Shallow-Low-Light Trees, and Tight Lower Bounds for Euclidean Spanners

Yefim Dinitz[†], Michael Elkin[*†], Shay Solomon[‡]
Department of Computer Science,
Ben-Gurion University of the Negev,
POB 653, Beer-Sheva 84105, Israel.
E-mail: {dinitz,elkinm,shayso}@cs.bgu.ac.il

Abstract

We show that for every n-point metric space M and positive integer k, there exists a spanning tree T with unweighted diameter $O(k)$ and weight $w(T) = O(k \cdot n^{1/k}) \cdot w(MST(M))$, and a spanning tree T' with weight $w(T') = O(k) \cdot w(MST(M))$ and unweighted diameter $O(k \cdot n^{1/k})$. Moreover, there is a designated point rt such that for every other point v, both $dist_T(rt, v)$ and $dist_{T'}(rt, v)$ are at most $(1+\epsilon) \cdot dist_M(rt, v)$, for an arbitrarily small constant $\epsilon > 0$.

We prove that the above tradeoffs are tight up to constant factors *in the entire range of parameters. Furthermore, our lower bounds apply to a basic one-dimensional Euclidean space. Finally, our lower bounds for the particular case of unweighted diameter $O(\log n)$ settle a long-standing open problem in Computational Geometry.*

1 Introduction

1.1 Background and Main Results

Spanning trees for finite metric spaces have been a subject of an ongoing intensive research since the beginning of the nineties [3, 11, 12, 18, 32, 29, 13, 37, 10, 46, 9, 49]. In particular, many researchers studied the notion of *shallow-light trees*, henceforth SLTs [13, 37, 10, 9, 49, 4, 46]. Roughly speaking, an SLT of an n-point metric space M is a spanning tree T of the complete graph corresponding to M whose total weight is close to the weight $w(MST(M))$ of the minimum spanning tree $MST(M)$ of M, and whose weighted diameter is close to that of M. (See Section 2 for relevant definitions.)

In addition to being an appealing combinatorial object, SLTs turned out to be useful for various data gathering and dissemination problems in the message-passing model of distributed computing [10], in approximation algorithms [49], for constructing spanners [9, 4], and for VLSI-circuit design [23, 25, 24]. Near-optimal tradeoffs between the weight and diameter of SLTs were established by Awerbuch et al. [9], and by Khuller et al. [39].

Even though the requirement that the spanning tree T will have a small weighted-diameter is a natural one, it is no less natural to require it to have a small *unweighted diameter* (also called *hop-diameter*). The latter requirement guarantees that any two points of the metric space will be connected in T by a path that consists of only a small *number of edges* or *hops*. This guarantee turns out to be particularly important for routing [36, 1], computing almost shortest paths [21, 22], and in other applications.

In this paper we investigate a related notion of *low-light trees*, henceforth LLTs, that combine small weight with small hop-diameter. We present near-tight upper and lower bounds on the parameters of LLTs. In addition, our constructions of LLTs have *optimal maximum degree*.

To specify our results, we need some notation. For a spanning subgraph G of a metric space M, let $H = H(G)$ denote the hop-diameter of G, and $\Psi = \Psi(G) = \frac{w(G)}{w(MST(M))}$ denote the ratio between its weight and the weight of the minimum spanning tree of M, henceforth the *lightness* of G. The *hop-radius* $h(G, rt)$ of G with respect to a distinguished vertex rt is the maximum unweighted distance between rt and some vertex v in G. Obviously, $h(G, rt) \leq H(G) \leq 2 \cdot h(G, rt)$. For a rooted tree (T, rt), the hop-radius (called also *depth*) of (T, rt) is the hop-radius of T with respect to rt. The hop-radius of G, denoted $h(G)$, is defined by $h(G) = \min\{h(G, rt) \mid rt \in V\}$.

We show the following bounds that are tight up to con-

[1]This research has been supported by the Israeli Academy of Science, grant 483/06. Additional funding was provided by the Lynn and William Frankel Center for Computer Sciences
[2]Partially supported by the Lynn and William Frankel Center for Computer Science.

stant factors in the *entire* range of the parameters.

1. For any sufficiently large integer n and positive integer h, and *any* n-point metric space M, there exists a spanning tree of M with hop-radius at most h and lightness at most $O(\Psi)$, for Ψ that satisfies the following relationship. If $h \geq \log n$ then ($h = O(\Psi \cdot n^{1/\Psi})$ and $\Psi = O(\log n)$). In the complementary range $h < \log n$, it holds that $\Psi = O(h \cdot n^{1/h})$.
   Moreover, this spanning tree is a binary one whenever $h \geq \log n$, and it has the *optimal* maximum degree $\lceil n^{1/h} \rceil$ whenever $h < \log n$. In addition, in the entire range of parameters the respective spanning trees can be constructed in optimal time $O(n^2)$.

2. For n and h as above, and $h \geq \log n$, there exists an n-point metric space $M^* = M^*(n)$ for which any spanning subgraph with hop-radius at most h has lightness at least $\Omega(\Psi)$, for some Ψ satisfying $h = \Omega(\Psi \cdot n^{1/\Psi})$.

3. For n and h as above, and $h < \log n$, any spanning subgraph with hop-radius at most h for $M^*(n)$ (see item 2) has lightness at least $\Psi = \Omega(h \cdot n^{1/h})$.

(Note that the equation $x \cdot n^{1/x} = \Theta(\log n)$ holds if and only if $x = \Theta(\log n)$.) See Figure 1 for an illustration of our results.

The small maximum degree of our LLTs may be helpful for various applications in which the degree of a vertex v corresponds to the load on a processor that is located in v. The requirement to achieve small maximum degree is particularly important for applications in Computational Geometry. (See [6, 14, 8, 36], and the references therein.)

We extend our constructions and devise trees that also provide a good approximation of all *weighted* distances from *any* given designated root vertex rt. The resulting spanning trees achieve small weight, hop-diameter, and weighted-diameter *simultaneously*! In other words, these trees combine the useful properties of SLTs and LLTs in *one construction*, and thus we call them *shallow-low-light-trees*, henceforth SLLTs. Specifically, for M and n as above, a positive real number $\epsilon > 0$, and a designated point rt, our construction provides a spanning tree T with the same (up to a factor of $O(\epsilon^{-1})$) optimal guarantees on the weight and hop-radius as in our construction of low-light trees, and which also satisfies that for every point $v \in M$, $dist_T(rt, v) \leq (1 + \epsilon) \cdot dist_M(rt, v)$. (However, the maximum degree of our SLLTs may be unbounded.)

Observe that to represent a general n-point metric space M one needs $O(n^2)$ space. Thus, the running time $O(n^2)$ of our algorithm is *linear* in the input size. Moveover, there is a variant of our algorithm that runs in time $O(n \cdot \log n)$ for *Euclidean* n-point metric spaces of any constant dimension.

We remark that our constructions of LLTs and SLLTs apply to metric spaces, and not to general graphs. We show that there are graphs with constant hop-diameter for which *any* spanning tree has either huge hop-diameter or huge weight, and thus our results cannot be extended to general graphs.

1.2 Lower Bounds for Euclidean Spanners

While our upper bounds apply to all finite metric spaces, our lower bounds apply to an extremely basic metric space $M^* = \vartheta_n$. Specifically, this metric space is the 1-dimensional Euclidean space with n points $v_1, v_2, \ldots, v_n$ lying on the x-axis with coordinates $1, 2, \ldots, n$, respectively. The basic nature of ϑ_n strengthens our lower bounds, as they are applicable even for very limited classes of metric spaces. One particularly important application of our lower bounds is in the area of Euclidean Spanners. For a set $\mathcal{U}$ of n points in $\mathbb{R}^2$, and a parameter α, $\alpha \geq 1$, a subset $\mathcal{H}$ of the $\binom{n}{2}$ segments connecting pairs of points from $\mathcal{U}$ is called a (Euclidean) α-*spanner* for $\mathcal{U}$, if for every pair of points $u, v \in \mathcal{U}$, the distance between them in $\mathcal{H}$ is at most α times the Euclidean distance between them in the plane. Euclidean spanner is a very fundamental geometric construct with numerous applications in Computational Geometry [6, 7, 8] and Network Design [36, 43]. (See also the recent book of Narasimhan and Smid [45] for a detailed account on Euclidean spanners and their applications.)

A seminal paper that was a culmination of a long line of research on Euclidean spanners was published by Arya et al. [6] in STOC'95. One of the main results of this paper is a construction of $(1 + \epsilon)$-spanners with $O(n)$ edges that also have lightness and hop-diameter both bounded by $O(\log n)$. As an evidence of the optimality of this combination of parameters, Arya et al. cited a result by Lenhof et al. [42]. Lenhof et al. showed that any construction of Euclidean spanners that employs well-separated pair decompositions cannot achieve a better combination of weight and hop-diameter. However, the fundamental question of whether this combination of parameters can be improved by other means was left open in Arya et al. [6]. A partial answer to this intriguing problem was given by Agarwal et al. [2] in SODA'05. Specifically, it is shown in [2] that any Euclidean spanner of ϑ_n with lightness (respectively, hop-diameter) $O(\log n)$ must have hop-diameter (resp., lightness) at least $\Omega(\frac{\log n}{\log \log n})$. Consequently, Agarwal et al. showed that the upper bound of Arya et al. is optimal up to a factor of $O(\log \log n)$. The problem of closing this gap is stated explicitly as an open problem in the book of Narasimhan and Smid ([45], open problem 18 on p. 480). A simple corollary of our lower bounds is that the result of Arya et al. is tight up to constant factors even for *one-dimensional* spanners! In other words, we show that if the lightness (respectively, hop-diameter) is $O(\log n)$ then the hop-diameter (resp., lightness) is $\Omega(\log n)$, settling the open

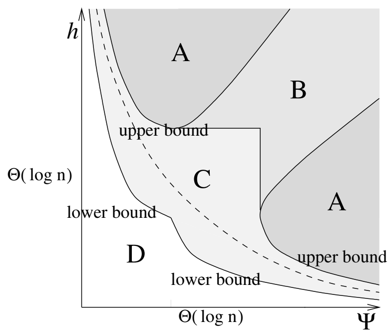

Figure 1. The dashed line separates two sets of pairs (Ψ, h). For a pair (Ψ, h) above the line, for any n-point metric space there exists a spanning tree with lightness at most Ψ and hop-radius at most h. For a pair (Ψ, h) below the line, there exist n-point metric spaces for which this property does not hold. The two areas A and the area B are contained in the former set, while the area D is contained in the latter one. The two areas A depict our upper bound constructions, and their extension by monotonicity is depicted by the area B. The area D represents our lower bounds. The area C represents the gap between our upper and lower bounds.

problem of [6, 2, 42, 45].

1.3 The h-hop MST problem

Our construction of LLTs can be also used to derive improved approximation algorithms for the *bounded diameter MST* (henceforth BDMST) and the *h-hop MST* problems for metric spaces. In these problems we are given a metric space M and a positive integer h. The objective in the BDMST problem is to minimize the weight of a spanning tree T of M with hop-diameter at most h. In the closely related h-hop MST problem the objective is to minimize the weight of a rooted spanning tree with hop-radius at most h. Both problems are among the classical and most well-studied NP-hard problems. (See the book of Garey and Johnson [34], page 206, and [40, 17, 5, 38, 41, 35].) Kortsarz and Peleg [40] and Charikar et al. [17] devised an $O(\log n)$-approximation algorithm for these problems when h is a constant, and an $O(n^{\epsilon})$-approximation algorithm for an arbitrarily small $\epsilon > 0$, which is applicable for all values of h. (These algorithms provide the same approximation guarantees for significantly more general problems.) Althaus et al. [5] devised a randomized $O(\log n)$-approximation algorithm for all values of h, but the algorithm of [5] requires a very high running time of $O(n^5 \cdot h)$. Kantor and Peleg [38] devised $2^{O(h)}$-approximation algorithms for these problems. Laue and Matijevic [41] presented a PTAS for 2-dimensional Euclidean metric spaces when h is a constant. These problems were also studied empirically. (See [35], and the references therein.)

Our constructions of LLTs give rise *directly* to improved approximation guarantees for these problems in the range $h = \Omega(\log n)$. Specifically, our approximation guarantee is $O(\Psi)$, where Ψ satisfies ($h = O(\Psi \cdot n^{1/\Psi})$ and $\Psi = O(\log n)$). In particular, for $h = \Theta(\log n)$ we obtain a deterministic $O(\log n)$-approximation algorithm. Though the approximation guarantee of our algorithm for $h = \Theta(\log n)$ is the same as that of Althaus et al. [5], its running time $O(n^2)$ is drastically better than the running time $O(n^5 \cdot \log n)$ of the algorithm of [5]. In addition, the algorithm of [5] is randomized, while our algorithm is deterministic. Moreover, for $\omega(\log n) = h = O(n)$ the approximation guarantee of our algorithm becomes *sublogarithmic*. Thus, in this range our algorithm improves the current state-of-the-art both in terms of the approximation guarantee and running time. Finally, when $h = n^{\epsilon}$, for some constant $\epsilon > 0$, our approximation guarantee becomes *constant*. See also Table 1.

To summarize, the problem of understanding the inherent tradeoff between different parameters of LLTs is a fundamental one in the investigation of spanning trees for metric spaces. In addition, this basic and combinatorially appealing problem has important applications in Computational Geometry and Approximation Algorithms. We believe that further investigation of LLTs will expose their additional applications, and connections to other areas.

1.4 Overview and Our Techniques

The most technically challenging part of our proof is the lower bound for the range of $h \geq \log n$. The proof of this lower bound consists of a number of components. First, we restrict our attention to binary trees. Second, we adapt a linear program for the minimum linear arrangement problem from the seminal paper of Even, Naor, Rao and

	$h = \Theta(\log n)$	$h = \log^c n$, $c > 1$	$h = 2^{c \cdot \sqrt{\log n}}$, $c > 0$	$h = n^{\epsilon}$, $\epsilon > 0$
Previous approximation	$O(\log n)$ [5] randomized	$O(\log n)$ [5] randomized	$O(\log n)$ [5] randomized	$O(\log n)$ [5] randomized
Our approximation	$\boldsymbol{O(\log n)}$ **deterministic**	$\boldsymbol{O((\log n)/(\log \log n))}$ **deterministic**	$\boldsymbol{O(\sqrt{\log n})}$ **deterministic**	$\boldsymbol{O(\epsilon^{-1})}$ **deterministic**
Previous runtime	$O(n^5 \cdot h)$	$O(n^5 \cdot h)$	$O(n^5 \cdot h)$	$O(n^5 \cdot h)$
Our runtime	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(n^2)}$	$\boldsymbol{O(n^2)}$

Table 1. A summary of previous and our results for the h-hop MST and BDMST problems for metric spaces, for different values of h, for $h = \Omega(\log n)$. Our results are indicated by bold font.

Schieber [31] on spreading metrics to our needs. Third, we analyze this linear program and show that the problem of providing a lower bound for its solution reduces to a clean combinatorial problem, and solve this problem. This enables us to establish the desired lower bounds for *binary trees*. Finally, we extend these lower bounds to general trees. This part of our proof demonstrates that techniques from the area of Low Distortion Embeddings (such as linear programs based on spreading metrics) may be useful for proving lower bounds in Computational Geometry. We anticipate that our approach will be applicable to other open problems in this area.

The proof of our lower bounds for $h < \log n$ combines some ideas from Agarwal et al. [2] with numerous new ideas. Specifically, Agarwal et al. reduce the problem from the general family of spanning subgraphs for ϑ_n to a certain restricted family of *stack graphs*. This reduction of [2] provides an elegant way for achieving somewhat weaker bounds, but it is inherently suboptimal. In our proof we tackle the general family of graphs directly. This more direct approach results in a more technically involved proof on the one hand, and in more accurate bounds on the other.

For upper bounds we essentially reduce the problem of constructing LLTs for general metric spaces to the same problem on ϑ_n.

1.5 Related work

SLTs have been extensively studied for the last twenty years [13, 37, 10, 9, 23, 25, 24, 39, 4]. However, all these constructions of SLTs may result in trees with very large hop-diameter, and the techniques used in these constructions appear to be inapplicable to the problem of constructing LLTs.

Euclidean spanners are also a subject of a recent extensive and intensive research (see [6, 27, 2, 7], and the references therein). However, the basic technique for constructing them relies heavily on the methodology of well-separated pair decomposition due to Callahan and Kosaraju [15]. This powerful methodology is, however, applicable only for Euclidean metric spaces of constant dimension, while our constructions apply to general metric spaces.

Tight lower bounds on the hop-diameter of Euclidean spanners with a given number of edges have been recently established by Chan and Gupta [16]. Specifically, it is shown in [16] that for any $\epsilon > 0$ there exists an n-point Euclidean metric space $M = M(n, \epsilon)$ for which any Euclidean $(1 + \epsilon)$-spanner with m edges has hop-diameter $\Omega(\alpha(m, n))$, where α is the functional inverse of the Ackermann's function. Moreover, the metric space M is 1-dimensional. (On the other hand, the space M is still not as restricted as ϑ_n.) However, this lower bound provides no indication whatsoever as to how *light* can be Euclidean spanners with low hop-diameter. In particular, the construction of Arya et al. [6] that provides matching upper bounds to the lower bounds of [16] produces spanners that may have very large weight.

In terms of the techniques, Chan and Gupta [16] start with proving their lower bounds for metric spaces induced by binary hierarchically-separated-trees (henceforth, HSTs), and then translate them into lower bounds for metric spaces induced by n points on the real line using known results. Their proof of the lower bound for HSTs is an extension of Yao's proof technique from [50]. As was discussed above, our lower bounds are achieved by completely different proof techniques that involve analyzing a linear program based on spreading metrics. In particular, our lower bounds are proved directly for ϑ_n.

The study of spanning trees of the 1-dimensional metric space ϑ_n is related to the well-studied problem of computing partial-sums. (See the papers of Yao [50], Chazelle and Rosenberg [20], Pătraşcu and Demaine [47], and the references therein.) For a discussion about the relationship between these two problems we refer to the introduction of [2]. We anticipate that our techniques will be useful in proving lower bounds for the problem of computing partial sums as well.

The spreading metric linear program for the minimum linear arrangement problem that we use for our lower bounds was studied in [31, 48]. There is an extensive literature on the minimum linear arrangement problem itself. (See [19, 33], and the references therein).

There is a large body of literature dealing with bicriteria approximation algorithms (see e.g., [44]). Despite the seeming similarity between the bicriteria setting and our setting, the bicriteria results are inapplicable to our problem. For an illustration, consider an algorithm from [44] that, given a metric space M and a positive integer h, constructs a spanning tree T of hop-diameter $O(\log n) \cdot h$ and weight $O(\log n) \cdot w(T_h^*)$, where T_h^* is the spanning tree of minimum weight among all spanning trees of hop-diameter at most h. If h is set to a constant, one obtains hop-diameter $O(\log n)$, but the weight bound of $O(\log n) \cdot w(T_h^*)$ may be much larger than $O(\log n) \cdot w(MST(M))$. On the other hand, if $h = h(n)$ tends to infinity as n grows, then the bound on hop-diameter will be greater than logarithmic. To summarize, the result of [44] does not imply the existence of an LLT with hop-diameter and lightness $O(\log n)$, which follows as an immediate corollary of our upper bounds.

1.6 The Structure of the Paper

Due to space limitations, in this extended abstract we focus on *binary* LLTs. Section 3 is devoted to lower bounds for binary trees, and Section 4 is devoted to upper bounds. At the end of Section 3 we outline the argument for deriving lower bounds for Euclidean spanners from our lower bounds for LLTs. Most proofs are omitted from this extended abstract.

2 Preliminaries

For a positive integer n, an *n-point metric space* $M = (V, dist_M)$ can be viewed as the complete graph $G = G(M) = (V, \binom{V}{2}, dist_M)$ in which for every pair of vertices $u, w \in V$, the weight of the edge $e = (u, w)$ between u and w in G is defined by $w(u, w) = dist_M(u, w)$. The distance function $dist_M$ is required to be non-negative, equal to zero when $u = w$, and to satisfy the triangle inequality ($dist_M(u, w) \leq dist_M(u, v) + dist_M(v, w)$, for every triple $u, w, v \in V$). A graph G' is called a *spanning subgraph* (respectively, *spanning tree*; *minimum spanning tree*) of M if it is a spanning subgraph (resp., spanning tree; minimum spanning tree) of $G(M)$.

For a weighted graph $G = (V, E, w)$, and a path P in G, its *(weighted) length* is defined as the sum of the weights of edges along P, and its *unweighted length* (or *hop-length*) is the number $|P|$ of edges (or *hops*) in P. For a pair of vertices $u, w \in V$, the *weighted* distance in G between u and w, denoted $dist_G(u, w)$, is the smallest weighted length of a path connecting between u and w in G. The *weighted* (respectively, *unweighted* or *hop-*) *diameter* of G is the maximum weighted (resp., unweighted) distance between a pair of vertices in V.

Whenever n can be understood from the context, we write ϑ as a shortcut for ϑ_n. We will use the notion *ϑ-tree* as an abbreviation for a "rooted spanning tree of ϑ".

Finally, for a pair of non-negative integers k and n, $k \leq n$, we denote the sets $\{k, k+1, \ldots, n\}$ and $\{1, 2, \ldots, n\}$ by $[k, n]$ and $[n]$, respectively.

3 Lower Bounds

In this section we devise lower bounds for binary ϑ-trees in the entire range of parameters.

We start with describing a relationship between the problem of constructing LLTs and the *minimum linear arrangement* (henceforth, *MINLA*) problem [31, 48].

The MINLA problem is defined as follows. Given an undirected graph $G = (V, E)$, we would like to find a permutation (called also a *linear arrangement*) of the nodes $\sigma : V \to \{1, \ldots, n = |V|\}$ that minimizes the cost of the linear arrangement σ, $LA(G, \sigma) = \sum_{(i,j) \in E} |\sigma(i) - \sigma(j)|$. The minimum linear arrangement of the graph G, denoted $MINLA(G)$, is defined as the minimum cost of a linear arrangement, i.e., $MINLA(G) = \min\{LA(G, \sigma) \mid \sigma \in S_n\}$, where S_n is the set of all permutations of $[n]$. Next, we study the family $B_n(h)$ of binary ϑ-trees of depth no greater than h, and show a lower bound on the value $MINLA(B_n(h)) = \min\{MINLA(G) \mid G \in B_n(h)\}$. It is easy to see that this lower bound will apply also to the minimum weight of a ϑ-tree from $B_n(h)$.

In a seminal work on spreading metrics, Even et al. [31] studied the following linear program relaxation $LP1$ for the MINLA problem. The variables of this linear program $\{\ell(e) \mid e \in E\}$ can be viewed as edge lengths. For a pair of vertices u and v, $dist_\ell(u, v)$ stands for the distance between u and v in the graph G equipped with length function $\ell(\cdot)$ on its edges.

$$
\begin{aligned}
LP1 \quad &: \min \sum_{e \in E} \ell(e) \text{ s.t. } \forall U \subseteq V, \forall v \in U : \\
&\sum_{u \in U} dist_\ell(v, u) \geq \frac{1}{4} \cdot (|U|^2 - 1), \\
&\text{and } \forall e \in E : \ell(e) \geq 0.
\end{aligned}
$$

It is well-known that the optimal solution of $LP1$ is a lower bound on $MINLA(G)$ [31, 48].

Next, we present a variant $LP2$ of $LP1$ which involves only a small subset of constraints that are used in $LP1$. Clearly, the optimal solution of $LP2$ is a lower bound on that of $LP1$. Consider a rooted tree $T = (T, rt)$ in $B_n(h)$.

For a vertex v in T, let T_v be the subtree of T rooted at v, and U_v be its vertex set. While in $LP1$ there is a constraint for each pair (U, v), $U \subseteq V$, $v \in U$, there are only the constraints that correspond to pairs (U_v, v) present in $LP2$ for T.

$$LP2 \quad : \min \sum_{e \in E} \ell(e) \text{ s.t. } \forall v \in V :$$
$$\sum_{u \in U_v} dist_\ell(v, u) \ \geq \ \frac{1}{4} \cdot (|U_v|^2 - 1),$$
$$\text{and } \forall e \in E \ : \ \ell(e) \ \geq \ 0.$$

For a vertex v in T, let $Ineq(v)$ be the inequality $\sum_{u \in U_v} dist_\ell(v, u) \geq \frac{1}{4}(|U_v|^2 - 1)$, and $Eq(v)$ be the corresponding equation $\sum_{u \in U_v} dist_\ell(v, u) = \frac{1}{4}(|U_v|^2 - 1)$. We replace all inequalities $Ineq(v)$ by equations $Eq(v)$ in $LP2$. By the following lemma, the value of $LP2$ remains intact.

Lemma 3.1 *For a binary tree T, in any optimal solution to $LP2$ all inequalities $\{Ineq(v) \mid v \in V\}$ hold as equalities.*

Intuitively, if one of the inequalities is strict for some solution to $LP2$, it allows some slack, and this slack can be used to decrease the value of the objective function. However, it is not a-priori clear that one can indeed exploit this slack without violating all the other constraints. Using the special tree structure of the constraints, we prove that this is indeed the case for $LP2$.

Consider a subtree T_v rooted at an inner vertex v. Without loss of generality, v has a left child v_L, and possibly a right child v_R, each being the root of the corresponding subtrees T_{v_L} and T_{v_R}, respectively.

Lemma 3.2 *For an optimal solution for $LP2$, $\ell(v, v_L) + \ell(v, v_R) \ > \ \frac{1}{2} \cdot (\min\{|U_{v_L}|, |U_{v_R}|\} + 1)$.*

Let $I = I(T)$ denote the set of inner vertices of T. By Lemma 3.2, for any optimal assignment for the values $\{\ell(e) \mid e \in E(T)\}$ of the linear program $LP2$ holds:

$$\sum_{e \in E(T)} \ell(e) \quad = \quad \sum_{v \in I(T)} (\ell(v, v_L) + \ell(v, v_R))$$
$$\geq \quad \frac{1}{2} \cdot \sum_{v \in I(T)} (\min\{|U_{v_L}|, |U_{v_R}|\} + 1).$$

We call the right-hand side expression the *cost* of T, and denote it $Cost(T)$. In the sequel we provide a lower bound for $\min\{Cost(T) \mid T \in B_n(h)\}$, which would apply to $MINLA(B_n(h))$ as well.

For a tree $T = (T, rt)$ in $B_n(h)$, we call the subtree rooted at the left (respectively, right) child of rt the *left subtree* (resp., *right subtree*) of T, and denote it by $T.left$ (resp., $T.right$). Also, let $|T|$ denote the *size* of the tree T, that is, the number of vertices in T. Consider the following cost function on binary trees: $Cost'(T) = Cost'(T.left) + Cost'(T.right) + \min\{|T.left|, |T.right|\}$. It is easy to verify that $Cost'(T)$ can be equivalently expressed as $Cost'(T) = \sum_{v \in I(T)} \min\{|U_{v_L}|, |U_{v_R}|\}$. By definition, for any binary tree T, $2 \cdot Cost(T) \geq Cost'(T)$. We will henceforth focus on proving a lower bound for $Cost'(T)$, and use the notion "cost" to refer to the function $Cost'$.

Fix a pair of positive integers n and h, $n - 1 \geq h$. A rooted binary tree on n vertices of depth at most h will be called an *(n,h)-tree*. Let $R(n, h)$ denote the minimum cost taken over all (n, h)-trees. Next, we outline the proof of the following theorem, which establishes lower bounds on $R(n, h)$, for all $h \geq \log n$.

Theorem 3.3 *1. If $\log n \leq h \leq 2\lfloor \log n \rfloor$, then $R(n, h) \geq \frac{2}{3} \cdot n \cdot \lfloor \frac{1}{8} \log n \rfloor$.*

2. For $2\lfloor \log n \rfloor < h \leq n - 1$, let $f(h)$ be the minimum integer such that $\binom{h+1}{f(h)} > \frac{2}{3} \cdot n$. Then $R(n, h) > \frac{2}{3} \cdot n \cdot (f(h) - 2)$.

Remark: Note that for $h > 2\lfloor \log n \rfloor$, $\binom{h+1}{\log n} > \frac{2}{3} \cdot n$, and thus $f(h)$ is well-defined in this range.
As was argued above, Theorem 3.3 implies the respective lower bounds for the weight of binary trees of any given depth.

Proof: Let n and h be non-negative integers. Given a binary rooted tree T, we restructure it *without changing its cost and depth*, so that for each vertex v in T, the size of its right subtree $v.right$ would not exceed the size of its left subtree $v.left$. Specifically, if in the original tree T it holds that $|v.left| \geq |v.right|$, then no adjustment occurs in v. However, if $|v.left| < |v.right|$, then the restructuring process exchanges between the left and right subtrees of v. We refer to this restructuring procedure as the *right-adjustment* of T, and denote the resulting binary tree by $\tilde{T}$. By construction, T and its *right-adjusted* tree $\tilde{T}$ have the same cost and depth. Thus we henceforth restrict our attention to right-adjusted trees. By definition, in a right-adjusted tree $\tilde{T}$, for any $v \in V(\tilde{T})$, it holds that $|v.right| \leq |v.left|$, and consequently, $Cost(\tilde{T}) = \sum_{v \in V(\tilde{T})} |v.right|$.

A set of n binary words with at most h bits each will be called an *(n,h)-vocabulary*. Next, we define an injection $\mathcal{S}$ from the set of (n, h)-trees to the set of (n, h)-vocabularies. For a vertex v in a binary tree T, denote by $P_v = (rt = v_0, v_1, \ldots, v_k = v)$ the path from rt to v in T, and define $B_v = b_1 b_2 \ldots b_k$ to be its corresponding binary word, where for $i \in \{1, 2, \ldots, k\}$, $b_i = 0$ if v_i is the left child of v_{i-1}, and $b_i = 1$ otherwise. Given an (n, h)-tree T, let $\mathcal{S}(T)$ be the (n, h)-vocabulary that consists of the $|T|$ binary words that correspond to the set of all root-to-vertex

paths in T, namely, $\mathcal{S}(T) = \{B_v \mid v \in V(T)\}$.
For a binary word α, denote its *Hamming weight* (the number of 1's in it) by $H(\alpha)$. For a set S of binary words, define its total Hamming weight, henceforth *Hamming cost*, $HCost(S)$, as the sum of Hamming weights of all words in S, namely, $HCost(S) = \sum_{B \in S} H(B)$. Finally, denote the minimum Hamming cost of a set of n distinct binary words with at most h bits each by $H(n,h)$. Observe that the function $H(n,h)$ is monotone non-increasing with h.

The next lemma shows that it is sufficient to prove the desired lower bound for $H(n,h)$. Its proof follows from the observation that for every right-adjusted tree $\tilde{T}$, $Cost(\tilde{T}) = HCost(\mathcal{S}(\tilde{T}))$.

Lemma 3.4 *For all positive integers n and h such that $h \le n-1$, $H(n,h) \le R(n,h)$.*

Consider a set $\mathcal{S}^* = \mathcal{S}^*(n,h)$ realizing $H(n,h)$, that is, a set that satisfies $HCost(\mathcal{S}^*) = H(n,h)$. For a non-negative integer i, $i \le h$, let $\mathcal{S}(h,i)$ be the set of all distinct binary words with at most h bits each, so that each word of which contains precisely i 1's. Observe that $|\mathcal{S}(h,i)| = \sum_{k=i}^{h} \binom{k}{i} = \binom{h+1}{i+1}$. To contain the minimum total number of 1's, the set $\mathcal{S}^*$ needs to contain all binary words with no 1's, all binary words that contain just a single 1, etc. In other words, there exists an integer $r = r(h)$ for which $\bigcup_{i=0}^{r} \mathcal{S}(h,i) \subset \mathcal{S}^* \subseteq \bigcup_{i=0}^{r+1} \mathcal{S}(h,i)$. Note that for a pair of distinct indices i and j, $0 \le i,j \le h$, the sets $\mathcal{S}(h,i)$ and $\mathcal{S}(h,j)$ are disjoint. Since $|\mathcal{S}^*| = n$, it holds that

$$\sum_{i=0}^{r} \binom{h+1}{i+1} \;<\; n \;\le\; \sum_{i=0}^{r+1} \binom{h+1}{i+1}.$$

Since for every non-negative integer i, $i \le h$, each word in $\mathcal{S}(h,i)$ contains precisely i 1's, it follows that $HCost(\mathcal{S}(h,i)) = i \cdot \binom{h+1}{i+1}$. Let

$$N \;=\; n - \sum_{i=0}^{r} \binom{h+1}{i+1} > 0$$

denote the number of words with Hamming weight $r+1$ in $\mathcal{S}^*$. Hence

$$\begin{aligned} H(n,h) &= HCost(\mathcal{S}^*) \\ &= \sum_{i=0}^{r} HCost(\mathcal{S}(h,i)) + N \cdot (r+1) \\ &= \sum_{i=0}^{r} i \cdot \binom{h+1}{i+1} + N \cdot (r+1). \qquad (1) \end{aligned}$$

A direct calculation shows that for $h \ge \lfloor 2\log n \rfloor$, $r \le \left\lfloor \frac{h+1}{4} \right\rfloor - 1$.

The next lemma establishes lower bounds on $H(n,h)$ for $h \ge \lfloor 2\log n \rfloor$. Since $H(n,h)$ is monotone non-increasing with h, it follows that the lower bound for $h = 2\lfloor \log n \rfloor$ applies for all smaller values of h.

Lemma 3.5 *(1) $H(n, 2\lfloor \log n \rfloor) \ge \frac{2}{3} \cdot n \cdot \lfloor \frac{1}{8} \log n \rfloor$. (2) For any $2\lfloor \log n \rfloor < h \le n-1$, $H(n,h) > \frac{2}{3} \cdot n \cdot (f(h) - 2)$.*

To prove this lemma we start with showing that $H(n,h) \ge \frac{2}{3} \cdot n \cdot r$. This inequality follows from (1) by a rather simple computation. Then the argument splits into two cases, depending on the value of h. (The cases are $h = \lfloor 2\log n \rfloor$ and $h > \lfloor 2\log n \rfloor$.) In both cases the proof proceeds by establishing a lower bound on r in terms of n and h. This lower bound, in turn, implies the lemma.

Lemmas 3.4 and 3.5 imply Theorem 3.3. ∎

The next theorem follows from Theorem 3.3.

Theorem 3.6 *For sufficiently large integers n and h, $h \ge \log n$, the minimum weight of a* binary *ϑ-tree that has depth at most h is at least $\Omega(\Psi \cdot n)$, for some Ψ satisfying $h = \Omega(\Psi \cdot n^{1/\Psi})$.*

We prove that this lower bound also applies to general trees. A particular case of this lower bound which is of special interest appears in the following corollary.

Corollary 3.7 *For a sufficiently large integer n, any spanning subgraph of ϑ that has hop-diameter at most $O(\log n)$ has lightness at least $\Omega(\log n)$, and vice versa.*

Corollary 3.7 implies that no construction that provides Euclidean spanner with hop-diameter $O(\log n)$ and lightness $o(\log n)$, or vice versa, is possible. This settles the open problem of [6, 2, 42, 45].

4 Upper Bounds

In this section we devise upper bounds for binary LLTs. Our upper bounds are tight up to constant factors in the entire range of parameters.

Consider a general n-point metric space M. Let T^* be an MST for M, and D be an in-order traversal of T^*, starting at an arbitrary vertex u. For every vertex x, remove from D all occurrences of x except for the first one. It is well-known ([26], ch. 36) that this way we obtain a Hamiltonian path $L = L(T)$ of M of total weight $w(L) = \sum_{e \in L} w(e) \le 2 \cdot w(MST(M))$. Let $(u_1, u_2, \ldots, u_n) = L$ be the order in which the points of M appear in L. Consider an edge $e' = (u_i, u_j)$ connecting two arbitrary points in M, and an edge $e = (u_q, u_{q+1}) \in E(L)$, $q \in [n-1]$. The edge e' is said to *load e (with respect to L)* if $i \le q < q+1 \le j$. When L is clear from the context, we write that e' *loads* e.

For a spanning tree T of M, the number of edges $e' \in E(T)$ that load an edge e of $E(L)$ is called the *load of e by T* and it is denoted $\xi(e) = \xi_T(e)$. The *load of the tree*

T, $\xi(T)$, is the maximum load of an edge $e \in E(L)$ by T, i.e., $\xi(T) = \max\{\xi_T(e) \mid e \in E(L)\}$. Observe that $w(T) \leq \sum_{e \in L(T)} \xi_T(e) \cdot w(e) \leq \xi(T) \cdot w(L)$, and so, $\xi(T) \geq \frac{w(T)}{w(L)} \geq \frac{1}{2} \cdot \Psi(T)$.

In the sequel we provide an upper bound for the load $\xi(T)$ of a tree T, which yields the same upper bound for the lightness $\Psi(T)$ of T, up to a factor of 2. It is not difficult to see that without loss of generality one can assume that u_i is located in the point i on the x-axis.

Consequently, the problem of providing upper bounds for general metric spaces reduces to the problem of providing upper bounds for ϑ. Next, we devise a construction of LLTs for ϑ. This construction exhibits tight up to constant factors tradeoff between load and depth, when the tree depth is at least logarithmic in the number of vertices. In addition, the constructed trees are *binary*.

We start by defining a certain composition of binary trees. Let n' and n'' be two positive integers, $n = n' + n''$. Let ϑ', ϑ'', and ϑ be the n'-, n''-, and n-point metric spaces $\vartheta_{n'}$, $\vartheta_{n''}$, and ϑ_n, respectively. Also, let $\{u'_1, u'_2, \ldots, u'_{n'}\}$, $\{u''_1, u''_2, \ldots, u''_{n''}\}$, and $\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_n\}$ denote the set of points of ϑ', ϑ'', and ϑ, respectively. Consider spanning trees T' and T'' of ϑ' and ϑ'', respectively. Let $u' = u'_i$ be a vertex of T'. Consider a tree $\tilde{T}$ that spans the vertex set $\{\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{n'+n''}\}$ formed out of the trees T' and T'' in the following way. The root rt'' of T'' is added as a right child of u' in $\tilde{T}$. The vertices $u'_1, u'_2, \ldots, u'_i = u'$ of T' retain their indices, and are translated into vertices $\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_i$ in $\tilde{T}$, respectively. The vertices $u''_1, u''_2, \ldots, u''_{n''}$ of T'' get the index i of u'_i, added to their indices, and are translated into vertices $\tilde{u}_{i+1}, \tilde{u}_{i+2}, \ldots, \tilde{u}_{i+n''}$ in $\tilde{T}$, respectively. Finally, the vertices $u'_{i+1}, u'_{i+2}, \ldots, u'_{n'}$ of T' get the number n'' of vertices of T'' added to their indices, and are translated into vertices $\tilde{u}_{i+1+n''}, \tilde{u}_{i+2+n''}, \ldots, \tilde{u}_{n'+n''}$ in $\tilde{T}$, respectively. We say that the tree $\tilde{T}$ is composed *by adding T'' as a right subtree to the vertex u' in T'*. (See Figure 2 for an illustration.)

Consider a family of binary ϑ-trees $T(\xi, h)$ with $n = N(\xi, h)$ vertices, load ξ and depth h, $h \geq \xi - 1$, $\xi \geq 1$. These trees are all rooted at the point 1. For $\xi = 1$, and $h = 0$, the tree $T(1, 0)$ is a singleton vertex, and so $N(1, 0) = 1$. For convenience, we define the load of $T(1, 0)$ to be 1. For $\xi = 1$ and $h \geq 1$, the tree $T(1, h)$ is the path $P_{h+1} = (u_1, u_2, \ldots, u_{h+1})$. Clearly, the depth of $T(1, h)$ is h, its size $N(1, h)$ is $h + 1$, and its load is 1.

For $\xi \geq 2$ and $h \geq \xi - 1$, the tree $T(\xi, h)$ is constructed as follows. Let $T' = P_{h+1}$ be the path $(u_1, u_2, \ldots, u_{h+1})$, and for each index i, $i \in [h]$, define for technical convenience $u'_i = u_i$. For each index i, $i \in [h]$, let T''_i be the tree $T(\xi''_i, h''_i)$, with $\xi''_i = \min\{\xi - 1, h - i + 1\}$, $h''_i = h - i$. Observe that for each $i \in [h]$, $h''_i \geq \xi''_i - 1$, and thus the tree T''_i is well-defined. For every $i \in [h]$, we add the tree T''_i as a right subtree of u'_i in T'.

Lemma 4.1 *The depth of the resulting tree $T(\xi, h)$ with respect to the vertex u'_1 is h, and its load is ξ.*

Finally, we analyze the number of vertices $N(\xi, h)$ in the tree $T(\xi, h)$. By construction, $N(\xi, h) = h + 1 + \sum_{i=1}^{h} N(\min\{\xi - 1, h - i + 1\}, h - i)$.

Lemma 4.2 *For $h \geq \xi - 1$, $N(\xi, h) \geq \binom{h}{\xi}$.*

Proof: The proof is by induction on ξ.
Base: For $\xi = 1$, $N(1, h) = h + 1 \geq \binom{h}{1}$, as required.
Step: For $\xi \geq 2$, and $i \leq h - \xi + 1$, $\min\{\xi - 1, h - i + 1\} = \xi - 1$. Hence

$$\begin{aligned} N(\xi, h) &= h + 1 + \\ &\quad \sum_{i=1}^{h} N(\min\{\xi - 1, h - i + 1\}, h - i) \\ &\geq \sum_{i=1}^{h-\xi+1} N(\min\{\xi - 1, h - i + 1\}, h - i) \\ &= \sum_{i=1}^{h-\xi+1} N(\xi - 1, h - i). \end{aligned}$$

Observe that for each index i, $i \in [h - \xi + 1]$, $h - i \geq \xi - 1$. Hence, by the induction hypothesis, the latter sum is at least $\sum_{i=1}^{h-\xi+1} \binom{h-i}{\xi-1} = \binom{h}{\xi}$. ∎

The following theorem summarizes the properties of the trees $T(\xi, h)$.

Theorem 4.3 *For a sufficiently large n, and h, $h \geq 2 \log n$, there exists a binary ϑ_n-tree that has depth at most h and load at most ξ, where ξ satisfies ($h = O(n^{1/\xi} \cdot \xi)$ and $\xi = O(\log n)$).*

The next corollary employs Theorem 4.3 to deduce an analogous result for general metric spaces.

Corollary 4.4 *For sufficiently large n and h, $h \geq 2 \log n$, there exists a binary spanning tree of M that has depth at most h and lightness at most $2 \cdot \Psi$, where Ψ satisfies ($h = O(n^{1/\Psi} \cdot \Psi)$ and $\Psi = O(\log n)$). Moreover, this binary tree can be constructed in time $O(n^2)$.*

We remark that if M is a Euclidean 2-dimensional metric space, the running time can be further improved to $O(n \cdot \log n)$. This is because the running time of this construction is dominated by the running time of the subroutine for constructing MST, and an MST of a Euclidean 2-dimensional metric space can be constructed in $O(n \cdot \log n)$ time [30]. By the same considerations, for Euclidean 3-dimensional spaces our algorithm can be implemented in a

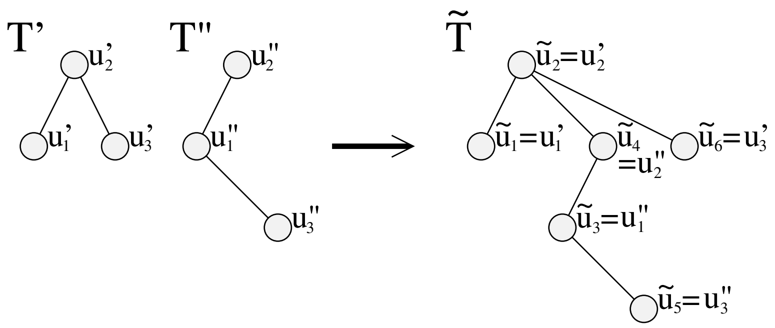

Figure 2. The trees T' and T'' depicted on the left are spanning trees of $\vartheta' = \vartheta'' = \vartheta_3$. The tree $\tilde{T}$ depicted on the right is a spanning tree of $\vartheta = \vartheta_6$. It is composed by adding T'' as a right subtree to the vertex u'_2 in T'.

randomized time of $O(n \cdot \log^{4/3} n)$, and more generally, for dimension $d = 3, 4, \ldots$ it can be implemented in deterministic time $O(n^{2-\frac{2}{\lceil d/2 \rceil + 1}+\epsilon})$, for an arbitrarily small $\epsilon > 0$ (cf. [30], page 5). Even better time bounds can be shown if one uses a $(1+\epsilon)$-approximation MST instead of the exact MST for Euclidean metric spaces (cf. [30], page 6). Specifically, this technique enables us to obtain running time of $O(n \cdot \log n)$ for any constant dimension d.

Corollary 4.4 provides tight upper bounds for the range of $h \geq 2 \log n$. A full balanced binary tree does the job for the range $\log n \leq h < 2 \log n$. Tight upper bounds for the range $h < \log n$ are achieved by full balanced d-regular trees with $d = \lceil n^{1/h} \rceil$. See the full version [28] for the details.

5 Acknowledgements

The second-named author thanks Michael Segal and Hanan Shpungin for approaching him with a problem in the area of wireless networks that is related to the problem of constructing LLTs. Correspondence with them triggered this research. Also, we are grateful to Guy Kortsarz for his helpful comments on a preliminary draft of this paper.

References

[1] I. Abraham and D. Malkhi. Compact routing on euclidian metrics. In *Proc. of 23rd Annual Symp. on Principles of Distributed Computing*, pages 141–149, 2004.

[2] P. K. Agarwal, Y. Wang, and P. Yin. Lower bound for sparse Euclidean spanners. In *Proc. of 16th SODA*, pages 670–671, 2005.

[3] N. Alon, R. M. Karp, D. Peleg, and D. B. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[4] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger. Prim-dijkstra tradeoffs for improved performance-driven routing tree design. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 14(7):890–896, 1995.

[5] E. Althaus, S. Funke, S. Har-Peled, J. Könemann, E. A. Ramos, and M. Skutella. Approximating k-hop minimum-spanning trees. *Oper. Res. Lett.*, 33(2):115–120, 2005.

[6] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.

[7] S. Arya, D. M. Mount, and M. H. M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. of 35th FOCS*, pages 703–712, 1994.

[8] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. *Algorithmica*, 17(1):33–54, 1997.

[9] B. Awerbuch and D. P. A. Baratz. Efficient broadcast and light-weight spanners. *Manuscript*, 1991.

[10] B. Awerbuch, A. Baratz, and D. Peleg. Cost-sensitive analysis of communication protocols. In *Proc. of 9th PODC*, pages 177–187, 1990.

[11] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proc. of 37th FOCS*, pages 184–193, 1996.

[12] Y. Bartal. On approximating arbitrary metrices by tree metrics. In *Proc. of 30th STOC*, pages 161–168, 1998.

[13] K. Bharath-Kumar and J. M. Jaffe. Routing to multiple destinations in computer networks. *IEEE Trans. on Commun.*, COM-31:343–351, 1983.

[14] P. Bose, J. Gudmundsson, and M. H. M. Smid. Constructing plane spanners of bounded degree and low weight. *Algorithmica*, 42(3-4):249–264, 2005.

[15] P. B. Callahan and S. R. Kosaraju. A decomposition of multi-dimensional point-sets with applications to k-nearest-neighbors and n-body potential fields. In *Proc. of 24th STOC*, pages 546–556, 1992.

[16] H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006.

[17] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *J. Algorithms*, 33(1):73–91, 1999.

[18] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proc. of 39th FOCS*, pages 379–388, 1998.

[19] M. Charikar, M. T. Hajiaghayi, H. J. Karloff, and S. Rao. l_2^2 spreading metrics for vertex ordering problems. In *Proc. of 17th SODA*, pages 1018–1027, 2006.

[20] B. Chazelle and B. Rosenberg. The complexity of computing partial sums off-line. *Int. J. Comput. Geom. Appl.*, 1:33–45, 1991.

[21] E. Cohen. Fast algrotihms for constructing t-spanners and paths with stretch t. In *Proc. of 34th FOCS*, pages 648–658, 1993.

[22] E. Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. In *Proc. of 26th STOC*, pages 16–26, 1994.

[23] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Performance-driven global routing for cell based ics. In *Proc. of IEEE International Conference on Computer Design: VLSI in Computer & Processors (ICCD)*, pages 170–173, 1991.

[24] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good algorithms for performance-driven global routing. In *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2240–2243, 1992.

[25] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good performance-driven global routing. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(6):739–752, 1992.

[26] T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition*. McGraw-Hill Book Company, Boston, MA, 2001.

[27] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *Proc. of 10th SOCG*, pages 132–139, 1994.

[28] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. *Manuscript, arXiv:0801.3581v1, available via http://arxiv.org/abs/0801.3581*, 2008.

[29] M. Elkin, Y. Emek, D. Spielman, and S. Teng. Lower stretch spanning trees. In *Proc. of 37th STOC*, pages 494–503, 2005.

[30] D. Eppstein. Spanning trees and spanners. *Technical report, Dept. of Information and Computer-Science, University of California, Irvine*, (96-16), 1996.

[31] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. In *Proc. of 36th FOCS*, pages 62–71, 1995.

[32] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. of 35th STOC*, pages 448–455, 2003.

[33] U. Feige and J. R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Inf. Process. Lett.*, 101(1):26–29, 2007.

[34] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[35] L. Gouveia and T. L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, 41(3):159–173, 2003.

[36] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. In *Proc. of 19th PODC*, pages 41–50, 2000.

[37] J. M. Jaffe. Distributed multi-destination routing: the constraints of local information. *SIAM J. Comput.*, 14:875–888, 1985.

[38] E. Kantor and D. Peleg. Approximate hierarchical facility location and applications to the shallow Steiner tree and range assignment problems. In *Proc. of 6th CIAC*, pages 211–222, 2006.

[39] S. Khuller, B. Raghavachari, and N. E. Young. Approximating the minimum equivalent diagraph. In *Proc. of 5th SODA*, pages 177–186, 1994.

[40] G. Kortsarz and D. Peleg. Approximating the weight of shallow Steiner trees. *Discrete Applied Mathematics*, 93(2-3):265–285, 1999.

[41] S. Laue and D. Matijevic. Approximating k-hop minimum spanning trees in Euclidean metrics. In *Proc. of 19th CCCG*, pages 117–120, 2007.

[42] H. P. Lenhof, J. S. Salowe, and D. E. Wrege. New methods to mix shortest-path and minimum spanning trees. manuscript, 1994.

[43] Y. Mansour and D. Peleg. An approximation algorithm for min-cost network design. *DIMACS Series in Discr. Math and TCS*, 53:97–106, 2000.

[44] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. H. III. Bicriteria network design problems. *J. Algorithms*, 28(1):142–171, 1998.

[45] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

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

[47] M. Pătraşcu and E. D. Demaine. Tight bounds for the partial-sums problem. In *Proc. of 15th SODA*, pages 20–29, 2004.

[48] S. Rao and A. W. Richa. New approximation techniques for some ordering problems. In *Proc. of 9th SODA*, pages 211–218, 1998.

[49] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi. Spanning trees short or small. In *Proc. of 5th SODA*, pages 546–555, 1994.

[50] A. C. Yao. Space-time tradeoff for answering range queries. In *Proc. of 14th STOC*, pages 128–136, 1982.