq k \cdot \frac{r}{2^{i-1}} \ .$$

Next, it follows that for every $1 \leq i \leq f$, $h[i]/\bar{h} = O(k)$, which allows us to conclude that $h/\bar{h} = O(k \log r)$, hence Lemma 5.3 and Corollary 5.4 are modified into

Lemma 5.14 *For every $1 \leq i \leq f$, $h[i]/\bar{h} < O(k)$.*

Corollary 5.15 $h/\bar{h} = O(k \log r)$.

And finally, analogously to Theorem 5.6 we conclude

Theorem 5.16 *For any graph $G \in SP_k(TREE)$, Algorithm Sparse_Spanner finds a k-spanner of $O(nk^2 \log r)$ edges.*

Remark: One factor of k follows from Lemma 3.3 and the other because the value of maximal density is not computed exactly but only approximated to a factor of $O(l) = O(k)$.

Corollary 5.17 *The k-spanner problem is $O(k^2 \log n)$-approximable for any k over the graph family $SP_k(TREE)$.*

Corollary 5.18 *The k-spanner problem is $O(\frac{\log^3 n}{(\log\log n)^2})$-approximable for any k over the graph family $SP_k(TREE)$.*

Proof: For $k = o(\frac{\log n}{\log\log n})$, Algorithm *Sparse_Spanner* supplies the required ratio. For $k = \Omega(\frac{\log n}{\log\log n})$ we obtain an $O(\log n)$ ratio by using $O(n^{1/k})$-approximation algorithm of [21]. ▮

Recall that $S_{TREE}(G)$ denotes the minimum stretch of any spanning tree T for the graph G. As mentioned earlier, the problem of finding such a tree is known to be $(1+\sqrt{5})/2$-inapproximable [22].

Theorem 5.19 *There is a polynomial time algorithm A that given a graph G constructs a k-spanner H satisfying the following two properties:*

1. $k \leq S_{TREE}(G)$, *and*
2. $|H| = \tilde{O}(n)$.

Proof: Algorithm A applies Algorithm *Sparse_Spanner* as a subroutine for k between 1 and $n-1$ in a binary search manner. It checks each time whether the size of the obtained spanner is $O(nk^2 \log r)$, and decreases k if it is, and increases it otherwise. Clearly, this algorithm provides an $O(S_{TREE}(G)^2 \log n)$ approximation. When $S_{TREE}(G) = o(\frac{\log n}{\log\log n})$, we obtain a spanner of size $O(n\frac{\log^3 n}{(\log\log n)^2})$. Otherwise, we just use the construction of [21] to obtain an $O(n \log n)$-size spanner with stretch $O(\frac{\log n}{\log\log n}) = O(S_{TREE}(G))$. ▮

6 Extension to client-server and directed k-spanners

In this section we show that Algorithm *Sparse_Spanner* with slightly modified subroutines for computing density is applicable to the CS k-spanner [10, 12], the directed k-spanner [21] and the k-DSS problems and yields similar approximation ratios for them. Hence we next focus on generalizing our analysis to these harder variants of the k-spanner problem, and, in particular, generalizing our constructions of the edge-dominating systems. Furthermore, we show that our analysis leads to the first approximation algorithms for these problems for $k > 2$ with a non-trivial (i.e, sublinear) approximation ratios (specifically, $\tilde{O}(n^{2/3})$). This complements a result of [21], that for certain n-vertex digraphs, any k-spanner requires $\Omega(n^2)$ edges.

The CS k-spanner problem is a generalization of the k-spanner problem in which as part of the input we are given two edge sets $\mathcal{C}$ and $\mathcal{S}$, called the *client edge set* and *server edge set* respectively. An *$\mathcal{S}$ k-spanner for $\mathcal{C}$* is a set $H \subseteq \mathcal{S}$ that k-spans all edges of $\mathcal{C}$, and the goal is to find the sparsest such spanner.

The basic notions required for handling the CS k-spanner problem are similar to those defined earlier, with but few modifications. The definition of $COV_k(\hat{T}, v, H^u)$ is the same as previously, but $\hat{T} \subseteq \mathcal{S}$ and $H^u \subseteq \mathcal{C}$. The set $T(U, v)$ is the set of all possible non-empty BFS trees rooted by v and contained in $\mathcal{S}(U)$. Analogously, $\hat{T}(U, v)$ is the set of all such trees (not necessarily BFS ones). By $\Gamma_l(v)$ we denote the l-neighborhood of v in $\mathcal{S}$. Now the functions ψ_k^l and $\hat{\psi}_k^l$ are defined by (2) and (3).

The proofs of Lemmas 2.2 and 2.3 are the same. For the proof of Lemma 2.7 to work, the definition of $\tilde{E}$ in (4) is changed to

$$\tilde{E} = \begin{cases} \mathcal{C}(\hat{\Gamma}_l(v)) \cap H^u, & k \text{ is even}, \\ \mathcal{C}(\hat{\Gamma}_l(v)) \cap H^u \setminus E(\Gamma_l(v)), & k \text{ odd}. \end{cases}$$

The definition of the (H,k)-dominating system is changed accordingly.

Definition 6.1 *For every graph G, edge sets $E = \mathcal{C} \cup \mathcal{S}$ and k-spanner $H \subseteq \mathcal{S}$ for G, the vertex $v \in V$* (H,k)-dominates *the edge $e = (u,z) \in \mathcal{C}$ if H contains a path P of length at most k and $v \in V(P)$.*

The definition (3.1) is changed analogously. The notion of $Sparsity(D,S)$ is defined in the same way. Both Construction B and the proof of Lemma 3.3 are not changed.

Define $CS - SP_k(TREE)$ to be the family of triples $(G,\mathcal{C},\mathcal{S})$ which admit a tree k-spanner contained in $\mathcal{S}$ for $\mathcal{C}$. Thus we conclude that

Theorem 6.2 *The CS k-spanner problem restricted to $CS - SP_k(TREE)$ admits $O(k^2 \log n)$-approximation algorithm.*

Analogously, Construction C works too. Thus we conclude the following.

Theorem 6.3 *For any instance of the CS 3-spanner problem $(G,\mathcal{C},\mathcal{S})$ and an optimal CS 3-spanner H^* the algorithm provides an $O(\log n \cdot a(H^*))$ approximation ratio for the CS 3-spanner problem on the instance.*

Theorem 6.4 *Let $0 < \beta \leq 1$ be a constant. The CS 3-spanner problem on instances with server set of size $|\mathcal{S}| = O(n^{1+\beta})$ admits an $\tilde{O}(n^{\frac{\beta+1}{3}})$-approximation ratio.*

Proof: If $a(\mathcal{S}) \leq n^{\frac{\beta+1}{3}}$ then Theorem 6.2 implies the result. Otherwise, let U be the densest vertex set of H^*. Then $|U|^2 \geq |H^*| \geq |U| a(H^*)$ and so $|U| \geq a(H^*)$, implying $|H^*| \geq a(H^*)^2$. Hence $|H^*| \geq a(\mathcal{S})^2 \geq n^{\frac{2\beta+2}{3}}$. Thus by taking all the edges we obtain $O(n^{1+\beta-\frac{2\beta+2}{3}}) = O(n^{\frac{\beta+1}{3}})$-approximation ratio. ▮

Corollary 6.5 *The CS 3-spanner problem admits an $\tilde{O}(n^{2/3})$-approximation ratio.*

Analogous considerations work for the directed 3-spanner and the directed CS 3-spanner problems. Given a directed graph (V,E) let us denote by $(V, under(E))$ the *underlying* graph (V,E), i.e,

$$under(E) = \{(v,u) \mid \langle v,u \rangle \in E \text{ s.t. } \langle u,v \rangle \in E\} .$$

$under(E) = \{(v,u) \mid \langle v,u \rangle \in E \text{ s.t. } \langle u,v \rangle \in E\}$. A subgraph (V,T) is a *quasi-tree* if the underlying graph $(V, under(T))$ is a tree. Let $SP_k(QTREE)$ be the family of digraphs that have a quasi-tree k-spanner. The analog of the k-spanner problem restricted to $SP_k(TREE)$ on the digraphs is the directed k-spanner problem restricted to $SP_k(QTREE)$. As shown in [6] the directed k-spanner problem on $SP_k(TREE)$ is polynomial, but the directed k-spanner problem on $SP_k(QTREE)$ is at least as hard as the tree k-spanner problem, i.e., is NP-hard for $k \geq 4$. Our algorithm adopted for the directed case leads to the following upper bound on the approximability of the problem.

Theorem 6.6 *The directed k-spanner problem on $SP_k(QTREE)$ admits an $O(k^2 \log n)$-approximation ratio algorithm.*

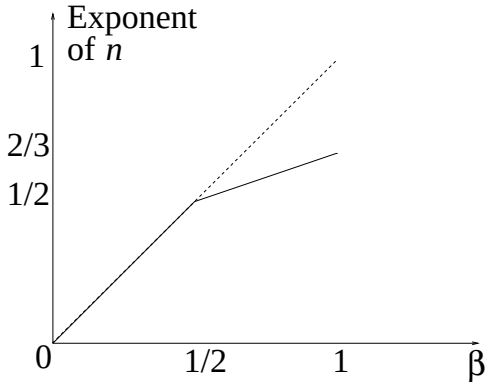

Figure 3: Upper bound on approximability threshold of directed 3-spanner, 3-DSS and directed 3-DSS problems as function of parameter β. Dotted line represents the previous upper bound and solid line represents the new one.

Theorem 6.7 *Let $0 < \beta \leq 1$ be a constant. The directed 3-spanner problem on graphs with $O(n^{1+\beta})$ edges and the directed CS 3-spanner problem on instances with server set of size $|\mathcal{S}| = O(n^{1+\beta})$ admit an $\tilde{O}(n^{\frac{\beta+1}{3}})$-approximation ratio.*

Corollary 6.8 *The directed 3-spanner and the directed CS 3-spanner problems admit an $\tilde{O}(n^{2/3})$-approximation ratio.*

A reduction from the k-DSS problem to the CS k-spanner problem was shown in [13]. Consequently we have

Theorem 6.9 *Let $0 < \beta \leq 1$ be a constant. The (directed) 3-DSS problem on graphs with $O(n^{1+\beta})$ edges admits an $\tilde{O}(n^{\frac{\beta+1}{3}})$-approximation ratio.*

Remark: The directed 3-spanner problem and the directed and undirected 3-DSS problem when restricted to graphs of size $O(n^{1+\beta})$ admit a trivial $O(n^{\beta})$-approximation ratio. Hence Theorems 6.7 and 6.9 yield better results only for $1/2 < \beta < 1$. Figure 3 plots the graph of an upper bound on the approximability of these problems as function of β.

Corollary 6.10 *The (directed) 3-DSS problem admits an $\tilde{O}(n^{2/3})$-approximation ratio.*

For the basic 3-spanner problem these considerations lead only to the $\tilde{O}(n^{1/2})$-approximation ratio algorithm, which is unfortunately no better than the ratio obtained from the $\tilde{O}(n^{3/2})$ universal construction of [9]. However, any improvement in Construction B, or in the bound of $O(a(H))$ on the sparsity of the dominating-system built in the Construction C (see Lemma 3.4), i.e., any construction of an $(H, 3)$-edge-dominating system (D, S) with $Sparsity(D, S) = O(a(H))^{(1-\delta)}$ for some $\delta > 0$ would enable to improve the approximation ratio for the basic 3-spanner problem beyond the $O(n^{1/2})$ ratio.

7 Bicriteria approximations

In this section we provide some bicriteria upper bounds on the CS k-spanner problem and on the k-DSS problem. We say that an algorithm is a *bicriteria $(\rho, (a, b))$-approximation algorithm* for

the CS k-spanner (respectively, k-DSS) problem if given an instance of the problem it returns a solution for the CS $(a \cdot k + b)$-spanner (resp., $(a \cdot k + b)$-DSS) problem of size at most ρ times larger than the optimal solution for the CS k-spanner (resp., k-DSS) problem. In particular, a *bicriteria* (ρ, a)*-approximation* (resp., $(\rho, +b)$*-approximation*) *algorithm* for the CS k-spanner and k-DSS problems is defined as a bicriteria $(\rho, (a, 0))$-approximation (resp., $(\rho, (1, b))$-approximation) algorithm for the problems.

Theorem 7.1 *The AC k-spanner problem admits an $(O(n^{1/2}), +2)$-bicriteria approximation algorithm.*

Proof: For a graph G the subset $H \subseteq E$ is called an *additive b-spanner* for G if for every pair of nodes $u, w \in V$, $d_H(u, w) \leq d_G(u, w) + 2$. We use the construction of [14] to get an additive 2-spanner H of the graph $(V, \mathcal{S})$ of size $|H| = O(|V(\mathcal{S})|^{3/2})$. We assume without loss of generality that the whole server set $\mathcal{S}$ k-spans the edge set E, because otherwise the instance is infeasible and it can be checked in a polynomial time at the beginning. Consider some edge $(u, w) \in E$. Let $(u = u_0, u_1, \ldots, u_l = w)$, $l \leq k$ be a spanning path in $\mathcal{S}$ of the edge e. Since H is a $(+2)$-spanner of $\mathcal{S}$, there is a path $P \subseteq H$ of length at most $l + 2 \leq k + 2$ between u and w Hence H is a $(k + 2)$-spanner of E. Since any t-spanner H' for of $\mathcal{S}$ for any integer t touchs every node of $V(E) = V$, and one edge can touch at most 2 new nodes, $|H'| \geq |V(E)|/2$. $\mathcal{S} \subseteq E$ implies $|V(\mathcal{S})| \leq |V(E)|$, i.e., H is an $(\tilde{O}(n^{1/2}), +2)$-bicriteria approximation of an optimal k-spanner for the instance. ▮

For any instance $(G = (V, E), \mathcal{C}, \mathcal{S})$ of the CS k-spanner problem, let $Diam(\mathcal{C}, \mathcal{S}) = \max_{(u,w)\in\mathcal{C}} dist_{\mathcal{S}}(u, w)$. The above proof yields the following.

Corollary 7.2 *The AC k-spanner problem restricted to the case of $Diam(E, \mathcal{S}) \leq k-2$ admits an $O(n^{1/2})$-approximation algorithm.*

As already mentioned, the k-DSS problem is reducible to the AC k-spanner problem. Subsequently, we have:

Corollary 7.3
1. *The k-DSS problem admits an $(O(n^{1/2}), +2)$-bicriteria approximation algorithm.*
2. *The k-DSS problem restricted to instances with $Diam(G) \leq k - 2$ admits an $O(n^{1/2})$-approximation algorithm.*

Analogously, constructions of $(1 + \epsilon, \beta(l, \epsilon))$-spanners with $O(n^{1+1/l})$ edges due to [14] can be used to extend the above results. Specifically, we have:

Theorem 7.4
1. *For any integers $l > 0$, $k \geq 3$, and any $\epsilon > 0$, the AC k-spanner and k-DSS problems admit $(O(n^{1/l}), (1 + \epsilon, \beta(l, \epsilon))$-bicriteria approximation algorithms, where $\beta(l, \epsilon) = l^{\log\log l - \log\epsilon}$.*
2. *For any integers $l > 0$ and $k \geq 3$, the AC k-spanner problem restricted to instances with $Diam(E, \mathcal{S}) \leq \frac{k-\beta(l,\epsilon)}{1+\epsilon}$ and the k-DSS problem restricted to instances with $Diam(G) \leq \frac{k-\beta(l,\epsilon)}{1+\epsilon}$ admit an $O(n^{1/l})$-approximation ratio algorithms.*

Corollary 7.5 *For any constant integer $l > 0$ and any constant $\epsilon, \delta > 0$, the AC n^{δ}-spanner and n^{δ}-DSS problems admit a bicriteria $(O(n^{1/l}), 1 + \epsilon)$-approximation algorithms.*

We remark that AC n^δ-spanner and n^δ-DSS problems are known to be $\Omega(2^{\log^{1-\mu} n})$-inapproximable for any $\mu > 0$, unless $NP \subseteq DTIME(n^{\text{polylog } n})$ [13]. To the best of our knowledge these problems constitute the first examples of $\Omega(2^{\log^{1-\mu} n})$-inapproximable problems that admit bicriteria $(O(n^{1+\eta}), 1+\epsilon)$-approximation for *any* constant $\epsilon, \eta > 0$.

8 Graphs that admit bounded-degree spanners

In this section we present an algorithm providing an $O(\Delta^{k-2} \log n)$ approximation algorithm for k-spanner problem, where Δ is the maximum degree of the optimum spanner. This is done by another extension of the algorithm of [18] for the 2-spanner problem.

Instead of considering the vertex density as in the 2-spanner case, we introduce the notion of the density of a *path*. The algorithm for the k-spanner problem will compute the maximum density of a path of length of $(k-2)$. Observe that in the 2-spanner case, a single vertex represents a path of length zero.

Following [12], for every subset $S \subseteq E$ we define the *neighborhood* of v in S as

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

The neighborhood of v with respect to the entire edge set E is defined as $N(v) \;=\; N(v, E)$. Let $G = (V, E)$ be a graph and let $P = (v_1, v_2, v_3, .., v_{l+1})$ be a path of length l in G. For every subset $S \subseteq E$ define the *neighborhood* of the path P in S as

$$N(P, S) \;=\; \bigcup_{1 \le i \le l+1} N(v_i, S) \; .$$

Similarly, $N(P) \;=\; N(P, E)$. Also let $deg(P, S)$ denote the size of $N(P, S)$. The algorithm constructs the spanner graph denoted by H. Let H^u be the set of edges that are still *uncovered*. Following the definitions of [12], for any subset Q of $N(e)$ and any path P let the set of *added edges* be

$$AE(Q, P, H) = \{(z, v) \in H \mid z \in Q, v \in V(P)\} \; .$$

Adding the edges of $AE(Q, P, H)$ to a spanner causes the covering of some new edges that were not covered before. The set of those *covered edges* is denoted by $COV(Q, P, H^u)$. These edges are partitioned into two disjoint classes, according the way they are covered. We denote these classes by $CE_1(Q, P, H^u)$ and $CE_3(Q, P, H^u)$ (the names are chosen for consistency with the notations of [12]). Hence

$$COV(Q, P, H^u, H) \;=\; CE_1(Q, P, H^u) \cup CE_3(Q, P, H^u, H) \; .$$

The set CE_1 consists of edges covered by the edges of the path P and two edges adjacent to the path, i.e.,

$$CE_1(Q, P, H^u) \;=\; \{(t, z) \in H^u \mid t, z \in Q\} \; .$$

The set CE_3 consists of edges that are used in the path, or edges with one endpoint in $V(P)$ and and a vertex from Q as another. This set is formally defined as

$$CE_3(Q, P, H^u) = P \cup \{(v, t) \in H^u \mid v \in V(P), t \in Q\} \; .$$

Following [12], for any subset Q of $N(P)$ we define the $\tilde{\varphi}$-density of the path P with respect to the set by

$$\tilde{\varphi}(Q, P, H^u, H) = \frac{cov(Q, P, H^u)}{ae(Q, P, H)} \ .$$

Finally, the φ-density of the path P is defined as

$$\varphi(P, H^u, H) = \max_{Q \subseteq N(P,H)} \tilde{\varphi}(Q, P, H^u, H) \ .$$

8.1 Density computation

We compute the φ-density by a reduction to the *provisioning problem*, formulated as follows.

Input: A collection of n items $\{1, .., n\}$ with costs $c_j > 0$ for $j \in \{1, .., n\}$, and m subsets of items $S_1, .., S_m$ with benefits $b_1, .., b_m$, where $S_i \subseteq \{1, .., n\}$ for $1 \leq i \leq m$ and $b_i > 0$.

Solution: A subset $R \subseteq \{1, .., n\}$. The *cost* of the solution R is $\sum_{j \in R} c_j$. The *benefit* of R is is the sum of benefits of subsets for which all their items were taken to the solution set, i.e., $\sum_{j=1}^{m} b_j \cdot I_j(R)$, where $I_j(R)$ is defined as

$$I_j(R) \ = \ 1 \ \text{ iff } \ S_j \subseteq R \ \text{ and } \ I_j(R) \ = \ 0 \ \text{ iff } \ S_j \not\subseteq R \ .$$

Objective: Maximize the difference D between the total benefit and the total cost of the solution.

This problem can be solved in polynomial time [19]. Let the *φ-decision problem* be the problem of deciding whether $\varphi(P, H^u, H) \geq k$, given a graph $G = (V, E)$, a path P, and an edge subsets H^u and H and an integer z. Clearly, the problem of computing $\varphi(v, H^u, H)$ is reducible to the φ-decision problem, because a binary search can be conducted over all possible values of $\varphi(P, H^u, H)$. The denominator of the density $\varphi(P, H^u, H)$ is bounded by n^{k-2}, the numerator by $m = |E|$ and so $\varphi(P, H^u, H)$ can be computed using at most $\log(n^{k-2} \cdot m)$ calls to a subroutine for the φ-decision problem.

Now we reduce the φ-decision problem to the provisioning problem in the following way.

Let $N(P)$ be the set of items and let their costs be be $c_j = z$. Let the subsets be $\{\{u\} \mid u \in N(P, H)\}$ and $\{\{u, w\} \mid u, w \in N(P, H) \mid (u, w) \in H^u\}$. Let their benefits be defined by

$$\begin{aligned} b(\{u\}) &= |\{v \in V(P) | (u, v) \in H^u\}| \ , \\ b(\{u, w\}) &= 1 \ . \end{aligned}$$

The intuition is that for every covered edge we gain one benefit unit. The provisioning problem will now maximize

$$D = B - C = cov(Q, P, H^u) - k \cdot ae(Q, P, H),$$

where Q is a subset of the items purchased (namely, the nodes taken into the densest subset). Now we compare D with 0 and answer "yes" iff $D > 0$. Indeed, this condition is equivalent to $\varphi(P, H^u, H) > z$.

8.2 The algorithm and its analysis

The algorithm will be the same as in Section 4, except that it will use the φ-density, instead the ψ-density. Specifically, in each iteration it will choose the densest path of length $k-2$ or less. Its correctness and termination follow from analogous considerations to those presented in Section 4. Its running time will now grow by n^{k-3} multiplicative factor, since in each iteration of the algorithm we now perform upto n^{k-2} density computations instead of n. But since k is a constant, the running time is still polynomial.

We use the same notion of r as in Section 5.1, and analogous definition of the phases (where the definition now refers to the φ-density instead of the ψ-density), and the sets H_j, H_j^c, H_j^u, $H[i]$, $H^c[i]$, $H^u[i]$ and $\hat{H}[i]$. Let X_i be the set of paths chosen during the ith phase. Also we denote by PH_i the set of indices of iterations in the ith phase (previously denoted by P_i). Let P_j be the path chosen in the jth iteration and Q_j be the corresponding densest set of neighbors of the path P_j.

For any subset $S \subseteq E$, denote by $\mathcal{P}_l(S)$ the set of all the paths in S of length l or less. I.e.,

$$\mathcal{P}_l(S) \;=\; \{P = ((v_1,v_2),(v_2,v_3),..,(v_r,v_{r+1})) \subseteq S \mid r \leq l\} \;.$$

Denote $\mathcal{P}_l = \mathcal{P}_l(E)$.

Similarly, for every path $P \in \mathcal{P}_l$, denote by $PH_i(P)$ the subset of PH_i containing only those ith phase iterations at which P was chosen, i.e.,

$$PH_i(P) \;=\; \{j \in PH_i \mid P_j = P\} \;.$$

Analogously,

$$\begin{aligned} H[i,P] &= \bigcup_{j \in P_i(P)} Q_j \;, \\ H^c[i,P] &= \bigcup_{j \in P_i(P)} COV(Q_j, P, H_j^u) \;. \end{aligned}$$

By the same considerations as in Section 5.1 we conclude that

$$\begin{aligned} h[i,P] &\leq \sum_{j \in P_i(P)} |Q_j| \;, \\ h^c[i,P] &= \sum_{j \in P_i(P)} cov(Q_j, P, H_j^u). \end{aligned}$$

Also for every path $P \in X_i$,

$$h^c[i] \geq h[i] \cdot \frac{r}{2^i} \;.$$

Now let $\bar{H}$ be some k-spanner for G and $\bar{H}_i$ be (a possibly Steiner) k-spanner for $H^u[i]$ of size not bigger than $\bar{H}$ and satisfying

$$\Delta_{\bar{H}_i} \leq \Delta_{\bar{H}} \;.$$

Specifically, $\bar{H}_i$ is allowed to use all E edges and not only those of $H^u[i]$. This implies the existence of such a spanner, since $\bar{H}$ itself in particular spans $H^u[i]$.

Lemma 8.1

$$h^u[i]/\bar{h} \leq \Delta^{k-2} \cdot \frac{r}{2^{i-1}} \ .$$

Proof: For every edge $e \in H^u[i]$ there exists a path P' in $\bar{H}_i$ of length at most k that spans the edge. Consider two cases. If $|P'| \leq k-2$ then $e \in COV(N(P', \bar{H}_i), P', H^u[i])$. Otherwise, $k-2 < |P'| \leq k$. Since $k \geq 3$, $|P'| \geq 2$. Thus we can represent the path P' as a concatenation of its first edge e with a (possibly empty) path P and its last edge e'. By definition of COV it follows that $e \in COV(N(P, \bar{H}_i), P, H^u[i])$. Hence we have proved that for every edge $e \in H^u[i]$ there exists a path P of length $k-2$ or less such that $e \in COV(N(P, \bar{H}_i), P, H^u[i])$.

Therefore

$$h^u[i] \leq \sum_{P \in \mathcal{P}_{k-2}(\bar{H}_i)} cov(N(P, \bar{H}_i), P, H^u[i]) \ .$$

Clearly, $\bar{h}_i \leq \bar{h}$. So

$$\frac{h^u[i]}{\bar{h}} \leq \frac{h^u[i]}{\bar{h}_i} \leq \frac{\sum_{v \in V} cov(N(P, \bar{H}_i), P, \hat{H}[i])}{\bar{h}_i}.$$

Note that

$$\frac{1}{2{\Delta_{\bar{H}_i}}^{k-2}} \cdot \sum_{P \in \mathcal{P}_{k-2}} deg(P, \bar{H}_i) \leq \frac{1}{2{\Delta_{\bar{H}_i}}^{k-2}} \cdot \sum_{v \in V} deg(v, \bar{H}_i)^{k-1} \leq \frac{1}{2} \sum_{v \in V} deg(v, \bar{H}_i) = \bar{h}_i \ .$$

Thus

$$\begin{aligned} \frac{h^u[i]}{\bar{h}} &\leq 2{\Delta_{\bar{H}_i}}^{k-2} \cdot \frac{\sum_{P \in \mathcal{P}_{k-2}(\bar{H}_i)} cov(N(P, \bar{H}_i), P, \hat{H}[i])}{\sum_{P \in \mathcal{P}_{k-2}(\bar{H}_i)} deg(P, \bar{H}_i)} \\ &= 2{\Delta_{\bar{H}_i}}^{k-2} \cdot \frac{\sum_{P \in \mathcal{P}_{k-2}(\bar{H}_i)} cov(N(P, \bar{H}_i), P, \hat{H}[i])}{\sum_{P \in \mathcal{P}_{k-2}} ae(N(P, \bar{H}_i), P, \bar{H}_i)} \ . \end{aligned}$$

The last equality follows because

$$ae(N(P, \bar{H}_i), v, \bar{H}_i) \ = \ |\{(z, v) \in \bar{H}_i \mid v \in V(P), z \in N(P, \bar{H}_i)\}| \ = \ deg(P, \bar{H}_i).$$

It follows that

$$\frac{h^u[i]}{\bar{h}} \leq 2{\Delta_{\bar{H}_i}}^{k-2} \cdot \max_{P \in \mathcal{P}_{k-2}(\bar{H}_i)} \{\tilde{\varphi}(N(P, \bar{H}_i), P, H^u[i], \bar{H}_i)\} = 2{\Delta_{\bar{H}_i}}^{k-2} \cdot \tilde{\varphi}(N(P_0, \bar{H}_i), P_0, H^u[i], \bar{H}_i)$$

for some specific path $P_0 \in \mathcal{P}_{k-2}(\bar{H}_i)$ that maximizes $\tilde{\varphi}$. By definition of φ we get

$$\frac{h^u[i]}{\bar{h}} \leq 2{\Delta_{\bar{H}_i}}^{k-2} \cdot \varphi(P_0, \hat{H}[i]) \leq 2{\Delta_{\bar{H}_i}}^{k-2} \frac{r}{2^i} = {\Delta_{\bar{H}_i}}^{k-2} \frac{r}{2^{i-1}} \ ,$$

using a bound analogous to (5). ∎

Lemmas analogous to the Lemma 5.3 and Corollary 5.4 can now be proven in the straightforward way.

Let denote by $\mathcal{H}(G)$ the family of all the 3-spanners of the graph G. We conclude

Theorem 8.2 *For any graph G the algorithm finds a k-spanner H' such that*

$$|H'| = O(\min_{H \in \mathcal{H}(G)} \left\{|H| \cdot {\Delta_H}^{k-2} \cdot \log r\right\}) \ .$$

References

[1] Baruch Awerbuch, Alan Baratz, and David Peleg. Efficient broadcast and light-weight spanners. Unpublished manuscript, November 1991.

[2] I. Althöfer, G. Das, D. Dobkin, and D. Joseph, Generating sparse spanners for weighted graphs, *Proc. 2nd Scandinavian Workshop on Algorithm Theory*, Lect. Notes in Comput. Sci., Vol. 447, pp. 26-37, Springer-Verlag, New York/Berlin, 1990.

[3] B. Bollobas, *Extremal Graph Theory*, Academic Press, New York, 1978.

[4] L.P. Chew, There is a planar graph almost as good as the complete graph, *Proc. ACM Symp. on Computational Geometry*, 1986, pp. 169-177

[5] L. Cai, NP-completeness of minimum spanner problems. *Discrete Applied Math.*, 48:187-194, 1994

[6] L. Cai and D.G. Corneil, Tree Spanners, *SIAM J. on Discrete Mathematics* **8**, (1995), 359-387.

[7] B. Chandra, G. Das, G. Narasimhan, J. Soares, New Sparseness Results on Graph Spanners, *Proc. 8th ACM Symp. on Computational Geometry*, pp. 192-201, 1992.

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

[9] D. Dor, S. Halperin, U. Zwick, All pairs almost shortest paths, *Proc. 37th IEEE Symp. on Foundations of Computer Science*, 1997, pp. 452-461.

[10] M.-L. Elkin and D. Peleg, The Hardness of Approximating Spanner Problems, *Proc. 17th Symp. on Theoretical Aspects of Computer Science*, Lille, France, Feb. 2000, 370-381.

[11] M.-L. Elkin and D. Peleg, Strong Inapproximability of the Basic k-Spanner Problem, *Proc. 27th International Colloquim on Automata, Languages and Programming*, Geneva, Switzerland, July 2000. See also Technical Report MCS99-23, the Weizmann Institute of Science, 1999.

[12] M.-L. Elkin and D. Peleg, The Client-Server 2-Spanner Problem and Applications to Network Design, Technical Report MCS99-24, the Weizmann Institute of Science, 1999.

[13] M.-L. Elkin, Additive Spanners and Diameter Problem, manuscript, 2000.

[14] M. -L. Elkin and D. Peleg, $(1+\epsilon, \beta)$-Spanners Constructions for General Graphs, to appear in *Proc. 33rd Annual ACM Symp. on Theory of Computing*, Crete, Greece, July, 2001.

[15] S. P. Fekete, J. Kremer, Tree Spanners in Planar Graphs, Angewandte Mathematik und Informatik Universitat zu Koln, Report No. 97.296

[16] S. Halperin, U. Zwick, Private communication, 1996.

[17] G. Kortsarz, On the Hardness of Approximating Spanners, *Proc. 1st Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, Lect. Notes in Comput. Sci., Vol. 1444, pp. 135-146, Springer-Verlag, New York/Berlin, 1998.

[18] G. Kortsarz and D. Peleg, Generating Sparse 2-Spanners. *J. Algorithms*, **17** (1994) 222-236.

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

[20] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*, SIAM, Philadelphia, PA, 2000.

[21] D. Peleg and A. Schäffer, Graph Spanners, *J. Graph Theory* **13** (1989), 99-116.

[22] D. Peleg and E. Reshef, A Variant of the Arrow Distributed Directory with Low Average Complexity, *Proc. 26th Int. Colloq. on Automata, Languages & Prog.*, Prague, Czech Republic, July 1999, 615–624.

[23] D. Peleg and J.D. Ullman, An optimal synchronizer for the hypercube, *SIAM J. Computing* **18** (1989), pp. 740-747.

[24] E. Scheinerman, The maximal integer number of graphs with given genus, *J. Graph Theory* **11** (1987), no. 3, 441-446.