2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

Everywhere-Sparse Spanners via Dense Subgraphs*

Eden Chlamtáč†
Ben Gurion University

Michael Dinitz ‡
Weizmann Institute of Science

Robert Krauthgamer‡
Weizmann Institute of Science

Abstract— The significant progressg in constructing graph spanners that are *sparse* (small number of edges) or *light* (low total weight) has skipped spanners that are *everywhere-sparse* (small maximum degree). This disparity is in line with other network design problems, where the maximum-degree objective has been a notorious technical challenge. Our main result is for the LOWEST DEGREE 2-SPANNER (LD2S) problem, where the goal is to compute a 2-spanner of an input graph so as to minimize the maximum degree. We design a polynomial-time algorithm achieving approximation factor $\tilde{O}(\Delta^{3-2\sqrt{2}}) \approx \tilde{O}(\Delta^{0.172})$, where Δ is the maximum degree of the input graph. The previous $\tilde{O}(\Delta^{1/4})$–approximation was proved nearly two decades ago by Kortsarz and Peleg [SODA 1994, SICOMP 1998].

Our main conceptual contribution is to establish a formal connection between LD2S and a variant of the DENSEST k-SUBGRAPH (DkS) problem. Specifically, we design for both problems strong relaxations based on the Sherali-Adams linear programming (LP) hierarchy, and show that "faithful" randomized rounding of the DkS-variant can be used to round LD2S solutions. Our notion of faithfulness intuitively means that all vertices and edges are chosen with probability proportional to their LP value, but the precise formulation is more subtle.

Unfortunately, the best algorithms known for DkS use the Lovász-Schrijver LP hierarchy in a non-faithful way [Bhaskara, Charikar, Chlamtac, Feige, and Vijayaraghavan, STOC 2010]. Our main technical contribution is to overcome this shortcoming, while still matching the gap that arises in random graphs by planting a subgraph with same log-density.

1. INTRODUCTION

The significant progress made over the years in constructing graph spanners shares, for the most part, two features: (1) the objective is to minimize the total number/weight of edges; and (2) the techniques are primarily combinatorial. This second feature has started to change recently, with the use of Linear Programming (LP) in several results [10], [18], [8]. One of the earliest uses of linear programming for spanners, though, was also one of the few examples of a different objective function: in 1994, Kortsarz and Peleg [31] considered the LOWEST DEGREE 2-SPANNER (LD2S) problem, where the goal is to find a 2-spanner of an input graph that minimizes the maximum degree, and used a natural LP relaxation to devise a polynomial-time algorithm achieving approximation factor $\tilde{O}(\Delta^{1/4})$ (where Δ is the maximum degree). They also showed that it is NP-hard to approximate LD2S within a factor smaller than $\Omega(\log n)$. We make the first progress on approximating LD2S since then, by designing a new approximation algorithm with an improved approximation factor.

Theorem 1. *For any arbitrarily small constant $\varepsilon > 0$, the* LD2S *problem can be approximated in polynomial time within factor $\tilde{O}(\Delta^{3-2\sqrt{2}+\varepsilon}) \leq \tilde{O}(\Delta^{0.172})$.*

Degree bounds have a natural mathematical appeal and are also useful in many applications. For example, one common use of spanners is in compact routing schemes (e.g. [42], [16]), which store small routing tables at every node. If we route on a spanner with large maximum degree, then *a priori* the node of large degree will have a large table, even if the total number of edges is small. Similarly, the maximum degree (rather than the overall number of edges) is what determines local memory constraints when using spanners to construct network synchronizers [37] or for efficient broadcast [5]. The literature on approximation algorithms includes recent exciting work on sophisticated LP rounding for network design problems involving degree bounds (e.g. [33], [39]).

Dense subgraphs. Our central insight involves the relationship between *sparse* spanners and finding *dense* subgraphs. Such an informal relationship has been folklore in the distributed computing and approximation algorithms communities; for instance, graph spanners are mentioned as the original motivation for introducing the DENSEST k-SUBGRAPH (DkS) problem [29], in that case in the context of minimizing the total number of edges in the spanner. Surprisingly, we show that there is a natural connection between DkS and the more challenging task of constructing spanners that have small maximum degree. We prove that certain types of "faithful" approximation algorithms for a variant of DkS which we call SMALLEST m-EDGE SUBGRAPH (or SmES) imply approximation algorithms for LD2S, and then show how to construct such an algorithm for SmES; combining these two together yields our improved approximation for LD2S.

We seem to be the first to formally define and study SmES, although it has been used in previous work (sometimes implicitly) as the natural minimization version of DkS, see

*Full version [13] available: http://arxiv.org/abs/1205.0144.

† This work was done while the first author was at Tel Aviv University. Work supported supported in part by an ERC Advanced grant. Email: chlamtac@post.tau.ac.il

‡Work supported in part by an Israel Science Foundation grant #452/08, a US-Israel BSF grant #2010418, and by a Minerva grant. Emails: {michael.dinitz, robert.krauthgamer}@weizmann.ac.il

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.61

e.g. [35], [25], [2]. In SmES we are given a graph G and a value m, and want to find the subgraph of G with at least m edges that has the fewest vertices. A straightforward argument shows that an f-approximation for SmES implies an $\tilde{O}(f^2)$-approximation for DkS. In the other direction, all that was known was that an f-approximation for DkS implies an $\tilde{O}(f)$-approximation for SmES. One contribution of this paper is a non-black box improvement: while the best-known approximation for DkS is $O(n^{1/4+\varepsilon})$, we give an algorithm with smaller approximation ratio for SmES.

Theorem 2. *For any arbitrarily small constant $\varepsilon > 0$,* SmES *can be approximated in polynomial time within factor $\tilde{O}(n^{3-2\sqrt{2}+\varepsilon}) \leq \tilde{O}(n^{0.172})$.*

This improvement is key to our main result about approximating LD2S.

LP hierarchies. The log-density framework introduced in [9] in the context of DkS (see Section 3.1) predicts, when applied to SmES, that current techniques would hit a barrier at $n^{3-2\sqrt{2}}$, precisely the factor achieved by our algorithm. Here, the use of strong relaxations (namely LP hierarchies) is crucial, since simple relaxations have large integrality gaps. For example, one can show that the natural SDP relaxation for SmES has an $\Omega(n^{1/4})$ integrality gap (for $G = G(n, n^{-1/2})$ and $m = n^{1/2}$), similarly to the $\Omega(n^{1/3})$-gap shown for DkS by Feige and Seltser [23].

While we borrow some of the algorithmic techniques developed for DkS by [9], the crucial need for a "faithful" approximation required us to develop new tools which represent a significant departure from previous work both in terms of the algorithm and its analysis. For example, our algorithm and analysis rely on the existence of consistent high-moment variables arising from the Sherali-Adams [38] hierarchy and not present in the Lovász-Schrijver [34] LP hierarchy (which was sufficient for [9]).

Basic terminology. We denote the (undirected)[1] input graph by $G = (V, E)$, and let $n = |V|$. For a vertex $v \in V$, let $\Gamma_G(v) = \{u : \{u, v\} \in E\}$ denote its neighbors in G. If the graph G is clear from context then we will drop the subscript and simply refer to $\Gamma(v)$. Recall that the maximum degree of vertices in G is denoted Δ. We suppress polylogarithmic factors by using the notation $\tilde{O}(f)$ as a shorthand for $f \cdot (\log n)^{O(1)}$.

As usual, a 2-*spanner* of G is a subgraph $H = (V, E_H)$ such that every $u, v \in V$ that are connected by an edge in G are also connected in H by a path of length at most 2. This is a special case of the more general notion of a k-*spanner*, which was introduced by Peleg and Schäffer [36] and has been studied extensively; see also Section 1.4.

[1]Our algorithm for LD2S also works for the directed case, though for simplicity we focus on undirected graphs.

1.1. LP-based approach for LD2S

The LP relaxation of LD2S used by Kortsarz and Peleg [31] is very natural: for each edge $\{u, v\} \in E$ it has a variable $x_{\{u,v\}} \in [0, 1]$, plus additional variables $x_{\{u,v\};w} \in [0, 1]$ for every $w \in \Gamma(u) \cap \Gamma(v)$ (i.e., whenever u, v, w form a triangle in G). The objective is to minimize λ, subject to a degree constraint

$$\sum_{v \in \Gamma(u)} x_{\{u,v\}} \leq \lambda \qquad \forall u \in V, \tag{1}$$

and the constraints that every edge in G (i.e. demand pair) is covered by either a 1-path or a 2-path in the spanner (subgraph):

$$x_{\{u,v\}} + \sum_{w \in \Gamma(u) \cap \Gamma(v)} x_{\{u,v\};w} \geq 1 \quad \forall \{u, v\} \in E. \tag{2}$$

$$x_{\{u,v\};w} \leq \min\{x_{\{u,w\}}, x_{\{v,w\}}\} \quad \forall \{u, v\} \in E, \; w \in \Gamma(u) \cap \Gamma(v). \tag{3}$$

This LP relaxation seems like a natural place to start, but it is actually quite weak, having integrality gap $\Omega(\sqrt{\Delta})$. Indeed, let G be a clique of size $\Delta+1$; observe that every 2-spanner of this G must have maximum degree at least $\sqrt{\Delta}$, while the LP has value $\lambda \leq 1$ (by setting all x variables to $1/\Delta$). The same argument works for a disjoint union of $n/(\Delta+1)$ such cliques. Kortsarz and Peleg [31] nevertheless managed to achieve $\tilde{O}(\Delta^{1/4})$ approximation (in polynomial-time). Their algorithm combines a relatively simple rounding of this LP with another partial solution that does not use the LP, and whose analysis relies on a combinatorial lower bound on the optimum.

Our approach is to look at the Kortsarz-Peleg LP above from the perspective of a single vertex w. Consider an integral solution H to the LP above, i.e. a valid 2-spanner. From the viewpoint of w, incident edges are included in H for two possible reasons: either to span an edge connecting two neighbors of w (i.e., including the edges $\{u, w\}$ and $\{v, w\}$ in order to span the edge $\{u, v\}$), or to span the edge itself. It's reasonable to focus on the case where H has significantly fewer edges than G, and therefore many edges in H are included because of the first reason. Let G_w be the subgraph of G induced by the neighbors of w, and let S be the subset of vertices of G_w that are adjacent to w in H. Then from the perspective of w, including the edges between w and S in H "covers" every demand formed by an edge (of G_w) that connects two vertices in S, namely $E' = \{\{u, v\} \in E : \; u, v \in S\}$. We can look at each neighborhood this way, and reinterpret LD2S as the problem of covering every demand in at least one neighborhood G_w, while minimizing the maximum degree.

This viewpoint naturally suggests an LP-based algorithm for LD2S: solve the Kortsarz-Peleg LP above (or some other relaxation), and for every $w \in V$, interpret $\sum_{\{u,v\} \in E : u,v \in \Gamma(w)} x_{\{u,v\};w}$ as the amount of "demand" that

w is supposed to cover locally, and $\sum_{u\in\Gamma(w)} x_{\{w,u\}}$ as w's "budget". Then for each $w \in V$ run a subroutine that covers the required amount of demand within the budget. Since in LD2S every demand *must* be covered, this subroutine should cover the required amount of demand but is free to somewhat violate the budget constraint; the amount of violation will correspond to the LD2S approximation guarantee. We thus need to solve the SMALLEST m-EDGE SUBGRAPH (SmES) problem: given a graph (in our case G_w) and a value m, choose as few vertices as possible subject to covering at least m edges, where an edge is covered if both its endpoints are chosen.

Unfortunately, this reduction from LD2S to SmES does not work. There are two main issues with it. First, if w chooses to add an edge to u (i.e. the SmES algorithm at G_w includes $u \in \Gamma(w)$) then this increases the degree of *both* w and u. So even if u stays within its own budget when G_u is processed, many of its neighbors might decide to add their edge to u, and the degree at u will be very large compared to its budget. Second, since we run a SmES algorithm at each vertex *separately*, they might make poorly-correlated choices as to which demands they cover. This may cause a high degree of overlap in the demands covered by different vertices, leading to much less total demand covered. Both of these problems stem from the same source: while we used the LP to define the total demand and budget at each vertex, we did not require the SmES algorithm to act in a way consistent with the LP. If we could force the SmES subroutine to make decisions that actually correspond to the fractional solution, then both of these problems would be solved. This is our motivation for defining faithfulness.

1.2. Faithful rounding

While our formal notion of faithfulness is somewhat technical and depends on the exact problem that we want to solve, the intuition behind it is natural and can apply to many problems. Suppose that we have an LP in which there are variables $\{x_e\}_{e\in U}$ (where U is a universe of elements) as well as variables $\{x_{e,e'}\}_{e,e'\in U}$. In our case, each e is a vertex in a SmES instance (i.e. an edge in LD2S) and each pair $\{e, e'\}$ is an edge in a SmES instance (a 2-path in LD2S). A standard way of interpreting fractional LP values is as *probabilities*, i.e. we think of x_e as the probability that e should be in the solution. This interpretation naturally leads to independent randomized rounding, where we take e into our solution with probability proportional to x_e. By this interpretation, $x_{e,e'}$ should be the probability that both e and e' are in the solution. But now we have a problem, since the natural constraints to force this type of situation in an integral setting, namely constraints such as $x_{e,e'} \le \min\{x_e, x_{e'}\}$, correspond poorly to the probabilities obtained by independent randomized rounding. For example, if $x_e = x_{e'} = x_{e,e'}$, then the LP "believes" that the probability that both e and e' are in the solution is $x_{e,e'}$, but under independent randomized rounding this event happens with probability $x_e \cdot x_{e'} = x_{e,e'}^2$, which could be much smaller. In a faithful rounding this does not happen: roughly speaking, faithfulness requires every element and pair of elements to be included in the solution with probability that is proportional to its LP value.

Many algorithms are naturally faithful, and indeed we suspect that one reason this notion has not been defined previously (to the best of our knowledge) is that in most cases it either falls out from the analysis "for free" or it is unnecessary. The connection we show between LD2S and faithful rounding for SmES might give one hope that the recent algorithmic breakthrough for DkS by Bhaskara, Charikar, Chlamtac, Feige and Vijayaraghavan [9] could imply better approximations for LD2S. However, their result heavily uses hierarchies, which creates a formidable obstacle for faithful rounding, as we discuss in Section 1.3.

1.3. LP hierarchies and faithful rounding

Following the lead of Bhaskara et al. [9], we employ a strong LP relaxation for SmES, which can be viewed as part of an LP hierarchy. In this context, a hierarchy is a sequence of increasingly tight relaxations to a 0-1 program, usually obtained via a general mechanism that works for any 0-1 program. Such hierarchies (for both LPs and SDPs) have been suggested by Sherali and Adams [38], Lovász and Schrijver [34], and Lasserre [32] (in our case, we use the Sherali-Adams hierarchy). A key property shared by these hierarchies is that they are locally integral; that is, the q-th relaxation in the hierarchy coincides exactly with the convex hull of feasible 0-1 solutions, when both are projected onto any q-dimensional subspace corresponding to q variables in the program.[2] Specifically for Sherali-Adams, the q-th relaxation for a given 0-1 linear program with variables $x_1, \ldots, x_N \in \{0, 1\}$, is obtained by extending the 0-1 program to include a variable x_S for every $S \subseteq \{1, \ldots, N\}$, $|S| \le q$, and then writing a "locally integral" relaxation for this extended 0-1 program to guarantee that $x_S = \prod_{i\in S} x_i$ (by convention $x_\emptyset = 1$). For more details, see the survey [15].

There has been a recent surge of interest in the study of hierarchies of LPs (or other convex programs), especially in connection with approximation algorithms for combinatorial optimization problems. Specifically, such strong relaxations can potentially lead to progress on problems whose approximability has persistent gaps, such as VERTEX-COVER and MINIMUM-BISECTION. This line of attack was probably first described explicitly in [3]. However, designing rounding procedures for these relaxations is often quite challenging.

[2]Consequently, if N denotes the number of initial 0-1 variables, then the N-th relaxation is exactly the convex hull of all 0-1 solutions, i.e., corresponds to solving the 0-1 optimization problem exactly. The q-th relaxation in the sequence can be written explicitly as a (linear) program of size $N^{O(q)}$, and thus solved in time $N^{O(q)}$.

Indeed, relatively few papers have managed to improve over state-of-the-art approximation algorithms using hierarchies. The few papers that do give improved approximation bounds using hierarchies include [11], [12], [7], [14], [9].[3] In particular, the last paper designs a rounding procedure for an LP hierarchy for DkS, which we adapt for SmES.

Our plan is to leverage the success of [9], but as mentioned before, we face a serious obstacle — their rounding procedure is not faithful. They essentially condition on a small set of events, for instance that the solution includes a small set S^* of *carefully chosen* elements, and then they use only the LP variables for sets containing this S^*, namely, a variable $x_{S^* \cup \{u\}}$ is now thought of as the LP variable for singleton u. But clearly that variable might have very little to do with the actual x_u, which is the quantity with respect to which we are trying to be faithful.

Our main technical contribution is to overcome this and design a faithful rounding for SmES based on Sherali-Adams. Our algorithm is loosely based on the DkS algorithm of [9], but numerous technical difficulties have to be resolved to make it faithful. This, together with our reduction from LD2S to faithful SmES, gives our new approximation algorithm for LD2S. We believe that our notion of faithful rounding is of independent interest, and might prove useful for other approximation algorithms, especially in the context of using hierarchies such as Sherali-Adams.

For comparison, we mention that recent algorithmic results, due to [6], [26], design rounding schemes for the Lasserre [32] hierarchy. Their rounding appears to be faithful (at least at an informal level), but it is not applicable to our context. First, their analysis holds only for expander-like graphs, and second, their rounding technique applies to problems such as constraint satisfaction and graph partitioning, with no connection to DkS.

1.4. Related work

Graph spanners, first introduced by Peleg and Schäffer [36] and Peleg and Ullman [37], have been studied extensively, with applications ranging from routing in networks (e.g. [4], [41]) to solving linear systems (e.g. [40], [20]). The foundational result on spanners is due to Althöfer, Das, Dobkin, Joseph and Soares [1], who gave an algorithm that, given a graph and an integer $k \geq 1$, constructs a $(2k-1)$-spanner with $n^{1+1/k}$ edges. Unfortunately this result obviously does not give anything nontrivial for 2-spanners, and indeed it is easy to see that there exist graphs for which every 2-spanner has $\Omega(n^2)$ edges, thus nontrivial absolute bounds on the size of a 2-spanner are not possible. Kortsarz and Peleg [30] were the first to consider relative bounds for spanners. They gave a greedy $O(\log |E|/|V|)$-approximation algorithm for the problem of finding a 2-spanner with the minimum number of edges. This was then extended to variants of 2-spanners, e.g. *client-server* 2*-spanner* [21] and *fault-tolerant* 2*-spanner* [18], [19] (for which only $O(\log \Delta)$ is known). All of these bounds are basically optimal, assuming P $\neq$ NP, due to a hardness result of Kortsarz [28].

[3]There are also papers that recover known approximation bounds, say a PTAS, while other ones show the limitations of these hierarchies by exhibiting integrality gaps for certain problems and hierarchies.

2. Outline of LP relaxation for LD2S and reduction to SmES

In this section we give an LP relaxation for LD2S that uses a relaxation of SmES as a black box, as well as an algorithm that shows how to use a faithful rounding for SmES to approximate LD2S. Both the relaxation and the algorithm presented here are simplifications that ignore some technical details; the full relaxation and algorithm, as well as all proofs, can be found in the full version [13].

We will actually give a relaxation for a slightly more general version of LD2S in which instead of spanning *all* edges we are given a subset $\hat{E} \subseteq E$ and are only required to span edges in $\hat{E}$. Note that the optimal solution for demands $\hat{E} \subseteq E$ has maximum degree that is at most the maximum degree of the optimal solution to the original LD2S problem (where all edges are demands). This will allow us to cover some demands, re-solve the LP with only the remaining demands, and repeat.

Our relaxation is a feasibility LP, so we will guess the optimal degree bound λ and use it as a constant in the LP. For each $u \in V$, let $G_u = (V_u, E_u)$ be the induced subgraph of $(V, \hat{E})$ on $\Gamma_G(u)$. Our relaxation includes a fractional SmES solution for each G_u: let **SmES-LP**(G_u) be a linear relaxation of SmES with variables $\{z^u_v\}_{v \in V_u} \cup \{z^u_e\}_{e \in E_u}$ with the property that $z^u_{\{w,v\}} \leq \min\{z^u_w, z^u_v\}$ for all $\{w,v\} \in E_u$. In a 0-1 solution this means an edge is covered only if both of its endpoints are chosen. Any polytope that includes this basic condition can be used, but obviously the tighter this relaxation is the tighter our LD2S relaxation will be, and in the end we will use a much stronger relaxation for SmES that is based on the Sherali-Adams hierarchy.

Our relaxation for LD2S with demands $\hat{E} \subseteq E$ is given by (4)-(7) in Figure 1. Constraint (4) requires that for each neighborhood graph G_u there is an associated fractional SmES solution.[4] Constraint (5) simply requires that for each edge, if either of the SmES instances at its endpoints include it in their solution then we include it in the overall solution. Constraint (6) gives the degree bound, and (7) is the main covering constraint, requiring that every demand is either included or is spanned by a 2-path. It is easy to see that this is a valid relaxation for LD2S: if we are given a 2-spanner H of G with maximum degree at most λ, for every edge $\{u,v\} \in E(H)$ we set $x_{\{u,v\}} = 1$ and $z^v_u = 1$ and $z^u_v = 1$.

[4]Our actual relaxation (see [13]) has a collection of SmES instances for each neighborhood graph based on the possible degrees in a bipartite decomposition of an optimal solution, and we allow the LP to fractionally "guess" which of these instances to use.

Figure 1. Relaxation for LD2S with demands $\hat{E}$

$$((z_v^u)_{v\in V(G_u)}, (z_e^u)_{e\in E(G_u)}) \in \textbf{SmES-LP}(G_u) \qquad \forall u \in V \tag{4}$$

$$\max\{z_v^u, z_u^v\} \le x_{\{u,v\}} \qquad \forall \{u,v\} \in E \tag{5}$$

$$\sum_{v\in\Gamma(u)} x_{\{u,v\}} \le \lambda \qquad \forall u \in V \tag{6}$$

$$\sum_{w\in\Gamma(u)\cap\Gamma(v)} z^w_{\{u,v\}} + x_{\{u,v\}} = 1 \qquad \forall \{u,v\} \in \hat{E} \tag{7}$$

For every edge $\{u,v\} \in E \setminus E(H)$ we arbitrarily choose some $w \in V$ so that $\{u,w\} \in E(H)$ and $\{w,v\} \in E(H)$ (some such w must exist since H is a 2-spanner) and set $z^w_{\{u,v\}} = 1$. All other variables are 0.

We now show that it is sufficient to design a rounding scheme for SmES that is faithful according to the following definition. Given a graph G, let $\mathcal{L}(G)$ be an LP that has a variable ζ_u for every $u \in V(G)$ and a variable ζ_e for every $e \in E(G)$ (we will later instantiate $\mathcal{L}(G)$ as various LP relaxations of SmES).

Definition 3. A randomized rounding algorithm $\mathcal{A}$ is a *factor f faithful rounding* for $\mathcal{L}(G)$ if, when given a valid solution $((\zeta_u)_{u\in V}, (\zeta_e)_{e\in E})$ to $\mathcal{L}(G)$, it produces a randomized (not necessarily induced) subgraph $H^* = (V^*, E^*)$ such that

a) $\Pr[v \in V^*] \le f \cdot \zeta_v$ for all $v \in V(G)$,
b) $\Pr[\{u,v\} \in E^*] \le \zeta_{\{u,v\}}$ for all $\{u,v\} \in E(G)$,
c) $|V^*| \le f \cdot \sum_{v\in V(G)} \zeta_v$ (with probability 1), and
d) $\mathbb{E}[|E^*|] \ge \tilde{\Omega}(\sum_{\{u,v\}\in E(G)} \zeta_{\{u,v\}})$.

Observe that if algorithm $\mathcal{A}$ is a factor f faithful rounding for a relaxation of SmES then it is also an f-approximation in the usual sense, simply by conditions c and d (up to a polylogarithmic loss in the amount of edges covered). The converse, however is not true: many rounding algorithms that give an f-approximation are not faithful, including [9].

We now show that if we are given an algorithm $\mathcal{A}$ that is a factor $f(n)$ faithful rounding for SmES (where n is the number of vertices in the SmES instance), there is an $\tilde{O}(f(\Delta))$-approximation algorithm for LD2S that uses algorithm $\mathcal{A}$ as a black box. The reduction is given as Algorithm 1. It begins with all edges as the demand set $\hat{E}$, and first solves the LP relaxation for LD2S with demand set $\hat{E}$. It adds every edge that has x value at least $1/4$, and then uses algorithm $\mathcal{A}$ to round each of the $|V|$ SmES instances in the relaxation. At the end of the loop it updates the demands $\hat{E}$ by removing edges that were successfully covered by this process, and repeats. Note that the *edges* covered by the SmES roundings are used only in the analysis; in the algorithm we take the *vertices* output by each SmES solution and include the appropriate edges in our spanner.

Algorithm 1: Approximation algorithm for LD2S

Input : Graph $G = (V, E)$, degree bound λ, factor $f(n)$ faithful rounding algorithm $\mathcal{A}$ for SmES
Output: 2-spanner $H = (V, E_H)$ of G

$\hat{E} \leftarrow E$, $E_H \leftarrow \emptyset$
while $\hat{E} \neq \emptyset$ **do**
  Compute a valid solution $\langle \vec{x}, \vec{z} \rangle$ for LP (4)-(7) on graph G with demands $\hat{E}$
  $E_x \leftarrow \{e \in E : x_e \ge 1/4\}$
  foreach $u \in V$ **do**
    $H_u^* \leftarrow \mathcal{A}(G_u, \vec{z^u})$; `// output of SmES rounding` $\mathcal{A}$
    $E_u \leftarrow \{\{u,v\} \in E \mid v \in V(H_u^*)\}$
  `// Add all edges found in above rounding`
  $E_H \leftarrow E_H \cup E_x \cup \left(\bigcup_{u\in V} E_u\right)$
  `// Remove satisfied demands`
  $\hat{E} \leftarrow \hat{E} \setminus \left(E_H \cup \left\{\{u,v\} \mid \exists w \in V : \{u,w\}, \{w,v\} \in E_H\right\}\right)$

Theorem 4. *Let algorithm $\mathcal{A}$ be a factor $f(n)$ faithful rounding for* SmES *(where n is the number of vertices in the* SmES *instance). Then there is a (randomized) $\tilde{O}(f(\Delta))$-approximation for* LD2S.

Proof: We provide only a sketch of the proof here, since as mentioned both the relaxation for LD2S and the rounding algorithm based on SmES are simplifications (see the full version [13] for the complete relaxation and algorithm). We may assume that our LD2S algorithm guesses some $\lambda \in [\mathrm{OPT}, 2\cdot\mathrm{OPT}]$ simply by trying the $O(\log\Delta)$ relevant values and reporting the best solution. In this case, LP (4)-(7) is guaranteed to have a feasible solution. We now use Algorithm 1 with this value of λ. It is easy to see that each iteration of the loop only increases the maximum degree by $\tilde{O}(f(\Delta)) \cdot \mathrm{OPT}$: adding edges in E_x only costs a constant factor more than the fractional solution, Definition 3(c) implies that rounding the SmES solution at u only increases the degree of u by $f(|V_u|) \cdot \mathrm{OPT} \le f(\Delta) \cdot \mathrm{OPT}$, and

Definition 3(a) together with the LP imply that rounding the $S m$ES solution at neighbors of u only increases the degree of u by $\tilde{O}(f(|V_u|)) \cdot \mathrm{OPT}$ (with high probability).

So we just need to show that the number of iterations is (with high probability) at most $\tilde{O}(1)$. To do this we first prove that in every iteration the expected number of satisfied demands is at least $\tilde{\Omega}(|\hat{E}|)$. Fix some iteration. We clearly cover enough demand if $|E_x| \geq \Omega(|\hat{E}|)$. Otherwise, summing (7) over all $\{u,v\} \in \hat{E}$ implies that the total amount of demand covered by $S m$ES instances (i.e. the z variables) is large; in particular,

$$\sum_{\{u,v\} \in \hat{E}} \sum_{w \in \Gamma(u) \cap \Gamma(v)} z^w_{\{u,v\}} \geq \Omega(|\hat{E}|). \tag{8}$$

For every $\{u,v\} \in \hat{E}$ and $w \in \Gamma(u) \cap \Gamma(v)$, let $p^w_{\{u,v\}}$ denote the probability that $\{u,v\}$ was covered by the $S m$ES rounding at w, i.e. the probability that $\{u,v\} \in E(H^*_w)$. Then Definition 3(d) and (8) imply that

$$\begin{aligned} \sum_{\{u,v\} \in \hat{E}} \sum_{w \in \Gamma(u) \cap \Gamma(v)} p^w_{\{u,v\}} &\geq \sum_{\{u,v\} \in \hat{E}} \tilde{\Omega}\left(\sum_{w \in \Gamma(u) \cap \Gamma(v)} z^w_{\{u,v\}} \right) \\ &\geq \tilde{\Omega}(|\hat{E}|). \end{aligned} \tag{9}$$

For any $\{u,v\} \in \hat{E}$, we know from Definition 3(b) and (7) that

$$\sum_{w \in \Gamma(u) \cap \Gamma(v} p^w_{\{u,v\}} \leq \sum_{w \in \Gamma(u) \cap \Gamma(v)} z^w_{\{u,v\}} \leq 1. \tag{10}$$

We can then deduce that the probability that we cover demand $\{u,v\} \in \hat{E}$ (in a single iteration) is at least $\frac{1}{2} \sum_{w \in \Gamma(u) \cap \Gamma(v)} p^w_{\{u,v\}}$, by simply using the following well-known argument: if t (pairwise) independent events occur with probabilities $q_1, \ldots, q_t$ that sum up to $\sum_{i=1}^t q_i \leq 1$, then by the Bonferroni inequality, the probability that at least one of these events occurs is at least

$$\sum_i q_i - \sum_{i<j} q_i q_j = \sum_i q_i - \tfrac{1}{2} \sum_{i \neq j} q_i q_j \geq \tfrac{1}{2} \sum_i q_i. \tag{11}$$

We thus obtain that the expected number of demands covered (in a single iteration) is at least

$$\tfrac{1}{2} \sum_{\{u,v\} \in \hat{E}} \sum_{w \in \Gamma(u) \cap \Gamma(v)} p^{w,}_{\{u,v\}} \geq \tilde{\Omega}(|\hat{E}|),$$

where the inequality is from (9), (10), and (11).

So we now know that in every iteration at least a $c' = \tilde{\Omega}(1)$ fraction of the remaining demands $\hat{E}$ are satisfied in expectation, or equivalently that in expectation the number of remaining unsatisfied demands is at most a $(1-c')$ fraction of the previous number of demands. To see that this is sufficient, note that by Markov's inequality with probability at most $1 - c'/2$ the number of remaining demands is at least a $\frac{1}{1-c'/2}(1-c') = 1 - \frac{c'/2}{1-c'/2}$ fraction of what it was. Equivalently, with probability at least $c'/2$ at least a $\frac{c'/2}{1-c'/2}$ fraction of demands are covered. Thus the probability that this does not happen after $(8/c') \ln n = \tilde{O}(1)$ iterations is at most $(1-c'/2)^{(8/c')\ln n} \leq 1/n^4$. So with high probability, after $\tilde{O}(1)$ rounds the number of unsatisfied demands is at most $1 - \frac{c'/2}{1-c'/2} \leq 1 - c'/2$ of what it was. Now if this happens $(4/c') \ln n = \tilde{O}(1)$ times then the number of remaining demands is at most $|E|(1-c'/2)^{(4/c')\ln n} < 1$, so the algorithm terminates. Thus with high probability the number of iterations is at most $(8/c') \ln n \cdot (4/c') \ln n = \tilde{O}(1)$, as required. ∎

3. A FAITHFUL ROUNDING ALGORITHM FOR $S m$ES

In this section we describe our faithful factor $n^{3-2\sqrt{2}+\varepsilon}$ rounding algorithm for $S m$ES. While the description of the full algorithm is rather lengthy, and therefore deferred to the full version, we give a high-level overview (with some technical details) and concentrate on a special case which illustrates the main ideas in the algorithm and its analysis.

3.1. DENSEST k-SUBGRAPH *and the log-density framework*

We follow the framework introduced in [9]. They begin by defining the notion of log-density of a graph as $\log_n(D_{\mathrm{avg}})$, where D_{avg} is the average degree and n is the number of nodes. They then asked the following question: how hard is it to distinguish between 1) a random graph, and 2) a graph containing a subgraph with roughly the same log-density as the first graph?

More formally, they pose the following DENSE VERSUS RANDOM promise problem, parameterized by k and constants $0 < \alpha, \beta < 1$: given a graph G, distinguish between the following two cases:

1) $G = G(n,p)$ where $p = n^{\alpha-1}$ (this graph has log-density concentrated around α).
2) G is adversarially chosen so that the densest k-subgraph has log-density β (where $k^{1+\beta} \gg pk$).

For certain ranges of parameters, it seems quite challenging to efficiently distinguish when $\beta < \alpha$. In fact, the following hypothesis is consistent with the current state of our knowledge:

Hypothesis 5. *For all $0 < \alpha < 1$, for all sufficiently small $\varepsilon > 0$, and for all $k \leq \sqrt{n}$, we cannot solve* DENSE VERSUS RANDOM *in polynomial time (w.h.p.) when $\beta \leq \alpha - \varepsilon$.*

The above hypothesis (if true) has immediate implications for the hardness of approximation of both DkS and $S m$ES. Concretely, for $S m$ES, let $m = k^{1+\beta}$ be the number of edges in k-subgraph in the second case. We know that in the first case w.h.p. the smallest m-edge subgraph has size at least $\widetilde{\Omega}(\min\{m, \sqrt{mn^{1-\alpha}}\})$. Thus, if we could achieve approximation ratio $\ll k/\min\{m, \sqrt{mn^{1-\alpha}}\}$, this would refute Hypothesis 5 for the corresponding parameters. For $k = n^{\sqrt{2}-1}$ and $\alpha = \sqrt{2}-1$, the hypothesis implies that there exists no $n^{3-2\sqrt{2}-\varepsilon}$-approximation for $S m$ES.

While [9] matches the gap predicted by the log-density model for DkS with an $n^{1/4+\varepsilon}$ approximation for DkS (even for general graphs), we also match the predicted gap for SmES with an $n^{3-2\sqrt{2}+\varepsilon}$-approximation for S$m$ES.

3.2. Parametrization and simplifications

In order to achieve a faithful rounding, we will make certain assumptions (which we later justify) about the structure of the intended solution to the LP relaxation. In particular, we will assume that the subgraph represented by the solution is regular, and that we are allowed to "guess" the size of the subgraph, k, and the degree in the subgraph, d, thus $m = \Theta(kd)$.

We also make the following simplifying assumptions. Let $f = f(n,k,d)$ be the intended approximation factor, which will be determined shortly. We may assume that $f \leq d$, since it is easy to achieve a faithful $O(d)$-approximation. We also assume that the maximum degree in the input graph is at most $D = nd/(kf^2)$. All of these assumptions are without loss of generality (see the full version [13] for details).

Let $\alpha = \log_n(D)$. We define our intended approximation f implicitly as the value which satisfies $f = n^{\alpha(1-\alpha)/(1+\alpha)}$ (together with the definition of D we can derive an explicit expression for α and f). Note that maximizing this expression over $\alpha \in [0,1]$ shows that $f \leq n^{3-2\sqrt{2}}$.

3.3. LP relaxation and faithful rounding for SmES

With the previous assumptions in mind, we can write a feasibility-LP relaxation (simplified for this overview) which is implied by q rounds of Sherali-Adams. This relaxation is presented in Figure 2. The variables are $\{z_T : T \subset V \cup E, |T| \leq q\}$ (in the intended 0-1 solution, $z_T = 1$ if and only if all vertices and edges in T are in the subgraph).

The algorithm in its full generality is based on the caterpillar structures introduced in [9] (where the caterpillar structure depends on α). Let us concentrate here on the case where $\alpha = 1/s$ for some (fixed) integer $s > 0$, in which case the caterpillar is simply a path of length s. At its core, the algorithm (for this value of α) relies on an LP-analogue of the following combinatorial argument. Fix a vertex v_0 in the optimum subgraph. For all $t = 1, \ldots, s$, let $P_t^{v_0}$ be the union of all (possibly self-intersecting) paths of length t in the subgraph starting at v_0, and let $V_t^{v_0}$ be final endpoints of those paths. Note that $|V_1^{v_0}| = d$ and that

$$|V_s^{v_0}| \leq k = \frac{dn}{f^2 D} = \frac{d}{f^2} \cdot n^{1-\alpha} = \frac{d}{f^2} f^{(1+\alpha)/\alpha} = df^{s-1}.$$

Therefore, there must be some $t \in \{1, \ldots, s-1\}$ for which $|V_{t+1}^{v_0}|/|V_t^{v_0}| \leq f$. Now consider the subgraph at this step

$$H_t^{v_0} = (V_t^{v_0}, V_{t+1}^{v_0}, \{\{v_t, v_{t+1}\} : \exists v_0 - \ldots - v_t - v_{t+1} \in P_{t+1}^{v_0}\}).$$

Since the vertices in $V_t^{v_0}$ all have degree d, the average degree of vertices in $V_{t+1}^{v_0}$ is at least d/f. It turns out that even without access to the optimum subgraph we can isolate a subgraph with average degree at least d/f and at most kf vertices (this is essentially because by the degree bound, the number of vertices at any intermediate stage is at most $D^{s-1} = n^{1-\alpha} = \frac{n}{D} = k \cdot \frac{f^2}{d} \leq kf$). This essentially gives an f-approximation for SmES (since we can repeat until accumulating m edges).

Here we come to the fundamental difficulty in adapting such an approach to achieve a faithful rounding. The combinatorial algorithm depends on choosing an initial vertex v_0 which is actually in the optimum subgraph. The analogous LP-rounding algorithm uses the LP values "conditioned on choosing v_0", that is, values of the form $z_{S \cup \{v_0\}}/z_{v_0}$ instead of the original z_S variables (where S corresponds to one or more vertices/edges along the path). However, it is the z_S variables (in particular for singleton sets S representing one vertex or one edge) which we want to be faithful to in our rounding.[5] Unfortunately, these two LP solutions might be almost completely unrelated.

To overcome this difficulty, we use a somewhat elaborate bucketing scheme, to ensure that all the relevant LP values are reasonably uniform, as follows. Denote by $\mathcal{P}_t^v$ the set of all length t paths in the graph starting at vertex v, and by z_p the variable for a path p (i.e., z_T where T is the set of edges and vertices in p, or by Constraint (15), T could equivalently be just the edges in p). The core of the analysis of the LP-analogue relies on the equality

$$\sum_v \sum_{p \in \mathcal{P}_t^v} z_p = \sum_v d^t z_{\{v\}} = kd^t,$$

obtained by Constraint (12) and repeated applications of (13), but in fact it can use any set of length-s paths $\mathcal{P}$ for which $\sum_{p \in \mathcal{P}} z_p = \widetilde{\Omega}(kd^s)$. Thus by partitioning the set of paths $\bigcup_v P_s^v$ into buckets and choosing a bucket $\mathcal{P}$ with the largest LP value, we can ensure that in every path $p = u_0 - u_1 - \ldots - u_s$ in the bucket $\mathcal{P}$ certain LP values (like the ones corresponding to entire paths, z_p, or the ones corresponding to path prefixes, $z_{\{\{u_{i-1},u_i\} | i \in [t]\}}$ for some $t \in [s-1]$, or to vertices in certain positions, $z_{\{u_t\}}$, or to "conditioned" values, $z_{\{u_0,u_t\}}/z_{\{u_0\}}$) are all independent of the choice of path (up to a constant factor). In other words, within the bucket $\mathcal{P}$ (say, vertices u_t for a fixed $t \in \{0, \ldots, s\}$), the corresponding LP values will be essentially uniform over the choice of starting vertex u_0 and path p.

Using the uniformity obtained via the above bucketing scheme, we can relate the algorithm (which is based on the conditioned LP values) to the original LP values. After some additional combinatorial bucketing, we can run the following algorithm: let $\mathcal{V}_0$ be the set of starting vertices u_0 (i.e. paths of length 0) that survive the bucketing, pick a starting vertex $u_0 \in \mathcal{V}_0$ uniformly at random, and for whichever level

[5] This problem is only exacerbated in the general case, when the caterpillar has additional leaves to condition on.

Figure 2. Relaxation for smES

$$\sum_{v\in V} z_{T\cup\{v\}} = kz_T \qquad \forall T \subset V \cup E,\ |T| \le q-1 \tag{12}$$

$$\sum_{u\in\Gamma(v)} z_{T\cup\{u\}} = dz_T \qquad \forall T \subset V \cup E : |T| \le q-1, \forall v \in T \cap V \tag{13}$$

$$z_T = z_{T\cup\{u\}} = z_{T\cup\{v\}} = z_{T\cup\{u,v\}} \qquad \forall T \subset V \cup E,\ |T| \le q-2, \forall \{u,v\} \in T \cap E \tag{14}$$

$$0 \le z_T \le z_{T'} \le z_\emptyset = 1 \qquad \forall T' \subseteq T \tag{15}$$

$t \in [s-1]$ that gives the approximation guarantee (it can be shown that such a t exists), output the level t subgraph $H_t^{u_0} = \{\{u_t, u_{t+1}\} \mid \exists p = u_0 - u_1 - \ldots - u_s \in \mathcal{P}\}$. Since LP values are uniform, the question essentially becomes, how do we guarantee that no bucketed vertex (or edge) is chosen with much higher probability than the rest (or the average)? This is where we crucially use the regularity Constraint (13) (as opposed to, say, a minimum degree constraint, as in [9]). Roughly speaking, individual vertices and edges cannot be reached by a disproportionately large fraction of vertices $u_0 \in \mathcal{V}_0$, because then the relative total LP weight of the corresponding paths (to such a vertex or edge) would exceed d^t.

For the sake of concreteness, let us consider one specific aspect of faithful rounding: the probability with which the level t vertices u_t are chosen. Let $\mathcal{P}_t$ be the set of length t prefixes of paths in $\mathcal{P}$, let $\mathcal{P}_t^{u_0}$ be the set of paths in $\mathcal{P}_t$ that start with the vertex $u_0 \in \mathcal{V}_0$, and let $\mathcal{V}_t$ (resp. $\mathcal{V}_t^{u_0}$) be the set of level t endpoints of paths in $\mathcal{P}_t$ (resp. in $\mathcal{P}_t^{u_0}$). By the approximation guarantee (via an LP analogue of the above combinatorial argument), we have

$$|\mathcal{V}_t^{u_0}| \le fk. \tag{16}$$

Suppose the bucketing also ensures that every $u_0 \in \mathcal{V}_0$ and $u_t \in \mathcal{V}_t^{u_0}$ are connected by roughly the same number of $\mathcal{P}_t$ paths (up to a constant factor), which we denote by h. Also, suppose the cardinalities $|\mathcal{V}_t^{u_0}|$ are roughly uniform for different choices of u_0. Then, abusing notation, we can write the number of paths as $|\mathcal{P}_t| \approx |\mathcal{V}_0| \cdot |\mathcal{V}_t^{u_0}| \cdot h$, and in particular, the total weight of paths $p \in \mathcal{P}_t$ is $z_p |\mathcal{V}_0| \cdot |\mathcal{V}_t^{u_0}| \cdot h \approx kd^t$. Now, by repeated applications of Constraint (13), we have that the total weight of paths leading to a specific vertex $u_t \in \mathcal{V}_t$ is $z_p |\{u_0 \mid u_t \in \mathcal{V}_t^{u_0}\}| h \le z_{u_t} d^t$ (note that this argument reverses the direction of paths in the algorithm and so crucially depends on the existence of consistent high-moment Sherali-Adams variables, which are not present in the Lovász-Schrijver hierarchy used in [9]). Combining this with the (approximate) equality above, we can bound the probability that a vertex u_t is included in the output (the level t subgraph) as

$$\frac{|\{u_0 \mid u_t \in \mathcal{V}_t^{u_0}\}|}{|\mathcal{V}_0|} \le \frac{d^t z_{u_t}}{z_p h |\mathcal{V}_0|} = \frac{|\mathcal{V}_t^{u_0}| z_{u_t}}{k} \le f z_{u_t},$$

by (16).

4. Discussion and Future Directions

Some features of our techniques might be applicable to other problems. Most obviously, this is perhaps the first time that LP hierarchies are applied to "local" parts of an LP, rather than to the entire LP. Can this approach be useful for other problems? Currently, it is not clear to us how this approach fares against one "global" application of an LP hierarchy to some basic relaxation: a global hierarchy could take advantage of non-locality in the constraints and solution, but on the other hand would not allow us to locally "guess" degrees (see e.g. footnote 4).

Persistent gaps in the approximability of other network design problems naturally call for a judicious use of LP hierarchies in order to obtain better approximation algorithms. For example, the BASIC k-SPANNER problem, in which the goal is to construct a k-spanner with as few edges as possible, is only known to admit approximation ratio $O(n^{1/\lfloor (k+1)/2 \rfloor})$ [1], while the best hardness of approximation is $2^{(\log^{1-\varepsilon} n)/k}$ for arbitrarily small constant $\varepsilon > 0$ [17]. An integrality gap that almost matches the upper bound (namely a gap of $n^{\Omega(1/k)}$) was recently shown by Dinitz and Krauthgamer [18], but stronger relaxations obtained via hierarchies can possibly have smaller integrality gaps. In particular, it is not at all clear what the best achievable approximation ratio is for the regime when k is constant; perhaps hierarchies will finally allow upper bounds that beat [1] (note that this has been done using other techniques for $k = 3$ by [8], who gave an $\tilde{O}(n^{1/3})$-approximation for BASIC 3-SPANNER). Similarly, for directed k-spanner the known upper bound is $\tilde{O}(\sqrt{n})$ [8], and there is an $\tilde{\Omega}(n^{1/3})$ integrality gap [18], but it only applies to a simple LP relaxation. Yet other relevant problems are DIRECTED STEINER TREE and DIRECTED STEINER FOREST, see [24], [8] and references therein. Perhaps hierarchies could help for some of these problems?

Finally, the connection we show between LD2S and SmES suggests an intriguing possibility for conditional lower bounds. The current hardness for LD2S is only $\Omega(\log n)$, while SmES is basically as hard as DkS, which is commonly thought to be difficult to approximate well (say within a polylogarithmic factor, although current hardness results rely on various complexity assumptions and give only a relatively small constant [22], [27]). A reduction in the other direction, i.e. from SmES to LD2S, could give partial evidence that LD2S cannot be approximated well, and could possibly even match the upper bound that we prove here. The same arguments about a formal connection to DkS obviously apply also to other network design problems, such as BASIC k-SPANNER.

REFERENCES

[1] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, "On sparse spanners of weighted graphs," *Discrete Comput. Geom.*, vol. 9, no. 1, pp. 81–100, 1993.

[2] B. Anthony, V. Goyal, A. Gupta, and V. Nagarajan, "A plant location guide for the unsure: Approximation algorithms for min-max location problems," *Math. Oper. Res.*, vol. 35, no. 1, pp. 79–101, Feb. 2010.

[3] S. Arora, B. Bollobás, and L. Lovász, "Proving integrality gaps without knowing the linear program," in *43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 313–322.

[4] B. Awerbuch and D. Peleg, "Online tracking of mobile users," *J. ACM*, vol. 42, no. 5, pp. 1021–1058, 1995.

[5] B. Awerbuch, A. Baratz, and D. Peleg, "Efficient broadcast and lightweight spanners," Weizmann Institute of Science, Tech. Rep. CS92-22, 1992.

[6] B. Barak, P. Raghavendra, and D. Steurer, "Rounding semidefinite programming hierarchies via global correlation," in *52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011.

[7] M. Bateni, M. Charikar, and V. Guruswami, "Maxmin allocation via degree lower-bounded arborescences," in *41st Annual ACM Symposium on Theory of Computing (STOC)*, 2009, pp. 543–552.

[8] P. Berman, A. Bhattacharyya, K. Makarychev, S. Raskhodnikova, and G. Yaroslavtsev, "Improved approximation for the directed spanner problem," in *38th International Colloquium on Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, vol. 6755. Springer, 2011, pp. 1–12.

[9] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan, "Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph," in *42nd ACM Symposium on Theory of Computing*, 2010, pp. 201–210.

[10] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff, "Transitive-closure spanners," in *20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009, pp. 932–941.

[11] E. Chlamtac, "Approximation algorithms using hierarchies of semidefinite programming relaxations," in *48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2007, pp. 691–701.

[12] E. Chlamtac and G. Singh, "Improved approximation guarantees through higher levels of SDP hierarchies," in *11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. Springer-Verlag, 2008, pp. 49–62.

[13] E. Chlamtac, M. Dinitz, and R. Krauthgamer, "Everywhere-sparse spanners via dense subgraphs," *CoRR*, vol. abs/1205.0144, 2012. [Online]. Available: http://arxiv.org/abs/1205.0144

[14] E. Chlamtac, R. Krauthgamer, and P. Raghavendra, "Approximating sparsest cut in graphs of bounded treewidth," in *13th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, ser. Lecture Notes in Computer Science, vol. 6302. Springer, 2010, pp. 124–137.

[15] E. Chlamtac and M. Tulsiani, "Convex relaxations and integrality gaps," in *Handbook on Semidefinite, Conic and Polynomial Optimization*, ser. International Series in Operations Research & Management Science, M. F. Anjos and J. B. Lasserre, Eds. Springer, 2012, vol. 166, pp. 139–169.

[16] M. Dinitz, "Compact routing with slack," in *26th Annual ACM Symposium on Principles of Distributed Computing*, 2007, pp. 81–88.

[17] M. Dinitz, G. Kortsarz, and R. Raz, "Label cover instances with large girth and the hardness of approximating basic k-spanner." in *ICALP (1)*, 2012, pp. 290–301.

[18] M. Dinitz and R. Krauthgamer, "Directed spanners via flow-based linear programs," in *43rd Annual ACM Symposium on Theory of Computing (STOC)*, 2011, pp. 323–332.

[19] ——, "Fault-tolerant spanners: better and simpler," in *30th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2011, pp. 169–178.

[20] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng, "Lower-stretch spanning trees," *SIAM J. Comput.*, vol. 38, no. 2, pp. 608–628, 2008.

[21] M. Elkin and D. Peleg, "The client-server 2-spanner problem with applications to network design," in *8th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2001, pp. 117–132.

[22] U. Feige, "Relations between average case complexity and approximation complexity," in *Proceedings of the 34th annual ACM Symposium on Theory of Computing (STOC)*, 2002, pp. 534–543.

[23] U. Feige and M. Seltser, "On the densest k-subgraph problem," Weizmann Institute of Science, Rehovot, Israel, Tech. Rep., 1997.

[24] M. Feldman, G. Kortsarz, and Z. Nutov, "Improved approximating algorithms for directed Steiner forest," in *20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009, pp. 922–931.

[25] A. Gupta, M. Hajiaghayi, V. Nagarajan, and R. Ravi, "Dial a ride from k-forest," in *European Symposium on Algorithms (ESA)*, 2007, pp. 241–252.

[26] V. Guruswami and A. K. Sinop, "Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives," in *52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011.

[27] S. Khot, "Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique," *SIAM J. Comput.*, vol. 36, no. 4, pp. 1025–1071, 2006.

[28] G. Kortsarz, "On the hardness of approximating spanners,"

Algorithmica, vol. 30, no. 3, pp. 432–450, 2001.

[29] G. Kortsarz and D. Peleg, "On choosing a dense subgraph," in *34th Annual Symposium on Foundations of Computer Science*, 1993, pp. 692 –701.

[30] ——, "Generating sparse 2-spanners," *J. Algorithms*, vol. 17, no. 2, pp. 222–236, 1994.

[31] ——, "Generating low-degree 2-spanners," *SIAM J. Comput.*, vol. 27, no. 5, pp. 1438–1456, 1998.

[32] J. B. Lasserre, "An explicit equivalent positive semidefinite program for nonlinear 0-1 programs," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 756–769, 2002.

[33] L. C. Lau, J. S. Naor, M. R. Salavatipour, and M. Singh, "Survivable network design with degree or order constraints," *SIAM J. Comput.*, vol. 39, no. 3, pp. 1062–1087, 2009.

[34] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization," *SIAM Journal on Optimization*, vol. 1, no. 12, pp. 166–190, 1991.

[35] Z. Nutov, "Approximating steiner networks with node-weights," *SIAM Journal on Computing*, vol. 39, no. 7, pp. 3001–3022, 2010.

[36] D. Peleg and A. A. Schäffer, "Graph spanners," *J. Graph Theory*, vol. 13, no. 1, pp. 99–116, 1989.

[37] D. Peleg and J. D. Ullman, "An optimal synchronizer for the hypercube," *SIAM J. Comput.*, vol. 18, pp. 740–747, August 1989.

[38] H. D. Sherali and W. P. Adams, "A hierarchy of relaxation between the continuous and convex hull representations," *SIAM J. Discret. Math.*, vol. 3, no. 3, pp. 411–430, 1990.

[39] M. Singh and L. C. Lau, "Approximating minimum bounded degree spanning trees to within one of optimal," in *STOC '07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, 2007, pp. 661–670.

[40] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *36th Annual ACM Symposium on Theory of Computing*. ACM, 2004, pp. 81–90.

[41] M. Thorup and U. Zwick, "Approximate distance oracles," *J. ACM*, vol. 52, no. 1, pp. 1–24, 2005.

[42] ——, "Compact routing schemes," in *13th annual ACM Symposium on Parallel Algorithms and Architectures*. ACM Press, 2001, pp. 1–10.