# Adaptive-Adversary-Robust Algorithms via Small Copy Tree Embeddings

**Bernhard Haepler**
Carnegie Mellon University, Pittsburgh, PA, USA
ETH Zürich, Switzerland

**D. Ellis Hershkowitz**
Carnegie Mellon University, Pittsburgh, PA, USA

**Goran Zuzic**
ETH Zürich, Switzerland

## Abstract

Embeddings of graphs into distributions of trees that preserve distances in expectation are a cornerstone of many optimization algorithms. Unfortunately, online or dynamic algorithms which use these embeddings seem inherently randomized and ill-suited against adaptive adversaries.

In this paper we provide a new tree embedding which addresses these issues by *deterministically* embedding a graph into a single tree containing $O(\log n)$ copies of each vertex while preserving the connectivity structure of every subgraph and $O(\log^2 n)$-approximating the cost of every subgraph. Using this embedding we obtain the first deterministic bicriteria approximation algorithm for the online covering Steiner problem as well as the first poly-log approximations for demand-robust Steiner forest, group Steiner tree and group Steiner forest.

**2012 ACM Subject Classification** Mathematics of computing → Trees; Mathematics of computing → Approximation algorithms; Mathematics of computing → Paths and connectivity problems; Theory of computation → Random projections and metric embeddings

**Keywords and phrases** Tree metrics, metric embeddings, approximation algorithms, group Steiner forest, group Steiner tree, demand-robust algorithms, online algorithms, deterministic algorithms

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2022.63

**Related Version** *Full Version*: https://arxiv.org/pdf/2102.05168.pdf

**Funding** 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) and Swiss National Foundation (project grant 200021-184735).
*D. Ellis Hershkowitz*: Supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0080.

## 1 Introduction

Probabilistic embedding of general metrics into distributions over trees are one of the most versatile tools in combinatorial and network optimization. The beauty and utility of these tree embeddings comes from the fact that their application is often simple, yet extremely powerful. Indeed, when modeling a network with length, costs, or capacities as a weighted graph, these embeddings often allow one to pretend that the graph is a tree. A common template for countless network design algorithms is to (1) embed the input weighted graph $G$ into a randomly sampled tree $T$ that approximately preserves the weight structure of $G$; (2) solve the input problem on $T$ and; (3) project the solution on $T$ back into $G$.


30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 63; pp. 63:1–63:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A long and celebrated line of work [32, 3, 6, 17] culminated in the embedding of Fakcharoenphol, Rao and Talwar [17] – henceforth the "FRT embedding" – which showed that any weighted graph on $n$ nodes can be embedded into a distribution over weighted trees in a way that $O(\log n)$-approximately preserves distances in expectation. Together with the above template this reduces many graph problems to much easier problems on trees at the cost of an $O(\log n)$ approximation factor. This has lead to a myriad of approximation, online, and dynamic algorithms with poly-logarithmic approximations and competitive ratios for NP-hard problems such as for $k$-server [5], metrical task systems [8], group Steiner tree and group Steiner forest [2, 36, 21], buy-at-bulk network design [4] and (oblivious) routing [37]. For many of these problems tree embeddings are the only known way of obtaining such algorithms on general graphs.

However, probabilistic tree embeddings have one drawback: Algorithms based on them naturally require randomization and their approximation guarantees only hold in expectation. For approximation algorithms – i.e., in the offline setting – there are derandomization tools, such as the FRT derandomizations given in [12, 17, 7], to overcome these issues. These derandomization results are so general that essentially any offline algorithm based on tree embeddings can be transformed into a deterministic algorithm with matching approximation guarantees (with only a moderate increase in running time). Unfortunately, these strategies are not applicable to online or dynamic settings where an adversary progressively reveals the input. Indeed, most online and dynamic algorithms that use FRT are randomized (e.g. [23, 28, 2, 19, 8, 36, 14, 15]).

This overwhelming evidence in the literature is driven by a well-known and fundamental barrier to the use of probabilistic tree embeddings in deterministic online and dynamic algorithms. More specifically and even worse, this is a barrier which prevents these algorithms from working against all but the weakest type of adversary. In particular, designing an online or dynamic algorithm which is robust to an oblivious adversary (which fixes all requests in advance, independently of the algorithm's randomness) is often much easier than designing an algorithm which is robust to an adaptive adversary (which chooses the next request based on the algorithm's current solution). As the actions of a deterministic algorithm can be fully predicted this distinction only holds for randomized algorithms – any deterministic algorithm has to always work against an adaptive adversary. For these reasons, many online and dynamic algorithms have exponentially worse competitive ratios in the deterministic or adaptive adversary setting than in the oblivious adversary setting. This is independent of computational complexity considerations.

The above barrier results from a repeatedly recognized and seemingly unavoidable phenomenon which prevents online algorithms built on FRT from working against adaptive adversaries. Specifically, there are graphs where every tree embedding must have many node pairs with polynomially-stretched distances [6]. There is nothing that prevents an adversary then from learning through the online algorithm's responses which tree was sampled and then tailoring the remainder of the online instance to pairs of nodes that have highly stretched distances. The exact same phenomenon occurs in the dynamic setting; see, for example, [23] and [28] for dynamic algorithms with expected cost guarantees that only hold against oblivious adversaries because they are based on FRT. In summary, online and dynamic algorithms that use probabilistic tree embeddings seem inherently randomized and seem to necessarily only work against adversaries oblivious to this randomness.

Similar, albeit not identical,[1] issues also arise in other settings, most notably demand-robust optimization. The demand-robust model is a well-studied model of optimization under uncertainty [13, 30, 18, 25, 26, 22] in which an algorithm first buys a partial solution given a large collection of potential problem instances. An "adaptive adversary" then chooses which of the potential instances must be solved and the algorithm must extend its partial solution to solve the selected instance at inflated costs. The adversary is adaptive in the sense that it chooses the final instance with full knowledge of the algorithm's partial solution. To thwart an algorithm which reduces a demand-robust problem to its tree version via a sampled FRT tree, the adversary can present a collection of potential instances which for every tree $T$ in the FRT distribution contains an instance for which $T$ is an arbitrarily bad approximation and then always choose the worst-case problem instance. The fact that there do not exist any demand-robust algorithms which use FRT despite this setting having received considerable attention seems at least partially due to the issues pointed out here.

Overall it seems fair to say that prior to this work tree embeddings seemed fundamentally incapable of enabling adaptive-adversary-robust and deterministic algorithms in several well-studied settings.

## 1.1 Our Contributions

We provide a new type of metric embedding – the copy tree embedding – which is deterministic and therefore also adaptive-adversary-robust. Specifically, we show that any weighted graph $G$ can be deterministically embedded into a single weighted tree with a small number of copies for each vertex. Any subgraph of $G$ will project onto this tree in a connectivity and approximate-cost preserving way.

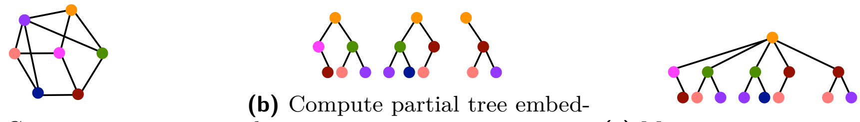
To precisely define our embeddings we define a copy mapping $\phi$ which maps a vertex $v$ to its copies.

▶ **Definition 1** (Copy Mapping). *Given vertex sets $V$ and $V'$ we say $\phi : V \to 2^{V'}$ is a copy mapping if every node has at least one copy (i.e. $|\phi(v)| \geq 1$ for all $v \in V$), copies are disjoint (i.e. $\phi(v) \cap \phi(u) = \emptyset$ for $u \neq v$) and every node in $V'$ is a copy of some node (i.e. for every $v' \in V'$ there is some $v \in V$ where $v' \in \phi(v)$). For $v' \in V'$, we use the shorthand $\phi^{-1}(v')$ to stand for the unique $v \in V$ such that $v' \in \phi(v)$.*

A copy tree embedding for a weighted graph $G$ now simply consists of a tree $T$ on copies of vertices of $G$ with one distinguished root and two mappings $\pi_{G\to T}$ and $\pi_{T\to G}$ which map subsets of edges from $G$ to $T$ and from $T$ to $G$ in a way that preserves connectivity and approximately preserves costs. We say that *two vertex subsets $U, W$ are connected* in a graph if there is a $u \in U$ and $w \in W$ such that $u$ and $w$ are connected. We also say that a mapping $\pi : 2^E \to 2^{E'}$ is *monotone* if for every $A \subseteq B$ we have that $\pi(A) \subseteq \pi(B)$. A rooted tree $T = (V, E, w)$ is *well-separated* if for all edges $e$ if $e'$ is a child edge of $e$ in $T$ then $w(e') \leq \frac{1}{2}w(e)$. In the below for $F \subseteq E$ we let $w(F) := \sum_{e\in F} w(e)$.

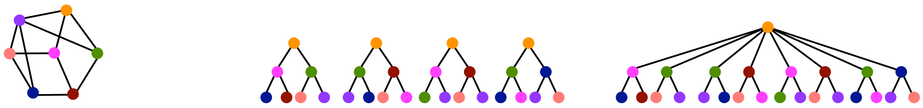
▶ **Definition 2** ($\alpha$-Approximate Copy Tree Embedding with Copy Number $\chi$). *Let $G = (V, E, w)$ be a weighted graph with some distinguished vertex $r \in V$ called the root. An $\alpha$-approximate copy tree embedding with copy number $\chi$ consists of a weighted rooted tree $T = (V', E', w')$, a copy mapping $\phi : V \to 2^{V'}$ and edge mapping functions $\pi_{G\to T} : 2^E \to 2^{E'}$ and $\pi_{T\to G} : 2^{E'} \to 2^E$ where $\pi_{T\to G}$ is monotone and:*

[1] We remark that, unlike the online and dynamic setting, the barrier to obtaining demand-robust algorithms which work against the "adaptive adversary" implicit in the setting is merely computational and thus seems potentially less inherent.



**(a)** Graph $G$. **(b)** Compute partial tree embeddings. **(c)** Merge trees.

**Figure 1** Illustration of our first construction where we merge $O(\log n)$ partial tree embeddings.



**(a)** Graph $G$. **(b)** Enumerate FRT support. **(c)** Merge trees.

**Figure 2** Illustration of our second construction where we merge the $O(n \log n)$ trees in the FRT support.

1. ***Connectivity Preservation:*** *For all $F \subseteq E$ and $u, v \in V$ if $u, v$ are connected by $F$, then $\phi(u), \phi(v) \subseteq V'$ are connected by $\pi_{G \to T}(F)$. Symmetrically, for all $F' \subseteq E'$ and $u', v' \in V'$ if $u'$ and $v'$ are connected by $F'$ then $\phi^{-1}(u')$ and $\phi^{-1}(v')$ are connected by $\pi_{T \to G}(F')$.*
2. ***$\alpha$-Cost Preservation:*** *For any $F \subseteq E$ we have $w(F) \leq \alpha \cdot w'(\pi_{G \to T}(F))$ and for any $F' \subseteq E'$ we have $w'(F') \leq w(\pi_{T \to G}(F'))$.*
3. ***Copy Number:*** *$|\phi(v)| \leq \chi$ for all $v \in V$ and $\phi(r) = \{r'\}$ where $r'$ is the root of $T$.*

*A copy tree embedding is efficient if $T$, $\phi$, and $\pi_{T \to G}$ are deterministically poly-time computable and well-separated if $T$ is well-separated.*

We emphasize that, whereas standard tree embeddings guarantee costs are preserved in expectation, our copy tree embeddings preserve costs deterministically. Also notice that for efficient copy tree embeddings we do not require that $\pi_{G \to T}$ is efficiently computable; this is because $\pi_{G \to T}$ will be used in our analyses but not in any of our algorithms. The idea of embeddings which map vertices to several copies has previously been explored by [9] and was recently explored in a concurrent work of [20]. The key difference between these works and our own is that the number of copies that each vertex is mapped to is unboundedly large (in the case of [9]) or only small in expectation (in the case of [20]). On the other hand, the analogue of $\alpha$-cost preservation in [9] ("path preservation") is stronger than our $\alpha$-cost preservation.

We first give two copy tree embedding constructions which trade off between the number of copies and cost preservation. Both constructions are based on the idea of merging appropriately chosen tree embeddings as pictured in Figure 1 and Figure 2 where we color nodes according to the node whose copy they are.

**Construction 1: Merging Partial Tree Embeddings (full version).** The cornerstone of our first construction is the idea of merging embeddings which give good *deterministic* distance preservation. If our goal is to embed the entire input metric into a tree this is impossible. However, it is possible to embed a random constant fraction of nodes in an input metric into a tree in a way that deterministically preserves distances of the embedded nodes; an embedding which we call a "partial tree embedding" (see also [24, 29]). We then use the method of conditional expectation to derandomize a node-weighted version of this random process and apply this derandomization $O(\log n)$ times, down-weighting nodes as they are embedded. The result of this process is $O(\log n)$ partial tree embeddings where a multiplicative-weights-type argument shows that each node appears in a constant fraction of these embeddings. Merging these $O(\log n)$ embeddings gives our copy tree while an Euler-tour-type proof shows that subgraphs of the input graph can be mapped to our copy tree in a cost and connectivity-preserving fashion. The following theorem summarizes our first construction.

▶ **Theorem 3.** *There is a poly-time deterministic algorithm which given any weighted graph $G = (V, E, w)$ and root $r \in V$ computes an efficient and well-separated $O(\log^2 n)$-approximate copy tree embedding with copy number $O(\log n)$.*

**Construction 2: Merging FRT Support (full version).** Our second construction follows from a known fact that the size of the support of the FRT distribution can be made $O(n \log n)$ and this support can be computed deterministically in poly-time [12]. Merging each tree in this support at the root and some simple probabilistic method arguments give a copy tree embedding that is $O(\log n)$-cost preserving but with an $O(n \log n)$ copy number. Equivalently, it can be inferred from [9]. The next theorem summarizes this construction.

▶ **Theorem 4.** *There is a poly-time deterministic algorithm which given any weighted graph $G = (V, E, w)$ and root $r \in V$ computes an efficient and well-separated $O(\log n)$-approximate copy tree embedding with copy number $O(n \log n)$.*

While our second construction achieves a slightly better cost bound than our first construction, it has the significant downside of a linear copy number. Notably, this linear copy number makes our second construction unsuitable for some applications, including, for example, our second application as described below. Moreover, our first construction also has several desirable properties which our second does not which we expect might be useful for future applications. These include: (1) $\pi_{G\to T}$ is monotone (in addition to $\pi_{T\to G}$ being monotone as stipulated by Definition 2); (2) if $u$ and $v$ are connected by $F \subseteq E$ then $\Omega(\log n)$ vertices of $\phi(u)$ are connected to $\Omega(\log n)$ vertices of $\phi(v)$ in $\pi_{G\to T}(F)$ (as opposed to just one vertex of $\phi(u)$ and one vertex of $\phi(v)$ as in Definition 2) and; (3) if $u$ is connected to $r$ by $F \subseteq E$ then every vertex in $\phi(u)$ is connected to $\phi(r)$ in $\pi_{G\to T}(F)$ (as opposed to just one vertex of $\phi(u)$ as in Definition 2).

We next apply our constructions to obtain new results for several online and demand-robust connectivity problems whose history we briefly summarize now. Group Steiner tree and group Steiner forest are two well-studied generalizations of set cover and Steiner tree. In the group Steiner tree problem, we are given a weighted graph $G = (V, E, w)$ and groups $g_1, \ldots, g_k \subseteq V$ and must return a subgraph of $G$ of minimum weight which contains at least one vertex from each group. The group Steiner forest problem generalizes group Steiner tree. Here, we are given $A_i, B_i \subseteq V$ pairs and for each $i$ we must connect some vertex from $A_i$ to some vertex in $B_i$. [2] and [36] each gave a poly-log approximation for online group Steiner tree and forest respectively but both of these approximation guarantees are randomized and only hold against oblivious adversaries because they rely on FRT. Indeed, [2] posed the existence of a deterministic poly-log approximation for online group Steiner tree as an open question which has since been restated several times [11, 10].

Another well-studied generalization of group Steiner tree is the covering Steiner problem which is defined as group Steiner tree but where we are additionally given a value $r_i \in (0, |g_i|]$ for each group $g_i$ and must connected at least an $r_i$ vertices of $g_i$ in our subgraph. This problem was introduced by [35] and further studied in several follow-up works [16, 27].

Similarly, while demand-robust minimum spanning tree and special cases of demand-robust Steiner tree have received considerable attention [13, 34, 33], there are no known poly-log approximations for demand-robust Steiner tree, group Steiner tree or group Steiner forest.

The reason our embeddings are well-suited to group Steiner problems and its generalizations is that mapping it onto a copy tree embedding simply results in another instance of the group Steiner tree problem, this time on a tree. Indeed, our embeddings almost immediately reduce the open question of [2] – solving online group Steiner tree and forest deterministically on a general graph – to its tree case (see the full version for details). Equivalently, this reduction be inferred from the embeddings [9], though [2] seems to have overlooked this connection.

**Application 1: Deterministic Online Covering Steiner (Section 3).** In our first application we make progress on the open question of [2] by showing that the online covering Steiner problem admits a *bicriteria* deterministic poly-log approximation. Specifically, note that the covering Steiner problem generalizes group Steiner tree but unlike group Steiner tree it admits a natural bicriteria relaxation: instead of connecting, for example, $\frac{1}{2}$ of the nodes in each group we could require that our algorithm only connects, say, $\frac{(1-\epsilon)}{2}$ of all nodes in each group for some $\epsilon > 0$. Thus, our result can be seen as showing that there is indeed a deterministic poly-log competitive algorithm for online group Steiner tree – as posed in the above open question of [2] – *provided the algorithm can be bicriteria* in the relevant sense. We use our embeddings for this application. Those of [9] or [20] are not suitable for our first application since this application requires a bound on the number of copies of each vertex. More formally, we obtain a deterministic poly-log bicriteria approximation for this problem which connects at least $\frac{1-\epsilon}{2}$ of the nodes in each group (notated "$(1-\epsilon)$-connection competitive" below) by using our copy tree embeddings and a "water-filling" algorithm to solve the tree case.

▶ **Theorem 5.** *There is a deterministic poly-time algorithm for online covering Steiner (on general graphs) which is $O(\frac{\log^3 n}{\epsilon} \cdot \max_i \frac{|g_i|}{r_i})$-cost-competitive and $(1-\epsilon)$-connection-competitive.*

As we later observe, providing a deterministic poly-log-competitive algorithm for the online covering Steiner problem with any constant bicriteria relaxation is strictly harder than providing a deterministic poly-log-competitive algorithm for online (non-group) Steiner tree. Thus, this result also generalizes the fact that a deterministic poly-log approximation is known for online (non-group) Steiner tree [31]. Additionally, as a corollary we obtain the first non-trivial deterministic approximation algorithm for online group Steiner tree – albeit one with a linear dependence on the maximum group size.[2]

▶ **Corollary 6.** *There is an $O(N \log^3 n)$-competitive deterministic algorithm for online group Steiner tree where $N := \max_i |g_i|$ is the maximum group size.*

We next adapt and apply our embeddings in the demand-robust setting.

**Application 2: Demand-Robust Steiner Problems (full version).** We begin by generalizing copy tree embeddings to demand-robust copy tree embeddings. Roughly, these are copy tree embeddings which simultaneously work well for every possible demand-robust scenario.

[2] We explicitly note here that this bicriteria guarantee does not yield a solution to the open problem of [2] of finding a poly-log deterministic approximation to the online group Steiner tree problem.

We then adapt our analysis from our previous constructions to show that these copy tree embeddings exist. Lastly, we apply demand-robust copy tree embeddings to give poly-log approximations for the demand-robust versions of several Steiner problems – Steiner forest, group Steiner tree and group Steiner forest – for which, prior to this work, nearly nothing was known. In particular, the only non-trivial algorithms known for demand-robust Steiner problems prior to this work are an algorithm for demand-robust Steiner tree [13] and an algorithm for demand-robust Steiner forest *on trees* with exponential scenarios [18] (which is, in general, incomparable to the usual demand-robust setting). To show these results, we apply our demand-robust copy tree embeddings to reduce these problems to their tree case. Thus, we also give our results on trees which are themselves non-trivial.

▶ **Theorem 7.** *There is a randomized poly-time $O(\log^2 n)$-approximation algorithm for the demand-robust group Steiner tree problem on weighted trees.*

▶ **Theorem 8.** *There is a randomized poly-time $O(D \cdot \log^3 n)$-approximation algorithm for the demand-robust group Steiner forest problem on weighted trees of depth $D$.*

▶ **Theorem 9.** *There is a randomized poly-time $O(\log^4 n)$-approximation algorithm for the demand-robust group Steiner tree problem on weighted graphs.*

▶ **Theorem 10.** *There is a randomized poly-time $O(\log^6 n)$-approximation for the demand-robust group Steiner forest problem on weighted graphs with polynomially-bounded aspect ratio.*

Demand-robust group Steiner forest generalizes demand-robust Steiner forest and prior to this work no poly-log approximations were known for demand-robust Steiner forest; thus the above result gives the first poly-log approximation for demand-robust Steiner forest. We solve the tree case of the above problems by observing a connection between demand-robust and online algorithms. In particular, we exploit the fact that for certain online rounding schemes a demand-robust problem can be seen as an online problem with two time steps provided certain natural properties are met. Notably, these properties will be met for these problems *on trees*. Thus, we emphasize that going through the copy tree embedding is crucial for our application – a more direct approach of using online rounding schemes on the general problem does not seem to yield useful results.

**Further Applications.** Lastly, we note that copy tree embeddings were integral to another recent work of [29], who gave the first poly-log approximations for the hop-constrained version of many classic network design problems, including hop-constrained Steiner forest [1], group Steiner tree and buy-at-bulk network design [4].

## 2 Graph Notation And Assumptions

Throughout this paper we will work with weighted graphs of the form $G = (V, E, w)$ where $V$ and $E$ are the vertex and edge sets of $G$ and $w : E \to \mathbb{R}_{\geq 1}$ gives the weight of edges. We typically assume that $n := |V|$ is the number of nodes and write $[n] = \{1, 2, \dots, n\}$. We will also use $V(G)$, $E(G)$ and $w_G$ to stand for the vertex set, edge set and weight function of $G$. Similarly, we will use $w_e$ to stand for $w(e)$ where convenient. For a subset of edges $F \subseteq E$, we use the notation $w(F) := \sum_{e \in F} w_G(e)$. We use $d_G : V \times V \to \mathbb{R}_{\geq 0}$ to give the shortest path metric according to $w$. We will talk about the diameter of a metric $(V, d)$ which is $\max_{u,v \in V} d(u, v)$; we notate the diameter with $D$. We use $B(v, x) := \{u \in V : d(v, u) \leq x\}$ to stand for the closed ball of $v$ of radius $x$ in metric $(V, d)$ and $B_G(v, x)$ if $(V, d)$ is the shortest path metric of $G$ and we need to disambiguate which graph we are taking balls with respect to. We will sometimes identify a graph with the metric which it induces.

Notice that we have assumed that edge weights are non-zero and at least 1. This will be without loss generality as for our purposes any 0 weight edges may be contracted and scaling of edge weights ensures that the minimum edge weight is at least 1.

# 3 Online Covering Steiner

In this section we give a deterministic bicriteria algorithm for the online covering Steiner problem which is the same as online group Steiner tree but where we must connect at least $r_i$ vertices from each group $g_i$ to the root. The algorithm is bicriteria in the sense that it relaxes both the $r_i$-connectivity guarantee and the cost.

As mentioned in the introduction, this problem generalizes group Steiner tree. Moreover, it is also easy to see that any deterministic bicriteria algorithm for online covering Steiner also gives a poly-log-competitive deterministic (unicriteria) algorithm for online (non-group) Steiner tree. In particular, given an instance of Steiner tree on weighted graph $G = (V, E, w)$ with root $r$ where we must connect terminals $A \subseteq V$ to $r$, it suffices to solve the covering Steiner problem where each vertex in $A$ is in a singleton group with any constant bicriteria relaxation. This is because connecting any $c > 0$ fraction of each group to $r$ will connect at least one vertex to $r$ by the integrality of the number of connected vertices. Thus, our result generalizes the fact that deterministic poly-log approximations are known for online (non-group) Steiner tree [31]. However, we do note that our (deterministic) poly-log-approximate bicriteria online covering Steiner problem algorithm does not imply there is a (deterministic) poly-log-approximate online (non-partial) group Steiner tree algorithm (due to the nature of the bicriteria guarantee).

**Offline Covering Steiner Problem.** In the covering Steiner problem we are given a weighted graph $G = (V, E, w)$ as well as pairwise disjoint groups $g_1, g_2, \ldots, g_k \subseteq V$, desired connected vertices $1 \le r_i \le |g_i|$ for each group $g_i$ and root $r \in V$. Our goal is to find a tree $T$ rooted at $r$ which is a subgraph of $G$ and satisfies $|T \cap g_i| \ge r_i$ for every $i$. We wish to minimize our cost, $w(T) := \sum_{e \in E(T)} w(e)$.[3]

**Online Covering Steiner Problem.** The online covering Steiner problem is the same as offline covering Steiner problem but where our solution need not be a tree and groups are revealed in time steps $t = 1, 2, \ldots$. That is, in time step $t$ an adversary reveals a new group $g_t$ and the algorithm must maintain a solution $T_t$ where: (1) $T_{t-1} \subseteq T_t$; (2) $T_t$ is feasible for the (offline) covering Steiner tree problem on groups $g_1, \ldots g_t$ and; (3) $T_t$ is cost-competitive with the optimal offline solution for this problem where the cost-competitive ratio of our algorithm is $\max_t w(T_t)/\mathrm{OPT}_t$ where $\mathrm{OPT}_t$ is the cost of the optimal offline covering Steiner problem solution on the first $t$ groups. We will give a bicriteria approximation for online covering Steiner; thus we say that an online solution is $\rho$-connection-competitive if for each $t$ we have $|T_t \cap g_i| \ge r_i \cdot \rho$ for every $i \le t$.

## 3.1 Online Covering Steiner on a Tree

We begin by giving a bicriteria deterministic online algorithm for covering Steiner on trees based on a "water-filling" approach. Informally, in iteration $t$ each unconnected vertex in each group will grow the solution towards the root at an equal rate until at least $r_i \cdot (1 - \epsilon)$ vertices in $g_t$ are connected to $r$.

[3] As with group Steiner tree the assumption that the tree is rooted and that the groups are pairwise disjoint is without loss of generality.

### 3.1.1 Problem

More formally we will solve a problem which is a slight generalization of covering Steiner on trees. We solve this problem on a tree rather than just covering Steiner on a tree because, unlike group Steiner tree, the "groupified" version of covering Steiner is not necessarily another instance of covering Steiner. Roughly, instead of groups we now have groups of groups, hence we call this problem 2-level covering Steiner.

**Offline 2-Level Covering Steiner Problem.** In the 2-level covering Steiner problem we are given a weighted graph $G = (V, E, w)$, root $r \in V$ and groups of groups $\mathcal{G}_1, \dots \mathcal{G}_k$ where $\mathcal{G}_i$ consists of groups $\{g_1^{(i)}, \dots g_{k_i}^{(i)}\}$ where each $g_j^{(i)} \subseteq V$. We are also given connectivity requirements $r_1, \dots, r_k$. Our goal is to compute a minimum-weight tree $T$ containing $r$ where for each $i \leq k$ we have $|\{g_j^{(i)} : g_j^{(i)} \cap T \neq \emptyset\}| \geq r_i$. We let $n_i := |\{v : \exists j \text{ s.t. } v \in g_j^{(i)}\}|$. Notice that covering Steiner is just 2-level covering Steiner where each $g_i^{(j)}$ is a singleton set.

**Online 2-Level Covering Steiner Problem.** Online 2-level covering Steiner is the same as the offline problem but where $\mathcal{G}_t$ is revealed in time step $t$ by an adversary. In particular, for each time step $t$ we must maintain a solution $T_t$ where: (1) $T_{t-1} \subseteq T_t$ for all $t$; (2) $T_t$ is feasible for the (offline) 2-level covering Steiner problem on $\mathcal{G}_1, \dots, \mathcal{G}_t$ with connectivity requirements $r_1, \dots, r_t$ and; (3) $T_t$ is cost-competitive with the optimal offline solution for this problem where the cost-competitive ratio of our algorithm is $\max_t w(T_t)/\text{OPT}_t$ where $\text{OPT}_t$ is the cost of the optimal offline 2-level covering Steiner problem solution on the first $t$ groups of groups.

We will give a bicriteria approximation for online 2-level covering Steiner problem on trees; thus we say that an online solution is $\rho$-connection-competitive if for each $t$ we have $|\{g_j^{(i)} : g_j^{(i)} \cap T \neq \emptyset\}| \geq \rho \cdot r_i$ for every $i \leq t$.

### 3.1.2 Algorithm

We now formally describe our algorithm for the 2-level covering Steiner problem on weighted tree $T = (V, E, w)$ given an $\epsilon > 0$. We will maintain a fractional variable $0 \leq x_e \leq w_e$ for each edge indicating the extent to which we buy $e$ where our $x_e$s will be monotonically increasing as our algorithm runs. Say that an edge $e$ is saturated if $x_e = w_e$.

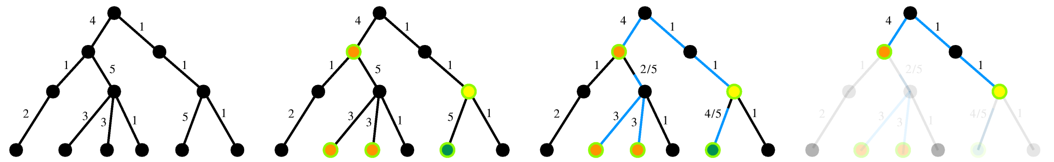
Let us describe how we update our solution in the $t$th time step. Let $T_t$ be the connected component of all saturated edges containing $r$. Then, we repeat the following until $|\{g_j^{(t)} : g_j^{(t)} \cap T_t \neq \emptyset\}| \geq r_t \cdot (1 - \epsilon)$. Let $\mathcal{G}'_t := \{g_j^{(t)} \in \mathcal{G}_t : g_j^{(t)} \cap T_t = \emptyset\}$ be all groups in $\mathcal{G}_t$ not yet connected and let $g'_t := \bigcup_{S \in \mathcal{G}'_t} S$ be all vertices in a group which have not yet been connected to $r$. We say that $e$ is on the frontier for $v \in g'_t$ if it is the first edge on the path from $v$ to $r$ which is not saturated. Similarly, let $r_e$ be the number of vertices in $g'_t$ for which $e$ is on the frontier for $v$. Then, for each edge $e$ we increase $x_e$ by $r_e \cdot \delta$ where $\delta = \min_e (w_e - x_e)/r_e$. Our solution in the $t$th time step is $T_t$ once $|\{g_j^{(t)} : g_j^{(t)} \cap T_t \neq \emptyset\}| \geq (1 - \epsilon) \cdot r_t$.

We illustrate one iteration of this algorithm in Figure 3.

### 3.1.3 Analysis

We proceed to analyze the above algorithm and give its properties.

▶ **Theorem 11.** *There is a deterministic poly-time algorithm for online 2-level covering Steiner on trees which is* $\frac{1}{\epsilon} \cdot (\max_i \frac{n_i}{r_i})$*-cost-competitive and* $(1 - \epsilon)$*-connection-competitive.*




**(a)** Graph $T$. **(b)** $\mathcal{G}_1$ arrives. **(c)** "Fill water." **(d)** Choose solution.

**Figure 3** Solution our algorithm gives after one group of groups, $\mathcal{G}_1$, is revealed where $r_1 = 2$. Nodes in groups in $\mathcal{G}_1$ outlined in green and nodes colored according to the group of $\mathcal{G}_1$ which contains them. Saturated edges given in blue and edges with $0 < x_e < w_e$ annoted with "$x_e/w_e$". All other edges labeled by $w_e$.

**Proof.** We begin by verifying that our algorithm returns a monotonically increasing and $(1-\epsilon)$-connection-competitive solution. First, notice that our solution is monotonically increasing since our $x_e$s are monotonically increasing and our solution only includes saturated edges. To see that our solution is $(1-\epsilon)$-connection-competitive notice that at least one new edge becomes saturated from each update to the $x_e$s (namely $\arg\min_e (w_e - x_e)/r_e$) and since if all edges are saturated then $T_t = T$ which clearly satisfies $|\{g_j^{(t)} : g_j^{(t)} \cap T_t \neq \emptyset\}| \geq (1-\epsilon)\cdot r_t$, this process will eventually halt with a $(1-\epsilon)$-connection-competitive solution in the $t$th iteration. For the same reason our algorithm is deterministic poly-time.

It remains to argue that our solution is $\frac{1}{\epsilon} \cdot (\max_i \frac{n_i}{r_i})$-cost-competitive. We will argue that we can uniquely charge each unit of increase of our $x_e$s to an appropriate cost portion of the optimal solution. Fix an iteration $t$. Next, let $\delta^{(i,j)}$ for $i \leq t$ be the value of $\delta$ in the $i$th iteration the $j$th time we increase the value of our $x_e$s. Similarly, let $\delta_x^{(i,j)}$ be the increase in $\sum_e x_e$ when we do so and let $\delta_y^{(i,j)}$ be the increase in $\sum_{e \in T_t^*} x_e$ where $T_t^*$ is the optimal offline solution to the 2-level covering Steiner problem we must solve in the $t$th iteration. Lastly, let $y := \sum_{i \leq t} \sum_j \delta_y^{(i,j)}$ be the value of $\sum_{e \in T_t^*} x_e$ at the end of the $t$th iteration; clearly we have $y \leq \text{OPT}_t$. We claim that it suffices to show that for each $i \leq t$ and each $j$ that $\delta_x^{(i,j)} \leq \frac{1}{\epsilon}\delta_y^{(i,j)} \frac{n_i}{r_i}$ since it would follow that at the end of iteration $t$ we have that

$$w(T_t) \leq \sum_e x_e = \sum_{i \leq t} \sum_j \delta_x^{(i,j)} \leq \frac{1}{\epsilon} \sum_{i \leq t} \sum_j \frac{n_i}{r_i} \delta_y^{(i,j)} \leq \frac{1}{\epsilon}\left(\max_i \frac{n_i}{r_i}\right) y \leq \frac{1}{\epsilon}\left(\max_i \frac{n_i}{r_i}\right) \text{OPT}_t.$$

We proceed to show that $\delta_x^{(i,j)} \leq \frac{1}{\epsilon}\delta_y^{(i,j)} \frac{n_i}{r_i}$ for each $i \leq t$ and $j$. We fix an $i$ and $j$ and for cleanliness of notation we will drop the dependence on $i$ and $j$ in our $\delta$s henceforth.

First, notice that we have that

$$\delta_x \leq n_i \cdot \delta \tag{1}$$

since each vertex $v \in g_i$ is uniquely responsible for up to a $\delta$ increase on $x_e$ where $e$ is the edge on $v$'s frontier.

On the other hand, notice that if a group in $\mathcal{G}_i$ is connected to $r$ by $T_t^*$ but is not yet connected by $T_i$ then such a group uniquely contributes at least $\delta$ to $\delta_y$. Since $T_t^*$ connects at least $r_i$ groups in $\mathcal{G}_i$ to $r$ but at the moment of our increase $T_i$ connects at most $(1-\epsilon)\cdot r_i$, there are at least $\epsilon \cdot r_i$ such groups in $\mathcal{G}_i$ which are connected to $r$ by $T_t^*$ but not by $T_i$. Thus, we have that

$$\delta_y \geq \epsilon \cdot r_i \cdot \delta \tag{2}$$

Combining Equations 1 and 2 shows $\delta_x \leq \frac{1}{\epsilon}\delta_y \frac{n_i}{r_i}$ as required. ◀

## 3.2 Online Covering Steiner on General Graphs

Next, we apply our first construction to give an algorithm for covering Steiner on general graphs. Crucially, the following result relies on a single copy tree embedding with polylogarithmic copy number, making our second construction unsuitable for this problem.

▶ **Theorem 5.** *There is a deterministic poly-time algorithm for online covering Steiner (on general graphs) which is* $O(\frac{\log^3 n}{\epsilon} \cdot \max_i \frac{|g_i|}{r_i})$*-cost-competitive and* $(1-\epsilon)$*-connection-competitive.*

**Proof.** We will use our copy tree embedding to produce a single tree on which we must deterministically solve online 2-level covering Steiner. We will then apply the algorithm from Theorem 11 to solve online 2-level covering Steiner on this tree.

More formally, consider an instance of online covering Steiner on weighted graph $G = (V, E, w)$ with root $r$. Then, we first compute a copy tree embedding $(T, \phi, \pi_{G\to T}, \pi_{T\to G})$ deterministically with respect to $G$ and $r$ as in Theorem 3 with cost approximation $O(\log^2 n)$ and copy number $O(\log n)$. Next, given our instance $I_t$ of covering Steiner on $G$ with groups $g_1, \dots g_t$ and connection requirements $r_1, \dots, r_t$ we let $I'_t$ be the instance of 2-level covering Steiner on $T$ with groups of groups $\mathcal{G}_1, \dots \mathcal{G}_t$ where $\mathcal{G}_i = \{\phi(v) : v \in g_i\}$, connection requirements $r_1, \dots, r_t$ and root $\phi(r)$. Then if the adversary has required that we solve instance $I_t$ in time step $t$, then we require that the algorithm in Theorem 11 solves $I'_t$ in time step $t$ and we let $H'_t$ be the solution returned by our algorithm for $I'_t$. Lastly, we return as our solution for $I_t$ in time step $t$ the set $H_t := \pi_{T\to G}(H'_t)$.

Let us verify that the resulting algorithm is indeed feasible (i.e. monotone and $(1-\epsilon)$-connection-competitive) and of the appropriate cost.

First, we have that $H_t \subseteq H_{t+1}$ for every $t$ since $H'_t \subseteq H'_{t+1}$ because our algorithm for trees returns a feasible solution for its online problem and $\pi_{T\to G}$ is monotone by definition of a copy tree embedding. Moreover, we claim that $H_t$ connects at least $(1-\epsilon)\cdot r_i$ vertices from $g_i$ to $r$ for $i \le t$ and every $t$. To see this, notice that there at least $(1-\epsilon)\cdot r_i$ groups from $\mathcal{G}_i$ containing a vertex connected to $r$ by $H'_t$. Since each such group consists of the copies of a distinct vertex, by the connectivity preservation properties of a copy tree it follows that $H_t$ connects at least $(1-\epsilon)\cdot r_i$ vertices from $g_i$ to $r$.

Next, we verify the cost of our solution. Let $\mathrm{OPT}'_t$ be the cost of the optimal solution to $I'_t$. Notice that since our copy number is $O(\log n)$, it follows that $n_i \le O(\log n \cdot |g_i|)$. Thus, by the guarantees of Theorem 11 we have

$$w_T(H'_t) \le \frac{1}{\epsilon} \cdot \left(\max_i \frac{n_i}{r_i}\right) \mathrm{OPT}'_t \le O\left(\frac{\log n}{\epsilon}\right) \cdot \left(\max_i \frac{|g_i|}{r_i}\right) \mathrm{OPT}'_t. \tag{3}$$

Next, we bound $\mathrm{OPT}'_t$. Let $H^*_t$ be the optimal solution to $I_t$. We claim that $\pi_{G\to T}(H^*_t)$ is feasible for $I'_t$. This follows because $H^*_t$ connects at least $r_i$ vertices from $g_i$ to $r$ for $i \le t$ and so by the connectivity preservation property of copy tree embeddings we know that there are at least $r_i$ groups in $\mathcal{G}_i$ with a vertex connected to $r$ by $\pi_{G\to T}(H^*_t)$. Thus, combining this with the $O(\log^2 n)$ cost preservation of our copy tree embedding we have

$$\mathrm{OPT}'_t \le w_T(\pi_{G\to T}(H^*_t)) \le O(\log^2 n) \cdot w_G(H^*_t). \tag{4}$$

Lastly, by the cost preservation property of our copy tree embedding we have that $w_G(H_t) \le w_T(H'_t)$ which when combined with Equations 3 and 4 gives

$$w_G(H_t) \le O\left(\frac{\log^3 n}{\epsilon} \cdot \max_i \frac{|g_i|}{r_i}\right) \cdot w_G(H^*_t).$$

thereby showing that our solution is within the required cost bound. ◀

Since group Steiner tree is exactly covering Steiner where $r_i = 1$ in which case $\max_i \frac{|g_i|}{r_i} \leq N$ where again $N$ is the maximum size of a group. Moreover, since any solution can only connect an integral number of vertices from each group, it follows that a $\frac{1}{2}$-connection-competitive solution for covering Steiner where $r_i = 1$ (i.e. for group Steiner tree) connects at least one vertex from each group. Thus, as a corollary of the above result we have the following deterministic algorithm for online group Steiner tree.[4]

▶ **Corollary 6.** *There is an $O(N \log^3 n)$-competitive deterministic algorithm for online group Steiner tree where $N := \max_i |g_i|$ is the maximum group size.*

## References


1 Ajit Agrawal, Philip Klein, and Ramamoorthi Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.

2 Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms (TALG)*, 2(4):640–660, 2006.

3 Noga Alon, Richard M Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, 1995.

4 Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Symposium on Foundations of Computer Science (FOCS)*, pages 542–547. IEEE, 1997.

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

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

7 Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *Annual European Symposium on Algorithms (ESA)*, pages 89–97. Springer, 2004.

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

9 Yair Bartal and Manor Mendel. Multi-embedding and path approximation of metric spaces. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 3, pages 424–433, 2003.

10 Marcin Bienkowski, Bjorn Feldkord, and Pawel Schmidt. A nearly optimal deterministic online algorithm for non-metric facility location. *arXiv preprint*, 2020. `arXiv:2007.07025`.

11 Niv Buchbinder and Joseph Naor. *The design of competitive online algorithms via a primal-dual approach*. Now Publishers Inc, 2009.

12 Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge Plotkin. Approximating a finite metric by a small number of tree metrics. In *Symposium on Foundations of Computer Science (FOCS)*, pages 379–388. IEEE, 1998.

13 Kedar Dhamdhere, Vineet Goyal, R Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *Symposium on Foundations of Computer Science (FOCS)*, pages 367–376. IEEE, 2005.


[4] We note that one can use an aforementioned property of our first construction – that if $u$ is connected to $r$ by $F \subseteq E$ then every vertex in $\phi(u)$ is connected to $\phi(r)$ in $\pi_{G \to T}(F)$ – to reduce the $O(\log^3 n)$s in this section to $O(\log^2 n)$s. In particular, if one were to use this property then when we map the solution to our covering Steiner problem on $G$ to our copy tree embedding, the resulting solution will connect at least $r_i$ groups in $\mathcal{G}_i$ at least $\Theta(\log n)$ times. It follows that when we run our water filling algorithm each time it increases $\sum_e x_e$ by 1 we know that it cover at least $\Omega(\log n)$ units of the optimal solution by weight rather than 1 unit of the optimal solution as in the current analysis.

**14** Matthias Englert and Harald Räcke. Reordering buffers with logarithmic diameter dependency for trees. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1224–1234. SIAM, 2017.

**15** Matthias Englert, Harald Räcke, and Matthias Westermann. Reordering buffers for general metric spaces. In *Annual ACM Symposium on Theory of Computing (STOC)*, 2007.

**16** Guy Even, Guy Kortsarz, and Wolfgang Slany. On network design problems: fixed cost flows and the covering steiner problem. *ACM Transactions on Algorithms (TALG)*, 1(1):74–101, 2005.

**17** Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

**18** Uriel Feige, Kamal Jain, Mohammad Mahdian, and Vahab Mirrokni. Robust combinatorial optimization with exponential scenarios. In *Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 439–453. Springer, 2007.

**19** Amos Fiat and Manor Mendel. Better algorithms for unfair metrical task systems and applications. *SIAM Journal on Computing*, 32(6):1403–1422, 2003.

**20** Arnold Filtser. Clan embeddings into trees, and low treewidth graphs. *arXiv preprint*, 2021. `arXiv:2101.01146`.

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

**22** Daniel Golovin, Vineet Goyal, and R Ravi. Pay today for a rainy day: improved approximation algorithms for demand-robust min-cut and shortest path problems. In *International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 206–217. Springer, 2006.

**23** Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. On the facility location problem in online and dynamic models. In *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

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

**25** Anupam Gupta, Viswanath Nagarajan, and R Ravi. Robust and maxmin optimization under matroid and knapsack uncertainty sets. *ACM Transactions on Algorithms (TALG)*, 12(1):1–21, 2015.

**26** Anupam Gupta, Viswanath Nagarajan, and Ramamoorthi Ravi. Thresholded covering algorithms for robust and max-min optimization. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 262–274. Springer, 2010.

**27** Anupam Gupta and Aravind Srinivasan. An improved approximation ratio for the covering steiner problem. *Theory of Computing*, 2(1):53–64, 2006.

**28** Varun Gupta, Ravishankar Krishnaswamy, and Sai Sandeep. Permutation strikes back: The power of recourse in online metric matching. In *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2020.

**29** Bernhard Haeupler, D Ellis Hershkowitz, and Goran Zuzic. Tree embeddings for hop-constrained network design. *Annual ACM Symposium on Theory of Computing (STOC)*, 2021.

**30** D Ellis Hershkowitz, R Ravi, and Sahil Singla. Prepare for the expected worst: Algorithms for reconfigurable resources under uncertainty. In *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2019.

**31** Makoto Imase and Bernard M Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991.

**32** Richard M Karp. A 2k-competitive algorithm for the circle. *Manuscript, August*, 5, 1989.

**33** Adam Kasperski and Paweł Zieliński. On the approximability of robust spanning tree problems. *Theoretical Computer Science*, 412(4-5):365–374, 2011.

34 Rohit Khandekar, Guy Kortsarz, Vahab Mirrokni, and Mohammad R Salavatipour. Two-stage robust network design with exponential scenarios. In *Annual European Symposium on Algorithms (ESA)*, pages 589–600. Springer, 2008.

35 Goran Konjevod, Ramamoorthi Ravi, and Aravind Srinivasan. Approximation algorithms for the covering steiner problem. *Random Structures & Algorithms*, 20(3):465–482, 2002.

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

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