THEORY OF COMPUTING, Volume 22 (2077), 2026, pp. 1–53
www.theoryofcomputing.org

SPECIAL ISSUE: APPROX-RANDOM 2021

Online Directed Spanners and Steiner Forests

Elena Grigorescu Young-San Lin Kent Quanrud

Received September 17, 2021; Revised May 19, 2023; Published April 25, 2026

Abstract. We present online algorithms for directed spanners and directed Steiner forests. These are well-studied network connectivity problems that fall under the unifying framework of online covering and packing linear programming formulations. This framework was developed in the seminal work of Buchbinder and Naor (Mathematics of Operations Research, 34, 2009) and is based on primal-dual techniques. Specifically, our results include the following:

- For the *pairwise spanner* problem, in which the pairs of vertices to be spanned arrive online, we present an efficient randomized algorithm with competitive ratio $\tilde{O}(n^{4/5})$ for graphs with general edge lengths, where n is the number of vertices of the given graph. For graphs with uniform edge lengths, we give efficient randomized algorithms with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$ and $\tilde{O}(k^{1/2+\varepsilon})$, where k is the number of terminal pairs. To the best of our knowledge, these are the first online algorithms for directed spanners. In the offline version, the current best approximation ratio for uniform edge lengths is $\tilde{O}(n^{3/5+\varepsilon})$, due to Chlamtáč, Dinitz, Kortsarz, and Laekhanukit (SODA 2017, TALG 2020).

An extended abstract of this paper [52], appeared in the International Conference on Approximation Algorithms for Combinatorial Optimization Problems, 2021.

ACM Classification: G.2.2

AMS Classification: 68W27, 68R10

Key words and phrases: online algorithms, linear programming, graph spanners

© 2026 Elena Grigorescu, Young-San Lin, and Kent Quanrud
Licensed under a Creative Commons Attribution License (CC-BY)
DOI: 10.4086/toc.2026.v022a2077

- For the *directed Steiner forest* problem with uniform costs, in which the pairs of vertices to be connected arrive online, we present an efficient randomized algorithm with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$. The state-of-the-art online algorithm for general costs is due to Chakrabarty, Ene, Krishnaswamy, and Panigrahi (SICOMP 2018) and is $\tilde{O}(k^{1/2+\varepsilon})$-competitive. In the offline version, the current best approximation ratio with uniform costs is $\tilde{O}(n^{4/7+\varepsilon})$, due to Abboud and Bodwin (SODA 2018).

To obtain *efficient* and *competitive* online algorithms, we observe that a small modification of the online covering and packing framework by Buchbinder and Naor implies a polynomial-time implementation of the primal-dual approach with separation oracles, which a priori might perform exponentially many calls to the oracle. We convert the online spanner problem into an online covering problem and complete the rounding-step analysis in a problem-specific fashion.

1 Introduction

We study online variants of directed network optimization problems. In an online problem, the input is presented sequentially, one item at a time, and the algorithm is forced to make irrevocable decisions in each step, without knowledge of the remaining part of the input. The performance of the algorithm is measured by its *competitive ratio*, which is the ratio between the value of the online solution and that of an optimal offline solution.

Our main results focus on *directed spanners*, which are sparse subgraphs that approximately preserve pairwise distances between vertices. Spanners are fundamental combinatorial objects with a wide range of applications, such as distributed computation [9, 69], data structures [5, 75], routing schemes [35, 67, 70, 72], approximate shorthest paths [18, 41, 42], distance oracles [18, 31, 68], and property testing [8, 22]. For a comprehensive account of the literature, we refer the reader to the excellent survey [2].

We also study related network *connectivity* problems, and in particular *directed Steiner forests*, which are sparse subgraphs that maintain connectivity between target terminal vertex pairs. Steiner forests are broadly used in different areas, such as multicommodity network design [49, 53], mechanism design and games [30, 63, 64, 73], computational biology [62, 71], and computational geometry [19, 24].

Our approaches are based on covering and packing linear programming (LP) formulations that fall into the unifying framework developed by Buchbinder and Naor [26], using the powerful primal-dual technique [51]. This unifying framework extends across widely different domains, and hence provides a general abstraction that captures the algorithmic essence of all online covering and packing formulations. In our case, to obtain *efficient* competitive algorithms for solving the LPs online, we observe that the algorithms in [26] can be slightly modified to significantly speed up the setting of our applications, in which the algorithm might otherwise make exponentially many calls to a separation oracle. This component is not tailored to the applications studied here and may be of independent interest. In particular, previous

approaches solving online covering and packing problems either focus on the competitiveness of the algorithm [4, 12, 16], or manage to leverage the specific structure of the problem for better time efficiency in a somewhat ad-hoc manner [3, 11, 15, 21, 25, 59]. Here, the solution may be viewed as a more *unified* framework that is also *efficient*.

1.1 Our contributions

1.1.1 Directed spanners

Let $G = (V, E)$ be a directed simple graph with n vertices. Each edge is associated with its *length* $\ell : E \to \mathbb{R}_{\geq 0}$. The edge lengths are *uniform* if $\ell(e) = 1$ for all $e \in E$. In spanner problems, the goal is to compute a minimum cardinality (number of edges) subgraph in which the distance between terminals is preserved up to some prescribed factor. In the most well-studied setting, called the *directed s-spanner* problem, there is a fixed value $s \geq 1$ called the *stretch*, and the goal is to find a minimum cardinality subgraph in which *every* pair of vertices has *distance* within a factor of s in the original graph. Here, we use the term distance to denote the sum of the edge lengths in a shortest path of two terminal vertices. For low stretch spanners, when $s = 2$, there is a tight $\Theta(\log n)$-approximation algorithm [44, 65]; when $s = 3, 4$ both with uniform edge lengths, there are $\tilde{O}(n^{1/3})$-approximation algorithms [20, 40]. When $s > 4$, the best known approximation factor is $\tilde{O}(n^{1/2})$ [20]. The problem is hard to approximate within an $O(2^{\log^{1-\varepsilon} n})$ factor for $3 \leq s = O(n^{1-\delta})$ and any $\varepsilon, \delta \in (0, 1)$, unless $NP \subseteq DTIME(n^{\text{polylog}\, n})$ [45].

A more general setting, called the *pairwise spanner* problem [34], and the *client-server* model [22, 44], considers an arbitrary set of terminals $D = \{(s_i, t_i) \mid i \in [k]\} \subseteq V \times V$. Each terminal pair (s_i, t_i) has its own target distance d_i. The goal is to compute a minimum cardinality subgraph in which for each i, the distance from s_i to t_i is at most d_i. For the pairwise spanner problem with uniform edge lengths, [34] obtains an $\tilde{O}(n^{3/5+\varepsilon})$-approximation.

In the online version, the graph is known ahead of time, and the terminal pairs and the corresponding target distances are received one by one in an online fashion. The distance requirement of the arriving terminal pair is satisfied by irrevocably including edges. There are no online algorithms for the pairwise spanner problem that we are aware of, even in the simpler and long-studied case of stretch s or graphs with uniform edge lengths.

For graphs with uniform edge lengths, we prove the following theorem in Section 2.

Theorem 1.1. *For the online pairwise spanner problem with uniform edge lengths, there exists a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(k^{1/2+\delta})$ for any constant $\delta > 0$.*

Next, we turn to graphs with general edge lengths and derive online algorithms with competitive ratios in terms of n. We present a generic algorithm (Algorithm 4) used for Theorem 1.2, Theorem 1.3, Theorem 1.4, and other variants of spanners introduced in Section 3. For graphs with general edge lengths, we show the following in Section 3.1.

Theorem 1.2. *For the online pairwise spanner problem, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{4/5})$.*

For graphs with uniform edge lengths without further assumptions, we use Theorem 1.1 and the generic algorithm to prove the following theorem in Section 3.4.

Theorem 1.3. *For the online pairwise spanner problem with uniform edge lengths, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$ for any constant $\varepsilon \in (0, 1/3)$.*

1.1.2 Directed Steiner forests

In the *directed Steiner forest* problem, we are given a directed graph $G = (V, E)$ with edge costs $w : E \to \mathbb{R}_{\geq 0}$, and a set of terminals $D = \{(s_i, t_i) \mid i \in [k]\} \subseteq V \times V$. The goal is to find a subgraph $H = (V, E')$ which includes an $s_i \rightsquigarrow t_i$ path for each terminal pair (s_i, t_i), and the total cost $\sum_{e \in E'} w(e)$ is minimized. The costs are *uniform* when $w(e) = 1$ for all $e \in E$.

In the online version, the graph is known ahead of time, and the terminal pairs arrive one by one in an online fashion. The connectivity requirement of the arriving terminal pair is satisfied by irrevocably including edges.

In the offline setting with general costs, the best known approximations are $O(k^{1/2+\varepsilon})$ by Chekuri et al. [32] and $O(n^{2/3+\varepsilon})$ by Berman et al. [20]. For the special case of uniform costs, there is an improved approximation factor of $\tilde{O}(n^{4/7+\varepsilon})$ by Abboud and Bodwin [1]. In the online setting, Chakrabarty et al. [28] give an $\tilde{O}(k^{1/2+\varepsilon})$ approximation for general costs. Their algorithm also extends to the more general buy-at-bulk version. We prove the following in Section 3.5.

Theorem 1.4. *For the online directed Steiner forest problem with uniform costs, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$ for any constant $\varepsilon \in (0, 1/3)$.*

We essentially improve the competitive ratio when the number of terminal pairs is $\omega(n^{4/3})$.

1.1.3 Summary

We summarize our main results for online pairwise spanners and directed Steiner forests in Table 1 by listing the competitive ratios and contrast them with the corresponding known competitive and approximation ratios. We note that offline $\tilde{O}(n^{4/5})$-approximate pairwise spanners for graphs with general edge lengths and offline $\tilde{O}(k^{1/2+\varepsilon})$-approximate pairwise spanners for graphs with uniform edge lengths can be obtained by our online algorithms.

1.2 An efficient online covering and packing framework

Before presenting our modification to the unified framework in [26] to obtain efficient online covering and packing LP solvers, we give an overview of the well-known primal-dual framework for approximating covering and packing LP's online. This framework is the main engine of our applications and it is important to establish some context before getting into the application for spanners and Steiner forests. We also introduce a discussion of certain technical nuances that arise for our application, and the small modification we propose to address it. A more formal description, including proofs and fully parameterized theorem statements, is fairly technical

Setting	Offline	Online
Pairwise Spanners	$\tilde{O}(n^{4/5})$ (implied by Thm 1.2)	$\tilde{O}(n^{4/5})$ (Thm 1.2)
	$\tilde{O}(n^{3/5+\varepsilon})$ (uniform lengths) [34]	$\tilde{O}(n^{2/3+\varepsilon})$ (uniform lengths, Thm 1.3)
	$\tilde{O}(k^{1/2+\varepsilon})$ (uniform lengths, implied by Thm 1.1)	$\tilde{O}(k^{1/2+\varepsilon})$ (uniform lengths, Thm 1.1)
Directed Steiner Forests	$\tilde{O}(n^{4/7+\varepsilon})$ (uniform costs) [1]	$\tilde{O}(k^{1/2+\varepsilon})$ [28]
	$O(n^{2/3+\varepsilon})$ [20]	$\tilde{O}(n^{2/3+\varepsilon})$ (uniform costs, Thm 1.4)
	$O(k^{1/2+\varepsilon})$ [32]	

Table 1: Summary of the competitive and approximation ratios. Here, n refers to the number of vertices and k refers to the number of terminal pairs. We include the known results for comparison. The text in gray refers to known results while the text in black refers to our contributions.

and therefore deferred to Section 4 *after* we have used these tools in the context of spanners and Steiner forests.

The primal-dual framework was first developed for the online set cover problem in the seminal work of [4]. The approach was extended to network optimization problems in undirected graphs in [3], then abstracted and generalized to a broad LP-based primal-dual framework in [26]. Our discussion primarily centers around the abstract framework in [26]. A number of previous results in online algorithms, such as ski rental [61] and paging [16], can be recovered from this approach and many new important applications have since been developed, such as the k-server problem [76]. We refer the reader to the excellent survey by Buchbinder and Naor [27].

These works develop a clean two-step approach to online algorithms based on 1) solving the LP online, and 2) rounding the LP online. Solving the LP online can be done in a generic fashion, while rounding tends to be problem-specific. The setting for the *covering* LP is the following.

$$\text{minimize } \mathbf{c}^T x \text{ over } x \in \mathbb{R}^n_{\geq 0} \text{ s.t. } Ax \geq \mathbf{b}. \tag{1.1}$$

Here, $A \in \mathbb{R}^{m \times n}_{\geq 0}$ consists of m covering constraints, $\mathbf{b} \in \mathbb{R}^n_{>0}$ is a positive lower bound of the covering constraints, and $\mathbf{c} \in \mathbb{R}^m_{>0}$ denotes the positive coefficients of the linear cost function. Each constraint can be normalized, so we focus on covering LP's in the following form.

$$\text{minimize } \mathbf{c}^T x \text{ over } x \in \mathbb{R}^n_{\geq 0} \text{ s.t. } Ax \geq \mathbf{1} \tag{1.2}$$

where $\mathbf{1}$ is a vector of all ones.

In the online covering problem, the cost vector $\mathbf{c}$ is given offline, and each of these covering constraints is presented one by one in an online fashion, that is, m can be unknown. The goal is to update x in a non-decreasing manner such that all the covering constraints are satisfied and the objective value $\mathbf{c}^T x$ is approximately optimal. An important idea in this line of work is to

simultaneously consider the dual *packing* problem:

$$\text{maximize } \mathbf{1}^T y \text{ over } y \in \mathbb{R}^m_{\geq 0} \text{ s.t. } A^T y \leq \mathbf{c} \tag{1.3}$$

where A^T consists of n packing constraints with an upper bound $\mathbf{c}$ given offline.

In the online packing problem, the *columns* of A^T and the corresponding variables are presented online taking initial value zero; one can either let the arriving variable remain zero, or irrevocably assign a positive value to the arriving variable. The goal is to approximately maximize the objective value $\mathbf{1}^T y$ with each constraint approximately satisfied.

Separation oracles in the online setting. The primal-dual framework in [26] simultaneously solves both LP (1.2) and LP (1.3), and crucially uses LP-duality and strong connections between the two solutions to argue that they are both nearly optimal. Here we give a sketch of the LP solving framework for reference in the subsequent discussion. We maintain solutions x and y for LP (1.2) and LP (1.3), respectively, in an online fashion. The covering solution x is a function of the packing solution y. In particular, each coordinate x_j is exponential in the *load* of the corresponding packing constraint in LP (1.3). Both x and y are monotonically increasing. The algorithm runs in phases, where each phase corresponds to an estimate for the optimal value revised over time. Within a phase we have the following. If the new covering constraint $i \in [m]$, presented online, is already satisfied, then there is nothing to be done. Otherwise, increase the corresponding coordinate y_i, which simultaneously increases the x_j's based on the magnitude of the coordinate a_{ij}, where a_{ij} is the i-th row j-th column entry of A. The framework in [26] increases y_i until the increased x_j's satisfy the new constraint. This naturally extends to the setting when the problem relies on a *separation oracle* to retrieve an unsatisfied covering constraint where the number of constraints can be unbounded [26]. However, while this approach will fix all violating constraints, each individual fix may require a diminishingly small adjustment that cannot be charged off from a global perspective. Consequently the algorithm may have to address exponentially many constraints.

A primal-dual bound on separation oracles. Our goal is to adjust the framework to ensure that we only address a polynomial number of constraints (per phase). For many concrete problems in the literature, this issue can be addressed directly based on the problem at hand (discussed in greater detail in Section 1.4). In our setting, we start with a combinatorially defined LP that is not a pure covering problem, and convert it to a covering LP. While having a covering LP is conducive to the online LP framework, the machinery generates a large number of covering constraints that are very unstructured. For example, we have little control over the coefficients of these constraints. This motivates us to develop a more generic argument to bound the number of queries to the separation oracle, based on the online LP framework, more so than the exact problem at hand. Here, when addressing a violated constraint i, we instead increase the dual variable y_i until the increased primal variables x (over-)satisfy the new constraint *by a factor of 2*. This forces at least one x_j to be doubled – and in the dual, this means we used up a substantial amount of the corresponding packing constraint. Since the packing solution

is already guaranteed to be feasible in each phase by the overall framework, this leads us to conclude that we only ever encounter polynomially many violating constraints.

For our modified online covering and packing framework, we show that 1) the approximation guarantees are identical to those in [26], 2) the framework only encounters polynomially many violating constraints for the online covering problem, and 3) only polynomially many updates are needed for the online packing problem.

Theorem 1.5. *(Informal) There exists an $O(\log n)$-competitive online algorithm for the covering LP (1.2) which encounters polynomially many violating constraints.*

Theorem 1.6. *(Informal) Given any parameter $B > 0$, there exists a $1/B$-competitive online algorithm for the packing LP (1.3) which updates y polynomially many times, and each constraint is violated within an $O(f(A)/B)$ factor ($f(A)$ is a logarithmic function that depends on the entries in A).*

We note that the competitive ratios given in [26] are tight, which also implies the tightness of the modified framework. The number of violating constraints depends not only on the number of covering variables and packing constraints n, but also on the number of bits used to present the entries in A and $\mathbf{c}$. The formal proof for Theorem 1.5 is provided in Section 4, while the formal proof for Theorem 1.6 provided in Section 5 is not directly relevant to this work, but may be of independent interest.

1.3 High-level technical overview for online network optimization problems

Online pairwise spanners. For this problem, a natural starting point is the flow-based LP approach for offline s-stretch directed spanners, introduced in [38]. The results of [34] adopt a slight tweak for this approach to achieve an $\tilde{O}(n/\sqrt{\mathsf{OPT}})$-approximation, where OPT is the size of the optimal solution. With additional ideas, the $\tilde{O}(n/\sqrt{\mathsf{OPT}})$-approximation is converted into an $\tilde{O}(n^{3/5+\varepsilon})$-approximation for pairwise spanners. One technical obstacle in the online setting is the lack of a useful lower bound for OPT. Another challenge is solving the LP for the spanner problem and rounding the solution in an online fashion, particularly as the natural LP is not a covering LP. We address these technical obstacles as discussed below in Section 3. Ultimately we obtain an $\tilde{O}(n^{4/5})$ competitive ratio for the online setting. The strategy here is to convert the LP for spanners into a covering LP, where the constraints are generated by an *internal* LP. The covering LP previously appeared in [38] implicitly, and in [39] explicitly.

Online pairwise spanners with uniform edge lengths. For the special case of uniform edge lengths, [34] obtains an improved bound of $\tilde{O}(n^{3/5+\varepsilon})$. It is natural to ask if the online bound of $\tilde{O}(n^{4/5})$ mentioned above can be improved as well. Indeed, we obtain an improved bound of $\tilde{O}(n^{2/3+\varepsilon})$ by replacing the greedy approach in the small OPT regime by using the $\tilde{O}(k^{1/2+\varepsilon})$-competitive online algorithm discussed in Section 2. This algorithm leverages ideas from [34] in reducing to label cover problems with ideas from the online network design algorithms of [28]. Some additional ideas are required to combine the existing tools and among others we had to formulate a new pure covering LP that can be solved online, to facilitate the transition.

Online Steiner forests with uniform costs. This problem is a special case of the online pairwise spanner problem where the distance requirement for each terminal pair is infinity and the edge lengths are uniform. The online algorithm for this problem has a similar structure to the one for pairwise spanners and similar obstacles to overcome.

1.4 Additional background and related work

Streaming, dynamic, and distributed algorithms for spanners. A model related to online algorithms is that of streaming algorithms. In the streaming model an input is also revealed sequentially, but the algorithm is only allowed to use some small amount of space, which is sublinear in the length of the stream, and is supposed to maintain an approximate solution. For this model, several papers consider spanner variants, such as undirected or weighted graphs, and additive or multiplicative stretch approximations, and the aim is to build spanners with small size or distortion [17, 60, 48]. In a related direction, spanners have also been studied in the setting of dynamic data structures, where the edges of a graph are inserted or removed one at a time and the goal is to maintain an approximate solution with small update time and space [43, 23]. A relevant model is that of distributed computation where nodes in the network communicate efficiently to build a solution [36, 37, 47]. As mentioned earlier, the survey by Ahmed et al. [2] gives a comprehensive account of the vast literature on spanners, and we refer the reader to the references within.

Connections to buy-at-bulk formulations. In the buy-at-bulk network design problem [10], each edge is associated with a sub-additive cost function of its *load*. Given a set of terminal demands, the goal is to route integral flows from each source to each sink concurrently to minimize the total cost of the routing. This problem is a generalization of various single-source or multicommodity network connectivity problems, including Steiner trees and Steiner forests, in which the cost of each edge is fixed once allocated. While most problems admit efficient polylogarithmic approximations in either the online or offline setting for undirected networks [32, 11, 21, 55], the problems are much harder for directed networks. In the offline setting, the current best approximation ratio is $O(k^{\varepsilon})$ in polynomial time and $\mathsf{polylog}(n)$ in quasi-polynomial time for the directed Steiner tree problem [77, 29], $O(\min\{k^{1/2+\varepsilon}, n^{2/3+\varepsilon}\})$ in polynomial time for the directed Steiner forest problem [20, 32], and $O(\min\{k^{1/2+\varepsilon}, n^{4/5+\varepsilon}\})$ in polynomial time for the directed buy-at-bulk problem [6]. In the online setting for directed networks, [28] showed that compared to offline, it suffices to pay an extra polylogarithmic factor, where the polylogarithmic term was later improved by [74]. The main contribution of [28] is essentially bringing the *junction-tree-based* approach into the online setting for connectivity problems. This is the main ingredient that improves the competitive ratio of our online algorithm for pairwise spanners from $\tilde{O}(n^{4/5+\varepsilon})$ when edge lengths are general to $\tilde{O}(n^{2/3+\varepsilon})$ when edge lengths are uniform. Our approach for online pairwise spanners with uniform edge lengths combines this ingredient and the ideas of the offline pairwise spanner framework [34] which tackles hard distance requirements.

Online LP's and separation oracles. As previously mentioned, generating separating constraints with an oracle in the online setting is not new. For example, this arises implicitly in early work on network optimization [3] and the oracle is discussed explicitly in [26]. As a recent example, [56] develops online algorithms for the multistage matroid maintenance problem, which requires solving a covering LP with box constraints online. [56] adjusts the separation oracle to only identify constraints that are violated by at least some constant. Because of the $\{0,1\}$-incidence structure of their LP, the sum of primal variables has to increase by a constant to satisfy such a constraint. Meanwhile the box constraints limit the total sum of primal variables to $O(n)$. This leads to an $O(n)$ bound on the number of separating constraints. While there are strong similarities to our approach, one difference is the use of the $\{0,1\}$-structure and box constraints to obtain their bound. Our comparably unstructured setting required us to develop an argument independent of concrete features such as these.

Other variants of online covering and packing problems. Beyond linear objectives, there are other variants of online covering and packing problems, which focus on different objectives with linear constraints. This includes optimizing convex objectives [14] and ℓ_q-norm objectives [74]. Other online problem-dependent variants include for instance mixed covering and packing programs [13], and sparse integer programs [54]. All these frameworks utilize the primal-dual technique, which updates the covering and packing solutions simultaneously with some judiciously selected growth rate, to guarantee nice competitive ratio. Instead, our modified framework focuses on the efficiency of online algorithms for fundamental covering and packing problems, which is amenable to applications with exponential or unbounded number of constraints, where a violating one can be searched by an efficient separation oracle.

1.5 Organization

Since the proof of Theorem 1.1 is the most involved contribution of this work, we start by presenting it in Section 2. In Section 3, we prove Theorem 1.2, Theorem 1.3, and Theorem 1.4 by designing and analyzing specific variants of a generic online algorithm. We show the modified online covering framework in Section 4, while the modified online packing framework is presented in Section 5.

2 Online Pairwise Spanners with Uniform Edge Lengths

In this section, we prove Theorem 1.1. Namely, we design a randomized online algorithm for the pairwise spanner problem with uniform edge lengths with competitive ratio $\tilde{O}(k^{1/2+\delta})$ for any constant $\delta > 0$. We recall that in the *pairwise spanner* problem, we are given a directed graph $G = (V, E)$ with edge length $\ell : E \to \mathbb{R}_{\geq 0}$, a general set of k terminals $D = \{(s_i, t_i) \mid i \in [k]\} \subseteq V \times V$, and a target distance d_i for each terminal pair (s_i, t_i), the goal is to output a subgraph $H = (V, E')$ of G such that for every pair $(s_i, t_i) \in D$ it is the case that $d_H(s_i, t_i) \leq d_i$, i. e., the distance of a shortest $s_i \rightsquigarrow t_i$ path is at most d_i in the subgraph H, and we want to minimize the number of edges in E'. The edge lengths are uniform if $\ell(e) = 1$ for

all $e \in E$. In the online setting, the directed graph G is given offline, while the vertex pairs in $D \subseteq V \times V$ arrive online one at a time. In the beginning, $E' = \emptyset$. Suppose (s_i, t_i) and its target distance d_i arrive in round i, we select some edges from E and irrevocably add them to E', such that in the subgraph $H = (V, E')$, $d_H(s_i, t_i) \leq d_i$.

2.1 Outline of the proof of Theorem 1.1

We start by describing the high-level approach of our proof of Theorem 1.1. While the proof combines ideas of the online buy-at-bulk framework in [28] and of the reduction from the pairwise spanner problem to a connectivity problem in [34], implementing the details require several new ideas. Specifically, we introduce a useful extension of the Steiner problem, called the *Steiner label cover* problem, and our main contribution is an online covering LP formulation for this problem. This approach allows us to not only capture the global approximation property in an online setting, as in [28], but also to handle distance constraints, as in [34]. The entire proof consists of three main ingredients:

1. We first show that there exists an $O(\sqrt{k})$-approximate solution consisting of *junction trees*. A junction tree is a subgraph consisting of an in-arborescence and out-arborescence rooted at the same vertex (see also Definition 2.1).

2. We then show a reduction from the online pairwise spanner problem to the online Steiner label cover problem on a forest with a loss of an $O(k^{1/2+\delta})$ factor. More precisely, an $O(\sqrt{k})$ factor comes from the junction tree approximation and an extra $O(k^{\delta})$ factor comes from the *height reduction* technique introduced in [32, 58]. The height reduction technique allows us to focus on low-cost trees of height $O(1/\delta)$ in order to recover a junction tree approximation.

3. Finally, we show a reduction from the online Steiner label cover problem to the online *undirected group Steiner forest* problem, with a loss of a $\mathsf{polylog}(n)$ factor. More precisely, we first formulate an online covering LP for the online Steiner label cover instance, then construct an online undirected group Steiner forest instance from the LP solution, with a loss of a factor of 2. By [28], the online undirected group Steiner forest problem on the forest that we construct can be solved with competitive ratio $\mathsf{polylog}(n)$.

Combining these three ingredients results in an $\tilde{O}(k^{1/2+\delta})$-competitive algorithm. We provide further intuition below. The detailed description of the first, second, and third ingredients are in Section 2.2, Section 2.3, and Section 2.4, respectively.

Junction tree approximation. Many connectivity problems, including Steiner forests, buy-at-bulk, and spanner problems, are usually solved using *junction trees* introduced in [33].

Definition 2.1. A *junction tree* rooted at $r \in V$ is a directed graph $G = (V, E)$, by taking the union of an in-arborescence rooted at r and an out-arborescence rooted at r.[1] A *junction tree solution* is a collection of junction trees rooted at different vertices, that satisfies all the terminal distance constraints.

Lemma 2.2. *There exists an $O(\sqrt{k})$-approximate junction tree solution for pairwise spanners.*

In Section 2.2, we present the proof of Lemma 2.2. At a high level, the proof follows by a standard *density* argument. A *partial solution* is a subgraph that connects a subset of the terminal pairs within the required distances. The density of a partial solution is the ratio between the number of edges used and the number of terminal pairs connected within the required distances. This argument is used for solving offline problems including the Steiner forest problem [32, 46, 20], the buy-at-bulk problem [6], the Client-Server s-spanner [22] problem, and the pairwise spanner problem [34], by greedily removing low-density partial solutions in an iterative manner. Fortunately, this iterative approach also guarantees a nice *global* approximation that consists of junction trees rooted at different vertices, which is amenable in the online setting.

Reduction to Steiner label cover. In Section 2.3, we reduce the pairwise spanner problem to the following extension of Steiner problem termed *Steiner label cover*.

Definition 2.3. In the *Steiner label cover* problem, we are given a (directed or undirected) graph $G = (V, E)$, non-negative edge costs $w : E \to \mathbb{R}_{\geq 0}$, and a collection of k disjoint vertex subset pairs (S_i, T_i) for $i \in [k]$ where $S_i, T_i \subseteq V$ and $S_i \cap T_i = \emptyset$. Each pair is associated with a relation (set of permissible pairs) $R_i \subseteq S_i \times T_i$. The goal is to find a subgraph $F = (V, E')$ of G, such that 1) for each $i \in [k]$, there exists $(s, t) \in R_i$ such that there is an $s \rightsquigarrow t$ path in F, and 2) the cost $\sum_{e \in E'} w(e)$ is minimized.

For the online Steiner label cover problem, (S_i, T_i) and R_i arrive online, and the goal is to irrevocably select edges to satisfy the first requirement and also approximately minimize the cost.

To reduce to the online Steiner label cover problem, we construct a directed graph G' that consists of disjoint layered graphs from the given graph $G = (V, E)$. Each vertex in G' is labeled by the distance to (from) the root of a junction tree. This allows us to capture distance constraints by a Steiner label cover instance with distance-based relations. From G', we further construct an undirected graph H which is a forest by the *height reduction* technique [32, 58]. In H, we define the corresponding Steiner label cover instance, where the solution is guaranteed to be a forest. The Steiner label cover instance on the forest H has a nice property. For each tree in H, the terminal vertices can be ordered in a way such that if an *interval* belongs to the relation, then any *subinterval* also belongs to the relation.

[1] A junction tree does not necessarily have a tree structure in directed graphs, i. e., an edge may be used twice, once in the in-arborescence and once in the out-arborescence. Nevertheless, we continue using this term because of historical reasons. A similar notion can also be used for undirected graphs, where a junction tree is indeed a tree.

Definition 2.4. The *ordered Steiner label cover problem on a forest* is defined as a special case of the Steiner label cover problem (see Definition 2.3) with the following properties.

1. G is an undirected graph consisting of disjoint union of trees $H_1, H_2, \ldots, H_n$ each of which has a distinguished root vertex r_j where $j \in [n]$.
2. For each (S_i, T_i) and R_i for $i \in [k]$ where S_i and T_i are disjoint subsets of the leaf vertices of G, and each tree H_j, the input also includes the total orderings $\prec_{i,j}$ such that:
   (a) For $S_i^j := S_i \cap V(H_j)$ and $T_i^j := T_i \cap V(H_j)$, the ordering $\prec_{i,j}$ is defined on $S_i^j \cup T_i^j$.
   (b) The root r_j separates S_i^j from T_i^j.
   (c) If $s \in S_i^j$ and $t \in T_i^j$ are such that $(s,t) \in R_i$, then for any $s' \in S_i^j$ and $t' \in T_i^j$ such that $s \preceq_{i,j} s' \prec_{i,j} t' \preceq_{i,j} t$, we have that $(s', t') \in R_i$.

We note that for the online ordered Steiner label cover problem on a forest, besides (S_i, T_i) and R_i, the orderings $\{\succ_{i,j}\}_{j\in[n]}$ also arrive online.

We employ a well-defined mapping between junction trees in G and forests in H by paying an $\tilde{O}(k^{1/2+\delta})$ factor for competitive online solutions. A crucial step for showing Theorem 1.1 is the following theorem.

Theorem 2.5. *For any constant $\delta > 0$, an α-competitive polynomial-time algorithm for online ordered Steiner label cover on a forest implies an $O(\alpha k^{1/2+\delta})$-competitive polynomial-time algorithm for the online pairwise spanner problem on a directed graph with uniform edge lengths.*

At a high level, the online pairwise spanner problem on a directed graph $G = (V, E)$ with uniform edge lengths reduces to an instance of online Steiner label cover on the forest H with the following properties.

1. H consists of disjoint trees H_r for each vertex $r \in V$.
2. $|V(H)| = n^{O(1/\delta)}$, $E(H) = n^{O(1/\delta)}$, and each tree H_r has depth $O(1/\delta)$.
3. For each arriving terminal pair (s_i, t_i) with distance requirement d_i, there is a corresponding pair of terminal sets $(\hat{S}_i, \hat{T}_i)$ and relation $\hat{R}_i$ with $|\hat{R}_i| = n^{O(1/\delta)}$, where $\hat{S}_i$ and $\hat{T}_i$ are disjoint subsets of vertices in H. Furthermore, we can generate total orderings $\prec_{i,r}$ based on the distance-based relations $\hat{R}_i$ such that the Steiner label cover instance is an ordered instance on the forest H.

This technique closely follows the one for solving offline pairwise spanners in [34]. The intermediary problem considered in [34] is the *minimum density Steiner label cover problem*. In this framework, the solution is obtained by selecting the partial solution with the lowest density among the junction trees rooted at different vertices and repeating. In the online setting, to capture the global approximation for pairwise spanners, we construct a forest H and consider all the possible roots simultaneously.

The constructions of G', H, $\hat{S}_i$, $\hat{T}_i$, $\hat{R}_i$, and the orderings $\prec_{i,r}$, the proofs of Theorem 2.5 and Theorem 1.1, and the reduction from the online pairwise spanner problem on the directed graph G to the online ordered Steiner label cover problem on the forest H, are presented in Section 2.3.

An online algorithm for Steiner label cover on H. In Section 2.4, the goal is to prove the following lemma.

Lemma 2.6. *For the online ordered Steiner label cover problem on a forest (see Definition 2.4), there is a randomized polynomial-time algorithm with competitive ratio* $\mathsf{polylog}(n)$.

We derive an LP formulation for the Steiner label cover instance on H. At a high level, the LP minimizes the total edge weight by selecting edges that cover paths with endpoint pairs which belong to the distance-based relation. We show that the LP for Steiner label cover can be converted into an online covering problem, which is efficiently solvable by Theorem 1.5.

The online rounding is based on the online LP solution for Steiner label cover. We extract the representative vertex sets $\tilde{S}_i$ and $\tilde{T}_i$ from the terminal sets $\hat{S}_i$ and $\hat{T}_i$, respectively, according to orderings $\prec_{i,r}$ and the contribution of the terminal vertex to the objective of the Steiner label cover LP. We show that the union of cross-products over partitions of $\tilde{S}_i$ and $\tilde{T}_i$ (based on the trees in H) is a subset of the distance-based relation $\hat{R}_i$. This allows us to reduce the online ordered Steiner label cover problem to the *online undirected group Steiner forest* problem defined below.

Definition 2.7. In the *online undirected group Steiner forest* problem, we are given an undirected graph $G = (V, E)$ with non-negative edge costs $w : E \to \mathbb{R}_0$, and a collection of k terminal vertex set pairs $(S_i, T_i) \subseteq V \times V$ for $i \in [k]$ that arrives online. The goal is to irrevocably pick edges from E to form a subgraph $F = (V, E')$ of G, such that 1) upon the arrival of pair $i \in [k]$, there exists $s \in S_i$ and $t \in T_i$ such that there is an $s \rightsquigarrow t$ path in F, and 2) the cost $\sum_{e \in E'} w(e)$ is approximately minimized.

This technique closely follows the one for solving offline pairwise spanners in [34]. The main difference is that in the offline pairwise spanner framework, the LP formulation is density-based and considers only one (fractional) junction tree. To globally approximate the online pairwise spanner solution, our LP formulation is based on the forest H and its objective is the total weight of a (fractional) forest.

The LP for the undirected group Steiner forest problem is roughly in the following form.

$$\begin{aligned} \min_{x} \quad & \sum_{e \in E(H)} w'(e) x_e & & \\ \text{subject to} \quad & x \text{ supports an } \tilde{S}_i\text{-}\tilde{T}_i \text{ flow of value 1} & & \forall i \in [k], \\ & x_e \geq 0 & & \forall e \in E(H). \end{aligned} \tag{2.1}$$

Here w' denotes the edge weights in H. We show that a solution of the undirected group Steiner forest LP (2.1) recovers a solution for the Steiner label cover LP by a factor of 2. The integrality gap of the undirected group Steiner forest LP is $\mathsf{polylog}(n)$ because the instance can be decomposed into single-source group Steiner tree instances by the structure of H [28, 50]. This implies that the online rounding for the Steiner label cover LP can be naturally done via solving the undirected group Steiner forest instance online by losing a $\mathsf{polylog}(n)$ factor [28].

Putting it all together. In Section 2.5, we summarize the overall $\tilde{O}(k^{1/2+\delta})$-competitive algorithm for online pairwise spanners when the given graph has uniform edge lengths. The reduction strategy is as follows:

1. Reduce the online pairwise spanner problem of the original graph G to the online Steiner label cover problem of the directed graph G' which consists of disjoint layered graphs.

2. Reduce the online Steiner label cover problem on G' to an online ordered Steiner label cover problem on H, where H is a forest.

3. In the forest H, reduce the online ordered Steiner label cover problem to the online undirected group Steiner forest problem.

We note that G' and H are constructed offline. The pairwise spanner in G is $O(\sqrt{k})$-approximated by junction trees according to Lemma 2.2. The graph G' preserves at most twice the cost of the pairwise spanner (junction tree solution) in G. The solution of the ordered Steiner label cover problem in graph H is a forest. One can map a forest in H to junction trees in G', via the height reduction technique by losing an $O(k^{\delta})$ factor. Finally, in the forest H, we solve the undirected group Steiner forest instance online and recover an ordered Steiner label cover solution by losing a $\mathsf{polylog}(n)$ factor. The overall competitive ratio is therefore $\tilde{O}(k^{1/2+\delta})$.

2.2 The junction-tree approximation

We can now proceed to the proof of Lemma 2.2. Our goal is to approximate a pairwise spanner solution by a collection of junction trees rooted at different vertices.

Lemma 2.8. *There exists an $O(\sqrt{k})$-approximate junction tree solution for pairwise spanners.*

Proof. We use a density argument to show a greedy procedure that implies a $O(\sqrt{k})$-approximate junction tree solution. The *density* of a partial solution is defined as follows.

Definition 2.9. Let J denote a junction tree in G and $D(J)$ denote the set of source-sink pairs (s_i, t_i) connected by J such that $d_J(s_i, t_i) \leq d_i$, then the density of J is $\rho(J) := |E(J)|/|D(J)|$, i. e., the number of edges used in J divided by the number of terminal pairs connected by J within required distances.

Intuitively, we are interested in finding low-density junction trees. Let OPT be the number of edges in an optimal pairwise spanner solution. We show that there always exists a junction tree with density at most a $\sqrt{k}$ factor of the optimal density. The proof of Lemma 2.10 closely follows the one for the directed Steiner network problem in [32] by considering whether there is a *heavy* vertex that lies in $s_i \rightsquigarrow t_i$ paths for distinct i or there is a simple path with low density. The case analysis also holds when there is a distance constraint d_i for each (s_i, t_i).

Lemma 2.10. *There exists a junction tree J such that $\rho(J) \leq \mathsf{OPT}/\sqrt{k}$.*

Proof. Let G^* (a subgraph of G) be the optimal pairwise spanner solution with OPT edges. The proof proceeds by considering the following two cases: 1) there exists a vertex $r \in V$ that belongs to at least $\sqrt{k}$ $s_i \rightsquigarrow t_i$ paths of distance at most d_i in G^* for distinct i, and 2) there is no such vertex $r \in V$.

For the first case, we consider the union of the $s_i \rightsquigarrow t_i$ paths in G^*, each of distance at most d_i, that passes through r. This subgraph in G^* contains an in-arborescence and an out-arborescence both rooted at r, whose union forms a junction tree. This junction tree has at most OPT edges and connects at least $\sqrt{k}$ terminal pairs, so its density is at most $\mathsf{OPT}/\sqrt{k}$.

For the second case, each vertex $r \in V$ appears in at most $\sqrt{k}$ $s_i \rightsquigarrow t_i$ paths in G^*. More specifically, each edge $e \in E$ also appears in at most $\sqrt{k}$ $s_i \rightsquigarrow t_i$ paths in G^*. By creating $\sqrt{k}$ copies of each edge, all terminal pairs can be connected by edge-disjoint paths. Since the overall duplicate cost is at most $\sqrt{k} \cdot \mathsf{OPT}$, at least one of these paths has cost at most $\sqrt{k} \cdot \mathsf{OPT}/k$. This path constitutes a junction tree whose density is at most $\mathsf{OPT}/\sqrt{k}$. □

Now we are ready to prove Lemma 2.2. Consider the procedure that finds a minimum density junction tree in each iteration, and continues on the remaining disconnected terminal pairs. Suppose there are t iterations, and after iteration $j \in [t]$, there are n_j disconnected terminal pairs. Let $n_0 = k$ and $n_t = 0$. After each iteration, the minimum number of edges used for connecting the remaining terminal pairs in the remaining graph is at most OPT, so the number of edges used by this procedure is upper-bounded by

$$\sum_{j=1}^{t} \frac{(n_{j-1} - n_j)\mathsf{OPT}}{\sqrt{n_{j-1}}} \leq \sum_{i=1}^{k} \frac{\mathsf{OPT}}{\sqrt{i}} \leq \int_{1}^{k+1} \frac{\mathsf{OPT}}{\sqrt{x}} dx = 2\mathsf{OPT}(\sqrt{k+1} - 1) = O(\sqrt{k})\mathsf{OPT}$$

where the first inequality uses the upper bound by considering the worst case when only one terminal pair is removed in each iteration of the procedure. □

By the proof of Lemma 2.10, Lemma 2.2 also accounts for *duplicate* edges from junction trees rooted at different vertices.

2.3 The Steiner label cover problem and height reduction

In this section, we give the construction of the graph G' that consists of disjoint layered graphs, the undirected graph H which is a forest, the terminal set pairs $(\hat{S}_i, \hat{T}_i)$, and the relations $\hat{R}_i$.

We reduce the pairwise spanner problem to the *Steiner label cover* problem. A similar technique is introduced in [34] by a density argument which is amenable for the offline setting. For our purpose, we instead focus on the global problem that is tractable in the online setting. In what follows we construct a relevant Steiner label cover instance (recall Definition 2.3).

Constructing the layered graph G'. We reduce the pairwise spanner problem to the Steiner label cover problem as follows. Given $G = (V, E)$, for each vertex $r \in V$, we construct

$G_r = (V_r, E_r)$. Let

$$V_r := ((V \setminus \{r\}) \times \bigcup_{j \in [n-1]} \{-j, j\}) \cup \{(r, 0)\}$$

and

$$E_r := \{(u, j) \to (v, j+1) \mid (u, j), (v, j+1) \in V_r, (u, v) \in E\}.$$

For each edge $e \in E_r$, let the edge weight be 1. For each vertex $(u, -j)$ (respectively (u, j)) in V_r where $j \in [n-1] \cup \{0\}$, add a vertex $(u^-, -j)$ (respectively (u^+, j)), and create an edge $(u^-, -j) \to (u, -j)$ (respectively $(u, j) \to (u^+, j)$) with weight 0. This concludes the construction of G_r (see Figure 1 for an illustration).

Let G' be the disjoint union of G_r for $r \in V$. Given a pairwise spanner instance on G, we can construct a Steiner label cover instance on G'. For explicitness, we denote the vertex (u, j) in V_r by $(u, j)_r$. For each (s_i, t_i), let $S_i = \{(s_i^-, -j)_r \mid j > 0, r \in V \setminus \{s_i\}\} \cup (s_i^-, 0)_{s_i}$, $T_i = \{(t_i^+, j)_r \mid j > 0, r \in V \setminus \{t_i\}\} \cup (t_i^+, 0)_{t_i}$, and $R_i = \{((s_i^-, -j_s)_r, (t_i^+, j_t)_r) \mid r \in V, j_s + j_t \le d_i\}$ where we recall that d_i is the distance requirement for pair (s_i, t_i). Intuitively, a copy of s_i and a copy of t_i belongs to the relation R_i if 1) they are connected by a junction tree with the same root r, and 2) the distance between them is at most d_i in this junction tree.

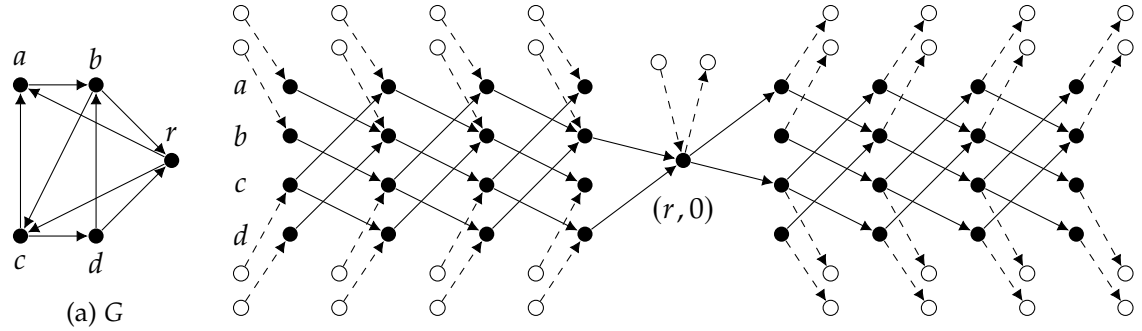


Figure 1: Construction of G_r given $G = (V, E)$ and $r \in V$

We can recover a pairwise spanner solution in G with at most the cost of the corresponding Steiner label cover solution in G'. Suppose we have the Steiner label cover solution in G', for each selected edge $(u, j) \to (v, j+1)$ in G_r, we select $u \to v$ for the pairwise spanner solution in G.

Claim 2.11. *There exists a junction tree solution for the pairwise spanner problem in G with at most the cost of a Steiner label cover solution in G'.*

We observe that any junction tree solution as a pairwise spanner in G corresponds to a Steiner label cover solution in G' and vice versa (see Figure 2 for an illustration). Let OPT_{junc} be the optimal value of a junction tree solution that counts duplicate edges from junction trees rooted at different vertices for the pairwise spanner problem in G. Then the optimum of the Steiner label cover solution in G' is at most $2\mathsf{OPT}_{junc}$. In each junction tree, each edge is used at most once in the in-arborescence and at most once in the out-arborescence.

Claim 2.12. *The optimal value of the Steiner label cover problem in G' is at most $2\mathsf{OPT}_{junc}$.*

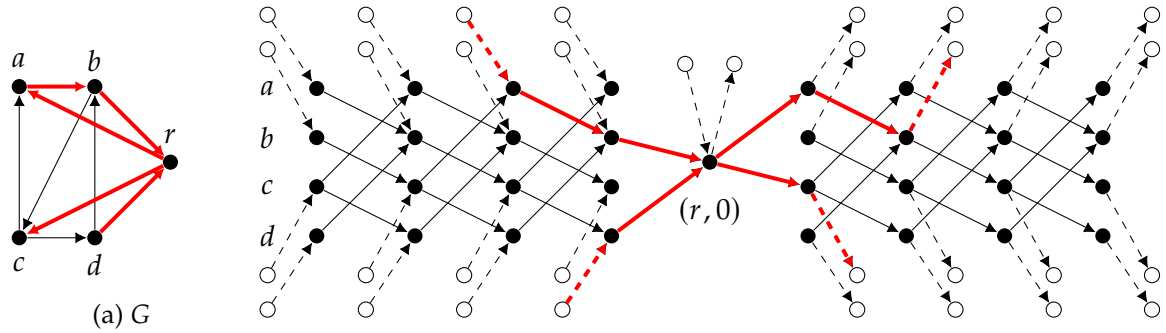


Figure 2: Suppose $(s_1, t_1) = (a, c)$, $d_1 = 3$, $(s_2, t_2) = (d, b)$, and $d_2 = 3$. We can find a junction tree rooted at r consisting of the red thick edges in G and G_r that connects the two terminal pairs and satisfies the distance requirements. Note that edge $a \to b$ is used in the in-arborescence and out-arborescence in G.

Constructing the Steiner label cover instance on the forest H. To find an approximate solution for the Steiner label cover instance, we further process the graph G' by the height reduction technique [32, 58]. The height reduction technique allows us to generate the forest H. The structure of H is later useful in Section 2.4 for deriving an LP formulation for the Steiner label cover problem.

Lemma 2.13. *([34]) Let $G = (V, E)$ be a directed graph where each edge is associated with a non-negative weight $w : E \to \mathbb{R}_{\geq 0}$. Let $r \in V$ be a root and $\sigma > 0$ be some parameter. Then we can efficiently construct an undirected tree T_r rooted at r' of height σ and size $|V|^{O(\sigma)}$ together with edge weights $w' : E(T_r) \to \mathbb{R}_{\geq 0}$ and a vertex mapping $\Psi : V(T_r) \to V$, such that*

1. *For any in-arborescence (out-arborescence) T in G rooted at r, there exists a subtree T' in T_r rooted at r' such that letting $L(T)$ and $L(T')$ be the set of leaves of T and T', respectively, we have $\Psi(L(T')) = L(T)$. Moreover,*
$$\sum_{e \in T} w(e) \leq O(\sigma |L(T)|^{1/\sigma}) \sum_{e \in T'} w'(e).$$
2. *Given any subtree T' in T_r rooted at r', we can efficiently find an in-arborescence (out-arborescence) T in G rooted at r, such that $\Psi(L(T')) = L(T)$ and $\sum_{e \in T'} w'(e) \leq \sum_{e \in T} w(e)$.*

For our purpose, for each $r \in V$, we construct a tree H_r from G_r by the height reduction technique. We recall that G_r is a layered graph with a center vertex $(r, 0)$. Let G_r^- denote the subgraph of G_r induced by the vertex set
$$\left(\bigcup_{v \in V \setminus \{r\}} \{v, v^-\} \times \{-j \mid j \in [n-1]\} \right) \cup \{(r, 0), (r^-, 0)\},$$

and similarly, let G_r^+ denote the subgraph of G_r induced by the vertex set

$$\left(\bigcup_{v \in V \setminus \{r\}} \{v, v^+\} \times [n-1]\right) \cup \{(r,0), (r^+,0)\}.$$

By Lemma 2.13, with $(r,0)$ being the root, we can construct a tree T_r^- rooted at r^- for G_r^- which approximately preserves the cost of any in-arborescence rooted at $(r,0)$ in G_r^- by a subtree in T_r^-, and similarly a tree T_r^+ rooted at r^+ for G_r^+ which approximately preserves the cost of any out-arborescence rooted at $(r,0)$ in G_r^+ by a subtree in T_r^+. We further add a super root r' and edges $\{r', r^-\}$ and $\{r', r^+\}$ both with weight 0. This concludes the construction of the weighted tree H_r (see Figure 3 for an illustration).

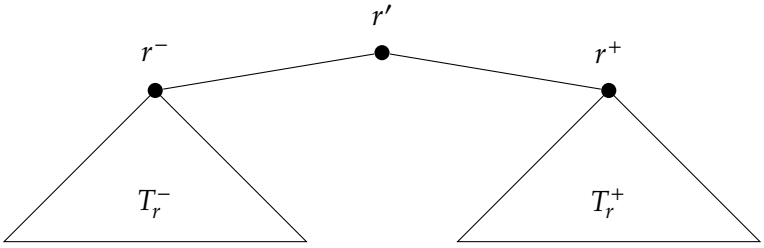


Figure 3: Construction of H_r

Let H be the disjoint union of H_r for $r \in V$. To prove Theorem 1.1, we show that we can achieve an $\tilde{O}(k^\delta)$-approximation for the Steiner label cover problem on graph H in an online manner. We set $\sigma = \lceil 1/\delta \rceil$ and apply Lemma 2.13 to obtain the weight $w' : E(H) \to \mathbb{R}_{\geq 0}$ and the mapping $\Psi_r : V(H_r) \to V(G_r)$ for all $r \in V$. Let Ψ be the union of the mappings for all $r \in V$. For each pair (s_i, t_i) in the original graph G, we recall that we focus on the vertex subset pair (S_i, T_i) of G' and its relation R_i which captures the distance requirement.

To establish the correspondence between the Steiner label cover instances, we clarify the mapping between the leaves of the arborescences in G' and H. Given a vertex $(s_i^-, -j)_r$ in $V(G')$ with a non-negative j, let $\Psi^{-1}((s_i^-, -j)_r)$ denote the set of leaves in T_r^- that maps to $(s_i^-, -j)_r$ by Ψ. Similarly, given a vertex $(t_i^+, j)_r$ in $V(G')$ with a non-negative j, let $\Psi^{-1}((t_i^+, j)_r)$ denote the set of leaves in T_r^+ that maps to $(t_i^+, j)_r$ by Ψ. The mapping Ψ^{-1} naturally defines the terminal sets of interest $\hat{S}_i := \Psi^{-1}(S_i) = \{\hat{s} \in V(H) \mid \Psi(\hat{s}) \in S_i\}$ and $\hat{T}_i := \Psi^{-1}(T_i) = \{\hat{t} \in V(H) \mid \Psi(\hat{t}) \in T_i\}$. The relation is also naturally defined: $\hat{R}_i := \{(\hat{s}, \hat{t}) \in \hat{S}_i \times \hat{T}_i \mid (\Psi(\hat{s}), \Psi(\hat{t})) \in R_i\}$. We note that for $\hat{s} \in \hat{S}_i$ and $\hat{t} \in \hat{T}_i$, $(\hat{s}, \hat{t})$ belongs to $\hat{R}_i$ only when $\hat{s}$ and $\hat{t}$ belong to the same tree H_r (see Figure 4 for an illustration).

Proving Theorem 2.5 and Theorem 1.1. We first show that the online Steiner label cover problem on H is indeed an ordered one. If we can obtain an α-competitive solution, then we can recover an $O(\alpha k^{1/2+\delta})$-competitive solution for the online pairwise spanner problem on G.

Having H, $(\hat{S}_i, \hat{T}_i)$, and $\hat{R}_i$ as aforementioned, we show how the orderings $\succ_{i,r}$ are generated so that we have the complete input for the ordered Steiner label cover problem. The ordering

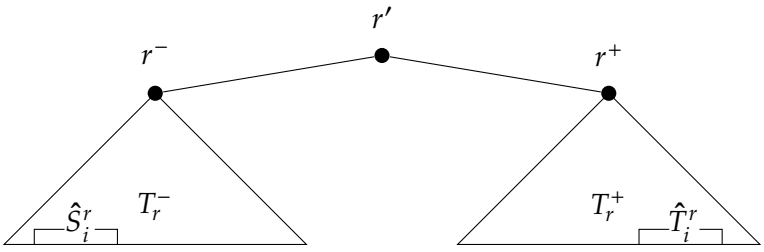


Figure 4: From pairwise spanner to Steiner label cover. In G, consider connecting s_i and t_i by using a junction tree rooted at r. In H_r, we consider terminal sets $\hat{S}_i^r := \hat{S}_i \cap H_r$ and $\hat{T}_i^r := \hat{T}_i \cap H_r$. A junction tree in G rooted at r corresponds to a subtree with leaves that belongs to $\hat{S}_i^r \cup \hat{T}_i^r$. The set $\hat{R}_i^r := \hat{R}_i \cap (\hat{S}_i^r \times \hat{T}_i^r)$ defines the admissible pairs, which maintains the distance requirement d_i in the layered graph G_r.

construction technique closely follows the one in [34]. The difference is that we generate an ordering for each tree for global approximation purpose while [34] focuses on one tree for greedy purpose.

In round i, let $\hat{S}_i^r := \hat{S}_i \cap H_r$ be the source leaves in $V(T_r^-)$ and $\hat{T}_i^r := \hat{T}_i \cap H_r$ be the sink leaves in $V(T_r^+)$. We have that $\hat{S}_i = \bigcup_{r \in V} \hat{S}_i^r$ and $\hat{T}_i = \bigcup_{r \in V} \hat{T}_i^r$ because H_r are disjoint for different $r \in V$ so $\{\hat{S}_i^r \mid r \in V\}$ forms a partition of $\hat{S}_i$ and a similar argument holds for $\hat{T}_i$. For the definition of the relation, let $\hat{R}_i^r := \hat{R}_i \cap (\hat{S}_i^r \times \hat{T}_i^r)$. Intuitively, we partition $\hat{S}_i$, $\hat{T}_i$, and $\hat{R}_i$ into $\hat{S}_i^r$, $\hat{T}_i^r$, and $\hat{R}_i^r$ based on the vertex $r \in V$, respectively.

We define the total ordering $\prec_{i,r}$ on $\hat{S}_i^r \cup \hat{T}_i^r$ as follows.

- For any $\hat{s} \in \hat{S}_i^r$ and any $\hat{t} \in \hat{T}_i^r$, we have that $\hat{s} \prec_{i,r} \hat{t}$.
- For any $\hat{s}, \hat{s}' \in \hat{S}_i^r$ where $\Psi(\hat{s}) = (s_i^-, -j)_r$ and $\Psi(\hat{s}') = (s_i^-, -j')_r$, we have that $\hat{s} \prec_{i,r} \hat{s}'$ if and only if $j > j'$, i. e., under the mapping Ψ, the one that is farther from $(r, 0)$ in G' has a lower rank according to $\prec_{i,r}$. Ties are broken arbitrarily.
- For any $\hat{t}', \hat{t} \in \hat{T}_i^r$ where $\Psi(\hat{t}') = (t_i^+, j')_r$ and $\Psi(\hat{t}) = (t_i^+, j)_r$, we have that $\hat{t}' \prec_{i,r} \hat{t}$ if and only if $j' < j$, i. e., under the mapping Ψ, the one that is farther from $(r, 0)$ in G' has a higher rank according to $\prec_{i,r}$. Ties are broken arbitrarily.

For ease of notion, we use $\preceq_{i,r}$ to denote inclusive lower order, i. e., when $\hat{s} \preceq_{i,r} \hat{s}'$, it is either $\hat{s} = \hat{s}'$ or $\hat{s} \prec_{i,r} \hat{s}'$. If the pair $(\hat{s}, \hat{t}) \in \hat{R}_i^r$ corresponds to an interval, then any subinterval pair $(\hat{s}', \hat{t}')$ where $\hat{s} \preceq_{i,r} \hat{s}'$ and $\hat{t}' \preceq_{i,r} \hat{t}$ also belongs to the relation $\hat{R}_i^r$. This is by the construction of the distance-based relations R_i and $\hat{R}_i$ which implies that the Steiner label cover problem is ordered by $\prec_{i,r}$.

Claim 2.14. *If $(\hat{s}, \hat{t}) \in \hat{R}_i^r$, then for any $\hat{s}' \in \hat{S}_i^r$ and $\hat{t}' \in \hat{T}_i^r$ such that $\hat{s} \preceq_{i,r} \hat{s}' \prec_{i,r} \hat{t}' \preceq_{i,r} \hat{t}$, we have that $(\hat{s}', \hat{t}') \in \hat{R}_i^r$.*

Now we are ready to show Theorem 2.5.

Theorem 2.5. *For any constant $\delta > 0$, an α-competitive polynomial-time algorithm for online ordered Steiner label cover on a forest implies an $O(\alpha k^{1/2+\delta})$-competitive polynomial-time algorithm for the online pairwise spanner problem on a directed graph with uniform edge lengths.*

Proof. Suppose we have an ordered Steiner label cover instance on H, $(\hat{S}_i, \hat{T}_i)$, $\hat{R}_i$, and $\succ_{i,r}$ as aforementioned and we have an α-competitive solution. From the structure of H, the online solution is guaranteed to be a forest. By Lemma 2.13, we can recover an online Steiner label cover solution in G' from that in H, by losing a factor of $O(k^\delta)$. These imply that we have an $O(\alpha k^\delta)$-competitive solution for the online Steiner label cover problem in G'.

We recall that by Claim 2.12, the Steiner label cover solution in G' pays at most twice of the optimal junction tree solution for pairwise spanners in G with value OPT_{junc} (which also counts the duplicate edges of junction trees rooted at different vertices). By Lemma 2.2, we have that OPT_{junc} is $O(\sqrt{k})\mathsf{OPT}$ where OPT is the optimal value of the pairwise spanner in G.

By Claim 2.11, we can recover a pairwise spanner solution in G from an online Steiner label cover solution in G'. Putting the results together, we obtain an online Steiner label cover solution in G' with cost at most $O(\alpha k^\delta)\mathsf{OPT}_{junc}$ which recovers an online pairwise spanner solution with cost at most $O(\alpha k^{1/2+\delta})\mathsf{OPT}$. □

The remaining is to show that there exists a randomized polylogarithmically competitive online algorithm for the ordered Steiner label cover problem on the forest H, with $(\hat{S}_i, \hat{T}_i)$ and $\hat{R}_i$ arriving online. We show the following lemma in Section 2.4.

Lemma 2.15. *For the online ordered Steiner label cover problem on a forest (see Definition 2.4), there is a randomized polynomial-time algorithm with competitive ratio* $\mathsf{polylog}(n)$.

Combining Lemma 2.6 and Theorem 2.5 implies Theorem 1.1.

2.4 Online ordered Steiner label cover on a forest

This section is devoted to proving Lemma 2.6.

We start with the high level sketch of the proof. In the original work for the offline setting [34], an LP-based approach is used for searching a minimum density junction tree, which picks the best solution among the roots $r \in V$ and repeats. For the online setting, we use an LP that captures the global approximate solution. The constraints of the LP involve the terminal pairs $(\hat{S}_i, \hat{T}_i)$, which is a non-cross-product relation that requires a more meticulous rounding scheme. Fortunately, we are able to use a similar technique to [34] that generates the terminal vertex orderings, extracts representatives of the sets of terminals according to the orderings, and reduces the ordered Steiner label cover problem to the undirected group Steiner forest problem. We employ the online buy-at-bulk framework [28] to solve the final undirected group Steiner forest instance on the forest with specific structure.

2.4.1 An LP-based approach for Steiner label cover

We recall that $(\hat{S}_i, \hat{T}_i)$ is the terminal pair that consists of leaves in the forest H, where H is the disjoint union of trees H_r for $r \in V$. Given $(\hat{s}, \hat{t}) \in \hat{R}_i$, let $r \in V$ be such that $\hat{s}$ and $\hat{t}$ are leaves of H_r. Let $r(\hat{s}) = r(\hat{t})$ denote the root r' of this tree H_r. The goal is to formulate an LP that fractionally picks the edges to pack the $\hat{s}$-$\hat{t}$ paths where $(\hat{s}, \hat{t})$ belongs to relation $\hat{R}_i$. We recall that H has $\sigma = \lceil 1/\delta \rceil$ layers, so $E(H) = n^{O(1/\delta)}$ and $|\hat{R}_i| = n^{O(1/\delta)}$. We use a natural LP relaxation for the problem described in Lemma 2.6:

$$
\begin{aligned}
\min_{x,y,z} \quad & \sum_{e \in E(H)} w'(e) x_e & & \\
\text{subject to} \quad & \sum_{(\hat{s},\hat{t}) \in \hat{R}_i} y_{\hat{s},\hat{t}} \geq 1 & & \forall i \in [k], \\
& \sum_{\hat{t} | (\hat{s},\hat{t}) \in \hat{R}_i} y_{\hat{s},\hat{t}} \leq z_{\hat{s}} & & \forall i \in [k], \hat{s} \in \hat{S}_i, \\
& \sum_{\hat{s} | (\hat{s},\hat{t}) \in \hat{R}_i} y_{\hat{s},\hat{t}} \leq z_{\hat{t}} & & \forall i \in [k], \hat{t} \in \hat{T}_i, \\
& x \text{ supports an } \hat{s}\text{-}r(\hat{s}) \text{ flow of value } z_{\hat{s}} & & \forall i \in [k], \hat{s} \in \hat{S}_i, \\
& x \text{ supports an } \hat{t}\text{-}r(\hat{t}) \text{ flow of value } z_{\hat{t}} & & \forall i \in [k], \hat{t} \in \hat{T}_i, \\
& x_e \geq 0 & & \forall e \in E(H), \\
& y_{\hat{s},\hat{t}} \geq 0 & & \forall i \in [k], (\hat{s},\hat{t}) \in \hat{R}_i.
\end{aligned}
\tag{2.2}
$$

This LP is a relaxation by considering an integral solution x, y, z where $x_e, y_{\hat{s},\hat{t}}, z_{\hat{s}}, z_{\hat{t}} \in \{0, 1\}$. x_e is an indicator of edge e, $y_{\hat{s},\hat{t}}$ is an indicator of the $\hat{s}$-$\hat{t}$ path, while $z_{\hat{s}}$ and $z_{\hat{t}}$ denote the $\hat{s}$-$r(\hat{s})$ and $\hat{t}$-$r(\hat{t})$ *flow* value, respectively. Suppose we have an integral solution for Steiner label cover, then for each $(\hat{S}_i, \hat{T}_i)$, there must exists a $y_{\hat{s},\hat{t}} = 1$ where $(\hat{s}, \hat{t}) \in \hat{R}_i$, which also indicates that all edges e of the $\hat{s}$-$\hat{t}$ path must satisfy $x_e = 1$ and there is an $\hat{s}$-$\hat{t}$ flow with value 1. The first constraint ensures that for each pair $(\hat{S}_i, \hat{T}_i)$, there must be an $\hat{s}$-$\hat{t}$ path that belongs to the relation $\hat{R}_i$, i. e., $\hat{s} \in \hat{S}_i$, $\hat{t} \in \hat{T}_i$, and $(\hat{s}, \hat{t}) \in \hat{R}_i$. In the second constraint, for each $i \in [k]$, $z_{\hat{s}}$ denotes an upper bound for the total number of paths that have $\hat{s}$ as the source, where each such $\hat{s}$-$\hat{t}$ path satisfies $(\hat{s}, \hat{t}) \in \hat{R}_i$. In the fourth constraint, the path upper bound $z_{\hat{s}}$ is subject to the capacity x, i. e., the edge indicator x is naturally an upper bound that packs the $\hat{s}$-$\hat{t}$ paths which belong to the relation. The third and the fifth constraints are defined similarly.[2]

In round i, $(\hat{S}_i, \hat{T}_i)$ and $\hat{R}_i$ arrive. The goal is to update x in a non-decreasing manner so that

[2]We thank the anonymous reviewer for suggesting to us the following simpler covering LP formulation similar to the one in [74]:

$$
\text{minimize} \sum_{e \in E(H)} w'(e) x_e \text{ over } x_e \geq 0 \quad \forall e \in E(H) \text{ s.t.} \sum_{(\hat{s},\hat{t}) \in \hat{R}_i} \min\{MC(\hat{s}, r(\hat{s})), MC(\hat{t}, r(\hat{t}))\} \geq 1 \quad \forall i \in [k]
$$

the constraints are satisfied with some underlying variables $y_{\hat{s},\hat{t}}$ (which is initially set to 0), and $z_{\hat{s}}$ and $z_{\hat{t}}$ where $(\hat{s},\hat{t}) \in \hat{R}_i$, such that the objective is approximately optimal.

We convert LP (2.2) into a covering LP. Suppose in round i, we are given the edge capacity x. Let $P(\hat{s})$ denote the set of edges in the $\hat{s}$-$r(\hat{s})$ path and $P(\hat{t})$ denote the set of edges in the $\hat{t}$-$r(\hat{t})$ path. By setting the first constraint as the objective, merging the second and fourth constraint, and merging the third and fifth constraint, we derive the following internal offline LP,

$$\begin{aligned} \max_{y} \quad & \sum_{(\hat{s},\hat{t})\in\hat{R}_i} y_{\hat{s},\hat{t}} \\ \text{subject to} \quad & \sum_{\hat{t}|(\hat{s},\hat{t})\in\hat{R}_i} y_{\hat{s},\hat{t}} \le x_e \quad \forall \hat{s} \in \hat{S}_i, \forall e \in P(\hat{s}), \\ & \sum_{\hat{s}|(\hat{s},\hat{t})\in\hat{R}_i} y_{\hat{s},\hat{t}} \le x_e \quad \forall \hat{t} \in \hat{T}_i, \forall e \in P(\hat{t}), \\ & y_{\hat{s},\hat{t}} \ge 0 \qquad\qquad \forall (\hat{s},\hat{t}) \in \hat{R}_i, \end{aligned} \tag{2.3}$$

and its dual

$$\begin{aligned} \min_{\alpha} \quad & \sum_{e\in E(H)} x_e \left(\sum_{\hat{s}|e\in P(\hat{s})} \alpha_{\hat{s},e} + \sum_{\hat{t}|e\in P(\hat{t})} \alpha_{\hat{t},e} \right) \\ \text{subject to} \quad & \sum_{e\in P(\hat{s})} \alpha_{\hat{s},e} + \sum_{e\in P(\hat{t})} \alpha_{\hat{t},e} \ge 1 \qquad \forall (\hat{s},\hat{t}) \in R_i, \\ & \alpha_{\hat{s},e} \ge 0 \qquad \forall \hat{s} \in \hat{S}_i, \forall e \in P(\hat{s}), \\ & \alpha_{\hat{t},e} \ge 0 \qquad \forall \hat{t} \in \hat{T}_i, \forall e \in P(\hat{t}). \end{aligned} \tag{2.4}$$

We can solve LP (2.3) and LP (2.4) directly because there are only polynomially many constraints. If the objective value is at least 1, then x is *good*, i. e., there exists an $\hat{s} \rightsquigarrow \hat{t}$ path. Otherwise, x is *bad*.

To solve LP (2.2), we check if x is good or bad by solving LP (2.4). If x is good, then there exists y and z such that all the constraints of LP (2.2) are satisfied, so we move on to the next round. Otherwise, x is bad, so we increment x until it becomes good, which implies that

$$\sum_{e\in E(H)} x_e \left(\sum_{\hat{s}|e\in P(\hat{s})} \alpha_{\hat{s},e} + \sum_{\hat{t}|e\in P(\hat{t})} \alpha_{\hat{t},e} \right) \ge 1$$

where $MC(u,v)$ denotes the minimum u-v cut value. This is a covering LP because one can consider all possible combinations by picking either $\hat{s}$ or $\hat{t}$ for each $(\hat{s},\hat{t}) \in \hat{R}_i$ as the constraints. A subroutine that computes the sum of the lowest min-cut combination can be used as a separation oracle. In LP (2.2), $y_{\hat{s},\hat{t}}$ was later used to extract representative terminal vertices for the undirected group Steiner forest problem. One can also use the above LP to get x then use LP (2.3) to get y to extract the representative terminal vertices.

for all feasible α in LP (2.4). Let A_i be the feasible polyhedron of LP (2.4) in round i. We derive the following LP which is equivalent to LP (2.2), by considering all the constraints of LP (2.4) from round 1 to round k.

$$\begin{aligned} \min_{x} \quad & \sum_{e\in E(H)} w'(e)x_e \\ \text{subject to} \quad & \sum_{e\in E(H)} x_e \left(\sum_{\hat{s}|e\in P(\hat{s})} \alpha_{\hat{s},e} + \sum_{\hat{t}|e\in P(\hat{t})} \alpha_{\hat{t},e} \right) \geq 1 \quad \forall i \in [k], \alpha \in A_i, \\ & x_e \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall e \in E(H). \end{aligned} \tag{2.5}$$

In round $i \in [k]$, the subroutine that solves LP (2.4) and checks if the optimum is good or not, is the separation oracle used for solving LP (2.5) online. Here we use Theorem 4.1 (the formal version of Theorem 1.5) to show that LP (2.5) can be solved online in polynomial time by paying an $O(\log n)$ factor. This requires that $\log(1/\alpha_{\hat{s},e})$, $\log(1/\alpha_{\hat{t},e})$, and $\log \mathsf{LP}^*$ where LP^* is the optimum of LP (2.5), can be represented by polynomial number of bits used for the edge weights w'. $\log \mathsf{LP}^*$ can be represented by polynomial number of bits. For $\log(1/\alpha_{\hat{s},e})$ and $\log(1/\alpha_{\hat{t},e})$, the subroutine that solves LP (2.4) returns a solution α which is represented by polynomial number of bits. By Theorem 4.1, we have the following Lemma.

Lemma 2.16. *There exists a polynomial-time $O(\log n)$-competitive online algorithm for LP (2.2).*

2.4.2 Online rounding

Now we are ready to show how to round the solution of LP (2.2) online. The rounding scheme consists of two main steps. First we extract the representative sets according to the orderings $\succ_{i,r}$ and the online solution of LP (2.2), then reduce the online ordered Steiner label cover instance to an online undirected group Steiner forest instance.

We recall that in round i, we have terminal sets $\hat{S}_i^r := \hat{S}_i \cap H_r$ and $\hat{T}_i^r := \hat{T}_i \cap H_r$, and the relations $\hat{R}_i^r := \hat{R}_i \cap (\hat{S}_i^r \times \hat{T}_i^r)$. The terminal sets and relations are partitioned based on the trees H_r where $r \in V$. We also have orderings $\succ_{i,r}$ on $\hat{S}_i^r \cup \hat{T}_i^r$.

Extracting representative sets. Let OPT be the optimum of the Steiner label cover instance on H. Our goal is to find an integral solution of LP (2.2) online with an objective value at most $\mathsf{polylog}(n)\mathsf{OPT}$. We first use the online LP solution to find the representative vertex sets, then use the representatives to reduce to an online undirected group Steiner forest instance.

For each $i \in [k]$ and $r \in V$, we need to find the representative sets $\tilde{S}_i^r \subseteq \hat{S}_i^r$ and $\tilde{T}_i^r \subseteq \hat{T}_i^r$ such that $\tilde{S}_i^r \times \tilde{T}_i^r \subseteq \hat{R}_i^r$. Suppose in round i, we have a fractional solution x, y, z of LP (2.2). The set pruning is accomplished by focusing on each $r \in V$ and its ordering $\prec_{i,r}$, and taking all terminal vertices according to their contribution to the objective value until we reach the one that cumulatively contributes half of the objective. Formally, for each $r \in V$:

- Let $\gamma_i^r := \sum_{(\hat{s},\hat{t})\in\hat{R}_i^r} y_{\hat{s},\hat{t}}$.

- Choose the median sets, i. e., define the boundary vertices

$$\hat{s}_{i,r} := \max_{\prec_{i,r}} \left\{ \hat{s} \in \hat{S}_i^r \;\middle|\; \sum_{\hat{s}' \succeq_{i,r} \hat{s}} \sum_{\hat{t} \mid (\hat{s}', \hat{t}) \in \hat{R}_i^r} y_{\hat{s}', \hat{t}} \geq \gamma_i^r / 2 \right\}$$

and

$$\hat{t}_{i,r} := \min_{\prec_{i,r}} \left\{ \hat{t} \in \hat{T}_i^r \;\middle|\; \sum_{\hat{t}' \preceq_{i,r} \hat{t}} \sum_{\hat{s} \mid (\hat{s}, \hat{t}') \in \hat{R}_i^r} y_{\hat{s}, \hat{t}'} \geq \gamma_i^r / 2 \right\},$$

and let $\tilde{S}_i^r := \{\hat{s} \mid \hat{s} \succeq_{i,r} \hat{s}_{i,r}\}$ and $\tilde{T}_i^r := \{\hat{t} \mid \hat{t} \preceq_{i,r} \hat{t}_{i,r}\}$.

The choice of the median sets guarantees that at least half of the LP value is preserved. We have to verify that the union of cross-products of these sets does not contain any pairs that are disallowed by the relation $\hat{R}_i$. The following lemma closely follows the pairwise spanner framework in [34]. The main difference is that the framework focuses on one tree for greedy purposes, while we focus on all trees for global approximation purpose.

Lemma 2.17. *The union of the cross-products of the representative sets $\tilde{S}_i^r$ and $\tilde{T}_i^r$ is a subset of $\hat{R}_i$, i. e.,*

$$\bigcup_{r \in V} \left(\tilde{S}_i^r \times \tilde{T}_i^r \right) \subseteq \hat{R}_i.$$

Proof. The special case of a single tree (i. e., a single r) was proven in [34]. That is, for fixed $r \in V$, [34] implies that $\tilde{S}_i^r \times \tilde{T}_i^r \subseteq \hat{R}_i^r$.

Lemma 2.18. *The cross-product of the representative sets $\tilde{S}_i^r$ and $\tilde{T}_i^r$ is a subset of $\hat{R}_i^r$, i. e.,*

$$\tilde{S}_i^r \times \tilde{T}_i^r \subseteq \hat{R}_i^r.$$

Proof. Let $U_i^r := \{\hat{s} \mid \hat{s} \preceq_{i,r} \hat{s}_{i,r}\}$ be the prefix set according to $\prec_{i,r}$. By the choice of $\hat{s}_{i,r}$ and the definition of γ_i^r, we have that

$$\sum_{\hat{s} \in U_i^r} \sum_{\hat{t} \mid (\hat{s}, \hat{t}) \in \hat{R}_i^r} y_{\hat{s}, \hat{t}} = \gamma_i^r - \sum_{\hat{s} \succ_{i,r} \hat{s}_{i,r}} \sum_{\hat{t} \mid (\hat{s}, \hat{t}) \in \hat{R}_i^r} y_{\hat{s}, \hat{t}} > \gamma_i^r - \frac{\gamma_i^r}{2} = \frac{\gamma_i^r}{2}.$$

Let $\hat{t}_{\max} := \max_{\prec_{i,r}} \{\hat{t} \mid (\hat{s}_{i,r}, \hat{t}) \in \hat{R}_i^r\}$. We have that for any $\hat{s} \in U_i^r$, if $\hat{t}$ is such that $(\hat{s}, \hat{t}) \in \hat{R}_i^r$, then $\hat{t} \preceq_{i,r} \hat{t}_{\max}$. Otherwise, suppose $\hat{t} \succ_{i,r} \hat{t}_{\max}$, then by Claim 2.14, $(\hat{s}, \hat{t}) \in \hat{R}_i^r$ implies that $(\hat{s}_{i,r}, \hat{t}) \in \hat{R}_i^r$, this contradicts to the definition of $\hat{t}_{\max}$. We have that

$$\sum_{\hat{t} \preceq_{i,r} \hat{t}_{\max}} \sum_{\hat{s} \mid (\hat{s}, \hat{t}) \in \hat{R}_i^r} y_{\hat{s}, \hat{t}} \geq \sum_{\hat{t} \preceq_{i,r} \hat{t}_{\max}} \sum_{\substack{\hat{s} \in U_i^r \\ (\hat{s}, \hat{t}) \in \hat{R}_i^r}} y_{\hat{s}, \hat{t}} = \sum_{\hat{s} \in U_i^r} \sum_{\hat{t} \mid (\hat{s}, \hat{t}) \in \hat{R}_i^r} y_{\hat{s}, \hat{t}} > \frac{\gamma_i^r}{2}$$

which implies that $\hat{t}_{\max} \succeq_{i,r} \hat{t}_{i,r}$ and $\tilde{T}_i^r \subseteq \{\hat{t} \mid \hat{t} \prec_{i,r} \hat{t}_{\max}\}$. By Claim 2.14, $(\hat{s}_{i,r}, \hat{t}_{\max}) \in \hat{R}_i^r$ implies that $\tilde{S}_i^r \times \tilde{T}_i^r \subseteq \tilde{R}_i^r$. □

Taking the union gives the desired claim. □

From ordered Steiner label cover to group Steiner forest. Given the representative sets, we reduce the online ordered Steiner label cover problem to the online undirected group Steiner forest problem as follows. For round i, let the representative sets be $\tilde{S}_i := \bigcup_{r \in V} \tilde{S}_i^r$ and $\tilde{T}_i := \bigcup_{r \in V} \tilde{T}_i^r$. We connect $\tilde{S}_i$ and $\tilde{T}_i$. It suffices to consider the cross product of $\tilde{S}_i$ and $\tilde{T}_i$ since H is partitioned by H_r, i. e., representative vertices from different trees in H cannot be connected.

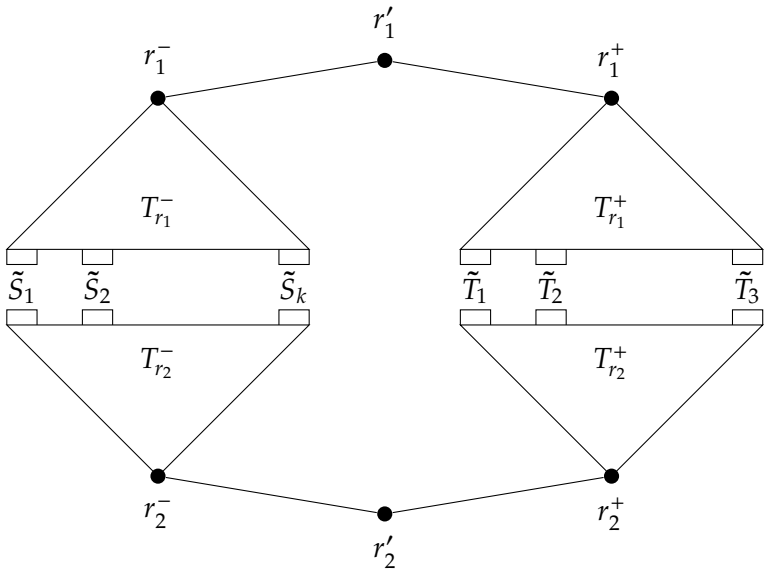


Figure 5: Connecting $\tilde{S}_i$ and $\tilde{T}_i$ in the undirected group Steiner forest problem on H

The goal is to select edges from $E(H)$, such that there is an $\hat{s}$-$\hat{t}$ path for $\hat{s} \in \hat{S}_i$ and $\hat{t} \in \hat{T}_i$ for all $i \in [k]$ and the total edge weight is approximately optimal.

The online buy-at-bulk framework [28] considers a height reduction forest and uses the single-source (sink) group Steiner tree problem [3] as a black-box to solve the online directed buy-at-bulk problem. Here, we apply this approach to our online undirected group Steiner forest instance.

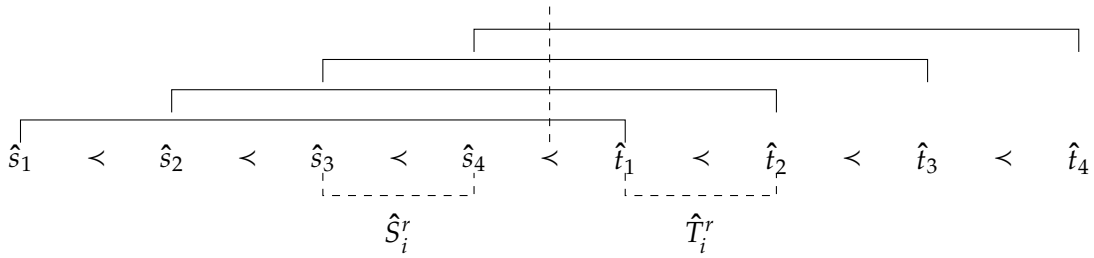
Theorem 2.19. *([28]) Given the graph H and the terminal sets $\tilde{S}_i$ and $\tilde{T}_i$ for $i \in [k]$ as the undirected group Steiner forest instance, there is a randomized polynomial-time algorithm with competitive ratio* $\mathsf{polylog}(n)$.

We recall that OPT is the optimum of the Steiner label cover instance on H with terminal set pairs $(\hat{S}_i, \hat{T}_i)$ and relations $\hat{R}_i$. Using Theorem 2.19, we show the following which completes the proof for Lemma 2.6.

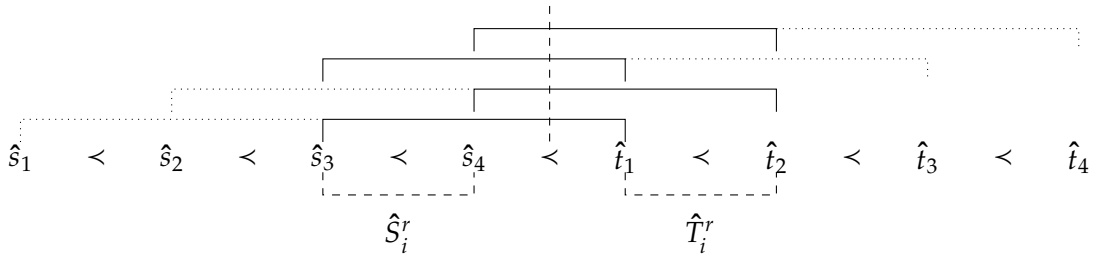
Lemma 2.20. *There exists a randomized polynomial-time online algorithm for LP (2.2) which returns an integral solution such that the objective value is at most* $\mathsf{polylog}(n)\mathsf{OPT}$.

Proof. Let x, y, and z be a feasible solution for LP (2.2) with objective value LP. By Claim 2.14 and Lemma 2.17, we can obtain a feasible solution x', y', and z' for LP (2.2) with objective value

at most 2LP as follows. For any $r \in V$ and $(\hat{s}, \hat{t}) \in \hat{R}_i^r$, there is a corresponding variable $y_{\hat{s},\hat{t}}$ for the $\hat{s}$-r'-$\hat{t}$ path. If $\hat{s} \notin \tilde{S}_i^r$, then we replace the $\hat{s}$-r' path by an $\hat{s'}$-r' path where $\hat{s'} \in \tilde{S}_i^r$; if $\hat{t} \notin \tilde{T}_i^r$, then we replace the r'-$\hat{t}$ path by an r'-$\hat{t'}$ path where $\hat{t'} \in \tilde{T}_i^r$. This shifts the path selection y towards the paths with representative endpoints $\hat{s} \in \tilde{S}_i^r$ and $\hat{t} \in \tilde{T}_i^r$ and addresses the new path assignment y' (see Figure 6 for an illustration). We obtain x' by adjusting the capacity. For each edge e that belongs to an $\hat{s}$-$\hat{t}$ path where $(\hat{s}, \hat{t}) \in \tilde{S}_i^r \times \tilde{T}_i^r$, we have that $x'_e \leq 2x_e$ because the representatives contribute at least half of the original LP path selection. If e does not belong the these paths, then $x'_e = 0$. We also adjust the flow value and obtain z' accordingly.



(a) The original path selection by LP (2.2)



(b) The adjusted path selection with representative endpoints for LP (2.2)

Figure 6: Shifting paths towards paths with representative endpoints

Now we consider the following LP for undirected group Steiner forest on H.

$$\begin{aligned}
\min_{x,\gamma} \quad & \sum_{e \in E(H)} w'(e) x_e \\
\text{subject to} \quad & \sum_{r \in V} \gamma_i^r \geq 1 && \forall i \in [k], \\
& x \text{ supports a } \tilde{S}_i^r\text{-}r' \text{ flow of value } \gamma_i^r && \forall i \in [k], \forall r \in V, \\
& x \text{ supports a } \tilde{T}_i^r\text{-}r' \text{ flow of value } \gamma_i^r && \forall i \in [k], \forall r \in V, \\
& x_e \geq 0 && \forall e \in E(H'), \\
& \gamma_i^r \geq 0 && \forall i \in [k], r \in V.
\end{aligned} \tag{2.6}$$

Let $\tilde{x}_e = x'_e$ for each $e \in E(H)$ and $\tilde{\gamma}_i^r = \sum_{(\hat{s},\hat{t}) \in \tilde{R}_i^r} y'_{\hat{s},\hat{t}}$. We observe that $\tilde{x}$ and $\tilde{\gamma}$ is a feasible

solution for LP (2.6) with objective value at most 2LP. This implies that for any solution of LP (2.2), there exists a corresponding solution of LP (2.6) with at most twice of the objective value.

Given a solution $\tilde{x}$ and $\tilde{\gamma}$ for LP (2.6), we can obtain a solution x', y', and z' for LP (2.2) as follows. We note that the objective values of these two solutions are the same.

- Let $x'_e = \tilde{x}_e$ for each $e \in E(H)$.
- Let $z'_{\hat{s}} = \min_{e \in P(\hat{s})}\{x'_e\}$, i. e., $z'_{\hat{s}}$ is the minimum capacity among the edges that belong to the path from $\hat{s}$ to its tree root $r(\hat{s})$.
- Let $z'_{\hat{t}} = \min_{e \in P(\hat{t})}\{x'_e\}$, i. e., $z'_{\hat{t}}$ is the minimum capacity among the edges that belong to the path from the tree root $r(\hat{t})$ to $\hat{t}$.
- Let $f^r_{s_i}$ be an $\tilde{S}^r_i$-r' flow of value γ^r_i and $f^r_{t_i}$ be an $\tilde{T}^r_i$-r' flow of value γ^r_i, both supported by edge capacity $\tilde{x}$.
- For each $i \in [k]$ and $(\hat{s}, \hat{t}) \in \tilde{R}_i$, let $y'_{\hat{s},\hat{t}}$ be the flow value along the $\hat{s}$-$\hat{t}$ path according to the flow f.

Now we use Theorem 2.19 to obtain an integral solution for LP (2.6), then transform this solution to the one for LP (2.2) as described above. Suppose we have a polylog(n)-competitive solution for the undirected group Steiner forest instance. For each edge $e \in E(H)$ that is selected, set $\tilde{x}_e = 1$, otherwise let $\tilde{x}_e = 0$. Let $\gamma^r_i = 1$ whenever there is an $\hat{s}$-r'-$\hat{t}$ path in H where $\hat{s} \in \tilde{S}_i$ and $\hat{t} \in \tilde{T}_i$, otherwise let $\gamma^r_i = 0$. We observe that every integral solution of LP (2.6) corresponds to an undirected group Steiner forest solution for H and (s_i, t_i). By the structure of H, the integrality gap of LP (2.6) is polylog(n) [50, 28]. This implies that the objective values obtained from the online undirected group Steiner forest solution for both LP (2.2) and LP (2.6) are upper-bounded by polylog(n)OPT. □

2.5 Summary

We summarize in Algorithm 1 the overall $\tilde{O}(k^{1/2+\delta})$-competitive randomized polynomial-time algorithm described in Section 2.3 and Section 2.4 for pairwise spanners on graphs with uniform edge lengths.

Algorithm 1 $\tilde{O}(k^{1/2+\delta})$-competitive pairwise spanner

1: Given a directed graph G, construct the directed graph G', which is a union of disjoint layered graphs. From G', construct the undirected graph H, which is a forest.
2: **for** an arriving pair (s_i, t_i) **do**
3: Generate $\hat{S}_i$, $\hat{T}_i$, and $\hat{R}_i$, and solve the Steiner label cover LP (2.2) online.[3]
4: Generate the orderings $\prec_{i,r}$. From the online solution of LP (2.2), extract the representative sets $\tilde{S}_i$ and $\tilde{T}_i$, and the corresponding relation $\tilde{R}_i$.
5: Solve the online Steiner forest instance for graph H and the terminal vertex set pair $\tilde{S}_i$ and $\tilde{T}_i$. Recover an integral solution for LP (2.2), obtain a solution which is a forest for the ordered Steiner label cover problem in H, and map it back to the collection of junction trees in G.
6: **end for**

We conclude that Algorithm 1 is $\tilde{O}(k^{1/2+\delta})$-competitive. We pay $O(\sqrt{k})$ for the junction tree approximation, $O(k^{\delta})$ for constructing the forest H by height reduction, and finally $\mathsf{polylog}(n)$ for solving LP (2.2) and the undirected group Steiner forest instance H, $\tilde{S}_i$, and $\tilde{T}_i$ online. The overall competitive ratio is $\tilde{O}(k^{1/2+\delta})$.

3 Online Pairwise Spanners

We recall that in the general *pairwise spanner* problem, we are given a directed graph $G = (V, E)$ with edge length $\ell : E \to \mathbb{R}_{\geq 0}$, a general set of k terminals $D = \{(s_i, t_i) \mid i \in [k]\} \subseteq V \times V$, and a target distance d_i for each terminal pair (s_i, t_i), the goal is to output a subgraph $H = (V, E')$ of G such that for every pair $(s_i, t_i) \in D$ it is the case that $d_H(s_i, t_i) \leq d_i$, i. e., the distance of a shortest $s_i \rightsquigarrow t_i$ path is at most d_i in the subgraph H, and we want to minimize the number of edges in E'.

3.1 An $\tilde{O}(n^{4/5})$-competitive online algorithm for pairwise spanners

In this section, we prove Theorem 1.2. Recall that in the online setting of the problem, the directed graph G is given offline, while the vertex pairs in $D \subseteq V \times V$ arrive online one at a time. In the beginning, $E' = \emptyset$. Suppose (s_i, t_i) and its target distance d_i arrive in round i, we select some edges from E and irrevocably add them to E', such that in the subgraph $H = (V, E')$, $d_H(s_i, t_i) \leq d_i$. The goal is to approximately minimize the total number of edges added to E'.

We start with a high-level sketch of an offline algorithm, which we will build on for the online setting. The randomized rounding framework in [34, 20] has two main steps. One step is to solve and round an LP for the spanner problem. The second is to uniformly sample vertices and add the shortest path in-arborescences and out-arborescences rooted at each of the sampled vertices. Terminal pairs are classified as either *thin* or *thick* and are addressed by one of the two steps above accordingly.

[3] One may derive an LP for the graph G' instead of H. However, the integrality gap might be large.

In the first step, the rounding scheme based on an LP solution for spanners ensures with high probability that for all thin terminal pairs the distance requirements are met. The second step ensures with high probability that for all thick terminal pairs the distance requirements are met. By selecting an appropriate threshold for classifying the thin and thick pairs, this leads to an $O(n/\sqrt{\mathsf{OPT}})$-approximation, where OPT is the number of edges in the optimal solution.

The main challenges in adapting this approach to the online setting are as follows: 1) OPT can be very small, and 2) the LP for spanners is not naturally a pure covering LP, which makes it difficult to solve online. In the previous work in the offline setting, the small-OPT case is addressed by sophisticated strategies that appear difficult to emulate online. Instead, we show that the optimal value (however small) is at least the square root of the number of terminal pairs that have arrived. Thus, if OPT is small and not many pairs have arrived, we can greedily add a path with the fewest edges subject to the distance requirement for each pair. To overcome the second challenge, we convert the LP for spanners into an equivalent covering LP as in [39], where violating covering constraints are generated by an auxiliary LP. Having transformed the LP into a purely covering one, we can solve the LP online, treating the auxiliary LP as a separation oracle.

3.1.1 A simple $\tilde{O}(n^{4/5})$-approximate offline algorithm based on [34]

For ease of exposition, we first design a simpler offline algorithm (slightly weaker than the state-of-the-art) that is more amenable to the online setting. This allows us to establish the main ingredients governing the approximation factor in a simpler setting, and then address the online aspects separately. The algorithm leverages the framework developed in [20, 34].

Let $\mathcal{P}_i$ denote the collection of $s_i \rightsquigarrow t_i$ paths of distance at most d_i consisting of edges in E. Let the *local graph* $G^i = (V^i, E^i)$ be the union of all vertices and edges in $\mathcal{P}_i$. A pair $(s_i, t_i) \in D$ is *t-thick* if $|V^i| \geq t$, otherwise (s_i, t_i) is *t-thin*. Consider the following standard LP relaxation (essentially the one in [38]).

$$\begin{aligned}
\min_{x,y} \quad & \sum_{e \in E} x_e & & \\
\text{subject to} \quad & \sum_{P \in \mathcal{P}_i} y_P \geq 1 & & \forall i \in [k], \\
& \sum_{P \mid e \in P \in \mathcal{P}_i} y_P \leq x_e & \forall e \in E, & \forall i \in [k], \\
& x_e \geq 0 & & \forall e \in E, \\
& y_P \geq 0 & \forall P \in \mathcal{P}_i \quad & \forall i \in [k].
\end{aligned} \tag{3.1}$$

Herein, x_e is an indicator of edge e and y_P is an indicator of path P. Suppose we have an integral feasible solution. Then the first set of constraints ensures that there is at least one $s_i \rightsquigarrow t_i$ path of distance at most d_i selected, and the second set of constraints ensures that if a path P is selected, then all its edges are selected.

We say that a pair $(s_i, t_i) \in D$ is *settled* if the selection of edges is such that there exists an

$s_i \rightsquigarrow t_i$ path of distance at most d_i. Applying a simple rounding scheme based on a solution of LP (3.1) settles the thin pairs with high probability, while sampling enough vertices and adding shortest path in-arborescences and out-arborescences rooted at each sampled vertex ensures with high probability that thick pairs are settled. Let OPT be the optimum value of the given pairwise spanner instance. Without loss of generality, we may assume that we know OPT since we can guess every value of OPT in $[|E|]$ in the offline setting. Now we are ready to describe Algorithm 2 in [34].

Algorithm 2 Offline pairwise spanner

1: $E' \leftarrow \emptyset$ and $t \leftarrow n/\sqrt{\mathsf{OPT}}$.
2: Solve LP (3.1) and add each edge $e \in E$ to E' with probability $\min\{1, x_e t \ln n\}$ independently.
3: Obtain a vertex set $W \subseteq V$ by sampling $(3n \ln n)/t$ vertices from V independently and uniformly at random. Add the edges of shortest path in-arborescences and out-arborescences rooted at w for each $w \in W$.

The approximation guarantee of Algorithm 2 is stated in the following lemma. The approximation (competitiveness) guarantee of the offline $\tilde{O}(n^{4/5})$-approximate algorithm (Algorithm 3) and the online $\tilde{O}(n^{4/5})$-competitive algorithm (Algorithm 4) presented later are based on the proof of this lemma. For this reason and the sake of completeness, we include the proof below.

Lemma 3.1. *([34]) Algorithm 2 is $\tilde{O}(n/\sqrt{\mathsf{OPT}})$-approximate.*

Proof. The following claim that settles thin pairs is due to [20]. Essentially, when n is large, for any $t > 2$, with high probability, all the t-thin pairs are settled by the rounding scheme suggested below.

Claim 3.2. *([20]) Let x, y be a feasible solution of LP (3.1) with objective value LP. By retaining every edge $e \in E$ independently with probability $\min\{1, x_e \cdot t \ln n\}$, all t-thin pairs are settled with probability at least $1 - \exp(-t \ln(n/t) + 2 \ln n)$, and the number of edges retained is at most $\tilde{O}(t)\mathsf{LP}$.*

To settle t-thick pairs, we obtain a vertex set $W \subseteq V$ by sampling $(3n \ln n)/t$ vertices from V independently and uniformly at random, then add edges of the shortest path in-arborescences and out-arborescences rooted at w for each $w \in W$. For each t-thick pair (s_i, t_i), the probability that $W \cup V^i = \emptyset$ is $(1 - t/n)^{-(3n \ln n)/t} \leq \exp(-3 \ln n) = 1/n^3$. There are at most n^2 t-thick pairs, so by a union bound, with probability at least $1 - 1/n$, all t-thick pairs in D are settled.

We note that there is a polynomial-time constant factor approximation algorithm for solving LP (3.1) according to [38] even though there are an exponential number of variables and constraints. Let LP^* denote the optimal objective value of LP (3.1). By Lemma 3.2, the t-thin pairs are settled in the second step and the number of edges retained is at most $\tilde{O}(t)\mathsf{LP}^*$. The t-thick pairs are settled by adding shortest path arborescences in the third step, and the number of edges retained is at most $2n \cdot 3n \ln n/t$ since each arborescence has at most n edges. It follows that the approximation guarantee is

$$\frac{\tilde{O}(t\mathsf{LP}^*) + \tilde{O}(n^2/t)}{\mathsf{OPT}} \leq \frac{\tilde{O}(t\mathsf{OPT}) + \tilde{O}(n^2/t)}{\mathsf{OPT}} = \frac{\tilde{O}(n\sqrt{\mathsf{OPT}})}{\mathsf{OPT}} = \tilde{O}(\frac{n}{\sqrt{\mathsf{OPT}}})$$

where the first inequality follows by $\mathsf{LP}^* \leq \mathsf{OPT}$. Namely, the optimal fractional solution is upper-bounded by the optimal integral solution of LP (3.1), and the second equality follows by the definition $t = n/\sqrt{\mathsf{OPT}}$. □

In the all-pairs spanner problem where OPT is $\Omega(n)$, Algorithm 2 is $\tilde{O}(\sqrt{n})$-approximate which matches the state-of-the-art approximation ratio given in [20]. For the pairwise spanner problem, the main challenge is the lack of a nice lower bound for OPT. In the offline setting, [34] achieves an $\tilde{O}(n^{3/5+\varepsilon})$-approximate solution by a careful case analysis when edges have uniform lengths. We give an alternative approach that is amenable to the online setting by considering two cases, where one resolves the issue when OPT does not have a nice lower bound, and the other uses a variant of Algorithm 2 given that OPT has a nice lower bound. This approach relies on the following observation.

Lemma 3.3. $\mathsf{OPT} \geq \sqrt{k}$.

Proof. We observe that when the spanner has ℓ edges, there are at most ℓ source vertices and ℓ sink vertices, so there are at most ℓ^2 terminal pairs. Therefore, when the spanner has OPT edges, there are at most OPT^2 terminal pairs, so $\mathsf{OPT} \geq \sqrt{k}$. □

Now we specify the simple offline algorithm given in Algorithm 3. In the beginning, we set two parameters T and t (which we will describe later), and set $E' = \emptyset$. An $s_i \rightsquigarrow t_i$ path is *cheapest feasible* if it meets the distance requirement d_i by using the minimum number of edges from E. We note that cheapest feasible paths can be found by Bellman-Ford algorithm.

Algorithm 3 Simple offline pairwise spanner

1: **if** $k < T$ **then**
2: Add the edges of a cheapest feasible $s_i \rightsquigarrow t_i$ path to E' for each $i \in [k]$.
3: **else**
4: Solve LP (3.1) and add each edge $e \in E$ to E' with probability $\min\{1, x_e t \ln n\}$ independently.
5: Obtain a vertex set $W \subseteq V$ by sampling $(3n \ln n)/t$ vertices from V independently and uniformly at random. Add the edges of shortest path in-arborescences and out-arborescences rooted at each vertex $w \in W$ to E'.
6: **end if**

Lemma 3.4. *Algorithm 3 is $\tilde{O}(n^{4/5})$-approximate when $T = t = n^{4/5}$.*

Proof. If $k < n^{4/5}$, we add the edges of a cheapest feasible $s_i \rightsquigarrow t_i$ path for each $(s_i, t_i) \in D$. Each cheapest feasible $s_i \rightsquigarrow t_i$ path contains at most OPT edges, so the ratio between this solution and OPT is $n^{4/5}$. The remaining case is $k \geq n^{4/5}$, which implies $\mathsf{OPT} \geq n^{2/5}$ by Lemma 3.3. Let LP^* be the optimal objective value of LP (3.1). By Claim 3.2, the rounding scheme settles the t-thin pairs with high probability. The t-thick pairs are settled by a collection of shortest path

in-arborescences and out-arborescences using a similar argument in the proof of Lemma 3.1. The approximation guarantee is

$$\frac{\tilde{O}(t\mathsf{LP}^*) + \tilde{O}(n^2/t)}{\mathsf{OPT}} \leq \frac{\tilde{O}(n^{4/5}\mathsf{OPT}) + \tilde{O}(n^{6/5})}{\mathsf{OPT}} = \tilde{O}(n^{4/5})$$

since the number of edges retained from the rounding scheme is at most $\tilde{O}(t)\mathsf{LP}^*$ and the number of edges retained by adding arborescences is at most $2n \cdot 3n \ln n/t$ since each arborescence has at most n edges. This summarizes the simple offline $\tilde{O}(n^{4/5})$-approximation algorithm. □

3.1.2 An $\tilde{O}(n^{4/5})$-competitive online algorithm

It remains to convert the simple offline algorithm to an online algorithm. We address the two main modifications.

1. We have to (approximately) solve LP (3.1) online, which is not presented as a covering LP.
2. We have to round the solution of LP (3.1) online.

For the first modification, LP (3.1) is converted to an equivalent covering LP (3.4) (which we will show later) and approximately solved in an online fashion. For the second modification, we use an online version of the rounding scheme in Algorithm 3, such that the overall probability (from round 1 to the current round) for the edge selection is consistent with the probability based on the online solution of LP (3.1), by properly scaling the probability based on a conditional argument.

Algorithm 4 Online pairwise spanner

for an arriving pair (s_i, t_i) **do**
  Convert the spanner LP (3.1) to the covering LP (3.4) and solve LP (3.4) online (see the proof of Theorem 1.2 for the details).
  if $i < T$ **then**
    Add the edges of a cheapest feasible $s_i \rightsquigarrow t_i$ path to E'.
  else if $i = T$ **then**
    Obtain a vertex set $W \subseteq V$ by sampling $(3n \ln n)/t$ vertices from V independently and uniformly at random. Add the edges of shortest path in-arborescences and out-arborescences rooted at each vertex $w \in W$ to E'.
    Add each edge e to E' independently with probability p_e^i for each edge $e \in E \setminus E'$.
  else ▷ $i > T$
    Add each edge e to E' independently with probability $(p_e^i - p_e^{i-1})/(1 - p_e^{i-1})$ for each edge $e \in E \setminus E'$.
  end if
end for

The online algorithm in round i is given in Algorithm 4. The same structure will be used later for other variants of the online pairwise spanner problem. All these algorithms rely

on the LP rounding scheme (Claim 3.2) to settle the t-thin pairs with high probability and a collection of shortest path in-arborescences and out-arborescences to settle t-thick pairs (similar to Lemma 3.1).

In the beginning, we pick a threshold parameter T and a thickness parameter t (which we will show later), and set $E' = \emptyset$. Let x_e^i denote the value of x_e in the approximate solution of LP (3.4) obtained in round i. Let $p_e^i := \min\{1, x_e^i t \ln n\}$. Algorithm 4 is the online version of Algorithm 3. A key insight is that when we add the arborescences in round T, it also settles the *future* thick terminal pairs with high probability. With the outline structure of the online algorithm, we are ready to prove Theorem 1.2.

Theorem 1.2. *For the online pairwise spanner problem, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{4/5})$.*

Proof. Suppose in the online setting, there are k rounds where k may be unknown. In round $i \in [k]$, the pair (s_i, t_i) and the distance requirement d_i arrive and we select some new edges from E to settle (s_i, t_i). We run Algorithm 4 by setting $T = t = \lfloor n^{4/5} \rfloor$. It suffices to show that 1) LP (3.1) can be solved online by losing a polylogarithmic factor, and 2) the overall probability of edge selection is consistent with the probability based on the online solution of LP (3.1).

Converting and solving LP (3.1) online. The goal is to update x in a non-decreasing manner upon the arrival of the pair (s_i, t_i) to satisfy all its corresponding constraints, so that the objective value is still approximately optimal. We convert LP (3.1) into a covering LP as follows.

First, we check in round i, given the *edge capacity* x, if there is a (fractional) $s_i \rightsquigarrow t_i$ path of distance at most d_i. This can be captured by checking the optimum of the following LPs,

$$\begin{aligned} \max_{y} \quad & \sum_{P \in \mathcal{P}_i} y_P \\ \text{subject to} \quad & \sum_{P \mid e \in P \in \mathcal{P}_i} y_P \le x_e \qquad \forall e \in E, \\ & y_P \ge 0 \qquad\qquad \forall P \in \mathcal{P}_i, \end{aligned} \tag{3.2}$$

and its dual

$$\begin{aligned} \min_{z} \quad & \sum_{e \in E} x_e z_e \\ \text{subject to} \quad & \sum_{e \in P} z_e \ge 1 \quad \forall P \in \mathcal{P}_i, \\ & z_e \ge 0 \qquad\quad \forall e \in E. \end{aligned} \tag{3.3}$$

We say that x is *good* if the optimum of LP (3.2) and LP (3.3) is at least 1, and it is *bad* otherwise. Namely, x is good if there is at least one (fractional) $s_i \rightsquigarrow t_i$ path of distance at most d_i. In LP (3.3), the feasibility problem is equivalent to the following problem. Given the local graph G^i and edge weight z, is there an $s_i \rightsquigarrow t_i$ path of distance at most d_i and weight less than 1? We note that with uniform lengths, this problem can be solved by Bellman-Ford algorithm

with d_i iterations, which computes the smallest weight among all $s_i \rightsquigarrow t_i$ paths of distance at most d_i in the local graph G^i.

Although this bicriteria path problem in general is NP-hard [7], an FPTAS is known to exist [57, 66], which gives an approximate separation oracle. We can verify in polynomial time that if there is a path of distance at most d_i and weight less than $1-\varepsilon$. We obtain a solution z' for LP (3.3) where each constraint is satisfied by a factor of $1-\varepsilon$ and set $z := z'/(1-\varepsilon)$ as the solution.

To solve LP (3.1), suppose in round i, we are given x. First, we check if x is good or bad by approximately solving LP (3.3). If x is good, then there exists y such that $\sum_{P\in\mathcal{P}_j} y_P \geq 1$, i. e., the solution is feasible for LP (3.1), so we move on to the next round. Otherwise, x is bad, so we increment x until it becomes good, which implies $\sum_{e\in E} x_e z_e \geq 1$ for all feasible z in LP (3.3). Let Z_i be the feasible polyhedron of LP (3.3) in round i. We derive the following LP (essentially the one in [39, 38]) which is equivalent to LP (3.1), by considering all the constraints of LP (3.3) from round 1 to round k.

$$\begin{aligned} \min_{x} \quad & \sum_{e\in E} x_e \\ \text{subject to} \quad & \sum_{e\in E} z_e x_e \geq 1 \quad \forall z \in Z_i \quad \forall i \in [k], \\ & x_e \geq 0 \qquad\qquad\qquad\qquad \forall e \in E. \end{aligned} \tag{3.4}$$

In round $i \in [k]$, the subroutine that approximately solves LP (3.3) and checks if the optimum is good or not, is the separation oracle used for solving LP (3.4) online. Here we use Theorem 4.1 (the formal version of Theorem 1.5) to show that LP (3.4) can be solved online in polynomial time by paying an $O(\log n)$ factor. This requires that both $\log(1/z_e)$ and $\log \mathsf{LP}^*$ are polynomial in the number of bits used for the edge lengths, where LP^* is the optimum of LP (3.4). Clearly, $\log \mathsf{LP}^* \leq \log |E|$ is in $\mathsf{poly}(n)$. For $\log(1/z_e)$, the subroutine that approximately solves LP (3.3) returns an approximate solution z which is represented by polynomial number of bits used for the edge lengths [57, 66]. By Theorem 4.1, we have the following Lemma.

Lemma 3.5. *There exists a polynomial-time $O(\log n)$-competitive online algorithm for LP (3.1).*

Conditional edge selection. After having a fractional solution of LP (3.1) in round i where $i \geq T = \lfloor n^{4/5} \rfloor$, we independently pick $e \in E \setminus E'$ with some scaled probability so that the edge selection is consistent with the probability based on the online solution of LP (3.1). More specifically, let $p_e := \min\{1, x_e t \ln n\}$ and let p_e^i be the value of p_e in round i. Let $\tilde{E}$ be the set of edges where each edge is neither selected while adding cheapest feasible paths prior to round T nor selected while adding in-arborescences and out-arborescences in round T. We show that each edge $e \in \tilde{E}$ has already been selected with probability p_e^i in round i. This can be proved by induction. According to Algorithm 4, the base case is round T, where $e \in \tilde{E}$ is selected with probability p_e^T. Now suppose $i > T$, if $e \in \tilde{E}$ has been selected, it is either selected prior to round i or in round i. For the former case, e must had already been selected in round $i-1$, with probability p_e^{i-1} by inductive hypothesis. For the later case, conditioned on e has not been

selected in round $i-1$, e is selected with probability $(p_e^i - p_e^{i-1})/(1-p_e^{i-1})$. Therefore, in round i, e has been selected with probability $p_e^{i-1} + (1-p_e^{i-1}) \cdot \frac{p_e^i - p_e^{i-1}}{1-p_e^{i-1}} = p_e^i$, which completes the proof. Intuitively, when $i > T$, conditioned on $e \in \tilde{E}$ was not picked from round 1 to round $i-1$, we pick e with probability $(p_e^i - p_e^{i-1})/(1-p_e^{i-1})$ at round i, so that the overall probability that e is picked from round 1 to round i is p_e^i.

Summary. We conclude the proof as follows. The overall algorithm is given in Algorithm 4. For the initialization, x is a zero vector, E' is an empty set, and $T = t = \lfloor n^{4/5} \rfloor$. The set E' is the solution. We pay an extra logarithmic factor for solving LP (3.1) online by Lemma 3.5. The competitive ratio remains $\tilde{O}(n^{4/5})$. □

3.2 Online pairwise spanners with maximum allowed distance d

Suppose the given graph has uniform edge lengths, and the diameter is bounded or it is guaranteed that the distances between the terminal pairs are bounded. Let $d = \max_{i \in [k]}\{d_i\}$ be the maximum allowed distance of any pair of terminals in the input. This setting is equivalent to the *d-diameter spanning subgraph* problem introduced in [22]. We assume that d is known offline and show the following result.

Theorem 3.6. *For the online pairwise spanner problem with uniform edge lengths and maximum allowed distance d, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(d^{1/3}n^{2/3})$.*

Proof. When the maximum allowed distance is d for the terminal pairs, we employ Algorithm 4 with $T = \lfloor d^{-4/3}n^{4/3} \rfloor$ and $t = d^{1/3}n^{2/3}$. If $k < T$, then for each $(s_i, t_i) \in D$, we add the edges of a shortest $s_i \rightsquigarrow t_i$ path. Each shortest $s_i \rightsquigarrow t_i$ path contains at most d edges. By Lemma 3.3, $\mathsf{OPT} \geq \sqrt{k}$, so the ratio between this solution and OPT is $\frac{dk}{\mathsf{OPT}} \leq d\sqrt{k} \leq d\sqrt{T} = O(d^{1/3}n^{2/3})$. If $k \geq T$, then $\mathsf{OPT} \geq d^{-2/3}n^{2/3}$. Let LP be the value of the online integral $O(\log n)$-competitive solution of LP (3.1) obtained by Algorithm 4. The approximation guarantee is

$$\frac{\tilde{O}(t\mathsf{LP}) + \tilde{O}(n^2/t)}{\mathsf{OPT}} \leq \frac{\tilde{O}(d^{1/3}n^{2/3}\mathsf{OPT}) + \tilde{O}(d^{-1/3}n^{4/3})}{\mathsf{OPT}} = \tilde{O}(d^{1/3}n^{2/3}).$$ □

3.3 Online quasimetric spanners and all-server spanners

Consider the case where the edge lengths form a *quasimetric*, i. e., for any two edges $u \to v$ and $v \to w$, there is also an edge $u \to w$ such that $\ell(u,w) \leq \ell(u,v) + \ell(v,w)$. This setting includes the class of transitive-closure graphs with uniform edge lengths, in which each pair or vertices connected by a directed path is also connected by a directed edge. The offline version of the *transitive-closure spanner* problem was formally defined in [22]. We obtain the following result.

Theorem 3.7. *For the online pairwise spanner problem where edge lengths are quasimetric, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3})$.*

In the setting of all-server spanners, the given graph has uniform edge lengths. For each terminal pair (s_i, t_i), there is also an edge $s_i \to t_i$ in E. This setting is equivalent to the *all-server spanner* problem introduced in [44]. We show the following theorem for this case.

Theorem 3.8. *For the online all-server spanner problem with uniform edge lengths, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3})$.*

Proof of Theorem 3.7 and Theorem 3.8. We employ Algorithm 4 with $T = \lfloor n^{4/3} \rfloor$ and $t = n^{2/3}$. For these special cases, we start by adding edges instead of paths. Then, in the initial greedy phase in line 4 of Algorithm 4, the cost of each step is upper-bounded by 1 instead of OPT. This enables us to obtain better competitive ratios. More formally, if $k < T$, then for each $(s_i, t_i) \in D$, we add the edge $s_i \to t_i$. By Lemma 3.3, $\mathsf{OPT} \geq \sqrt{k}$, so the ratio between this solution and OPT is

$$\frac{k}{\mathsf{OPT}} \leq \sqrt{k} \leq \sqrt{T} = O(n^{2/3}).$$

If $k \geq T$, then $\mathsf{OPT} \geq n^{2/3}$. Let LP be the value of the online integral $O(\log n)$-competitive solution of LP (3.1) obtained by Algorithm 4. The approximation guarantee is

$$\frac{\tilde{O}(t\mathsf{LP}) + \tilde{O}(n^2/t)}{\mathsf{OPT}} \leq \frac{\tilde{O}(n^{2/3}\mathsf{OPT}) + \tilde{O}(n^{4/3})}{\mathsf{OPT}} = \tilde{O}(n^{2/3}). \qquad \square$$

3.4 Online pairwise spanners with uniform edge lengths

In this section, we prove Theorem 1.3.

Theorem 1.3. *For the online pairwise spanner problem with uniform edge lengths, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$ for any constant $\varepsilon \in (0, 1/3)$.*

Proof. We employ Algorithm 4 with a slight tweak and set $T = \lfloor n^{4/3-4\varepsilon} \rfloor$ and $t = n^{2/3+\varepsilon}$.

If $k < T$, instead of adding edges of a shortest $s_i \rightsquigarrow t_i$ path, we use Theorem 1.1 to find an $\tilde{O}(n^{2/3+\varepsilon})$ competitive solution.

Theorem 1.1. *For the online pairwise spanner problem with uniform edge lengths, there exists a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(k^{1/2+\delta})$ for any constant $\delta > 0$.*

For any $\varepsilon \in (0, 1/3)$, there exists δ such that $4\delta/(9 + 12\delta) = \varepsilon$. By picking this δ, we have that

$$\begin{aligned}(\frac{4}{3} - 4\varepsilon)(\frac{1}{2} + \delta) &= (\frac{4}{3} - \frac{16\delta}{9 + 12\delta})(\frac{1}{2} + \delta) = \frac{2}{3} + \frac{4\delta}{3} - \frac{8\delta + 16\delta^2}{9 + 12\delta} \\ &= \frac{2}{3} + \frac{12\delta + 16\delta^2 - 8\delta - 16\delta^2}{9 + 12\delta} = \frac{2}{3} + \frac{4\delta}{9 + 12\delta} = \frac{2}{3} + \varepsilon.\end{aligned}$$

Hence, the ratio between the solution obtained by Theorem 1.1 and OPT is

$$k^{1/2+\delta} \leq n^{(4/3-4\varepsilon)(1/2+\delta)} = n^{2/3+\varepsilon} = \tilde{O}(n^{2/3+\varepsilon}).$$

If $k \geq T$, then $\mathsf{OPT} \geqslant n^{2/3-2\varepsilon}$. Let LP be the online integral solution of LP (3.1) obtained by Algorithm 4. The approximation guarantee is

$$\frac{\tilde{O}(t\mathsf{LP}) + \tilde{O}(n^2/t)}{\mathsf{OPT}} \leq \frac{\tilde{O}(n^{2/3+\varepsilon}\mathsf{OPT}) + \tilde{O}(n^{4/3-\varepsilon})}{\mathsf{OPT}} = \tilde{O}(n^{2/3+\varepsilon}). \tag{3.5}$$

$\square$

3.5 Online directed Steiner forests with uniform costs

This problem is a special case of pairwise spanners with uniform edge lengths and infinite target distances. We show the following theorem.

Theorem 1.4. *For the online directed Steiner forest problem with uniform costs, there is a randomized polynomial-time algorithm with competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$ for any constant $\varepsilon \in (0, 1/3)$.*

Proof. The structure of the online algorithm is the same as that for online pairwise spanners with uniform edge lengths. We employ Algorithm 4 and set $T = \lfloor n^{4/3-4\varepsilon} \rfloor$ and $t = n^{2/3+\varepsilon}$. If $k < T$, instead of adding edges of a shortest $s_i \rightsquigarrow t_i$ path, we use Theorem 1.1 to find an $\tilde{O}(n^{2/3+\varepsilon})$ competitive solution.[4] If $k \geq T$, then the algorithm is $\tilde{O}(n^{2/3+\varepsilon})$-competitive by (3.5). □

4 Online Covering in Polynomial Time

This section is devoted to proving the formal version of Theorem 1.5. We recall that the problem of interest is to solve the covering LP (1.2) online:

$$\text{minimize } \mathbf{c}^T x \text{ over } x \in \mathbb{R}^n_{\geq 0} \text{ s.t. } Ax \geq \mathbf{1}$$

where $A \in \mathbb{R}^{m \times n}_{\geq 0}$ consists of m covering constraints, $\mathbf{1} \in \mathbb{R}^m_{>0}$ is a vector of all ones treated as the lower bound of the covering constraints, and $\mathbf{c} \in \mathbb{R}^n_{>0}$ denotes the positive coefficients of the linear cost function.

In the online covering problem, the cost vector $\mathbf{c}$ is given offline, and each of these covering constraints is presented one by one in an online fashion, that is, m can be unknown. In round $i \in [m]$, $\{a_{ij}\}_{j \in [n]}$ (where a_{ij} denotes the i-th row j-th column entry of A) is revealed, and we have to monotonically update x so that the constraint $\sum_{j \in [n]} a_{ij} x_j \geq 1$ is satisfied. We always assume that there is at least one positive entry a_{ij} in each round i, otherwise constraint i cannot be satisfied since all the row entries are zeros. The goal is to update x in a non-decreasing manner and approximately minimize the objective value $\mathbf{c}^T x$.

We recall that an important idea in this line of work is to simultaneously consider the dual packing problem LP (1.3):

$$\text{maximize } \mathbf{1}^T y \text{ over } y \in \mathbb{R}^m_{\geq 0} \text{ s.t. } A^T y \leq \mathbf{c}$$

where A^T consists of n packing constraints with an upper bound $\mathbf{c}$ given offline.

The primal-dual framework in [26] simultaneously solves both LP (1.2) and LP (1.3), and crucially uses LP-duality and strong connections between the two solutions to argue that they are both nearly optimal. The modified framework closely follows the *guess-and-double* scheme in [26]. Specifically, the scheme runs in phases where each phase estimates a lower bound for the optimum. When the first constraint arrives, the scheme generates the first lower bound

$$\alpha(1) \leftarrow \min_{j \in [n]} \{ \frac{c_j}{a_{1j}} \mid a_{1j} > 0 \} \leq \mathsf{OPT}$$

[4] One can also use the $\tilde{O}(k^{1/2+\delta})$-competitive online algorithm in [28].

where c_j is the j-th entry of $\mathbf{c}$ and OPT is the optimal value of LP (1.2).

During phase r, we always assume that the lower bound of the optimum is $\alpha(r)$ until the online objective $\mathbf{c}^T x$ exceeds $\alpha(r)$. Once the online objective exceeds $\alpha(r)$, we start the new phase $r+1$ from the current violating constraint (let us call it constraint i_{r+1}, in particular, $i_1 = 1$), and double the estimated lower bound, i. e., $\alpha(r+1) \leftarrow 2\alpha(r)$.[5] We recall that x must be updated in a non-decreasing manner, so the algorithm maintains $\{x_j^r\}$, which denotes the value of each variable x_j in each phase r, and the value of each variable x_j is actually set to $\max_r\{x_j^r\}$.

In Algorithm 5, we describe one round of the modified scheme in phase r. When a covering constraint i arrives, we introduce a packing variable $y_i = 0$. If the constraint is violated, we increment each x_j according to an exponential function of y_i until the constraint is satisfied *by a factor of 2*. This is the main difference between the modified framework and [26], which increments the variables until the constraint is satisfied.

Algorithm 5 Online Covering

for arriving covering constraint i **do**
  $y_i \leftarrow 0$. ▷ the packing variable y_i is used for the analysis
  if $\sum_{j=1}^n a_{ij}x_j^r < 1$ **then** ▷ if constraint i is not satisfied
    while $\sum_{j=1}^n a_{ij}x_j^r < 2$ **do** ▷ update until constraint i is satisfied by a factor of 2
      Increase y_i continuously.
      Increase each variable x_j^r by the following increment function:

$$x_j^r \leftarrow \frac{\alpha(r)}{2nc_j}\exp\left(\frac{\ln(2n)}{c_j}\sum_{k=i_r}^{i} a_{kj}y_k\right).$$

    end while
  end if
end for

Although the augmentation is in a continuous fashion, it is not hard to implement it in a discrete way for any desired precision by binary search. Therefore, to show that the modified framework is efficient, it suffices to bound the number of violating constraints it will encounter. The performance of the modified scheme is analyzed in Theorem 4.1 (the formal version of Theorem 1.5).

Theorem 4.1. *There exists an $O(\log n)$-competitive online algorithm for the covering LP (1.2) which encounters* $\mathsf{poly}(n, \log \mathsf{OPT}, \log(1/\alpha(1)))$ *violating constraints.*

Proof. The proof for the $O(\log n)$-competitiveness closely follows the one in [26]. Let $X(r)$ and $Y(r)$ be the covering and packing objective values, respectively, generated during phase r. The following claims are used to show that Algorithm 5 is $O(\log n)$-competitive.

[5] In [26], the scheme starts all over again from the first constraint. We start from the current violating constraint because it is more amenable when violating constraints are generated by a separation oracle. There is no guarantee for the order of arriving violating constraints in such settings.

(i) x is feasible.

(ii) For each *finished* phase r, $\alpha(r) \le 4\ln(2n) \cdot Y(r)$.

(iii) y generated during phase r is feasible.

(iv) The sum of the covering objective generated from phase 1 to r is at most $2\alpha(r)$.

(v) Let r' be the last phase, then the covering objective $\mathbf{c}^T x \le 2\alpha(r')$.

From these five claims together with weak duality, we conclude that

$$\mathbf{c}^T x \le 2\alpha(r') = 4\alpha(r'-1) \le 16\ln(2n) \cdot Y(r'-1) \le 16\ln(2n) \cdot \mathsf{OPT}.$$

Proof of (i). In phase r, suppose constraint i arrives, then either it is already satisfied by x^r, or it is violated and we update x^r until constraint i is satisfied by a factor of 2 (or start another phase until we reach a phase that satisfies constraint i by a factor of 2 without exceeding the estimated lower bound α). x satisfies all the covering constraints from 1 to i because it is the coordinate-wise maximum of $\{x^k\}_{k\in[r]}$. □

Proof of (ii). In the beginning of phase r, $x_j^r = \alpha(r)/(2nc_j)$ so $X(r)$ is initially at most $\alpha(r)/2$. The total increase of $X(r)$ is at most $\alpha(r)/2$ because $X(r) \ge \alpha(r)$ when phase r ends. Therefore, it suffices to show that

$$\frac{\partial X(r)}{\partial y_i} \le 2\ln(2n) \cdot \frac{\partial Y(r)}{\partial y_i}.$$

This follows because

$$\begin{aligned}
\frac{\partial X(r)}{\partial y_i} &= \sum_{j=1}^{n} c_j \frac{\partial x_j^r}{\partial y_i} \\
&= \sum_{j=1}^{n} \frac{c_j \ln(2n) a_{ij}}{c_j} \frac{\alpha(r)}{2nc_j} \cdot \exp\left(\frac{\ln(2n)}{c_j} \sum_{k=i_r}^{i} a_{kj} y_k\right) \\
&= \ln(2n) \sum_{j=1}^{n} a_{ij} x_j^r \le 2\ln(2n) = 2\ln(2n) \cdot \frac{\partial Y(r)}{\partial y_i} \qquad (4.1)
\end{aligned}$$

where (4.1) holds because $\sum_{j=1}^{n} a_{ij} x_j^r \le 2$ and $Y(r) = \mathbf{1}^T y$ implies that $\partial Y(r)/\partial y_i = 1$. □

Proof of (iii). We observe that during phase r of Algorithm 5, $x_j^r \le \alpha(r)/c_j$, since otherwise phase r is finished. Therefore,

$$x_j^r = \frac{\alpha(r)}{2nc_j} \exp\left(\frac{\ln(2n)}{c_j} \sum_{k=i_r}^{i} a_{kj} y_k\right) \le \frac{\alpha(r)}{c_j}$$

which implies that $\sum_{k=i_r}^{i} a_{kj} y_k \le c_j$. □

Proof of (iv). The sum of the covering objective generated from phase 1 to r is at most

$$\sum_{k=1}^{r} \alpha(k) = \sum_{k=1}^{r} \frac{\alpha(r)}{2^{k-r}} \leq 2\alpha(r). \qquad \square$$

Proof of (v). In the last phase r', x is feasible because it is the coordinate-wise maximum of $\{x^r\}_{r\in[r']}$. We have that

$$\mathbf{c}^T x = \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n}\sum_{r=1}^{r'} c_j x_j^r = \sum_{r=1}^{r'}\left(\sum_{j=1}^{n} c_j x_j^r\right) \leq \sum_{r=1}^{r'} \alpha(r) \leq 2\alpha(r')$$

where the first inequality holds because $x_j = \max_{r\in[r']}\{x_j^r\} \leq \sum_{r=1}^{r'} x_j^r$, the second inequality is by the fact that the covering objective $\sum_{j=1}^{n} c_j x_j^r$ cannot exceed the estimated lower bound $\alpha(r)$, while the last inequality is by (iv). $\square$

Now we show that Algorithm 5 encounters $\mathsf{poly}(n, \log \mathsf{OPT}, \log(1/\alpha(1)))$ violating constraints. We first show that there are $O(\log\log n + \log \mathsf{OPT} + \log(1/\alpha(1)))$ phases. The estimated lower bound α doubles when we start a new phase. Suppose there are r' phases, then $\alpha(1)\cdot 2^{r'-1} = O(\log n)\mathsf{OPT}$ because Algorithm 5 is $O(\log n)$-competitive. This implies that $r' = O(\log\log n + \log \mathsf{OPT} + \log(1/\alpha(1)))$.

In each phase, when a violating constraint arrives, we increment x so that the constraint is satisfied by a factor of 2. This implies that at least one variable x_j is doubled. $x_j = O(\log n)\mathsf{OPT}/c_j$ because $c_j x_j \leq \mathbf{c}^T x = O(\log n)\mathsf{OPT}$. At the start of phase r, $x_j = \alpha(r)/(2nc_j) \geq \alpha(1)/(2nc_j)$. Suppose x_j has been doubled t times in phase r, then

$$\frac{\alpha(1)}{2nc_j}\cdot 2^t \leq \frac{\alpha(r)}{2nc_j}\cdot 2^t \leq x_j = O(\log n)\frac{\mathsf{OPT}}{c_j}$$

which indicates that $t = O(\log n + \log \mathsf{OPT} + \log(1/\alpha(1)))$.

There are n variables and r' phases, and in each phase, each variable is doubled at most t times. Therefore, Algorithm 5 encounters $\mathsf{poly}(n, \log \mathsf{OPT}, \log(1/\alpha(1)))$ violating constraints. $\square$

As stated in [26], the online scheme naturally extends to the setting with unbounded number of constraints where covering constraints do not appear explicitly, but are detected by a separation oracle. However, in the previous work, there was no guarantee about the efficiency, i. e., how fast the algorithm reaches an approximately optimal solution. In the worst-case scenario, it is possible that a detected constraint is slightly violated, thus hindering the growth of the covering variables since they are incremented until the constraint is *just* satisfied. The modified framework, on the other hand, increments the variables until the constraint is satisfied by a factor of 2. The sufficient growth of the covering variables ensures a polynomial upper bound for the number of violating constraints that the framework will encounter.

5 Online Packing in Polynomial Time

In this section, we prove the formal version of Theorem 1.6. The problem of interest is to solve the packing LP (1.3) online:

$$\text{maximize } \mathbf{1}^T y \text{ over } y \in \mathbb{R}^m_{\geq 0} \text{ s.t. } A^T y \leq \mathbf{c}$$

where $A^T \in \mathbb{R}^{n\times m}_{\geq 0}$ consists of n packing constraints with an upper bound $\mathbf{c}$ given offline.

In the online packing problem, the columns $i \in [m]$ of A^T and the corresponding variables y_i's are presented online as zeros, one at a time, where m can be unknown. Let a_{ij} be the i-th row j-th column entry of the matrix A and c_j be the j-th entry of $\mathbf{c}$. In round i, $\{a_{ij}\}_{j\in[n]}$, i. e., the i-th column of A^T is revealed, and the goal is to either let the arriving variable y_i remain zero, or irrevocably assign a positive value to it, such that 1) the objective value $\mathbf{1}^T y$ is approximately optimal, and 2) each constraint $\sum_{k=1}^{i} a_{kj} y_k \leq c_j$ for each $j \in [n]$ is approximately satisfied.

An important idea for solving the online packing problem is to simultaneously consider the dual online covering problem LP (1.2):

$$\text{minimize } \mathbf{c}^T x \text{ over } x \in \mathbb{R}^n_{\geq 0} \text{ s.t. } Ax \geq \mathbf{1}.$$

In this problem, the cost vector $\mathbf{c}$ is given offline, and each of these covering constraints is presented one by one in an online fashion. More specifically, in round i, $\{a_{ij}\}_{j\in[n]}$ is revealed. The goal is to monotonically update x so that the arriving constraint $\sum_{j=1}^{n} a_{ij} x_j \geq 1$ is satisfied and the objective value $\mathbf{c}^T x$ remains approximately optimal.

Similar to Section 4, we employ the primal-dual framework that simultaneously solves both LP (1.2) and LP (1.3), and crucially uses LP-duality and strong connections between the two solutions, to argue that they are both nearly optimal.

In the beginning of Algorithm 6, a parameter $B > 0$ is given as an input, and x is initialized to a zero vector. In round i, y_i is introduced with the i-th column of A^T, i. e., $\{a_{ij}\}_{j\in[n]}$ is revealed. In the corresponding online covering problem, constraint i arrives and is presented in the form $\sum_{j=1}^{n} a_{ij} x_j \geq 1$. Without loss of generality, we always assume that there is at least one positive entry a_{ij} in round i, otherwise, the packing objective $\mathbf{1}^T y$ is unbounded. If the arriving constraint is violated, we increase the value of the new packing variable y_i and each covering variable x_j simultaneously until the new covering constraint is satisfied *by a factor of 2*. We use an augmenting method in a continuous fashion, which can be implemented in a discrete way with any desired accuracy by binary search. The x_j's are incremented according to an exponential function of y_i. We note that y_i is only increased in round i and fixed after then, and the x_j's never decrease. The performance of the modified scheme is analyzed in Theorem 5.1 (the formal version of Theorem 1.6).

Algorithm 6 Online packing

for an arriving covering constraint i **do**
  $y_i \leftarrow 0$ and $a_j^{\max} \leftarrow \max_{k\in[i]}\{a_{kj}\}$ for each $j \in [n]$.
  if $\sum_{j=1}^n a_{ij}x_j < 1$ **then** ▷ if constraint i is not satisfied
    while $\sum_{j=1}^n a_{ij}x_j < 2$ **do** ▷ update until constraint i is satisfied by a factor of 2
      Increase y_i continuously.
      Increase each variable x_j where $a_j^{\max} > 0$ by the following increment function:

$$x_j \leftarrow \max\left\{x_j, \frac{1}{na_j^{\max}}\left(\exp\left(\frac{B}{3c_j}\sum_{k=1}^{i} a_{kj}y_k\right) - 1\right)\right\}.$$

    end while
  end if
end for

Theorem 5.1. *For any $B > 0$, there exists a $1/B$-competitive online algorithm for the packing LP (1.3) which updates y $\mathsf{poly}(n, \log B', \log \mathsf{OPT}, \log \alpha)$ times. Moreover, for each constraint $j \in [n]$, the following holds*

$$\sum_{i=1}^{m} a_{ij}y_i = c_j \cdot O\left(\frac{\log n + \log(a_j^{\max}/a_j^{\min})}{B}\right).$$

Here, $a_j^{\max} := \max_{i\in[m]}\{a_{ij}\}$ and $a_j^{\min} := \min_{i\in[m]}\{a_{ij} \mid a_{ij} > 0\}$ for each $j \in [n]$, $\alpha := \max_{j\in[n]}\{a_j^{\max}/c_j\}$, and $B' := 3\max_{j\in[n]}\{\ln(2na_j^{\max}/a_j^{\min} + 1)\}$.

Proof. The proof for the $1/B$-competitiveness closely follows the one in [26]. We provide the proof for completeness. Let $X(i)$ and $Y(i)$ be the covering and packing objective values, respectively, in round $i \in [m]$. The following claims are used to show that Algorithm 6 is $1/B$-competitive.

(i) x is feasible.

(ii) In each round $i \in [m]$, $X(i)/B \le Y(i)$.

(iii) For any packing constraint $j \in [n]$,

$$\sum_{i=1}^{m} a_{ij}y_i \le c_j \frac{3\ln(2na_j^{\max}/a_j^{\min} + 1)}{B}.$$

From the first two claims together with weak duality, we conclude that Algorithm 6 is B-competitive, while the third claim directly proves that each packing constraint is approximately satisfied.

Proof of (i). This clearly holds because if constraint i is already satisfied, we do nothing, otherwise, update variables x_j's until constraint i is satisfied by a factor of 2. □

Proof of (ii). We observe that at the beginning of each round, when $a_j^{\max}$ increases, x_j and $X(i)$ do not change. $X(i)$ increases only when $Y(i)$ increases. Initially, both $X(i)$ and $Y(i)$ are zero. Consider round i in which y_i is increased continuously. We show that $\partial X(i)/\partial y_i \le B \cdot \partial Y(i)/\partial y_i$ and conclude that $X(i)/B \le Y(i)$.

$$\begin{aligned}
\frac{\partial X(i)}{\partial y_i} &= \sum_{j=1}^{n} c_j \frac{\partial x_j}{\partial y_i} \\
&\le \sum_{j=1}^{n} \frac{c_j}{n a_j^{\max}} \frac{B a_{ij}}{3c_j} \exp\left(\frac{B}{3c_j} \sum_{k=1}^{i} a_{kj} y_k\right) && (5.1) \\
&= \frac{B}{3} \sum_{j=1}^{n} a_{ij} \left(\frac{1}{n a_j^{\max}} \left(\exp\left(\frac{B}{3c_j} \sum_{k=1}^{i} a_{kj} y_k\right) - 1\right) + \frac{1}{n a_j^{\max}}\right) \\
&\le \frac{B}{3} \sum_{j=1}^{n} a_{ij} \left(x_j + \frac{1}{n a_j^{\max}}\right) \le \frac{B}{3}(2+1) = B = B \frac{\partial Y(i)}{\partial y_i} && (5.2)
\end{aligned}$$

where (5.1) follows by taking the partial derivative, and (5.2) follows because

1. $\sum_{j=1}^{n} a_{ij} x_j \le 2$ while incrementing x_j in round i.
2. $x_j \ge 1/(n a_j^{\max})(\exp(B/(3c_j) \sum_{k=1}^{i} a_{kj} y_k) - 1)$.
3. $\sum_{j=1}^{n} a_{ij}/(n a_j^{\max}) \le 1$. □

Proof of (iii). We recall that $a_j^{\max} := \max_{i \in [m]}\{a_{ij}\}$ and $a_j^{\min} := \min_{i \in [m]}\{a_{ij} \mid a_{ij} > 0\}$ for each $j \in [n]$. During the run of Algorithm 6, $x_j \le 2/a_j^{\min}$, since otherwise each covering constraint $i \in [m]$ with $a_{ij} > 0$ is already satisfied by a factor of 2. Consider the packing constraint j, we have

$$\frac{1}{n a_j^{\max}} \left(\exp\left(\frac{B}{3c_j} \sum_{i=1}^{m} a_{ij} y_i\right) - 1\right) \le x_j \le \frac{2}{a_j^{\min}}$$

which implies that

$$\sum_{i=1}^{m} a_{ij} y_i \le c_j \frac{3 \ln(2 n a_j^{\max}/a_j^{\min} + 1)}{B}.$$ □

Now we show that Algorithm 6 updates y $\mathsf{poly}(n, \log B', \log \mathsf{OPT}, \log \alpha)$ times where OPT is the objective value of LP (1.3). It suffices to show that Algorithm 6 encounters $\mathsf{poly}(n, \log B', \log \mathsf{OPT}, \log \alpha)$ violating covering constraints. We recall that

$$B' := 3 \max_{j \in [n]}\{\log(2 n a_j^{\max}/a_j^{\min} + 1)\} \text{ and } \alpha := \max_{j \in [n]}\{a_j^{\max}/c_j\}.$$

Suppose we scale the packing solution y by B/B' such that all the packing constraints $j \in [n]$ are satisfied. Then by weak duality, since By/B' is feasible, we have that

$$\frac{c_j x_j}{B'} \leq \frac{X(i)}{B'} \leq \frac{BY(i)}{B'} \leq \mathsf{OPT}$$

for every $j \in [n]$. Therefore, $x_j = O(B'\mathsf{OPT}/c_j)$.

We show that for each arriving covering constraint i that is violated, one of the following cases must hold when the constraint is satisfied by a factor of 2 after the variable update: 1) there exists a *large* variable $x_j \geq 1/(2na_j^{\max})$ that is updated to at least $3x_j/2$, or 2) there exists a *small* variable $x_j < 1/(2na_j^{\max})$ that becomes large, i. e., x_j is updated to at least $1/(2na_j^{\max})$. Let L and S be the set of large and small variable subscript labels before the update, respectively, and x'_j be the value of x_j after the update. If none these two cases holds, then

$$\sum_{j=1}^{n} a_{ij}x'_j < \frac{3}{2}\sum_{j\in L} a_{ij}x_j + \sum_{j\in S}\frac{a_{ij}}{2na_j^{\max}} < \frac{3}{2} + \frac{1}{2} = 2$$

where the second inequality is by the fact that constraint i is violated and $a_{ij} \leq a_j^{\max}$. This implies that constraint i is not satisfied by a factor of 2, a contradiction.

Suppose x_j has been updated t times by a factor of $3/2$ since it was large, then

$$\frac{1}{2na_j^{\max}}(\frac{3}{2})^t = O(\frac{B'\mathsf{OPT}}{c_j})$$

which implies $t = O(\log n + \log \mathsf{OPT} + \log B' + \log(a_j^{\max}/c_j))$.

There are n variables, each variable can be updated from small to large once and updated t times by a factor of $3/2$ since it was large. Hence, Algorithm 6 encounters $\mathsf{poly}(n, \log B', \log \mathsf{OPT}, \log \alpha)$ violating covering constraints. □

6 Conclusions and Open Problems

In this work, we present the first online algorithm for pairwise spanners with competitive ratio $\tilde{O}(n^{4/5})$ for general lengths and $\tilde{O}(n^{2/3+\varepsilon})$ for uniform lengths. We also improve the competitive ratio for the online directed Steiner forest problem with uniform costs to $\tilde{O}(n^{2/3+\varepsilon})$ when $k = \omega(n^{4/3})$. We also show an efficient modified framework for online covering and packing. Our work raises several open questions that we state below.

Online pairwise spanners. An intriguing open problem is improving the competitive ratio for online pairwise spanners. For graphs with uniform edge lengths, there is a small polynomial gap between the state-of-the-art offline approximation ratio $\tilde{O}(n^{3/5+\varepsilon})$ and the online competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$. For graphs with general edge lengths, we are not aware of any studies about the pairwise spanner problem. Our $\tilde{O}(n^{4/5})$-competitive online algorithm intrinsically suggests

an $\tilde{O}(n^{4/5})$-approximate offline algorithm. As the approach in [34] achieves an $\tilde{O}(n/\sqrt{\mathsf{OPT}})$-approximation, we believe that the approximation ratio can be improved for the offline pairwise spanner problem, by judicious case analysis according to the cardinality of OPT.

Approximating Steiner forests online in terms of n. The state-of-the-art online algorithm for Steiner forests with general costs is $\tilde{O}(k^{1/2+\varepsilon})$-competitive [28]. A natural open question is designing an $o(n)$-competitive online algorithm when k is large, and potentially extending this result to the more general buy-at-bulk network design problem. The currently best-known offline approximation for Steiner forests with general costs is $O(n^{2/3+\varepsilon})$ [20], by case analysis that settles thick and thin terminal pairs separately. However, the approach in [20] for settling thin pairs is essentially a greedy procedure which is inherently offline. Our approach utilizes the uniform cost assumption to obtain a useful lower bound for the optimal solution, which is incompatible with general costs. It would be interesting to resolve the aforementioned obstacles and have an $o(n)$-competitive online algorithm for directed Steiner forests with general edge costs. One open problem for uniform costs is to improve the competitive ratio, as there is a polynomial gap between the state-of-the-art offline approximation ratio $\tilde{O}(n^{4/7+\varepsilon})$ and the online competitive ratio $\tilde{O}(n^{2/3+\varepsilon})$.

7 Acknowledgements

We thank the anonymous reviewers for comments and suggestions that helped improve the presentation. We thank Anupam Gupta and Greg Bodwin for bringing to our attention references that we missed in previous versions of the write-up.

References

[1] AMIR ABBOUD AND GREG BODWIN: Reachability preservers: New extremal bounds and approximation algorithms. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1865–1883. SIAM, 2018. [doi:10.1137/1.9781611975031.122] 4, 5

[2] REYAN AHMED, GREG BODWIN, FARYAD DARABI SAHNEH, KEATON HAMM, MOHAMMAD JAVAD LATIFI JEBELLI, STEPHEN KOBOUROV, AND RICHARD SPENCE: Graph spanners: A tutorial review, 2020. [doi:10.1016/j.cosrev.2020.100253] 2, 8

[3] 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. [doi:10.1145/1198513.1198522] 3, 5, 9, 25

[4] NOGA ALON, BARUCH AWERBUCH, YOSSI AZAR, NIV BUCHBINDER, AND JOSEPH NAOR: The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009. [doi:10.1137/060661946] 3, 5

[5] NOGA ALON AND BARUCH SCHIEBER: *Optimal preprocessing for answering on-line product queries*. Citeseer, 1987. 2

[6] SPYRIDON ANTONAKOPOULOS: Approximating directed buy-at-bulk network design. In *Proceedings of the 8th International Workshop on Approximation and Online Algorithms (WAOA)*, pp. 13–24. Springer, 2010. [doi:10.1007/978-3-642-18318-8_2] 8, 11

[7] ESTHER M ARKIN, JOSEPH SB MITCHELL, AND CHRISTINE D PIATKO: Bicriteria shortest path problems in the plane. In *Proceedings of the 3rd Canadian Conference on Computational Geometry*, pp. 153–156, 1991. 34

[8] PRANJAL AWASTHI, MADHAV JHA, MARCO MOLINARO, AND SOFYA RASKHODNIKOVA: Testing lipschitz functions on hypergrid domains. *Algorithmica*, 74(3):1055–1081, 2016. [doi:10.1007/s00453-015-9984-y] 2

[9] BARUCH AWERBUCH: Communication-time trade-offs in network synchronization. In *Proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, p. 272–276, New York, NY, USA, 1985. ACM. [doi:10.1145/323596.323621] 2

[10] BARUCH AWERBUCH AND YOSSI AZAR: Buy-at-bulk network design. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 542–547. IEEE, 1997. [doi:10.1109/SFCS.1997.646143] 8

[11] BARUCH AWERBUCH, YOSSI AZAR, AND YAIR BARTAL: On-line generalized steiner problem. *Theoretical Computer Science*, 324(2-3):313–324, 2004. [doi:10.1016/j.tcs.2004.05.021] 3, 8

[12] BARUCH AWERBUCH, YOSSI AZAR, AND SERGE PLOTKIN: Throughput-competitive on-line routing. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 32–40. IEEE, 1993. [doi:10.1109/SFCS.1993.366884] 3

[13] YOSSI AZAR, UMANG BHASKAR, LISA FLEISCHER, AND DEBMALYA PANIGRAHI: Online mixed packing and covering. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 85–100. SIAM, 2013. [doi:10.1137/1.9781611973105.6] 9

[14] YOSSI AZAR, NIV BUCHBINDER, TH HUBERT CHAN, SHAHAR CHEN, ILAN REUVEN COHEN, ANUPAM GUPTA, ZHIYI HUANG, NING KANG, VISWANATH NAGARAJAN, JOSEPH NAOR, ET AL.: Online algorithms for covering and packing problems with convex objectives. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 148–157. IEEE, 2016. [doi:10.1109/FOCS.2016.24] 9

[15] NIKHIL BANSAL, NIV BUCHBINDER, AND JOSEPH NAOR: Randomized competitive algorithms for generalized caching. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 235–244, 2008. [doi:10.1145/1374376.1374412] 3

[16] NIKHIL BANSAL, NIV BUCHBINDER, AND JOSEPH NAOR: A primal-dual randomized algorithm for weighted paging. *Journal of the ACM (JACM)*, 59(4):1–24, 2012. [doi:10.1145/2339123.2339126] 3, 5

[17] SURENDER BASWANA: Streaming algorithm for graph spanners - single pass and constant processing time per edge. *Information Processing Letters*, 106(3):110–114, 2008. [doi:10.1016/j.ipl.2007.11.001] 8

[18] SURENDER BASWANA AND TELIKEPALLI KAVITHA: Faster algorithms for all-pairs approximate shortest paths in undirected graphs. *SIAM Journal on Computing*, 39(7):2865–2896, 2010. [doi:10.1137/080737174] 2

[19] MOHAMMADHOSSEIN BATENI AND MOHAMMADTAGHI HAJIAGHAYI: Euclidean prize-collecting steiner forest. *Algorithmica*, 62(3-4):906–929, 2012. [doi:10.1007/s00453-011-9491-8] 2

[20] PIOTR BERMAN, ARNAB BHATTACHARYYA, KONSTANTIN MAKARYCHEV, SOFYA RASKHODNIKOVA, AND GRIGORY YAROSLAVTSEV: Approximation algorithms for spanner problems and directed steiner forest. *Information and Computation*, 222:93–107, 2013. [doi:10.1016/j.ic.2012.10.007] 3, 4, 5, 8, 11, 28, 29, 30, 31, 45

[21] PIOTR BERMAN AND CHRIS COULSTON: On-line algorithms for steiner tree problems (extended abstract). In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 344–353, 1997. [doi:10.1145/258533.258618] 3, 8

[22] ARNAB BHATTACHARYYA, ELENA GRIGORESCU, KYOMIN JUNG, SOFYA RASKHODNIKOVA, AND DAVID P WOODRUFF: Transitive-closure spanners. *SIAM Journal on Computing*, 41(6):1380–1425, 2012. [doi:10.1137/110826655] 2, 3, 11, 35

[23] GREG BODWIN AND VIRGINIA VASSILEVSKA WILLIAMS: Better distance preservers and additive spanners. *ACM Transactions on Algorithms (TALG)*, 17(4):1–24, 2021. [doi:10.1145/3490147] 8

[24] GLENCORA BORRADAILE, PHILIP N KLEIN, AND CLAIRE MATHIEU: A polynomial-time approximation scheme for euclidean steiner forest. *ACM Transactions on Algorithms (TALG)*, 11(3):1–20, 2015. [doi:10.1145/2629654] 2

[25] NIV BUCHBINDER AND JOSEPH NAOR: Improved bounds for online routing and packing via a primal-dual approach. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 293–304. IEEE, 2006. [doi:10.1109/FOCS.2006.39] 3

[26] NIV BUCHBINDER AND JOSEPH NAOR: Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009. [doi:10.1287/moor.1080.0363] 2, 4, 5, 6, 7, 9, 37, 38, 40, 42

[27] NIV BUCHBINDER AND JOSEPH SEFFI NAOR: The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3(2–3):93–263, 2009. [doi:10.1561/0400000024] 5

[28] DEEPARNAB CHAKRABARTY, ALINA ENE, RAVISHANKAR KRISHNASWAMY, AND DEBMALYA PANIGRAHI: Online buy-at-bulk network design. *SIAM Journal on Computing*, 47(4):1505–1528, 2018. [doi:10.1137/16M1117317] 4, 5, 7, 8, 10, 13, 20, 25, 27, 37, 45

[29] MOSES CHARIKAR, CHANDRA CHEKURI, TO-YAT CHEUNG, ZUO DAI, ASHISH GOEL, SUDIPTO GUHA, AND MING LI: Approximation algorithms for directed steiner problems. *Journal of Algorithms*, 33(1):73–91, 1999. [doi:10.1006/jagm.1999.1042] 8

[30] SHUCHI CHAWLA, TIM ROUGHGARDEN, AND MUKUND SUNDARARAJAN: Optimal cost-sharing mechanisms for steiner forest problems. In *Proceedings of the 2nd International Workshop on Internet and Network Economics (WINE)*, pp. 112–123. Springer, 2006. [doi:10.1007/11944874_11] 2

[31] SHIRI CHECHIK: Approximate distance oracles with improved bounds. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1–10, 2015. [doi:10.1145/2746539.2746562] 2

[32] CHANDRA CHEKURI, GUY EVEN, ANUPAM GUPTA, AND DANNY SEGEV: Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Transactions on Algorithms (TALG)*, 7(2):1–17, 2011. [doi:10.1145/1921659.1921664] 4, 5, 8, 10, 11, 14, 17

[33] CHANDRA CHEKURI, MOHAMMAD TAGHI HAJIAGHAYI, GUY KORTSARZ, AND MOHAMMAD R SALAVATIPOUR: Approximation algorithms for nonuniform buy-at-bulk network design. *SIAM Journal on Computing*, 39(5):1772–1798, 2010. [doi:10.1137/090750317] 10

[34] EDEN CHLAMTÁČ, MICHAEL DINITZ, GUY KORTSARZ, AND BUNDIT LAEKHANUKIT: Approximating spanners and directed steiner forest: Upper and lower bounds. *ACM Transactions on Algorithms (TALG)*, 16(3):1–31, 2020. [doi:10.1145/3381451] 3, 5, 7, 8, 10, 11, 12, 13, 15, 17, 19, 20, 24, 28, 29, 30, 31, 45

[35] LENORE COWEN AND CHRISTOPHER G. WAGNER: Compact roundtrip routing in directed networks. *Journal of Algorithms*, 50(1):79–95, 2004. [doi:10.1016/j.jalgor.2003.08.001] 2

[36] BILEL DERBEL, CYRIL GAVOILLE, AND DAVID PELEG: Deterministic distributed construction of linear stretch spanners in polylogarithmic time. In ANDRZEJ PELC, editor, *DISC*, volume 4731 of *Lecture Notes in Computer Science*, pp. 179–192. Springer, 2007. [doi:10.1007/978-3-540-75142-7_16] 8

[37] BILEL DERBEL, CYRIL GAVOILLE, DAVID PELEG, AND LAURENT VIENNOT: On the locality of distributed sparse spanner construction. In *Proceedings of the 27th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 273–282, 2008. [doi:10.1145/1400751.1400788] 8

[38] MICHAEL DINITZ AND ROBERT KRAUTHGAMER: Directed spanners via flow-based linear programs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 323–332, 2011. [doi:10.1145/1993636.1993680] 7, 29, 30, 34

[39] MICHAEL DINITZ, YASAMIN NAZARI, AND ZEYU ZHANG: Lasserre integrality gaps for graph spanners and related problems. In *Proceedings of the 19th International Workshop on Approximation and Online Algorithms (WAOA)*, pp. 97–112. Springer, 2021. [doi:10.1007/978-3-030-80879-2_7] 7, 29, 34

[40] MICHAEL DINITZ AND ZEYU ZHANG: Approximating low-stretch spanners. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 821–840. SIAM, 2016. [doi:10.1137/1.9781611974331.ch59] 3

[41] DORIT DOR, SHAY HALPERIN, AND URI ZWICK: All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000. [doi:10.1137/S0097539797327908] 2

[42] MICHAEL ELKIN: Computing almost shortest paths. *ACM Transactions on Algorithms (TALG)*, 1(2):283–323, 2005. [doi:10.1145/1103963.1103968] 2

[43] MICHAEL ELKIN: Streaming and fully dynamic centralized algorithms for constructing and maintaining sparse spanners. *ACM Transactions on Algorithms (TALG)*, 7(2):1–17, 2011. [doi:10.1145/1921659.1921666] 8

[44] MICHAEL ELKIN AND DAVID PELEG: The client-server 2-spanner problem with applications to network design. In FRANCESC COMELLAS, JOSEP FÀBREGA, AND PIERRE FRAIGNIAUD, editors, *SIROCCO 8, Proceedings of the 8th International Colloquium on Structural Information and Communication Complexity, Vall de Núria, Girona-Barcelona, Catalonia, Spain, 27-29 June, 2001*, volume 8 of *Proceedings in Informatics*, pp. 117–132. Carleton Scientific, 2001. 3, 36

[45] MICHAEL ELKIN AND DAVID PELEG: The hardness of approximating spanner problems. *Theory of Computing Systems*, 41(4):691–729, 2007. [doi:10.1007/s00224-006-1266-2] 3

[46] MORAN FELDMAN, GUY KORTSARZ, AND ZEEV NUTOV: Improved approximation algorithms for directed steiner forest. *Journal of Computer and System Sciences*, 78(1):279–292, 2012. [doi:10.1016/j.jcss.2011.05.009] 11

[47] MANUEL FERNÁNDEZ V, DAVID P WOODRUFF, AND TAISUKE YASUDA: Graph spanners in the message-passing model. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. [doi:10.4230/LIPIcs.ITCS.2020.77] 8

[48] ARNOLD FILTSER, MICHAEL KAPRALOV, AND NAVID NOURI: Graph spanners by sketching in dynamic streams and the simultaneous communication model. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1894–1913. SIAM, 2021. [doi:10.1137/1.9781611976465.11] 8

[49] LISA FLEISCHER, JOCHEN KÖNEMANN, STEFANO LEONARDI, AND GUIDO SCHÄFER: Simple cost sharing schemes for multicommodity rent-or-buy and stochastic steiner tree. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 663–670, 2006. [doi:10.1145/1132516.1132609] 2

[50] NAVEEN GARG, GORAN KONJEVOD, AND RAMAMOORTHI RAVI: A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000. [doi:10.1006/jagm.2000.1096] 13, 27

[51] MICHEL X. GOEMANS AND DAVID P. WILLIAMSON: A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995. [doi:10.1137/S0097539793242618] 2

[52] ELENA GRIGORESCU, YOUNG-SAN LIN, AND KENT QUANRUD: Online directed spanners and steiner forests. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2021. [doi:10.4230/LIPIcs.APPROX/RANDOM.2021.5] 1

[53] ANUPAM GUPTA, AMIT KUMAR, MARTIN PÁL, AND TIM ROUGHGARDEN: Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 606–615. IEEE, 2003. [doi:10.1109/SFCS.2003.1238233] 2

[54] ANUPAM GUPTA AND VISWANATH NAGARAJAN: Approximating sparse covering integer programs online. *Mathematics of Operations Research*, 39(4):998–1011, 2014. [doi:10.1287/moor.2014.0652] 9

[55] ANUPAM GUPTA, R RAVI, KUNAL TALWAR, AND SEEUN WILLIAM UMBOH: Last but not least: Online spanners for buy-at-bulk. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 589–599. SIAM, 2017. [doi:10.1137/1.9781611974782.38] 8

[56] ANUPAM GUPTA, KUNAL TALWAR, AND UDI WIEDER: Changing bases: Multistage optimization for matroids and matchings. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 563–575. Springer, 2014. [doi:10.1007/978-3-662-43948-7_47] 9

[57] REFAEL HASSIN: Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 17(1):36–42, 1992. [doi:10.1287/moor.17.1.36] 34

[58] CHRISTOPHER S HELVIG, GABRIEL ROBINS, AND ALEXANDER ZELIKOVSKY: An improved approximation scheme for the group steiner problem. *Networks: An International Journal*, 37(1):8–20, 2001. [doi:10.1002/1097-0037(200101)37:1<8::AID-NET2>3.0.CO;2-R] 10, 11, 17

[59] MAKOTO IMASE AND BERNARD M WAXMAN: Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991. [doi:10.1137/0404033] 3

[60] MICHAEL KAPRALOV AND DAVID P. WOODRUFF: Spanners and sparsifiers in dynamic streams. In MAGNÚS M. HALLDÓRSSON AND SHLOMI DOLEV, editors, *Proceedings of the 33rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 272–281. ACM, 2014. [doi:10.1145/2611462.2611497] 8

[61] ANNA R. KARLIN, MARK S. MANASSE, LYLE A. MCGEOCH, AND SUSAN S. OWICKI: Competitive randomized algorithms for non-uniform problems. *Algorithmica*, 11(6):542–571, 1994. [doi:10.1007/BF01189993] 5

[62] VIKRAM KHURANA, JIAN PENG, CHEE YEUN CHUNG, PAVAN K AULUCK, SARANNA FANNING, DANIEL F TARDIFF, THERESA BARTELS, MARTINA KOEVA, STEPHEN W EICHHORN, HADAR BENYAMINI, ET AL.: Genome-scale networks link neurodegenerative disease genes to α-synuclein through specific molecular pathways. *Cell systems*, 4(2):157–170, 2017. [doi:10.1016/j.cels.2016.12.011] 2

[63] JOCHEN KÖNEMANN, STEFANO LEONARDI, GUIDO SCHÄFER, AND STEFAN VAN ZWAM: From primal-dual to cost shares and back: a stronger lp relaxation for the steiner forest problem. In *Proceedings of the 32nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 930–942. Springer, 2005. [doi:10.1007/11523468_75] 2

[64] JOCHEN KÖNEMANN, STEFANO LEONARDI, GUIDO SCHÄFER, AND STEFAN HM VAN ZWAM: A group-strategyproof cost sharing mechanism for the steiner forest game. *SIAM Journal on Computing*, 37(5):1319–1341, 2008. [doi:10.1137/050646408] 2

[65] GUY KORTSARZ: On the hardness of approximating spanners. *Algorithmica*, 30:432–450, 2001. [doi:10.1007/s00453-001-0021-y] 3

[66] DEAN H LORENZ AND DANNY RAZ: A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001. [doi:10.1016/S0167-6377(01)00069-4] 34

[67] JAKUB PACHOCKI, LIAM RODITTY, AARON SIDFORD, ROEI TOV, AND VIRGINIA VASSILEVSKA WILLIAMS: Approximating cycles in directed graphs: Fast algorithms for girth and roundtrip spanners. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1374–1392. SIAM, 2018. [doi:10.1137/1.9781611975031.91] 2

[68] MIHAI PATRASCU AND LIAM RODITTY: Distance oracles beyond the Thorup-Zwick bound. *SIAM Journal on Computing*, 43(1):300–311, 2014. [doi:10.1137/11084128X] 2

[69] DAVID PELEG AND ALEJANDRO A. SCHÄFFER: Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. [doi:10.1002/jgt.3190130114] 2

[70] DAVID PELEG AND JEFFREY D. ULLMAN: An optimal synchronizer for the hypercube. *SIAM Journal on Computing*, 18(4):740–747, 1989. [doi:10.1137/0218050] 2

[71] LEILA PIRHAJI, PAMELA MILANI, MATHIAS LEIDL, TIMOTHY CURRAN, JULIAN AVILA-PACHECO, CLARY B CLISH, FOREST M WHITE, ALAN SAGHATELIAN, AND ERNEST FRAENKEL: Revealing disease-associated pathways by network integration of untargeted metabolomics. *Nature methods*, 13(9):770–776, 2016. [doi:10.1038/nmeth.3940] 2

[72] IAM RODITTY, MIKKEL THORUP, AND URI ZWICK: Roundtrip spanners and roundtrip routing in directed graphs. *ACM Transactions on Algorithms (TALG)*, 4(3):1–17, 2008. [doi:10.1145/1367064.1367069] 2

[73] TIM ROUGHGARDEN AND MUKUND SUNDARARAJAN: Optimal efficiency guarantees for network design mechanisms. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pp. 469–483. Springer, 2007. [doi:10.1007/978-3-540-72792-7_35] 2

[74] XIANGKUN SHEN AND VISWANATH NAGARAJAN: Online covering with ℓ_q-norm objectives and applications to network design. *Mathematical Programming*, 184, 2020. [doi:10.1007/s10107-019-01409-9] 8, 9, 21

[75] ANDREW CHI-CHIH YAO: Space-time tradeoff for answering range queries (extended abstract). In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)*, 1982. [doi:10.1145/800070.802185] 2

[76] NEAL YOUNG: The k-server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, 1994. [doi:10.1007/BF01189992] 5

[77] ALEXANDER ZELIKOVSKY: A series of approximation algorithms for the acyclic directed steiner tree problem. *Algorithmica*, 18(1):99–110, 1997. [doi:10.1007/BF02523690] 8

AUTHORS

Elena Grigorescu
Associate professor
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA
elena-g@purdue.edu
https://www.cs.purdue.edu/homes/egrigore/

Young-San Lin
Postdoctoral Research Fellow
Melbourne Business School
Carlton, Victoria, Australia
y.lin@mbs.edu
https://mbs.edu/faculty-and-research/faculty/young-san-lin

Kent Quanrud
Assistant Professor
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA
krq@purdue.edu
https://kentquanrud.com/

ABOUT THE AUTHORS

ELENA GRIGORESCU is an associate professor in the Computer Science department at Purdue University, which she joined in 2012. She graduated from MIT CSAIL in 2010, where her advisor was Madhu Sudan. After that she spent two years as a CIF postdoctoral fellow at Georgia Tech. Her research interests have focused on designing algorithms that use only sublinear time or sublinear space, in computational models relevant to large data sets, and on understanding the complexity of fundamental problems on error-correcting codes and point lattices. She is broadly interested in understanding information-theoretical limits in diverse computational models. She is also interested in sighting a yellow cardinal.

YOUNG-SAN LIN is a postdoctoral research fellow at Melbourne Business School, which he joined in 2022. He completed his Ph. D.in the Computer Science Department at Purdue University. His research interests are in the interdisciplinary field of theoretical computer science, economics, and operations research, with a focus on market design, revenue management, resource allocation, and online algorithms. He enjoys cooking and playing volleyball during his leisure time.

KENT QUANRUD is an assistant professor of Computer Science at Purdue University. He received his Ph. D.from the University of Illinois at Urbana-Champaign in 2019, where he was advised by Chandra Chekuri and Sariel Har-Peled. He was born in Japan and grew up in California. His research interests include approximation algorithms, randomized algorithms, combinatorial optimization, continuous optimization, online learning, and computational geometry. He is particularly interested in extremely scalable algorithms for fundamental problems in optimization