Tree Embeddings for Hop-Constrained Network Design[*]

Bernhard Haeupler
haeupler@cs.cmu.edu
Carnegie Mellon University &
ETH Zürich
USA & Switzerland

D. Ellis Hershkowitz
dhershko@cs.cmu.edu
Carnegie Mellon University
USA

Goran Zuzic
goran.zuzic@inf.ethz.ch
ETH Zürich
Switzerland

ABSTRACT

Network design problems aim to compute low-cost structures such as routes, trees and subgraphs. Often, it is natural and desirable to require that these structures have small hop length or hop diameter. Unfortunately, optimization problems with hop constraints are much harder and less well understood than their hop-unconstrained counterparts. A significant algorithmic barrier in this setting is the fact that hop-constrained distances in graphs are very far from being a metric.

We show that, nonetheless, hop-constrained distances can be approximated by distributions over "partial tree metrics." We build this result into a powerful and versatile algorithmic tool which, similarly to classic probabilistic tree embeddings, reduces hop-constrained problems in general graphs to hop-unconstrained problems on trees. We then use this tool to give the first poly-logarithmic bicriteria approximations for the hop-constrained variants of many classic network design problems. These include Steiner forest, group Steiner tree, group Steiner forest, buy-at-bulk network design as well as online and oblivious versions of many of these problems.

CCS CONCEPTS

• **Theory of computation** → *Routing and network design problems*; *Sparsification and spanners*; *Online algorithms*.

KEYWORDS

tree embeddings, hop-constrained distances, approximation algorithms, hop-constrained network design

ACM Reference Format:
Bernhard Haeupler, D. Ellis Hershkowitz, and Goran Zuzic. 2021. Tree Embeddings for Hop-Constrained Network Design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451053

[*]Supported in part by NSF grants CCF-1527110, CCF-1618280, CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808, a Sloan Research Fellowship, funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (ERC grant agreement 949272), Swiss National Foundation (project grant 200021-184735) and the Air Force Office of Scientific Research under award number FA9550-20-1-0080.

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06.
https://doi.org/10.1145/3406325.3451053

1 INTRODUCTION

The field of network design studies how to efficiently construct and use large networks. Over the past several decades researchers have paid particular attention to the construction of low-cost computer and transportation networks that enable specified communication and delivery demands.

Formally, these problems require computation of low-cost structures in graphs, such as paths, trees or subgraphs, that satisfy specified connectivity requirements. For example, there has been extensive work on how, given a weighted graph $G = (V, E, w)$ with n nodes, one can compute a subgraph $H \subseteq G$ of minimum weight that connects: all vertices (*minimum spanning tree (MST)*); all vertices in an input $S \subseteq V$ (*Steiner tree*); at least k nodes (*k-MST*); at least k terminals from an input $S \subseteq V$ (*k-Steiner tree*); at least one vertex from each set S_i for a given collection of vertex sets $S_1, \ldots, S_k \subset V$ (*group Steiner tree*); $s_i \in V$ to $t_i \in V$ for every pair in $\{(s_i, t_i)\}_i$ (*Steiner forest*); and some vertex in $S_i \subseteq V$ to some vertex in $T_i \subseteq V$ for every pair in $\{(S_i, T_i)\}_i$ (*group Steiner forest a.k.a. generalized connectivity*). To model the uncertainty and dynamic nature of networks, these problems are often generalized to their *online* variants where the demands to be connected are revealed over discrete time steps. An even stronger model of uncertainty is the *oblivious* setting where an algorithm must specify how it will satisfy each possible demand before it even knows the demands; demands are then revealed and the algorithm buys its pre-specified solution.

However, connectivity alone is often not sufficient for fast and reliable networks. Indeed, we often also desire that our networks be *hop-constrained*; namely we desire that demands are not just appropriately connected but connected with a path consisting of a low number of edges (a.k.a. hops). By reducing the number of traversed edges, hop constraints facilitate fast communication [6, 25]. Furthermore, low-hop networks tend to also be more reliable: if a transmission over an edge fails with some probability, the greater the number of hops between the source and destination, the greater the probability that this transmission fails [60, 63].

Unfortunately, adding hop constraints to network design problems makes them significantly harder. MST is solvable in polynomial time but MST with hop constraints is known to admit no $o(\log n)$ poly-time approximation algorithm [14]. Similarly, Steiner forest has a constant approximation [5] but hop-constrained Steiner forest has no poly-time $o(2^{\log^{1-\varepsilon} n})$-approximation for any constant $\varepsilon > 0$ [29].[1] Indeed, although there has been extensive work on approximation algorithms for simple connectivity problems like spanning tree and Steiner tree with hop constraints [9, 42, 43, 46, 47, 49, 52, 59], nothing is known regarding algorithms for

[1]Under standard complexity assumptions.

many well-studied generalizations of these problems with hop constraints. For instance, no non-trivial approximation algorithms are known for Steiner forest, group Steiner tree, group Steiner forest or online Steiner tree with hop constraints.

By allowing an algorithm to "pretend" that the input graph is a tree, probabilistic tree embeddings have had enormous success as the foundation of many poly-log approximation algorithms for network design; thus, we might naturally expect them to be useful for hop-constrained network design. Specifically, a long and celebrated line of work [8, 15, 33, 44] culminated in the embedding of Fakcharoenphol, Rao and Talwar [33]—henceforth "FRT"—which showed that any metric can be $O(\log n)$-approximated by a distribution $\mathcal{D}$ over trees.[2] Consequently, a typical template for many network design algorithms is to (1) embed the metric induced by weighted graph G into a $T \sim \mathcal{D}$; (2) solve the input problem on T (which is typically much easier than the problem on G) and; (3) project the solution on T back into G. For example, such a template gives poly-log approximations for group Steiner tree and group Steiner forest [35, 56]. In the h-hop-constrained setting for some $h \geq 1$, the natural notion of distance to consider between vertices u and v is the h-hop-constrained distance—the length of the shortest path between u and v according to w with at most h hops. Thus, to use tree embeddings for hop-constrained network design we must first understand how to approximate these distances with trees.

1.1 Our Contributions

In this paper we initiate the study of metric approximations for hop-constrained distances and their use in algorithms for hop-constrained network design. Broadly, our results fall into four categories.

1.1.1 Impossibility of Approximating Hop-Constrained Distances with Metrics. We begin by observing that hop-constrained distances are inapproximable by metrics (Section 4.1). Not only are hop-constrained distances not a metric (since they do not satisfy the triangle inequality) but given a hop constraint there are weighted graphs where any metric that approximates hop-constrained distances does so with an $\Omega(L)$ multiplicative error where L is the aspect ratio (Lemma 4.3). This lower bound is matched by a trivial upper bound (Lemma 4.4). Since the expected distance between two nodes in a distribution over metrics is itself a metric, our impossibility result also rules out approximating hop-constrained distances with distributions over metrics as in FRT. This observation is proved by careful analysis of a simple example: a path graph.

1.1.2 Approximating Hop-Constrained Distances with Partial Tree Metrics. Despite these apparent roadblocks, we show that—somewhat surprisingly—it is indeed possible to to approximate hop-constrained distances with trees (Sections 4.2, 4.3). We show that a distribution over "partial tree metrics" can approximate hop-constrained distances with an expected distance stretch of $O(\log n \log \log n)$ and a worst-case distance stretch of $O(\log^2 n)$ with an $O(\log^2 n)$ relaxation in the hop constraint (Theorem 4.8). This result differs from FRT in two notable ways: (1) our partial tree metrics are partial in the sense that they contain only a constant fraction of nodes from the input graph—indeed, this is what allows us to overcome the impossibility of approximating hop-constrained distances with metrics; (2) our result provides a worst-case guarantee, unlike FRT which only gives a guarantee in expectation. We show this result by first proving a decomposition lemma (Lemma 4.11), which applies padded decompositions to a "mixture metric" that combines hops and (unconstrained) distances. We then recursively apply this decomposition, using different combinations of hops and distances in our recursive calls.

1.1.3 h-Hop Partial Tree Embeddings. We next build embeddings for hop-constrained network design from our metric approximations (Section 5). Specifically, we show that one can construct a distribution over "h-hop partial tree embeddings" of hop-constrained distances with expected distance stretch $O(\log n \log \log n)$ and a worst-case distance stretch $O(\log^2 n)$ with an $O(\log^3 n)$ relaxation in the hop constraint (Theorem 5.6). Further, we show that these embeddings can be used for hop-constrained network design as in the above template for network design that uses FRT. Notably, our embeddings reduce many hop-constrained network design problems to their *non-hop-constrained* versions on trees. Since our embeddings, like our partial tree metrics, are also partial, we build on these embeddings by constructing "h-hop copy tree embeddings," which represent many draws from our distribution over partial tree embeddings as a single tree.[3] Like our tree metrics and unlike FRT, our tree embeddings are partial and give worst-case guarantees. Moreover, our embeddings follow almost immediately from our metric approximations. However, a notable difference between our embeddings and those of FRT is that demonstrating that they can be used for hop-constrained network design requires a non-trivial amount of work. In particular, while appropriately projecting from an input graph to a tree embedding is trivial in the FRT case, the partialness of our embeddings makes this projection significantly more troublesome. Thus, we develop a projection theorem (Theorem 5.8), which informally shows that a natural projection from G to one of our tree embeddings appropriately preserves cost and connectivity. We prove our projection theorem using "h-hop-connectors" which are, informally, a hop-constrained version of Euler tours. We emphasize that this projection theorem is only used in the analysis of our algorithms.

1.1.4 Applications to Hop-Constrained Network Design. Lastly, we use our embeddings to develop the first non-trivial approximation algorithms for the hop-constrained versions of many classic network design problems (Sections 6). As detailed in Table 1, we give numerous (poly-log, poly-log) bicriteria algorithms for hop-constrained network design problems that relax both the cost and hop constraint of the solution. As noted above, bicriterianess is necessary for any poly-log approximation for Steiner forest and its generalizations. Furthermore, while the results in Table 1 are stated in utmost generality, many special cases of our results were to our knowledge not previously known. For example, our algorithm for hop-constrained oblivious Steiner forest immediately gives new algorithms for hop-constrained Steiner forest, hop-constrained online Steiner tree and hop-constrained online Steiner forest, as well as min-cost h-spanner. Similarly, our algorithm for oblivious

[2] See Section 2 for a formal statement.

[3] In a previous version of this work "h-hop copy tree embeddings" were called "h-hop repetition tree embeddings."

network design immediately gives new algorithms for the hop-constrained version of the well-studied buy-at-bulk network design problem [11]. All of our algorithms for these problems use the above mentioned tree embedding template with either our h-hop partial tree embeddings or our h-hop copy tree embeddings.

Table 1: Our bicriteria approximation results. All results are for poly-time algorithms that succeed with high probability (at least $1 - \frac{1}{\text{poly}(n)}$). For some of the problems we assume certain parameters are poly(n) to simplify presentation. All results are new except for the k-Steiner tree result which is implied by [46].

Hop-Constrained Problem	Cost Apx.	Hop Apx.
Offline Problems		
Relaxed k-Steiner Tree	$O(\log^2 n)$	$O(\log^3 n)$
k-Steiner Tree	$O(\log^3 n)$	$O(\log^3 n)$
Group Steiner Tree	$O(\log^5 n)$	$O(\log^3 n)$
Group Steiner Forest	$O(\log^7 n)$	$O(\log^3 n)$
Online Problems		
Group Steiner Tree	$O(\log^6 n)$	$O(\log^3 n)$
Group Steiner Forest	$O(\log^8 n)$	$O(\log^3 n)$
Oblivious Problems		
Steiner Forest	$O(\log^3 n)$	$O(\log^3 n)$
Network Design	$O(\log^4 n)$	$O(\log^3 n)$

In this short version of our paper we omit several minor proofs as well as our hop-constrained copy tree embeddings and all but one of our applications. For these results please see the full version of our paper, available on arXiv.

2 RELATED WORK

Before proceeding we give a brief overview of additional work on approximation algorithms for hop-constrained network design and tree embeddings. We later give related work in each of our application sections.

2.1 Hop-Constrained Network Design

For some simple hop-constrained network design problems non-trivial (unicriteria) approximation algorithms are known. [9] gave an $O(\log n)$ approximation for minimum depth spanning tree on metrics. [47] gave a $O(\sqrt{\log n})$ for the degree-bounded minimum diameter spanning tree problem. [49] gave a $O(d \log n)$ approximation for computing a minimum cost Steiner tree with depth at most d. [43] gave a constant approximation for the minimum depth Steiner tree problem on a metric.

However, hop constraints often make otherwise easy problems so challenging that the only non-trivial approximation algorithms known or possible are bicriteria. The apparent necessity of bicriterianess in hop-constrained optimization is highlighted by the existence of many bicriteria algorithms. For example, [59] and [52] gave an $(O(\log n), O(\log n))$ bicriteria approximation algorithms for MST and Steiner tree with hop constraints.[4] Similarly, [42] gave a $(O(\log^4 n), O(\log^2 n))$-bicriteria algorithm for k-Steiner tree with hop constraints which was later improved to $(O(\log^2 n), O(\log n))$ by [46]; here the first term is the approximation in the cost while the second term is the approximation in the hop constraint.

Lastly, hop-constrained network design has also received considerable attention from the operations research community; see, for example, [6, 21–23, 25–27, 36–39, 51, 60–63] among many other papers.

2.2 Tree Embeddings

The celebrated embedding of [33] showed that for any metric (V, d) there is a distribution $\mathcal{D}$ of weighted trees on V so that for any $u, v \in V$ we have $d(u, v) \le d_T(u, v)$ for any tree T in the support of $\mathcal{D}$ and $\mathbb{E}_{T\sim\mathcal{D}}\, d_T(u, v) \le O(\log n \cdot d(u, v))$; here, d_T indicates the distance according to the weight function in T. Using these tree embeddings with the above template reduces many graph problems to their tree versions at the cost of $O(\log n)$ in the quality of the resulting solution. This has lead to many algorithms with poly-log approximations and competitive ratios for NP-hard problems including, among many others, the k-server [13], metrical task systems [17], offline and online group Steiner tree and group Steiner forest [7, 35, 56], buy-at-bulk network design [11] and oblivious routing problems [58].

There has also been considerable work on extending the power of tree embeddings to a variety of other settings including tree embeddings for planar graphs [48], online tree embeddings [18], dynamic tree embeddings [24, 34], distributed tree embeddings [45] and tree embeddings where the resulting tree is a subgraph of the input graph [1, 4, 8, 31, 50]. Lastly, the notion of Ramsey trees and Ramsey tree covers has been extensively used for metric-type problems. Specifically, it is known that for every metric (V, d) and k there is some subset $S \subseteq V$ of size at least $n^{1-1/k}$ which embeds into a tree—a so-called Ramsey tree—with distortion $O(k)$ [2, 20, 53, 55]. Iterating (a slight strengthening of) this fact shows that there exist collections of Ramsey trees—so-called Ramsey tree covers—where each vertex v has some "home tree" in which the distances to v are preserved. The guarantees of Ramsey trees and Ramsey tree covers are insufficient for our purposes for at least two reasons: (1) there is the obvious issue that they are only known to apply to metrics which h-hop distances are not and; (2) since we are interested in connectivity problems we cannot just preserve distances on one home tree, but rather, must preserve h-hop distances on all of the nodes in each h-hop partial tree embedding.

3 PRELIMINARIES

Before proceeding to our formal results we define conventions we use throughout this work.

General: We let $[k] := \{1, 2, \ldots, k\}$ for any non-negative integer k. We let $A \sqcup B$ denote the disjoint union of A and B. We often use the Iverson bracket notation $\mathbb{I}[\text{condition}]$ which evaluates to 1 when the condition is true and 0 otherwise.

Graphs: Given a graph $G = (V, E)$ we denote its vertex set by $V(G)$ and $E(G)$, or simply V and E if G is clear from context. We let

[4] A later paper of [57] claimed to improve this result to a $(O(\log n), 2)$-approximation but it is our understanding that this paper was retracted due to a bug.

$n := |V|$. All graphs considered in this paper are undirected. Most commonly, we consider undirected weighted graphs $G = (V, E, w)$ with weights $w : E \to \{1, 2, \ldots, L\}$. The value L is called the aspect ratio and throughout this paper we assume $L = \text{poly}(n)$. We will let w_G be G's weight function if G is not clear from context. Generally, weighted graphs in this paper are assumed to be complete, i.e., $E = \binom{V}{2}$. In the context of this paper this is without loss of generality. In particular, one can transform any non-complete weighted graph $G = (V, E, w)$ with aspect ratio L into an equivalent complete graph G' with aspect ratio $L' = n^2 \cdot L$ which gives a weight of L' to any edge not in E without affecting any of the results in this paper.

Subgraphs: Given a weighted graph $G = (V(G), E(G), w_G)$, we will often consider a subgraph $H = (V(H), E(H))$ where $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Unlike G, such subgraphs will not necessarily be complete. We will often identify a subset of edges $E' \subseteq E(G)$ of a graph G with the subgraph induced by these edges, i.e., the subgraph H with $E(H) = E'$ and $V(H) = \bigcup_{e \in E(H)} e$. Given a collection of vertices $U \subseteq V(G)$ we will let $G[U]$ be the "induced" subgraph with vertex set U and edge set $\{\{u, v\} : u, v \in U \text{ and } \{u, v\} \in E(G)\}$. We define the weight of a subgraph $w_G(H) := \sum_{e \in E(H)} w_G(e)$ as the sum of weights of its edges.

Well-Separated Trees: We will often work with well-separated rooted trees. We say that a weighted rooted tree $T = (V, E, w)$ with root $r \in V$ is well-separated if every root-to-leaf path has weights that are decreasing powers of 2. That is, if e' is a child edge of e in T then $w(e') = \frac{1}{2} w(e)$.

Distances and Metrics: For a set V we call any positive real function $d : V \times V \to R_{\geq 0}$ which is symmetric, i.e., satisfies $d(u, v) = d(v, u)$ for all $u, v \in V$, and satisfies the identity of indiscernibles, i.e., $d(u, v) = 0 \Leftrightarrow u = v$, a distance function (such a function is also often called a semimetric). If d also satisfies the triangle inequality $d(u, w) \leq d(u, v) + d(v, u)$ for all $u, v, w \in V$ then d is called a metric. We also extend the definition of d to sets in the standard way: $d(U, U') := \min_{u \in U, u' \in U'} d(u, u')$.

Paths, Path Length, and Hop Length: A sequence of nodes $P = (v_0, v_1, \ldots, v_\ell)$ in a graph G is called a path if for all $i \in [\ell]$ we have $\{v_{i-1}, v_i\} \in E(G)$ and we say $E(P) := \bigcup_i \{\{p_{i-1}, p_i\}\} \subseteq E(G)$ is the edge set of P. If the nodes in P are distinct we say that P is simple. In this paper paths are not assumed to be simple. We denote the number of hops in P with $\text{hop}(P) := \ell$ and call $\text{hop}(P)$ the hop length of P. If $G = (V, E, w)$ is weighted, we define the weight of a path P in G to be the sum of weights of its edges: $w(P) := \sum_{e \in E(P)} w(e)$.

Hop Distance and Hop Diameter: For a (non-complete) subgraph $H = (V(H), E(H))$ of a (complete) graph G we let $\text{hop}_H(u, v)$ be the minimum number of edges of a path between u and v in H (i.e., using only the edges $E(H)$). We also define the hop diameter of H as $\text{hop}(H) := \max_{u, v \in V(H)} \text{hop}_H(u, v)$.

Shortest-Path Metric and Tree Metric: For a weighted graph G the distance between any two nodes $u, v \in V$ is defined as $d_G(u, v) := \min\{w(P) \mid \text{path } P \text{ between } u, v\}$. It is easy to verify that d_G is a metric on V and for this reason d_G is called the shortest path metric of G. Any metric d on a set V which is identical to a shortest path metric of a weighted tree $T = (V, E, w)$ is called a tree metric; for this reason we will sometimes conflate a tree metric with its corresponding tree.

4 APPROXIMATING HOP-CONSTRAINED DISTANCES

In this section we show that even though hop-constrained distances are not well-approximated by any metric, they are approximated by a distribution over what we call partial tree metrics. More specifically, we consider hop-constrained distances defined as follows.

Definition 4.1 (Hop-Constrained Distances). For a (complete) weighted graph $G = (V, E, w)$ and a hop constraint $h \geq 1$ we define the h-hop distance between any two nodes $u, v \in V$ as

$$d_G^{(h)}(u, v) := \min\{w(P) \mid \text{path } P \in G \text{ between } u, v \text{ s.t. } \text{hop}(P) \leq h\}$$

As we have assumed that our graph G is complete without loss of generality (see Section 3), the above is always well-defined for any $u, v \in V$.

4.1 Hop-Constrained Distances Are Inapproximable by Metrics

We begin by observing that, not only is $d_G^{(h)}$ not a metric, but it is, in general, innaproximable by any metric.

It is easy to verify that $d_G^{(h)}$ is a valid distance function on $V(G)$. Indeed $d_G^{(h)}$ is clearly symmetric, i.e., $d^{(h)}(u, v) = d^{(h)}(v, u)$, and satisfies the identity of indiscernibles, i.e., $d^{(h)}(u, v) = 0 \Leftrightarrow u = v$. However, it is also simple to see that hop-constrained distances are not necessarily metrics since they do not obey the triangle inequality. Indeed, the existence of a short h-hop path from u to v and a short h-hop path from v to w does not imply that the existence of a short h-hop path between u and w. More formally it is possible that $d^{(h)}(u, w) \gg d^{(h)}(u, v) + d^{(h)}(v, w)$. See [10] for a similar observation.

Of course with a factor 2 relaxation in the hop constraint the relaxed triangle inequality $d^{(2h)}(u, w) \leq d^{(h)}(u, v) + d^{(h)}(v, w)$ holds for any graph G and any $u, v, w \in V(G)$. This suggests—albeit incorrectly—that one might be able to approximate hop-constrained distance by allowing constant slack in the hop constraint and length approximation as in the following definition.[5]

Definition 4.2. A distance function $\tilde{d}$ approximates the h-hop constrained distances $d_G^{(h)}$ for a weighted graph $G = (V, E, w)$ where $h \geq 1$ with *distance stretch* $\alpha \geq 1$ and *hop stretch* $\beta \geq 1$ if for all $u, v \in V$ we have

$$d_G^{(\beta h)}(u, v) \leq \tilde{d}(u, v) \leq \alpha \cdot d_G^{(h)}(u, v).$$

As we next observe, no metric provides such an approximation without a very large hop or distance stretch.

Lemma 4.3. *For any hop constraint $h \geq 1$, distance stretch α, hop stretch β and any $L > 1$, there exists a graph $G = (V, E, w)$ with*

[5]In addition to naturally arising when trying to get hop-constrained distances to satisfy the triangle inequality, relaxing hop distances is further motivated by the following. As we later show, a relaxation in the hop constraint in a hop-constrained tree embedding propagates through to an approximation on the hop diameter of the solutions we find for our network design problems. Since, as mentioned above, some amount of approximation on the hop diameter is necessary for many of our problems to admit a poly-log approximation in the cost, relaxing our hop constraints in our notion of approximating hop-constrained distances is a natural way we can set ourselves up for success when aiming for poly-log cost approximations for our problems.

aspect ratio L such that if a metric $\tilde{d}$ approximates $d_G^{(h)}$ with distance stretch α and hop stretch β then $\alpha(\beta h+1) \geq L$.

Indeed, an approximation with the above large stretch is always trivially attainable. In particular, no metric can approximate $d^{(h)}$ any better than the trivial approximation by the scaled shortest-path metric $\alpha \cdot d_G$ which gives value $\alpha \cdot d_G(u,v)$ to each $u, v \in V$, as shown by the following.

Lemma 4.4. *Given any graph $G = (V, E, w)$ with aspect ratio L and a distance stretch α and hop stretch β satisfying $\alpha(\beta h+1) \geq L$, we have that $\alpha \cdot d_G$ approximates $d_G^{(h)}$ with distance stretch α and hop stretch β.*

Thus, hop-constrained distances can be maximally far from *any* metric in the sense that the only way to approximate them by a metric requires so much slack in the hop and distance stretch that the approximation becomes trivial. Moreover, since the expected distance between two nodes in a distribution over metrics is itself a metric, the above result also rules out approximating $d^{(h)}$ in a non-trivial way with distributions over metrics as in FRT. This impossibility remains even when one allows for relaxations of the hop constraint.

4.2 Distances Induced by Distributions Over Partial Metrics

While Lemma 4.3 shows that no metric can approximate $d_G^{(h)}$ on all vertices, it does not rule out the possibility that some metric approximates $d_G^{(h)}$ on a large subset of V. Thus, we introduce the following concept of partial metrics.

Definition 4.5 (Partial Metric). Any metric d defined on a set V_d is called a *partial metric* on V if $V_d \subseteq V$.

We will often talk about how partial metric d approximates $d_G^{(h)}$ on V_d with hop and distance stretches α and β by which we mean that the inequality of Definition 4.2—$d_G^{(\beta h)}(u,v) \leq d(u,v) \leq \alpha \cdot d_G^{(h)}(u,v)$—holds for every $u, v \in V_d$. Of course, a partial metric on the empty set trivially approximates $d_G^{(h)}$ and we are ultimately interested in estimating $d^{(h)}$ on all pairs of nodes. For this reason, we give the following notions of exclusion probability and how a distribution over partial metrics can induce a distance function between all nodes.

Definition 4.6 (Distances of Partial Metric Distributions). Let $\mathcal{D}$ be a distribution of partial metrics of V for weighted graph $G = (V, e, w)$. We say $\mathcal{D}$ has *exclusion probability* ε if for all $v \in V$ we have $\Pr_{d\sim\mathcal{D}}[v \in V_d] \geq 1 - \varepsilon$. If $\epsilon \leq \frac{1}{3}$ then we say that $\mathcal{D}$ induces the distance function $d_{\mathcal{D}}$ on V, defined as

$$d_{\mathcal{D}}(u,v) := \mathop{\mathbb{E}}_{d\sim\mathcal{D}}\left[d(u,v) \cdot \mathbb{I}[u, v \in V_d]\right].$$

It is easy to verify that $d_{\mathcal{D}}$ is indeed a distance function. In particular, we trivially have that $d_{\mathcal{D}}(v,v) = 0$ since $d(v,v) = 0$ for all d in the support of $\mathcal{D}$. An exclusion probability bounded above by $\frac{1}{2}$ guarantees that $\Pr_{d\sim\mathcal{D}}[u, v \in V_d] > 0$ for any $u, v \in V$. This guarantees that $d_{\mathcal{D}}(u,v) > 0$ for $u \neq v$ which makes $d_{\mathcal{D}}$ a valid distance function.

Since we are treating the distance between u and v as 0 in trees which only contain one of u or v it may happen that $d_{\mathcal{D}}(u,v) < d(u,v)$ which may seem strange. However, provided ε is at most some fixed constant, the above notion of distance (up to constants) is equal to the arguably more natural notion of distance $\mathbb{E}_{d\sim\mathcal{D}_{uv}}[d(u,v)]$ where $\mathcal{D}_{uv}$ is $\mathcal{D}$ conditioned on both u and v being in the drawn partial metric; for any $u, v \in V$ this distance is always at least $d(u,v)$. Thus, at the loss of constants the reader may think of $d_{\mathcal{D}}(u,v)$ as a conditional expected distance where we condition on u and v both being in the metric drawn from $\mathcal{D}$. We choose the above notion of distance as opposed to the conditional expectation version as it simplifies our exposition but we emphasize that since these two notions only differ by constants this choice does not impact any of our results.

With these definitions in place we can define what it means for a distribution of partial metrics to approximate hop-constrained distances.

Definition 4.7 (Stretch of Partial Metric Distribution). A distribution $\mathcal{D}$ of partial metrics on V with exclusion probability at most $\frac{1}{3}$ approximates $d^{(h)}$ on weighted graph $G = (V, E, w)$ for hop constraint $h \geq 1$ with *worst-case distance stretch* $\alpha_{WC} \geq 1$ and *hop stretch* $\beta \geq 1$ if each d in the support of $\mathcal{D}$ approximates $d_G^{(h)}$ on V_d with distance stretch α_{WC} and hop stretch β, i.e. for each d in the support of $\mathcal{D}$ and all $u, v \in V_d$ we have

$$d_G^{(\beta h)}(u,v) \leq d(u,v) \leq \alpha \cdot d_G^{(h)}(u,v).$$

Furthermore, $\mathcal{D}$ has *expected distance stretch* $\alpha_{\mathbb{E}}$ if for all $u, v \in V$ we have

$$d_{\mathcal{D}}(u,v) \leq \alpha_{\mathbb{E}} \cdot d_G^{(h)}(u,v).$$

4.3 Approximating Hop-Constrained Distances with Partial Tree Metrics

Even though h-hop distances are generally inapproximable by distributions over metrics, we now show that they are well-approximated by distributions over very simple partial metrics, namely well-separated partial tree metrics.

Theorem 4.8. *For any (complete) weighted graph G, any hop-constraint $h \geq 1$, and any $0 < \varepsilon < \frac{1}{3}$ there is a distribution $\mathcal{D}$ over well-separated tree metrics each of which is a partial metric on $V(G)$ such that $\mathcal{D}$ has exclusion probability at most ε and approximates $d_G^{(h)}$ with expected distance stretch $\alpha_{\mathbb{E}} = O(\log n \cdot \log \frac{\log n}{\varepsilon})$, worst-case distance stretch $\alpha_{\mathrm{WC}} = O(\frac{\log^2 n}{\varepsilon})$ and hop stretch $\beta = O(\frac{\log^2 n}{\varepsilon})$.*

The rest of Section 4.3 is dedicated to the proof of Theorem 4.8. In Section 4.3.1 we define simple "mixture metrics" and show how combining these metrics with padded decompositions leads to random decompositions with desirable properties. In Section 4.3.2 we show how recursively refining these partitions gives a random partial tree metric which proves Theorem 4.8.

4.3.1 Mixture Metrics and Padded Decompositions. To better understand the structure of hop-constrained distances, we develop a decomposition lemma which gives structure both in terms of weights and hops. In particular, we call a collection of disjoint vertex sets $C_1 \sqcup C_2 \sqcup \ldots \sqcup C_k$ a *partial vertex partition*; $C_1 \sqcup C_2 \sqcup \ldots \sqcup C_k$ is a *complete vertex partition* if $\bigcup_i C_i = V$. In a nutshell, we decompose the vertices of a weighted graph G into a partial vertex partition where (1) both the hop diameter and weight diameter of all C_i's is small, (2) C_i and C_j for $i \neq j$ are well-separated both in terms of hops and weight and (3) almost every vertex is in the partial vertex partition. Our decomposition combines two simple ingredient.

Our first ingredient is what we call the mixture metric which is obtained by mixing together hop lengths and weights in the following way.

Definition 4.9 (Mixture Metric). Given a weighted graph $G = (V, E, w)$, a hop scale $h > 0$, and a weight scale $b > 0$, we define a *mixture weight* $w' : E \to \mathbb{R}_{\geq 0}$ of an edge $e \in E$ as $w'(e) := 1/h + w(e)/b$. The shortest path metric induced by w' is called the *mixture metric* $d' : V \times V \to \mathbb{R}_{\geq 0}$.

The utility of the mixture metric is given by three easy to verify facts: It is a metric and so is amenable to standard metric decomposition theorems; if $d'(u,v) \leq \alpha$ in the mixture metric with hop scale h and weight scale b, then $d^{(\alpha \cdot h)}(u,v) \leq \alpha \cdot b$; if $d'(u,v) > \alpha$, then $d^{(\alpha \cdot h/2)}(u,v) > \alpha \cdot b/2$.

Our second ingredient is the well-studied padded decomposition [3, 41]. Given a metric space (V, d) we denote the ball of radius $r \geq 0$ around $x \in V$ with $B_d(x,r) := \{y \in V \mid d(x,y) \leq r\}$. Next, let $C = C_1 \sqcup \ldots \sqcup C_k$ be a (partial or complete) vertex partition. Then, for a subset $U \subseteq V$, we say that U *is broken in* C if $|\{i \mid U \cap C_i \neq \emptyset\}| > 1$. We also denote this event by $U \not\subseteq C$ and its logical negation by $U \subseteq C$. With this notation, we define padded decompositions:

Definition 4.10 (Padded Decompositions). Let (V, d) be a metric space and let $\mathcal{C}$ be a distribution over complete vertex partitions. $\mathcal{C}$ is a $(\rho_{\text{pad}}, \Delta)$-padded decomposition if:

1. *Diameter:* $\max_{u,v \in C_i} d(u,v) \leq \Delta$ for each $C = C_1 \sqcup C_2 \sqcup \ldots \sqcup C_k$ in the support of $\mathcal{C}$ and $i \in [k]$.
2. *Paddedness:* $\Pr_{C \sim \mathcal{C}}[B_d(v,r) \not\subseteq C] < \frac{r \cdot \rho_{\text{pad}}}{\Delta}$ for each $v \in V$ and every $r > 0$.

In other words, each part of a partition in $\mathcal{C}$ has diameter at most Δ and the probability of a node being within r from a node in a different part is at most $\frac{r\rho_{\text{pad}}}{\Delta}$. The value ρ_{pad} is known as the *padding parameter*.[6] Combining padded decompositions with our mixture metric and its properties as observed above gives our decomposition lemma.

LEMMA 4.11. *Let $G = (V, E, w)$ be a weighted graph with padding parameter ρ_{pad}. For any hop constraint $h > 0$, weight diameter $b > 0$, and exclusion probability $\gamma > 0$, there exists a distribution $\mathcal{C}$ over partial vertex partitions where for every $C = C_1 \sqcup \ldots \sqcup C_k$ in the support of $\mathcal{C}$:*

1. Hop-Constrained Diameter: $d_G^{(h)}(u,v) \leq b$ *for* $i \in [k]$ *and* $u, v \in C_i$;
2. Hop-Constrained Paddedness: $d_G^{(h \frac{\gamma}{2\rho_{\text{pad}}})}(u,v) \geq b \cdot \frac{\gamma}{2\rho_{\text{pad}}}$ *for every* $u \in C_i$ *and* $v \in C_j$ *where* $i \neq j$.

And:

3. Exclusion probability: $\Pr_{C \sim \mathcal{C}}[v \notin \bigcup_{i \in [k]} C_i] \leq \gamma$ *for each* $v \in V$ *where* $C = C_1 \sqcup \ldots \sqcup C_k$;
4. Path preservation: $\Pr_{C \sim \mathcal{C}}[V(P)$ *is broken in* $C]$ *is at most* $(\text{hop}(P)/h + w(P)/b) \cdot \rho_{\text{pad}}$ *for each path* P.

PROOF. Let d' be the mixture metric of G with hop scale h and weight scale b and let $\Delta := 2\rho_{\text{pad}}$. We first take a (distribution over) $(\rho_{\text{pad}}, \Delta)$-padded decompositions $C' = C'_1 \sqcup C'_2 \sqcup \ldots \sqcup C'_k$ using d' as the underlying metric. Next, we construct $C_i \subseteq C'_i$ by starting with $C_i := C'_i$ and removing all vertices $v \in C'_i$ where $B_{d'}(v, 2\gamma) \not\subseteq C'_i$. Now $\Pr[v \notin \bigcup_{i \in [k]} C_i] \leq \frac{2\gamma \cdot \rho_{\text{pad}}}{\Delta} \leq \gamma$ for each vertex $v \in V$, as stipulated by (3).

Fix $u, v \in C_i$. Since every C'_i has d'-diameter at most Δ, there exists a sequence of edges $P = (e_1, e_2, \ldots, e_\ell)$ between u and v whose d'-length is at most Δ. Therefore:

$$\Delta \geq \sum_{i=1}^{\ell} \left(\frac{\Delta}{h} + \frac{\Delta \cdot w(e_i)}{b} \right) = \frac{\Delta \cdot \text{hop}(P)}{h} + \frac{\Delta \cdot w(P)}{b}.$$

In other words, $\text{hop}(P) \leq h$ and $w(P) \leq b$, implying that $d_G^{(h)}(u,v)$ is at most b for any $u, v \in C'_i$. Therefore, the same claim holds for $u, v \in C_i \subseteq C'_i$, giving (1).

For $u \in C_i$ and $v \in C_j$ where $i \neq j$ we argue that $d^{(\gamma h/\Delta)}(u,v) > \gamma b/\Delta$, i.e. (2). Suppose for the sake of contradiction that the distance $d_G^{(\gamma h/\Delta)}(u,v)$ is at most $\gamma b/\Delta$. It follows that there exists a path P with $\text{hop}(P) \leq \gamma h/\Delta$ and $w(P) \leq \gamma b/\Delta$. However, the d'-length of P is at most $\frac{\text{hop}(P)\Delta}{h} + \frac{w(P)\Delta}{b} \leq 2\gamma$. Thus, we have contradicted how we constructed C_i from C'_i. Hence $d_G^{(\gamma h/2\rho_{\text{pad}})}(u,v) > \gamma b/2\rho_{\text{pad}}$ since $\Delta = 2\rho_{\text{pad}}$.

Finally, consider a path P from u to v and let $\delta' := \text{hop}(P)\Delta/h + w(P)\Delta/b$. If P is broken in $C_1 \sqcup \ldots \sqcup C_k$ then $B_{d'}(u, \delta') \not\subseteq P$. We therefore have (4), namely we have $\Pr[P$ is broken in $C_1 \sqcup \ldots \sqcup C_k]$ is at most

$$\Pr[B_{d'}(u, \delta') \not\subseteq P] < \frac{\delta' \cdot \rho_{\text{pad}}}{\Delta} = \frac{\delta'}{2} \leq (\text{hop}(P)/h + w(P)/b) \cdot \rho_{\text{pad}}.$$

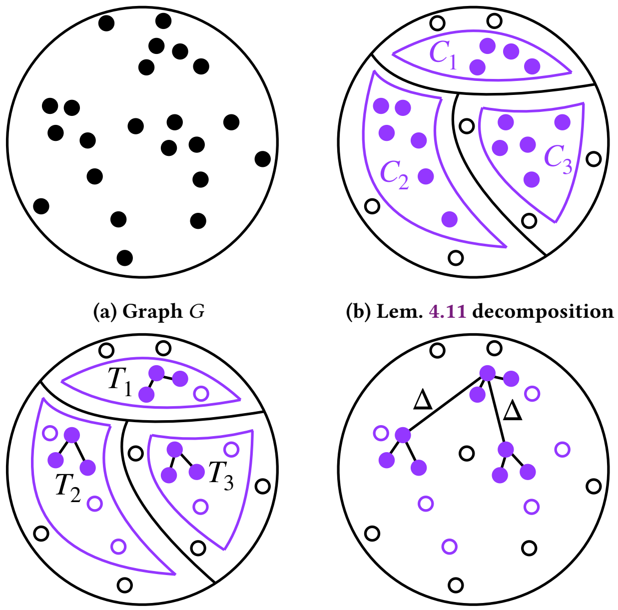
□

Lastly, we note that it is known that every metric has padded decompositions with padding parameter $O(\log n)$ and so our decomposition lemma holds with $\rho_{\text{pad}} = O(\log n)$.

LEMMA 4.12 ([3, 41]). *Every metric on n points admits a $(\rho_{\text{pad}}, \Delta)$-padded decomposition for $\rho_{\text{pad}} = O(\log n)$ and any $\Delta > 0$. Furthermore, such a decomposition can be computed in polynomial time.*

4.3.2 Constructing Tree Metrics for Hop-Constrained Distances and the Proof of Theorem 4.8. Next, we recursively apply the random partial vertex partitions of Lemma 4.11 to obtain a distribution over families of laminar subsets of nodes of G. This distribution will naturally correspond to a distribution over well-separated tree metrics which approximate h-hop constrained distances. In particular, a rough outline of our construction is as follows: we start with a large weight diameter $\Delta \leq \text{poly}(n)$ and hop constraint about h and compute the partial vertex partition $C_1 \sqcup \ldots \sqcup C_k \subseteq V(G)$ of Lemma 4.11. We remove from our process any vertices not in our partial vertex partition. We then recurse on each part C_i while keeping our hop constraint constant but shrinking Δ by a factor of 2. We combine the recursively constructed trees by hanging the

[6]We note that our definition of ρ_{pad} slightly differs from that of other papers, albeit only by a constant factor.



(a) Graph G (b) Lem. 4.11 decomposition (c) Recursing (d) Merging recursions

Figure 1: An illustration of the top-level recursive call of the embedding of Theorem 4.8 on graph G (edges omitted from illustration). Vertices in the partial vertex partition of Lemma 4.11 given in purple. Vertices removed from the process given as empty circles and all other vertices given as filled-in circles.

roots of the returned trees off of the root of a fixed but arbitrary tree with edges of length Δ. The recursion stops when each C_i is a singleton. The resulting tree metric is partial since each application of Lemma 4.11 removes a small fraction of nodes. We illustrate our construction in Figure 1 and proceed to prove Theorem 4.8.

Proof of Theorem 4.8. We describe a recursive and randomized procedure that induces a distribution over well-separated and rooted trees where each tree can be interpreted as a partial tree metric with the required properties. Given hop constraint h', weight diameter Δ and vertex set $V' \subseteq V$ where $d_G^{(h')}(u,v) \le \Delta$, our procedure returns a rooted tree $(V(T), E(T), w_T)$ satisfying $V(T) \subseteq V'$. Let ρ_{pad} be the padding parameter of G; we will give our proofs in terms of ρ_{pad} and then conclude by applying Lemma 4.12. We fix $h' := h \cdot \kappa$ where we define $\kappa := O(\varepsilon^{-1}\rho_{\text{pad}} \log n)$ throughout the procedure. We emphasize that h' will also be the same for all of our recursive calls. The construction procedure is initially invoked with the parameters $V' := V$ and weight scale Δ equal to the smallest power of 2 which is at least the aspect ratio $L \le \text{poly}(n)$. That is, $\Delta \in [L, 2L) \ge \max_{u,v} d_G^{(h')}(u,v)$.

Construction procedure: We will use the Lemma 4.11 decomposition with hop constraint h', weight diameter $\Delta/2$, and exclusion probability $\gamma := \varepsilon/O(\log n)$ (for a sufficiently large hidden constant) to obtain a partial vertex partition $C_1 \sqcup C_2 \sqcup \ldots \sqcup C_k \subseteq V'$ where, plugging in our choice of parameters and the guarantees of Lemma 4.11, we have:

(1) $\max_{u,v \in C_i} d_G^{(h')}(u,v) \le \Delta/2$;
(2) $d_G^{(h)}(C_i, C_j) = d_G^{(h'/\kappa)}(C_i, C_j) \ge \Delta/(2\kappa)$ for each $i, j \in [k]$ where $j \ne i$;
(3) $\Pr[v \notin \bigcup_{i=1}^k C_i] \le \frac{\varepsilon}{O(\log n)}$ for all $v \in V$.

We recursively construct k rooted trees $T_1 = (V_1, E_1, w_1), \ldots, T_k = (V_k, E_k, w_k)$ by calling the same procedure with our distance scale set to $\Delta' \leftarrow \Delta/2$ on sets $C_1, \ldots, C_k$. We construct the tree $T = (V(T), E(T), w_T)$ returned by the procedure by connecting the roots of $T_2, \ldots, T_k$ to the root of T_1 via a tree edge of weight Δ. The procedure is stopped when the set of nodes V' is a singleton, at which point the trivial one-node tree is returned.

Exclusion probability analysis: Consider a recursive call with $v \in V'$ and suppose that the partial vertex partition in the call is $C_1 \sqcup \ldots \sqcup C_k$. By the properties of the partition, $\Pr[v \notin \bigcup_{i=1}^k C_i] \le \varepsilon/O(\log n)$ (for a sufficiently large constant). First, we note that $v \notin V(T)$ if and only if there is a recursive call where $v \in V' \setminus (\bigcup_i C_i)$, which happens with probability $\varepsilon/O(\log n)$. Since v is in a unique recursive call on each level and there are $O(\log n)$ levels, we conclude via a union bound that this happens in at least one level with probability at most ε, proving that the exclusion probability of each node $v \in V$ is a most ε.

Worst-case distance stretch and hop stretch analysis: In the final tree T, for two nodes $u, v \in V(T)$ let $e_{u,v} := \arg\max\{w_T(e) \mid e \in T_{u,v}\}$ be the heaviest weight tree edge on the unique tree path between u and v. The weights w_T are strictly decreasing powers of 2 on any root-leaf path. Therefore, $w_T(e_{u,v}) \le d_T(u,v) \le O(w_T(e_{u,v}))$. Edge $e_{u,v}$ was created via a recursive call with the parameters V' and Δ where $V' \subseteq V$, $u, v \in V'$ and $d^{(h')}(u', v') \le \Delta = w_T(e_{u,v})$ for all $u', v' \in V'$. Let $C_1 \sqcup \ldots \sqcup C_k$ be the partial vertex partition created by this recursive call where each C_i has weight diameter $\Delta/2$ and $d^{(h)}(C_i, C_j) \ge \Delta/(2\kappa)$ when $i \ne j$. Since $u, v \in V(T)$ we have that $u \in C_i$ and $v \in C_j$ for $i \ne j$ (since otherwise $e_{u,v}$ would not be created by this recursive call), hence $d_G^{(h)}(u,v) \ge \frac{\Delta}{2\kappa} = \frac{w_T(e_{u,v})}{2\kappa} = \Theta(\frac{d_T(u,v)}{\kappa})$. Consequently, $d_T(u,v) \le O(\kappa \cdot d_G^{(h)}(u,v))$. Furthermore, since $u, v \in V'$ we have that $d^{(h')}(u,v) \le \Delta \le d_T(u,v)$, which can be rewritten as $d^{(\beta h)}(u,v) \le d_T(u,v)$ for $\beta := O(\kappa)$. Combining the two bounds on d_T we have that both the worst-case distance stretch α_{WC} and hop stretch β are $O(\kappa) = O(\varepsilon^{-1} \log n \cdot \rho_{\text{pad}})$ which gives the desired bound when we plug in the $\rho_{\text{pad}} = O(\log n)$ padded decomposition of Lemma 4.12.

Expected distance stretch analysis: Let Δ_l be the weight diameter of recursive calls at level $l \in [O(\log n)]$. In particular, $\Delta_1 \in (L, 2L]$ and $\Delta_{l+1} = \Delta_l/2$. Fix $u, v \in V$, let P be a path in G between u and v with at most h hops and weight $\delta := d_G^{(h)}(u,v)$ and let $e_{u,v}$ be defined—as in the worst-case distance stretch analysis—as the heaviest weight tree edge between u and v. As in the worst-case stretch analysis, it suffices to bound $w_T(e_{u,v})$. We now partition the $O(\log n)$ levels into three phases $H_1 \sqcup H_2 \sqcup H_3$ where $l \in H_1$ iff $\Delta_l > \delta \cdot (2\kappa)$, $l \in H_3$ iff $\Delta_l \le \delta \cdot (2\rho_{\text{pad}})$ and $l \in H_2$ in the remaining case where $\Delta_l \in (\delta \cdot 2\rho_{\text{pad}}, \delta \cdot 2\kappa]$. We proceed to bound the probability that $e_{u,v}$ is created by a recursive call in H_1, H_2 and H_3 which, in turn, gives a bound on the expected distance between u and v.

We begin with calls at levels in H_1. In particular, we argue that a call at level $l \in H_1$ cannot create the edge $e_{u,v}$ (i.e., it cannot be that $\Delta_l = w_T(e_{u,v})$). This follows from the worst-case distance stretch analysis, which stipulates that $d_G^{(h)}(u,v) \geq \Delta_l/(2\kappa)$. However, this would yield $d_G^{(h)}(u,v) > \delta$, a contradiction. Thus, the contribution of edges corresponding to levels in H_1 to $w_T(e_{u,v})$ is 0:

$$\sum_{l \in H_1} \Pr[e_{u,v} \text{ created by level } l \text{ call}] \cdot \Delta_l \cdot \mathbb{I}[u,v \in V(T)] = 0$$

Next, suppose that $l \in H_2$ and suppose $e_{u,v}$ was created via a level l call with the vertex set V' and partial vertex partition $C_1 \sqcup \ldots \sqcup C_k \subseteq V'$. If this is the case, the path P between u and v is broken in $C_1 \sqcup \ldots \sqcup C_k$, which by Lemma 4.11 happens with probability at most

$$\rho_{\text{pad}}\left(\frac{\text{hop}(p)}{h'} + \frac{w_G(p)}{\Delta_l/2}\right) \leq \rho_{\text{pad}}\left(\frac{h}{h'} + \frac{\delta}{\Delta_l/2}\right) = \frac{\rho_{\text{pad}}}{\kappa} + \frac{\rho_{\text{pad}}\delta}{\Delta_l/2}.$$

Moreover, note that $|H_2| = O(\log(\kappa/\rho_{\text{pad}})) = O(\log(\varepsilon^{-1}\log n))$ since $\Delta_{l+1} = \Delta_l/2$. Therefore we have that the term $\sum_{l \in H_2} \Pr[e_{u,v} \text{ created by level } l \text{ call}] \cdot \Delta_l \cdot \mathbb{I}[u,v \in V(T)]$ is at most

$$\begin{aligned}\sum_{l \in H_2}\left(\frac{\rho_{\text{pad}}}{\kappa} + \frac{\rho_{\text{pad}}\delta}{\Delta_l/2}\right) \cdot \Delta_l \cdot 1 &\leq \sum_{l \in H_2}\left(\Delta_l \cdot \frac{\rho_{\text{pad}}}{\kappa} + 2\rho_{\text{pad}}\delta\right)\\ &\leq \frac{\rho_{\text{pad}}}{\kappa} \cdot (\delta \cdot 2\kappa) + 2\rho_{\text{pad}}\delta|H_2|)\\ &\leq \delta \cdot O(\rho_{\text{pad}}\log(\varepsilon^{-1}\log n)).\end{aligned}$$

Lastly, for H_3 notice that we can coarsely upper bound the terms $\sum_{l \in H_3} \Pr[e_{u,v} \text{ created by level } l \text{ call}] \cdot \Delta_l \cdot \mathbb{I}[u,v \in V(T)]$ as $\sum_{l \in H_3} \Delta_l$ which is at most $\delta \cdot 4\rho_{\text{pad}}$ by our choice of H_3 and the fact that our weight diameters are geometrically decreasing.

Combining our upper bounds on the probability that $e_{u,v}$ is created in each level gives an upper bound on the expectation of $w_T(e_{u,v})$, which in turn bounds the expected value of $d_T(u,v)$ since $d_T(u,v) = O(w_T(e_{u,v}))$. In the following we let $(\ldots)$ stand for $\Pr[e_{u,v} \text{ created by level } l \text{ call}] \cdot \Delta_l \cdot \mathbb{I}[u,v \in V(T)]$. We have that $\mathbb{E}[w_T(e_{u,v}) \cdot \mathbb{I}[u,v \in V(T)]]$ is at most

$$\begin{aligned}&\sum_{l=1}^{O(\log n)} \Pr[e_{u,v} \text{ created by level } l \text{ call}] \cdot \Delta_l \cdot \mathbb{I}[u,v \in V(T)]\\ &\leq \sum_{l \in H_1}(\ldots) + \sum_{l \in H_2}(\ldots) + \sum_{l \in H_3}(\ldots)\\ &\leq 0 + \delta \cdot O(\rho_{\text{pad}}\log(\varepsilon^{-1}\log n)) + \delta \cdot (4\rho_{\text{pad}})\\ &= \delta \cdot O(\rho_{\text{pad}}\log(\varepsilon^{-1}\log n))\end{aligned}$$

Plugging in the padded decompositions of Lemma 4.12, we conclude that the expected distance stretch is $O(\rho_{\text{pad}}\log(\varepsilon^{-1}\log n)) = O(\log n \log(\varepsilon^{-1}\log n))$, as required. □

5 h-HOP PARTIAL TREE EMBEDDINGS

In the preceding section we demonstrated that hop-constrained distances can be well-approximated by distributions over partial tree metrics. In this section we describe how this result gives embeddings which can be used for hop-constrained network design problems. In particular, in Section 5.1 we will define h-hop partial tree embeddings which are partial tree metrics along with a mapping of each edge in the tree metric to a path in G. As an (almost) immediate corollary of our results in the previous section, we have that one can produce such an embedding where h-hop distances are approximately preserved by T and each path to which we map an edge has a low number of hops and less weight than the corresponding edge in T.

However, ultimately we are interested in using these embeddings to instantiate the usual tree embedding template and the above properties alone are not sufficient to do so. In particular, recall that in the usual tree embedding template for network design we embed our input graph into a tree, solve our problem on the tree and then project our solution back onto the input graph. If the problem which we solve on the tree has a much greater cost than the optimal solution on our input graph then our solution has no hope of being competitive with the optimal solution. Thus, we require some way of projecting the optimal solution of G onto our embeddings in a way that produces low-cost, feasible solutions for our tree problems.

When tree embeddings are not partial—as in FRT—such a projection is trivial. However, the partial nature of our embeddings along with the fact that we must preserve "h-hop connectivity" makes arguing that such a low cost solution exists significantly more challenging than in the FRT case. Somewhat surprisingly, we show that a natural projection of the optimal solution onto T produces an appropriate subgraph of T, despite the fact that an FRT-like charging argument seems incapable of proving such a result. Our proofs will be based on what may be viewed as a hop-constrained version of Euler tours which we call h-hop connectors. We give further intuition and details in Section 5.2. Thus, while Section 5.1 is a straightforward extension of our results from the previous section, the primary technical contribution of this section is the projection result of Section 5.2 which shows that, indeed, these embeddings may be used for tree-embedding algorithms in the usual way.

5.1 Defining h-Hop-Partial Tree Embeddings

We begin by defining our partial tree embeddings and proceed to argue that we can map from the trees in these embeddings to our graphs in a weight and connectivity-preserving fashion.

Definition 5.1 (Partial Tree Embedding). A partial tree embedding on weighted graph $G = (V(G), E(G), w_G)$ consists of a rooted and weighted tree $T = (V(T), E(T), w_T)$ with $V(T) \subseteq V(G)$ and a path $T_e^G \subseteq G$ for every $e \in E(T)$ between e's endpoints satisfying $w_G(T_e^G) \leq w_T(e)$.

We extend the notation from Definition 5.1 to nodes in T which are not adjacent: for any two vertices $u, v \in V(T)$, if e_i is the ith edge in T_{uv} (ordered, say, from u to v) then $T_{uv}^G := T_{e_1}^G \oplus T_{e_2}^G \oplus \ldots$ where $\oplus$ is concatenation.

We now define hop and distance stretch of partial tree embeddings analogously to how we defined these concepts for partial metrics.

Definition 5.2 (h-Hop Partial Tree Embedding). A partial tree embedding $(T, \{T_e^G\}_{e \in E(T)})$ is an h-hop partial tree embedding with *distance stretch* $\alpha \geq 1$ and *hop stretch* $\beta \geq 1$ for graph $G = (V(G), E(G), w_G)$ if

(1) $d_G^{(\beta h)} \leq d_T(u,v) \leq \alpha \cdot d^{(h)}(u,v)$ for all $u, v \in V(T) \subseteq V(G)$;

(2) $\text{hop}(T^G_{uv}) \leq \beta h$ for all $u, v \in V(T) \subseteq V(G)$.

Notice that the above definitions show that one can map subgraphs of a partial tree embedding $(T, \{T^G_e\}_{e \in E(T)})$ for G to subgraphs of G in a cost and connectivity preserving way. In particular, given a $T' \subseteq T$ we have that $H := \bigcup_{e \in E(T')} T^G_e$ satisfies (1) $w_G(H) \leq w_T(T')$ and (2) if u and v are connecting in T' then $\text{hop}_H(u, h) \leq \beta h$. In the next section we give a much more involved and interesting proof showing that one can also project from subgraphs of G to T in a cost and connectivity preserving way.

The next observation confirms that, up to an $O(\log n)$, hop stretch and distance stretch for h-hop partial tree embeddings and partial metrics are equivalent, provided the relevant trees are well-separated.

Lemma 5.3. *Let G be a weighted graph and let $h \geq 1$ be a hop constraint.*

- *If $(T, \{T^G_e\}_{e \in E(T)})$ is a partial tree embedding with distance stretch α and hop stretch β then T is a partial tree metric which approximates $d^{(h)}_G$ with distance stretch α and hop stretch β.*
- *Conversely, if T is a partial tree metric with hop diameter $D_T := \text{hop}(T)$ which approximates $d^{(h)}_G$ with distance stretch α and hop stretch β then there is a collection of paths $\{T^G_e\}_{e \in E(T)}$ where $(T, \{T^G_e\}_{e \in E(T)})$ is a partial tree embedding with distance stretch α and hop stretch $D_T \cdot \beta$.*

Analogously to our results for partial metrics, we will also talk about the exclusion probability of distributions over partial tree, the distances they induce and how well they approximate hop-constrained distances; in particular, the following definitions are analogous to Definition 4.6 and Definition 4.7 respectively. For the sake of presentation, here and later in the paper we let $(T, \cdot)$ be shorthand for $(T, \{T^G_e\}_{e \in E(T)})$.

Definition 5.4 (Distances of Partial Tree Embedding Distributions). Let $\mathcal{D}$ be a distribution of partial tree embeddings on weighted graph $G = (V, E, w)$. We say $\mathcal{D}$ has *exclusion probability* ε if for all $v \in V$ we have $\Pr_{(T,\cdot)\sim\mathcal{D}}[v \in V(T)] \geq 1 - \varepsilon$. If $\epsilon \leq \frac{1}{3}$ then we say that $\mathcal{D}$ induces the distance function $d_{\mathcal{D}}$ on V, defined as

$$d_{\mathcal{D}}(u, v) := \mathop{\mathbb{E}}_{(T,\cdot)\sim\mathcal{D}} \left[d_T(u, v) \cdot \mathbb{I}[u, v \in V(T)] \right].$$

Definition 5.5 (Stretch of Partial Tree Embedding Distribution). A distribution $\mathcal{D}$ of h-hop partial tree embeddings on V with exclusion probability at most $\frac{1}{3}$ approximates $d^{(h)}$ on weighted graph $G = (V, E, w)$ for hop constraint $h \geq 1$ with *worst-case distance stretch* $\alpha_{WC} \geq 1$ and *hop stretch* $\beta \geq 1$ if each $(T, \cdot)$ in the support of $\mathcal{D}$ approximates $d^{(h)}_G$ on $V(T)$ with distance stretch α_{WC} and hop stretch β, i.e. for each $(T, \cdot)$ in the support of $\mathcal{D}$ and all $u, v \in V(T)$ we have

$$d^{(\beta h)}_G(u, v) \leq d_T(u, v) \leq \alpha \cdot d^{(h)}_G(u, v).$$

Furthermore, $\mathcal{D}$ has *expected distance stretch* $\alpha_{\mathbb{E}}$ if for all $u, v \in V$ we have

$$d_{\mathcal{D}}(u, v) \leq \alpha_{\mathbb{E}} \cdot d^{(h)}_G(u, v).$$

Concluding, we have that there exists an efficiently-computable distribution over partial tree embeddings with poly-logarithmic stretches.

Theorem 5.6. *Given weighted graph $G = (V, E, w)$, $0 < \epsilon < \frac{1}{3}$ and root $r \in V$, there is a poly-time algorithm which samples from a distribution over h-hop partial tree embeddings whose trees are well-separated and rooted at r with exclusion probability ε, expected distance stretch $\alpha_{\mathbb{E}} = O(\log n \cdot \log \frac{\log n}{\epsilon})$, worst-case distance stretch $\alpha_{WC} = O(\frac{\log^2 n}{\epsilon})$ and hop stretch $\beta = O(\frac{\log^3 n}{\epsilon})$.*

Proof. We begin by remarking that Theorem 4.8 can be adapted so that all trees are rooted at r in the following way. First, we can assume that $r \in V(T)$ by resampling trees until r is in $V(T)$. By a union bound, this increases the exclusion probability by a factor of at most 2, leaves the hop stretch and worst-case distance stretch unchanged, and increases the expected distance stretch by a factor of at most $\frac{1}{1-\varepsilon} = O(1)$; these modifications to our sampling process leave the statement of our theorem unchanged.

Now, suppose that a sampled tree has $r \in V(T)$; we will observe that r can be assumed to be the root of T. In particular, recall that in the construction of T in Theorem 4.8 we recursively constructs trees $T_1, \ldots, T_k$ on the parts of a partial vertex partition and then outputs a tree by connecting the root of $T_2, \ldots, T_k$ to the root of T_1. We note that T_1 is chosen arbitrarily, and so we can choose T_1 to be the tree containing r. Since we may assume inductively that r is the root of T_1, the tree we return has r as its root. Choosing a root in this way does not change the guarantees of our partial tree metrics.

Our result then follows immediately from Lemma 5.3, Theorem 4.8, the observation that the construction procedures of Theorem 4.8 and Lemma 5.3 are poly-time and the fact that well-separated trees have hop diameter $O(\log n)$,. □

5.2 Projecting From The Graph to h-Hop Partial Tree Embeddings

In this section we show how to project the optimal solution for a hop-constrained problem onto a partial tree embedding to get a low-cost subgraph which will be feasible for the optimization problems on trees which we later solve. In particular, we show that it is possible to project *any* subgraph $H \subseteq G$ onto an h-hop partial tree embedding $(T, \cdot)$ with worst-case distance stretch α in a way that α-approximately preserves the cost of H and preserves "h-hop connectivity": that is, the projection of H will have cost at most $O(\alpha \cdot w_G(H))$ and if u and v are within h hops in H then they will be connected by the projection of H onto our embedding.

In the (non-partial) tree embedding setting where we typically only care about the connectivity structure of nodes—as in FRT—such a projections is trivial. In particular, if T is a tree drawn from the FRT distribution then an edge $e \in E(G)$ can be projected onto the simple tree path $T_{uv} \subseteq T$ between u and v in T and the resulting path will have expected weight $O(\log n \cdot w_G(e))$. Thus, we can project a subgraph $H \subseteq G$ to $T(H) := \bigcup_{\{u,v\} \in E(H)} T_{uv}$. If u and v are connected in H then they are connected in $T(H)$ and so the connectivity of nodes is preserved. Moreover, we can upper bound the weight of $T(H)$ by summing up $w_T(T_{uv})$ over all $\{u, v\} \in E(H)$ to get that, in expectation, $w_T(T(H)) \leq O(\log n \cdot w_G(H))$ and so the cost of the projection is appropriately low.

We might naturally try to use the same projection as is used in the FRT case but only for the nodes embedded by T. Specifically, suppose that T is now the tree of a partial tree embedding with worst-case distance stretch α. Then, we could project H to $T(H) := \bigcup T_{uv}$ where the $\bigcup$ is taken over all u, v such that $\{u, v\} \in E(H)$ *and* $u, v \in V(T)$. Although we trivially have that $w_T(T(H)) \leq \alpha \cdot w_G(H)$ by summing up over edges in $E(H)$, such a projection has no hope of preserving h-hop-connectivity as required: if, for example, u and v are connected by exactly one path in H with h hops then if there is even a single node along this path which is not in $V(T)$ then u and v may not be connected in $T(H)$.

We could try to fix these connectivity issues by forcing all vertices in T which are within h hops in H to be connected in T as captured by the following definition.

Definition 5.7 ($T(H,h)$). Let $(T, \cdot)$ be a partial tree embedding. Then $T(H,h) := \bigcup T_{uv}$ where the $\bigcup$ is taken over u, v such that $u, v \in V(T)$ and $\text{hop}_H(u, v) \leq h$.

$T(H,h)$ trivially preserve h-hop connectivity as needed: if u and v are connected by an h-hop path in H then they will be connected in $T(H,h)$. However, while $T(H,h)$ preserves h-hop connectivity, it seems to yield a subgraph of T of potentially unboundededly-bad cost. For example, let $h = 3$ and suppose H is a spider graph with $O(n)$ nodes in which one leg connects vertex r to center c with a cost 1 edge and the remaining ith leg connects c to u_i to v_i with a sufficiently small $\epsilon > 0$ cost edge. Further, suppose that $V(T)$ consists of r and all v_i. $T(H,h)$ will buy T_{rv_i} for every i since there is an h-hop path from r to v_i. Our worst-case distance guarantee ensures that $w_T(T_{rv_i})$ is at most $\alpha \cdot d_G(v_i, r) \approx \alpha$ and so we might hope to bound the cost of $T(H,h)$ as within $O(\alpha)$ times $w_G(H)$. However, if we try to apply the usual FRT-type proof and upper bound the cost of $T(H,h)$ in T as $\sum d_T(r, v_i)$ then our sum comes out to $O(\alpha \cdot n)$. On the other hand, $w_G(H) \approx 1$ and so $w_T(T(H,h))$ is a factor of $O(n \cdot \alpha)$ larger than $w_G(H)$ while we would like it to only be an $O(\alpha)$ factor larger. Thus, whereas FRT can charge each path in the projection of H to a unique edge of H, the partialness of our embedding means that we must charge paths in $T(H,h)$ to paths in H. These paths in H may induce large congestion which causes us to "overcharge" edges of H.

Surprisingly, in what follows we show that, while the above naive charging argument cannot succeed, a more nuanced proof shows that the above $T(H,h)$ is, in fact, competitive with the optimal solution up to small constants in the hop and distance stretch.

Theorem 5.8. *Fix $h \geq 1$, let H be a subgraph of weighted graph $G = (V, E, w_G)$ and let $(T, \cdot)$ be an $8h$-hop partial tree embedding of G with worst-case distance stretch α. Then $w_T(T(H,h)) \leq 4\alpha \cdot w_G(H)$.*

The basic idea of our proof will be to identify a collection of low congestion paths in H to which we can charge $T(H,h)$.

5.2.1 Warm-Up: Low Diameter Tree Case. To illustrate this idea we begin by showing how to prove Theorem 5.8 in the simple case where G is a tree with diameter at most h. In particular, on a tree of diameter at most h we can mitigate the congestion of charged paths by buying an Euler tour restricted to our embedded nodes; conveniently $T(H,h)$ will also be a subgraph of the projection of such an Euler tour onto T.

More specifically, suppose G is a tree with diameter at most h and let $(T, \cdot)$ be a partial tree embedding of G. Let G_2 be the multigraph of G where each edge is doubled. Let $t = (v_1, v_2, \ldots)$ be an Euler tour of G_2 and let $t' = (w_1, w_2, \ldots)$ be the vertices of $V(T)$ visited by this tour in the order in which they are visited. That is, t' is gotten from t be deleting from it all vertices not in $V(T)$ while leaving the ordering of the remaining vertices unchanged. Notice that vertices in $V(T)$ might occur multiple times in t'. We let P_ℓ be the path in G between w_ℓ and $w_{\ell+1}$ and let $\mathcal{P} := \{P_\ell\}_\ell$. Next, consider $T(\mathcal{P})$ which is the union of T_{uv} for every u, v where u and v form the endpoints of some path in $\mathcal{P}$.

First, notice that $T(H,h) \subseteq T(\mathcal{P})$. This follows since every $u, v \in V(T)$ which are within h hops (namely all $u, v \in V(T)$) are also visited by t' and so if T_{uv} is included in $T(H,h)$ then it will also be included in $T(\mathcal{P})$. Next, notice that $w_T(\mathcal{P}) \leq 2\alpha \cdot w_G(H)$ since our Euler tour when projected onto G visited each edge at most twice. This proves Theorem 5.8 for the h-diameter tree case.

5.2.2 h-Hop Connectors. The key observation of the above warm-up is that Euler tours allow us to mitigate the congestion induced in our charging arguments by providing a low-congestion collection of paths. We abstract such a collection of paths out in the form of what we call h-hop connectors.

For undirected and unweighted graph $G = (V, E)$ with $W \subseteq V$, we let $\mathcal{P}^{(h)}(W)$ be all simple paths between vertices in W with at most h hops. That is, each $P \in \mathcal{P}^{(h)}(W)$ has vertices in W as its first and last vertices and satisfies $|P \cap W| = 2$ and $\text{hop}(P) \leq h$. Given a collection of paths $\mathcal{P}$ in G between vertices in W, we abuse notation and let $(W, \mathcal{P})$ be the graph with vertex set W and an edge $\{u, v\}$ iff there is a $P \in \mathcal{P}$ with endpoints $\{u, v\}$. We will refer to $(W, \mathcal{P}^{(h)}(W))$ as the h-hop connectivity graph of W. We let $c_e(\mathcal{P}) := |P \in \mathcal{P} : e \in P|$ be the congestion of e with respect to a collection of paths $\mathcal{P}$. With this notation in hand, we give our definition of h-hop connectors which we illustrate in Figure 2.

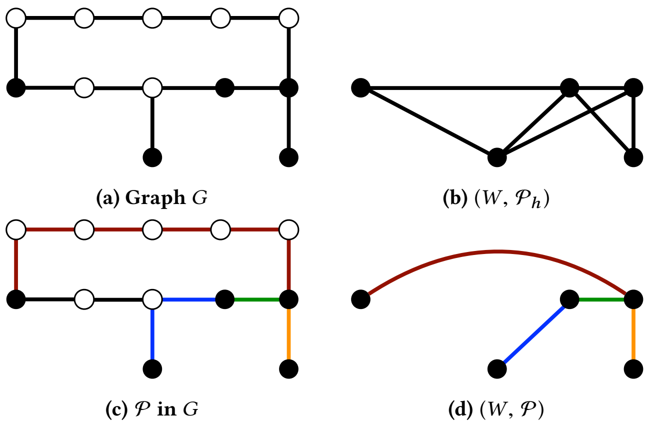
Definition 5.9 (h-Hop Connector). Let $G = (V, E)$ be an undirected and unweighted graph, let $h \geq 1$ and let $W \subseteq V$. An h-hop connector $\mathcal{P}$ of W with congestion C and hop stretch β is a collection of paths in G between vertices of W such that:

1. *Connecting:* if $u, v \subseteq W$ are connected in $(W, \mathcal{P}^{(h)}(W))$ then they are connected in $(W, \mathcal{P})$;
2. *Edge Congestion:* For all $e \in E$ we have $c_e(\mathcal{P}) \leq C$;
3. *Hop Stretch:* $\text{hop}(P) \leq \beta \cdot h$ for all $P \in \mathcal{P}$.

It is easy to observe that the existence of good h-hop connectors are sufficient to show Theorem 5.8.

Lemma 5.10. *Fix $h \geq 1$, let $H \subseteq G$ be a subgraph of weighted graph $G = (V, E, w_G)$ and let $(T, \cdot)$ be a (βh)-hop partial tree embedding of G with worst-case distance stretch α. If H has an h-hop connector on $V(T)$ with hop stretch β and congestion C then $w_T(T(H,h))$ is at most $C\alpha \cdot w_G(H)$.*

Proof. Let $\mathcal{P}$ be the stated h-hop connector, let $S := \{(u, v) : (u, \ldots, v) \in \mathcal{P}\}$ be the endpoints of its path and also let $T(\mathcal{P}) := \bigcup_{(u,v) \in S} T_{uv}$ be the subgraph of T corresponding to $\mathcal{P}$. By the connecting property of our h-hop connector any u, v which are within h hops in H must also be connected in $T(\mathcal{P})$ and so $T(H,h) \subseteq T(\mathcal{P})$. Combining this with the edge congestion and hop stretch of our h-hop connector with the worst-case distance stretch of T we have $w_T(T(H,h)) \leq w_T(T(\mathcal{P})) \leq C\alpha \cdot w_G(H)$. ◻



(a) Graph G (b) $(W, \mathcal{P}_h)$

(c) $\mathcal{P}$ in G (d) $(W, \mathcal{P})$

Figure 2: An illustration of an h-hop connector with congestion 1 and hop stretch 2 on a graph G for a vertex set $W \subseteq V(G)$ with $h = 3$. Vertices of W given as solid black circles; all other vertices of G given as white circles. Edges in $(W, \mathcal{P})$ and paths in $\mathcal{P}$ colored according to their correspondence.

Thus, we devote the remainder of this section to showing that every graph has an h-hop connector with hop stretch 8 and congestion 4.

A simple proof similar to the above warm-up shows that trees with low diameter have good h-hop connectors.

Lemma 5.11. *Let $G = (V, E)$ be a tree with diameter at most βh for $h \geq 1$. Then, G has an h-hop connector with congestion at most 2 and hop stretch at most β for every $W \subseteq V$.*

Proof. Suppose G is a tree. Let G_2 be the multigraph of G where each edge is doubled. Let $t = (v_1, v_2, \ldots)$ be an Euler tour of G_2 and let $t' = (w_1, w_2, \ldots)$ be the vertices of W visited by this tour in the order in which they are visited. That is, t' is gotten from t be deleting from it all vertices not in W while leaving the ordering of the remaining vertices unchanged. Notice that vertices in W might occur multiple times in t'. We let P_ℓ be the path in G between w_ℓ and $w_{\ell+1}$ and let $\mathcal{P} := \{P_\ell\}_\ell$.

Since every vertex in W occurs at least once in t' we have that all vertices in W are connected in $(W, \mathcal{P})$. Since t used each edge of G_2 once, it follows that $c_e(\mathcal{P}) \leq 2$. Lastly, $\text{hop}(P_\ell) \leq \beta h$ for all $P_\ell \in \mathcal{P}$ since each P_ℓ is a simple path in a tree with diameter at most βh. □

We proceed to show how to construct an h-hop connector with congestion 4 and hop stretch 8 on any graph. We first reduce the general graph case to the forest case: we show that, up to a factor of 2 in the hop stretch, every graph G has as a subgraph a forest F where an h-hop connector for F is an h-hop connector for G. We then reduce the forest case to the low diameter tree case by cutting each tree in F at $O(h)$-spaced annuli from an arbitrary root so that the resulting trees have low diameter. We apply Lemma 5.11 to the resulting low-diameter trees. More specifically, we perform these cuts and applications of Lemma 5.11 twice with two different offsets to get back paths $\mathcal{P}_1$ and $\mathcal{P}_2$; we then take our h-hop connector to be $\mathcal{P} := \mathcal{P}_1 \cup \mathcal{P}_2$.

We begin with a simple technical lemma which shows that the graphs induced by the connected components of the h-hop connectivity graph are disjoint. For a collection of paths $\mathcal{P}$ in G we let $G[\mathcal{P}] := G[\bigcup_{P \in \mathcal{P}} V(P)]$ be the graph induced by the union of all such paths.

Lemma 5.12. *Let $G = (V, E)$ be a graph, let $W \subseteq V$, and let U and U' be the vertices of two distinct connected components of $(W, \mathcal{P}^{(h)}(W))$. Then $G[\mathcal{P}^{(h)}(U)]$ and $G[\mathcal{P}^{(h)}(U')]$ are vertex-disjoint.*

Applying the above lemma, we show that, up to a factor of 2 in the hop stretch, we may assume that our graph is a forest. We let $\mathcal{P}_G^{(h)}(W)$ be all paths with at most h hops between vertices in W *in graph* G.

Lemma 5.13. *Let $G = (V, E)$ be a graph, let $W \subseteq V$. Then there exists a subgraph $F \subseteq G$ which is a forest where $u, v \in W$ are connected in $(W, \mathcal{P}_G^{(h)}(W))$ iff u, v are connected in $(W, \mathcal{P}_F^{(2h)}(W))$.*

Proof. We will iteratively construct F. Specifically, for each connected component of $(W, \mathcal{P}_G^{(h)}(W))$ with vertex set U we will maintain a collection of paths $\mathcal{P}_U$ where these paths are all contained in $G[\mathcal{P}^{(h)}(U)]$ and F is the graph induced by the union of all these paths. It follows that by Lemma 5.12 if $G[\mathcal{P}_U]$ is a tree then the connected components of our final solution are indeed a forest. We will maintain the following invariants for our $\mathcal{P}_U$s where $\text{hop}_G(v, U) := \min_{u \in U} \text{hop}_G(v, u)$:

(1) $U' := U \cap V(G[\mathcal{P}_U])$ is connected in $(U', \mathcal{P}_U)$;
(2) $G[\mathcal{P}_U]$ is a tree;
(3) $\text{hop}(P) \leq 2h$ for every $P \in \mathcal{P}_U$;
(4) $\text{hop}_{G[\mathcal{P}_U]}(v, U) \leq h$ for every $v \in V(G[\mathcal{P}_U])$.

We initialize $\mathcal{P}_U$ to contain a path consisting of exactly one (arbitrary) vertex in U. Notice that our construction trivially satisfies these invariants initially.

Next, we repeat the following until $U' = U$. Let u be a vertex in $U \setminus U'$ where u has a path P of at most h hops to a vertex in U'; such a u and P must exist by the definition of U. Let x be the first vertex in $P \cap G[\mathcal{P}_U]$ where we imagine that P starts at u and let P_{ux} be the subpath of P from u to x. By invariant 4 we also know there is some path in $G[\mathcal{P}_U]$ from x to a $u' \in U'$ with at most h hops; call this path $P_{xu'}$ and let P' be the concatenation of P_{ux} and $P_{xu'}$; we add P' to $\mathcal{P}_U$. Notice that this adds u to U' and so this process will eventually terminate at which point $U' = U$.

Let us argue that our invariants hold. Our first invariant holds since before adding u to U', U' was connected and after adding u to U', u is connected to u' by P'. Our second invariant holds since x was the first vertex in $G[\mathcal{P}_U]$ incident to P. Our third invariant holds since P_{ux} and $P_{xu'}$ were each of at most h hops. Our fourth invariant holds since the only new vertices we add to $G[\mathcal{P}_U]$ are the vertices of P_{ux}, all of which are within h hops of u.

Lastly, notice that once $U' = U$ for every U, our claim follows from invariants 1,2 and 3. □

By turning our graph into a forest with Lemma 5.13 and then cutting the constituent trees at $O(h)$-spaced level sets with two different initial offsets, we can conclude that every graph has h-hop connectors with constant congestion and hop stretch.

Lemma 5.14. *Let $G = (V, E)$ be a graph. Then G has an h-hop connector with congestion 4 and hop stretch 8 for every $W \subseteq V$.*

Proof. By Lemma 5.13 we know that there is a forest F such that u, v are connected in $(W, \mathcal{P}_G^{(h)}(W)$ iff u, v are connected in $(W, \mathcal{P}_F^{(2h)}(W))$. Let T be a tree in this forest and notice that to get an h-hop connector on G with hop stretch 8 and congestion 4, it suffices to find a $2h$-hop connector on T with hop stretch 4 and congestion 4.

We do so as follows. Root T arbitrarily at root r and let $T_1, T_2, \ldots$ be the subtrees resulting from cutting T once every $4h$ levels and let $T_1', T_2', \ldots$ be the subtrees resulting from cutting T every $4h$ levels with an initial offset of $2h$. That is, $T_i = T[V(T_i)]$ and $v \in V(T_i)$ iff $4h(i-1) \leq d_T(v, r) < 4h \cdot i$ and $T_i' = T[V(T_i')]$ where $v \in V(T_i')$ iff $\max(4h(i-1)-2h, 0) \leq d_T(v, r) < 4h \cdot i - 2h$. Notice that each T_i and T_i' has diameter at most $4(2h)$. Thus, by Lemma 5.11 we know that each T_i and T_i' have $2h$-hop connectors $\mathcal{P}_i$ and $\mathcal{P}_i'$ with congestion at most 2 and hop stretch at most 4. Thus, we let $\mathcal{P}_1 := \{\mathcal{P}_i\}_i$ and $\mathcal{P}_2 := \{\mathcal{P}_i'\}_i$ and we let our h-hop connector for T be $\mathcal{P} := \mathcal{P}_1 \cup \mathcal{P}_2$.

Let us argue that $\mathcal{P}$ is a $2h$-hop connector on T with hop stretch 8 and congestion 4. The bound on the congestion is immediate from Lemma 5.11 and the fact that each edge occurs in at most 2 trees among all T_i and T_i'. To see why $\mathcal{P}$ is connecting notice that if u, v are within $2h$ hops of one another in T by some path P then this path must be fully contained in some T_i or T_i'; it follows that u and v will be connected in some $\mathcal{P}_i$ or $\mathcal{P}_i'$ and so connected in $\mathcal{P}$. Lastly, our hop bound is immediate by Lemma 5.11 since each T_i and T_i' has diameter at most $4(2h)$. □

Combining Lemma 5.14 with Lemma 5.10 immediately gives Theorem 5.8.

Before proceeding to our applications, we remark on a subtle issue regarding independence and expected distance stretch versus worst case distance stretch. Theorem 5.8 bounded the cost of projecting a subgraphs of G onto a partial tree embedding of G based on the tree embedding's worst-case distance stretch; one might naturally wonder if similar results are possible in terms of the expected distance stretch of a distribution over partial tree embeddings. Here, dependence issues and the partialness of our embeddings work against us. Specifically, one would have to argue that $T(H, h)$—and, in particular, the relevant h-hop connector for $T(H, h)$—has low cost in expectation where $(T, \cdot)$ is drawn from a distribution. However, while it is true that for a fixed H and T the relevant h-hop connector for H and T has low cost in expectation over the entire distribution of tree embeddings, it need not be the case that this h-hop connector has low cost when we condition on the fact that T is the tree we drew from our distribution. In short, Lemma 5.10 seems to fail to hold for the expectation case.

6 APPLICATION OF h-HOP PARTIAL TREE EMBEDDINGS TO OBLIVIOUS HOP-CONSTRAINED STEINER FOREST

In this section we apply our embeddings of $d^{(h)}$ to give approximation algorithms for hop-constrained (oblivious) Steiner forest. While we give our results for oblivious hop-constrained Steiner forest, it is easy to see that an approximation algorithm for the oblivious version gives an approximation algorithm with the same approximation ratios for the online and offline versions of the problem; to our knowledge nothing was known for any of these variants prior to our work.

In Steiner forest we are given a weighted graph $G = (V, E, w)$.

- *Offline:* In offline Steiner forest we are also given a collection of pairs of nodes $\{(s_i, t_i)\}_i$. Our goal is to find a subgraph $H \subseteq G$ so that every s_i is connected to every t_i in H.
- *Online:* In online Steiner forest in each time step $t = 1, 2, \ldots$ a new pair of vertices (u_t, v_t) is revealed and we must maintain a solution H_t for each t where $H_{t-1} \subseteq H_t$ which connects pairs in $\{(u_1, v_1), \ldots, (u_t, v_t)\}$.
- *Oblivious:* In oblivious Steiner forest we must specify a path P_{uv} for each pair of vertices $(u, v) \in V \times V$ before seeing any demands. The demands $\{(s_i, t_i)\}_i$ are then revealed, inducing our solution $H := \bigcup_i P_{s_i t_i}$.

In all three problems the cost of our solution H is defined as $w(H) := \sum_{e \in E(H)} w(e)$. In the oblivious and offline versions, our approximation ratio is $w(H)/\mathrm{OPT}$ where OPT is the cost of the optimal offline solution for the given demand pairs. The competitive ratio of our solution in the online case is $\max_t w(H_t)/\mathrm{OPT}_t$ where OPT_t is the minimum cost subgraph of G connecting pairs in $\{(u_1, v_1), \ldots, (u_t, v_t)\}$.

In the hop-constrained versions of each of these problems we are additionally given a hop constraint $h \geq 1$ and if (s_i, t_i) is a demand pair then our solution H must satisfy $\mathrm{hop}_H(s_i, t_i) \leq h$ for all i. The optimal solution against which we measure our approximation ratio is similarly hop-constrained. Notice that, unlike in the Steiner forest problem where we may assume without loss of generality that each connected component of H is a tree, in hop-constrained Steiner forest each connected component of H might not be a tree.

We give some brief highlights from work in Steiner forest and hop-constrained Steiner forest: while NP-hard [5] gave the first constant approximation for offline Steiner forest; [19] gave an (optimal) $O(\log k)$ approximation for online Steiner forest and [40] gave the first non-trivial approximation algorithm for oblivious Steiner forest, an $O(\log^2 n)$ approximation. There has also been quite a bit of work on approximation algorithms for h-spanners which can be seen as a special case of offline hop-constrained Steiner forest; see, for example, [30] and references therein. Notably for our purposes, [32] and [28] show that unless it holds that $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(2^{\mathrm{poly}\log n})$ hop-constrained Steiner forest admits no $O(2^{\log^{1-\epsilon} n})$ approximation; this immediately rules out the possibility of a poly-log (unicriteria) approximation for hop-constrained Steiner forest. We also note that a recent work of [12] gave results for hop-constrained Steiner forest from a parameterized complexity perspective.

Roughly, our algorithm follows the tree-embedding template: we first apply our h-hop partial tree embeddings to reduce oblivious hop-constrained Steiner forest to oblivious Steiner forest on a tree; we then observe that oblivious Steiner forest is trivially solvable on trees and project our solution back to G. The only minor caveats are: (1) since our tree embeddings will only embed a constant fraction of nodes, we must repeat this process $O(\log n)$ times and (2) for each tree embedding we must use Theorem 5.8 to argue that there is a cheap, feasible solution for the relevant Steiner forest problem on each tree.

Formally, our algorithm to compute our solution H is as follows. We begin by applying Theorem 5.6 to sample $8h$-hop partial tree embeddings $T_1, T_2, \ldots, T_k$ where $k := O(\log n)$ for a sufficiently large hidden constant, $\varepsilon = .1$ and an arbitrary root. Given $u, v \in V$, assign the pair (u, v) to an arbitrary T_j such that $u, v \in V(T_j)$ (we will argue that such a T_j exists with high probability). Next, we let our path for u, v be $P_{uv} := (T_j)^G_{uv}$ the projection of the tree path between u and v onto G. We now analyze this algorithm.

Theorem 6.1. *There is a poly-time algorithm which given an instance of h-hop-constrained oblivious Steiner forest returns a collection of paths such that the induced solution H for any demand set satisfies $w(H) \leq O(\mathrm{OPT} \cdot \log^3 n)$ and $\mathrm{hop}_H(s_i, t_i) \leq O(h \cdot \log^3 n)$ with high probability.*

Proof. We use the above algorithm. We begin by arguing that H connects every s_i to t_i for every i with high probability with a path of at most $O(\log^3 n \cdot h)$ edges. Fix a vertex v. A standard Chernoff-and-union-bound-type argument shows that v is in at least $.8k$ of the T_j with high probability. Specifically, let X_j be the random variable which indicates if v is in $V(T_j)$, let $X := \sum_j X_j$ and apply a Chernoff bound to X.

Taking a union bound over all v we have that with high probability every v is in at least $.8k$ of the T_j. Since we have k total T_j, by the pigeonhole principle it follows that any pair of vertices (s_i, t_i) simultaneously occur in at least $.6k$ of the T_j, meaning that for each such pair there is a T_j where we buy $(T_j)^G_{s_i t_i}$ and so s_i will be connected to t_i in our solution. Since $\mathrm{hop}((T_j)^G_{s_i t_i}) \leq O(h \cdot \log^3 n)$ by Theorem 5.6, it follows that $\mathrm{hop}_H(s_i, t_i) \leq O(h \cdot \log^3 n)$.

We next argue that our solution satisfies the stated cost bound. Let H_{T_j} be the minimal subgraph of T_j connecting all pairs assigned to T_j and let $H_j := \bigcup_{e \in H_{T_j}} (T_j)^G_e$ be the projection of H_{T_j} onto G. Notice that it suffices to argue that $w_{T_j}(H_{T_j}) \leq O(\mathrm{OPT} \cdot \log^2 n)$ for every j since if this held we would have by Theorem 5.6 that the cost of our solution is $w(H) \leq \sum_j w(H_j) \leq \sum_j \sum_{e \in H_{T_j}} w((T_j)^G_e) \leq \sum_j \sum_{e \in H_{T_j}} w_{T_j}(e) = \sum_j w_{T_j}(H_{T_j}) \leq O(\mathrm{OPT} \cdot \log^3 n)$. However, applying Theorem 5.8 to the optimal solution H^* on G shows that $T(H^*, h)$ is a feasible solution for the Steiner forest problem on T_j which connects all pairs assigned to T_j with cost at most $O(\log^2 n \cdot \mathrm{OPT})$. Since H_{T_j} is the optimal solution for such a Steiner forest problem, it follows that $w_{T_j}(H_{T_j}) \leq O(\mathrm{OPT} \cdot \log^2 n)$. □

7 CONCLUSION AND FUTURE WORK

In this work we showed that, while far from any metric, hop-constrained distances are well-approximated by partial tree metrics. We used this fact to develop new embeddings for hop-constrained distances which we then used to give the first bicriteria (poly-log, poly-log) approximation algorithms for many classic network design problems.

We conclude by giving directions for future work. Reducing the stretch in our embeddings, or proving lower bounds stronger than those immediately implied by the FRT lower bounds is our main open question. Improving the upper bounds in our embeddings—as in the FRT setting—has the benefit that doing so immediately improves the approximation ratios for the many algorithms we gave in this paper. We note that, like the embeddings of [16], our embeddings are built around the paddedness of certain decompositions and these embeddings were later improved by FRT [33]. One might naturally wonder, then, if an FRT-like analysis might improve our stretch guarantees; from what we can tell no such FRT-type proof seems capable of improving our bounds. Another point to note is that we lose an $O(\log n)$ in the hop stretch when moving from partial tree metrics to partial tree embeddings. This loss does not seem to have an analogue in the (non-partial) tree embedding setting and it is not clear if such a loss is necessary.

Moreover, while tree embeddings have proven useful for many network design problems, there are many other problems such as k-server [13], metrical task systems [17] and requirement cuts [54] where tree embeddings enabled the first poly-log approximations. Thus, while the focus of our paper has been on the hop-constrained versions of network design problems, we expect that our embeddings will prove useful for the hop-constrained versions of many of these other problems.

Lastly, as we discussed at the end of Section 5, our h-hop partial tree embeddings are built on the worst-case stretch guarantees of our partial metrics; it would be interesting if it were possible to construct embeddings based on the expected stretch guarantees of our partial metrics. Such a result would immediately give several randomized algorithms for hop-constrained problems with low expected cost.

REFERENCES

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. 2008. Nearly tight low stretch spanning trees. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 781–790.
[2] Ittai Abraham, Shiri Chechik, Michael Elkin, Arnold Filtser, and Ofer Neiman. 2018. Ramsey spanning trees and their applications. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1650–1664.
[3] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. 2019. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM J. Comput.* 48, 3 (2019), 1120–1145.
[4] Ittai Abraham and Ofer Neiman. 2012. Using petal-decompositions to build a low stretch spanning tree. In *Annual ACM Symposium on Theory of Computing (STOC)*. 395–406.
[5] Ajit Agrawal, Philip Klein, and Ramamoorthi Ravi. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.* 24, 3 (1995), 440–456.
[6] İbrahim Akgün and Barbaros Ç Tansel. 2011. New formulations of the hop-constrained minimum spanning tree problem via Miller–Tucker–Zemlin constraints. *European Journal of Operational Research* 212, 2 (2011), 263–276.
[7] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. 2006. A general approach to online network optimization problems. *ACM Transactions on Algorithms (TALG)* 2, 4 (2006), 640–660.
[8] Noga Alon, Richard M Karp, David Peleg, and Douglas West. 1995. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.* 24, 1 (1995), 78–100.
[9] Ernst Althaus, Stefan Funke, Sariel Har-Peled, Jochen Könemann, Edgar A Ramos, and Martin Skutella. 2005. Approximating k-hop minimum-spanning trees. *Operations Research Letters* 33, 2 (2005), 115–120.
[10] Alexandr Andoni, Clifford Stein, and Peilin Zhong. 2020. Parallel approximate undirected shortest paths via low hop emulators. In *Annual ACM Symposium on Theory of Computing (STOC)*. 322–335.
[11] Baruch Awerbuch and Yossi Azar. 1997. Buy-at-bulk network design. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 542–547.
[12] Amy Babay, Michael Dinitz, and Zeyu Zhang. 2018. Characterizing demand graphs for (fixed-parameter) shallow-light steiner network. *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)* (2018).

[13] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. 2011. A polylogarithmic-competitive algorithm for the k-server problem. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 267–276.

[14] Judit Bar-Ilan, Guy Kortsarz, and David Peleg. 2001. Generalized submodular cover problems and applications. *Theoretical Computer Science* 250, 1-2 (2001), 179–200.

[15] Yair Bartal. 1996. Probabilistic approximation of metric spaces and its algorithmic applications. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 184–193.

[16] Yair Bartal. 1998. On approximating arbitrary metrices by tree metrics. In *Annual ACM Symposium on Theory of Computing (STOC)*. 161–168.

[17] Yair Bartal, Avrim Blum, Carl Burch, and Andrew Tomkins. 1997. A polylog (n)-competitive algorithm for metrical task systems. In *Annual ACM Symposium on Theory of Computing (STOC)*. 711–719.

[18] Yair Bartal, Nova Fandina, and Seeun William Umboh. 2020. Online probabilistic metric embedding: a general framework for bypassing inherent bounds. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1538–1557.

[19] Piotr Berman and Chris Coulston. 1997. On-line algorithms for Steiner tree problems. In *Annual ACM Symposium on Theory of Computing (STOC)*. 344–353.

[20] Guy E. Blelloch, Yan Gu, and Yihan Sun. 2017. Efficient Construction of Probabilistic Tree Embeddings. In *International Colloquium on Automata, Languages and Programming (ICALP)*, Vol. 80. 26:1–26:14.

[21] Andreas Bley, S Mehdi Hashemi, and Mohsen Rezapour. 2013. IP modeling of the survivable hop constrained connected facility location problem. *Electronic Notes in Discrete Mathematics* 41 (2013), 463–470.

[22] Quentin Botton, Bernard Fortz, and Luis Gouveia. 2015. On the hop-constrained survivable network design problem with reliable edges. *Computers & Operations Research* 64 (2015), 159–167.

[23] Quentin Botton, Bernard Fortz, Luis Gouveia, and Michael Poss. 2013. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS journal on computing* 25, 1 (2013), 13–26.

[24] Shiri Chechik and Tianyi Zhang. 2020. Dynamic low-stretch spanning trees in subpolynomial time. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 463–475.

[25] Jérôme De Boeck and Bernard Fortz. 2018. Extended formulation for hop constrained distribution network configuration problems. *European Journal of Operational Research* 265, 2 (2018), 488–502.

[26] Ibrahima Diarrassouba, Virginie Gabrel, Ali Ridha Mahjoub, Luís Gouveia, and Pierre Pesneau. 2016. Integer programming formulations for the k-edge-connected 3-hop-constrained network design problem. *Networks* 67, 2 (2016), 148–169.

[27] Ibrahima Diarrassouba, Meriem Mahjoub, A Ridha Mahjoub, and Hande Yaman. 2018. k-node-disjoint hop-constrained survivable networks: polyhedral analysis and branch and cut. *Annals of Telecommunications* 73, 1-2 (2018), 5–28.

[28] Michael Dinitz, Guy Kortsarz, and Ran Raz. 2012. Min-Rep instances with large supergirth and the hardness of approximating basic spanners. In *International Colloquium on Automata, Languages and Programming (ICALP)*.

[29] Michael Dinitz, Guy Kortsarz, and Ran Raz. 2015. Label cover instances with large girth and the hardness of approximating basic k-spanner. *ACM Transactions on Algorithms (TALG)* 12, 2 (2015), 1–16.

[30] Michael Dinitz and Zeyu Zhang. 2016. Approximating low-stretch spanners. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 821–840.

[31] Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. 2008. Lower-stretch spanning trees. *SIAM J. Comput.* 38, 2 (2008), 608–628.

[32] Michael Elkin and David Peleg. 2000. Strong inapproximability of the basic k-spanner problem. In *International Colloquium on Automata, Languages and Programming (ICALP)*. Springer, 636–648.

[33] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2004. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.* 69, 3 (2004), 485–497.

[34] Sebastian Forster, Gramoz Goranci, and Monika Henzinger. 2020. Dynamic Maintanance of Low-Stretch Probabilistic Tree Embeddings with Applications. *arXiv preprint arXiv:2004.10319* (2020).

[35] Naveen Garg, Goran Konjevod, and R Ravi. 2000. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms* 37, 1 (2000), 66–84.

[36] Luis Gouveia. 1995. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Computers & Operations Research* 22, 9 (1995), 959–970.

[37] Luis Gouveia. 1996. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research* 95, 1 (1996), 178–190.

[38] Luis Gouveia and Thomas L Magnanti. 2003. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks: An International Journal* 41, 3 (2003), 159–173.

[39] Luis Gouveia and Cristina Requejo. 2001. A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem. *European Journal of Operational Research* 132, 3 (2001), 539–552.

[40] Anupam Gupta, Mohammad T Hajiaghayi, and Harald Räcke. 2006. Oblivious network design. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 970–979.

[41] Anupam Gupta, Robert Krauthgamer, and James R Lee. 2003. Bounded geometries, fractals, and low-distortion embeddings. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 534–543.

[42] Mohammad Taghi Hajiaghayi, Guy Kortsarz, and Mohammad R Salavatipour. 2009. Approximating buy-at-bulk and shallow-light k-Steiner trees. *Algorithmica* 53, 1 (2009), 89–103.

[43] Erez Kantor and David Peleg. 2009. Approximate hierarchical facility location and applications to the bounded depth Steiner tree and range assignment problems. *Journal of Discrete Algorithms* 7, 3 (2009), 341–362.

[44] Richard M Karp. 1989. A 2k-competitive algorithm for the circle. *Manuscript, August* 5 (1989).

[45] Maleq Khan, Fabian Kuhn, Dahlia Malkhi, Gopal Pandurangan, and Kunal Talwar. 2012. Efficient distributed approximation algorithms via probabilistic tree embeddings. *Distributed Computing* 25, 3 (2012), 189–205.

[46] M Reza Khani and Mohammad R Salavatipour. 2011. Improved approximations for buy-at-bulk and shallow-light k-Steiner trees and (k, 2)-subgraph. In *Annual International Symposium on Algorithms and Computation (ISAAC)*. Springer, 20–29.

[47] Jochen Könemann, Asaf Levin, and Amitabh Sinha. 2005. Approximating the degree-bounded minimum diameter spanning tree problem. *Algorithmica* 41, 2 (2005), 117–129.

[48] Goran Konjevod, R Ravi, and F Sibel Salman. 2001. On approximating planar metrics by tree metrics. *Information Processing Letters (IPL)* 80, 4 (2001), 213–219.

[49] Guy Kortsarz and David Peleg. 1997. Approximating shallow-light trees. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 103–110.

[50] Ioannis Koutis, Gary L Miller, and Richard Peng. 2011. A nearly-m log n time solver for sdd linear systems. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 590–598.

[51] Markus Leitner. 2016. Layered graph models and exact algorithms for the generalized hop-constrained minimum spanning tree problem. *Computers & Operations Research* 65 (2016), 1–18.

[52] Madhav V Marathe, Ramamoorthi Ravi, Ravi Sundaram, SS Ravi, Daniel J Rosenkrantz, and Harry B Hunt III. 1998. Bicriteria network design problems. *Journal of algorithms* 28, 1 (1998), 142–171.

[53] Manor Mendel and Assaf Naor. 2006. Ramsey partitions and proximity data structures. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 109–118.

[54] Viswanath Nagarajan and Ramamoorthi Ravi. 2005. Approximation algorithms for requirement cut on graphs. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 209–220.

[55] Assaf Naor and Terence Tao. 2012. Scale-oblivious metric fragmentation and the nonlinear Dvoretzky theorem. *Israel Journal of Mathematics* 192, 1 (2012), 489–504.

[56] Joseph Naor, Debmalya Panigrahi, and Mohit Singh. 2011. Online node-weighted steiner tree and related problems. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 210–219.

[57] Joseph Naor and Baruch Schieber. 1997. Improved approximations for shallow-light spanning trees. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 536–541.

[58] Harald Racke. 2002. Minimizing congestion in general networks. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 43–52.

[59] R Ravi. 1994. Rapid rumor ramification: Approximating the minimum broadcast time. In *Symposium on Foundations of Computer Science (FOCS)*. IEEE, 202–213.

[60] André Rossi, Alexis Aubry, and Mireille Jacomino. 2012. Connectivity-and-hop-constrained design of electricity distribution networks. *European journal of operational research* 218, 1 (2012), 48–57.

[61] Babacar Thiongane, Jean-François Cordeau, and Bernard Gendron. 2015. Formulations for the nonbifurcated hop-constrained multicommodity capacitated fixed-charge network design problem. *Computers & Operations Research* 53 (2015), 1–8.

[62] Stefan Voß. 1999. The Steiner tree problem with hop constraints. *Annals of Operations Research* 86 (1999), 321–345.

[63] Kathleen A Woolston and Susan L Albin. 1988. The design of centralized networks with reliability and availability constraints. *Computers & Operations Research* 15, 3 (1988), 207–217.