Online Weighted Degree-Bounded Steiner Networks via Novel Online Mixed Packing/Covering*

Sina Dehghani[1], Soheil Ehsani[2], Mohammad Hajiaghayi[3], Vahid Liaghat[4], Harald Räcke[5], and Saeed Seddighin[6]

1 **University of Maryland, College Park, USA**
dehghani@cs.umd.edu
2 **University of Maryland, College Park, USA**
ehsani@cs.umd.edu
3 **University of Maryland, College Park, USA**
hajiagha@cs.umd.edu
4 **Stanford University, Palo Alto, USA**
vliaghat@stanford.edu
5 **Technische Universität München, München, Germany**
raecke@in.tum.de
6 **University of Maryland, College Park, USA**
saeedrez@cs.umd.edu

Abstract

We design the first online algorithm with poly-logarithmic competitive ratio for the *edge-weighted degree-bounded Steiner forest* (EW-DB-SF) problem and its generalized variant. We obtain our result by demonstrating a new generic approach for solving mixed packing/covering integer programs in the online paradigm. In EW-DB-SF, we are given an edge-weighted graph with a degree bound for every vertex. Given a root vertex in advance, we receive a sequence of terminal vertices in an online manner. Upon the arrival of a terminal, we need to augment our solution subgraph to connect the new terminal to the root. The goal is to minimize the total weight of the solution while respecting the degree bounds on the vertices. In the offline setting, *edge-weighted degree-bounded Steiner tree* (EW-DB-ST) and its many variations have been extensively studied since early eighties. Unfortunately, the recent advancements in the online network design problems are inherently difficult to adapt for degree-bounded problems. In particular, it is not known whether the fractional solution obtained by standard primal-dual techniques for mixed packing/covering LPs can be rounded online. In contrast, in this paper we obtain our result by using structural properties of the optimal solution, and reducing the EW-DB-SF problem to an exponential-size mixed packing/covering integer program in which every variable appears only once in covering constraints. We then design a generic *integral* algorithm for solving this restricted family of IPs.

As mentioned above, we demonstrate a new technique for solving mixed packing/covering integer programs. Define the *covering frequency* k of a program as the maximum number of covering constraints in which a variable can participate. Let m denote the number of packing constraints. We design an online *deterministic integral algorithm* with competitive ratio of $O(k \log m)$ for the mixed packing/covering integer programs. We prove the tightness of our result by providing a matching lower bound for any randomized algorithm. We note that our solution solely depends on m and k. Indeed, there can be exponentially many variables. Furthermore,

* Supported in part by NSF AF:Small grant CCF-1216698, NSF CAREER award CCF-1053605, NSF BIGDATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 42; pp. 42:1–42:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

our algorithm directly provides an integral solution, even if the integrality gap of the program is unbounded. We believe this technique can be used as an interesting alternative for the standard primal-dual techniques in solving online problems.

1998 ACM Subject Classification I.1.2 Algorithms, G.2.2 [Graph Theory] Network problems

Keywords and phrases Online, Steiner Tree, Approximation, Competitive ratio

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.42

1 Introduction

Degree-bounded network design problems comprise an important family of network design problems since the eighties. Aside from various real-world applications such as vehicle routing and communication networks [6, 32, 38], the family of degree-bounded problems has been a testbed for developing new ideas and techniques. The problem of *degree-bounded spanning tree*, introduced in Garey and Johnson's *Black Book* of NP-Completeness [29], was first investigated in the pioneering work of Fürer and Raghavachari [15]. In this problem, we are required to find a spanning tree of a given graph with the goal of minimizing the maximum degree of the vertices in the tree. Let b^* denote the maximum degree in the optimal spanning tree. Fürer and Raghavachari give a parallel approximation algorithm which produces a spanning tree of degree at most $O(\log(n)b^*)$. This result was later generalized by Agrawal, Klein, and Ravi [1] to the case of degree-bounded Steiner tree (DB-ST) and degree bounded Steiner forest (DB-SF) problem. In DB-ST, given a set of terminal vertices, we need to find a subgraph of minimum maximum degree that connects the terminals. In the more generalized DB-SF problem, we are given pairs of terminals and the output subgraph should contain a path connecting each pair. Fürer and Raghavachari [16] significantly improved the result for DB-SF by presenting an algorithm which produces a Steiner forest with maximum degree at most $b^* + 1$.

The study of DB-ST and DB-SF was the starting point of a very popular line of work on various degree-bounded network design problems; e.g. [28, 31, 27, 22, 13] and more recently [14, 13]. One particular variant that has been extensively studied was initiated by Marathe et al. [28]: In the *edge-weighted degree-bounded spanning tree* problem, given a weight function over the edges and a degree bound b, the goal is to find a minimum-weight spanning tree with maximum degree at most b. The initial results for the problem generated much interest in obtaining approximation algorithms for the edge-weighted degree-bounded spanning tree problem [11, 10, 17, 23, 24, 25, 26, 34, 35, 36]. The groundbreaking results obtained by Goemans [18] and Singh and Lau [37] settle the problem by giving an algorithm that computes a minimum-weight spanning tree with degree at most $b+1$. Singh and Lau [27] generalize their result for the *edge-weighted Steiner tree* (EW-DB-ST) and *edge-weighted Steiner forest* (EW-DB-SF) variants. They design an algorithm that finds a Steiner forest with cost at most twice the cost of the optimal solution while violating the degree constraints by at most three.

Despite these achievements in the offline setting, it was not known whether degree-bounded problems are tractable in the *online setting*. The online counterparts of the aforementioned Steiner problems can be defined as follows. The underlying graph and degree bounds are known in advance. The demands arrive one by one in an online manner. At the arrival of a demand, we need to augment the solution subgraph such that the new demand is satisfied. The goal is to be competitive against an offline optimum that knows the demands in advance.

Recently, Dehghani et al. [12] explore the tractability of the Online DB-SF problem by showing that a natural greedy algorithm produces a solution in which the degree bounds are violated by at most a factor of $O(\log n)$, which is asymptotically *tight*. They analyze their algorithm using a dual fitting approach based on the combinatorial structures of the graph such as the toughness[1] factor. Unfortunately, they can also show that greedy methods are not competitive for the edge-weighted variant of the problem. Hence, it seems unlikely that the approach of [12] can be generalized to EW-DB-SF.

The *online edge-weighted Steiner connectivity* problems (with no bound on the degrees) have been extensively studied in the last decades. Imase and Waxman [21] use a dual-fitting argument to show that the greedy algorithm has a competitive ratio of $O(\log n)$, which is also asymptotically tight. Later the result was generalized to the EW SF variant by Awerbuch et al. [4] and Berman and Coulston [7]. In the past few years, various primal-dual techniques have been developed to solve the more general node-weighted variants [2, 30, 19], prize-collecting variants [33, 20], and multicommodity buy-at-bulk [9]. These results are obtained by developing various primal-dual techniques [2, 19] while generalizing the application of combinatorial properties to the online setting [30, 20, 9]. In this paper however, we develop a primal approach for solving *bounded-frequency mixed packing/covering integer programs*. We believe this framework would be proven useful in attacking other online packing and covering problems.

1.1 Our Results and Techniques

In this paper, we consider the online Steiner tree and Steiner forest problems at the presence of both edge weights and degree bounds. In the Online EW-DB-SF problem, we are given a graph $G = (V, E)$ with n vertices, edge-weight function w, degree bound b_v for every $v \in V$, and an online sequence of connectivity demands (s_i, t_i). Let w_{opt} denote the minimum weight subgraph which satisfies the degree bounds and connects all demands. Let $\rho = \frac{\max_e w(e)}{\min_{e:w(e)>0} w(e)}$.

▶ **Theorem 1.** *There exists an online deterministic algorithm which finds a subgraph with total weight at most $O(\log^3 n)w_{opt}$ while the degree bound of a vertex is violated by at most a factor of $O(\log^3(n)\log(n\rho))$.*

If one favors the degree bounds over total weight, one can find a subgraph with degree-bound violation $O(\log^3(n)\frac{\log(n\rho))}{\log\log(n\rho)})$ and total cost $O(\log^3(n)\frac{\log(n\rho))}{\log\log(n\rho)})w_{opt}$.

We note that the logarithmic dependency on ρ is indeed necessary. It follows from the result of [12] that the competitive ratio of any algorithm is either $\Omega(n)$ or $\Omega(\log \rho)$.

Our technical contribution for solving the EW-DB-SF problem is twofold. First by exploiting a structural result and massaging the optimal solution, we show a formulation of the problem that falls in the restricted family of *bounded-frequency mixed packing/cover IPs*, while losing only logarithmic factors in the competitive ratio. We then design a generic online algorithm with a logarithmic competitive ratio that can solve any instance of the bounded-frequency packing/covering IPs. In what follows, we describe our results in detail.

1.1.1 Massaging the optimal solution

Initiated by work of Alon et al. [2] on online set cover, Buchbinder and Naor developed a strong framework for solving packing/covering LPs *fractionally* online. For the applications

[1] The toughness of a graph is defined as $\min_{X\subseteq V} \frac{|X|}{|\mathrm{CC}(G\setminus X)|}$; where for a graph H, $\mathrm{CC}(H)$ denotes the collection of connected components of H.

of their general framework in solving numerous online problems, we refer the reader to the survey in [8]. Azar et al. [5] generalize this method for the fractional *mixed* packing and covering LPs. The natural linear program relaxation for EW-DB-SF, commonly used in the literature, is a special case of mixed packing/covering LPs: one needs to select an edge from every cut that separates the endpoints of a demand (covering constraints), while for a vertex we cannot choose more than a specific number of its adjacent edges (packing constraints). Indeed, one can use the result of Azar et al. [5] to find an online *fractional* solution with polylogarithmic competitive ratio. However, doing the rounding in an online manner seems very hard.

Offline techniques for solving degree-bounded problems often fall in the category of iterative and dependent rounding methods. Unfortunately, these methods are inherently difficult to adapt for an online settings since the underlying fractional solution may change dramatically in between the rounding steps. Indeed, this might be the very reason that despite many advances in the online network design paradigm in the past two decades, the natural family of degree-bounded problems has remained widely open. In this paper, we circumvent this by reducing EW-DB-ST to a novel formulation beyond the scope of standard online packing/covering techniques and solving it using a new online integral approach.

The crux of our IP formulation is the following structural property: Let (s_i, t_i) denote the i^{th} demand. We need to augment the solution Q_{i-1} of previous steps by buying a subgraph that makes s_i and t_i connected. Let G_i denote the graph obtained by contracting the pairs of vertices s_j and t_j for every $j < i$. Note that any $(s_i - t_i)$-path in G_i corresponds to a feasible augmentation for Q_{i-1}. Some edges in G_i might be already in Q_{i-1} and therefore by using them again we can save both on the total weight and the vertex degrees. However, in Section 2 we prove that there always exists a path in G_i such that even without sharing on any of the edges in G_i and therefore paying completely for the increase in the weight and degrees, we can approximate the optimal solution up to a logarithmic factor. This in fact, enables us to have a formulation in which the covering constraints for different demands are *disentangled*. Indeed, we only have one covering constraint for each demand. Unfortunately, this implies that we have exponentially many variables, one for each possible path in G_i. This may look hopeless since the competitive factors obtained by standard fractional packing/covering methods introduced by Buchbinder and Naor [8] and Azar et al. [5], depend on the logarithm of the number of variables. Therefore we come up with a new approach for solving this class of mixed packing/covering integer programs (IP).

1.1.2 Bounded-frequency mixed packing/covering IPs

We derive our result for EW-DB-ST by demonstrating a new technique for solving mixed packing/covering *integer* programs. We believe this approach could be applicable to a broader range of online problems. The integer program $\mathbb{IP}1$ describes a general mixed packing/covering IP with the set of integer variables $\mathbf{x} \in \mathbb{Z}^n_{\geq 0}$ and α. The packing constraints are described by a $m \times n$ non-negative matrix P. Similarly, the $q \times n$ matrix C describes the covering constraints. The *covering frequency* of a variable x_i is defined as the number of covering constraints in which x_i has a positive coefficient. The covering frequency of a mixed packing/covering program is defined as the maximum covering frequency of its variables.

$$\begin{aligned} \text{minimize} \quad & \alpha \ , \\ s.t. \quad & P\mathbf{x} \leq \alpha \ . \\ & C\mathbf{x} \geq 1 \ . \\ & \mathbf{x} \in \mathbb{Z}_{\geq 0}, \alpha \in \mathbb{R}_{>0} \ . \end{aligned} \tag{$\mathbb{IP}1$}$$

In the online variant of mixed packing and covering IP, we are given the packing constraints in advance. However the covering constraints arrive in an online manner. At the arrival of each covering constraint, we should *increase* the solution $\mathbf{x}$ such that it satisfies the new covering constraint. We provide a deterministic algorithm for solving online mixed packing/covering IPs.

▶ **Theorem 2.** *Given an instance of the online mixed packing/covering IP, there exists a deterministic integral algorithm with competitive ratio $O(k \log m)$, where m is the number of packing constraints and k is the covering frequency of the IP.*

We note that the competitive ratio of our algorithm is independent of the number of variables or the number of covering constraints. Indeed, there can be exponentially many variables.

Our result can be thought of as a generalization of the work of Aspnes et al. [3] on virtual circuit routing. Although not explicit, their result can be massaged to solve mixed packing/covering IPs in which all the coefficients are zero or one, and the covering frequency is one. They show that such IPs admit a $O(\log(m))$-competitive algorithms. Theorem 2 generalizes their result to the case with arbitrary non-negative coefficients and any bounded covering frequency.

We complement our result by proving a matching lower bound for the competitive ratio of any *randomized* algorithm. This lower bound holds even if the algorithm is allowed to return fractional solutions.

▶ **Theorem 3.** *Any randomized online algorithm A for integral mixed packing and covering is $\Omega(k \log m)$-competitive, where m denotes the number of packing constraints, and k denotes the covering frequency of the IP. This even holds if A is allowed to return a fractional solution.*

As mentioned before, Azar et al. [5] provide a fractional algorithm for mixed packing/-covering LPs with competitive ratio of $O(\log m \log d)$ where d is the maximum number of variables in a single constraint. They show an almost matching lower bound for deterministic algorithms. We distinguish two advantages of our approach compared to that of Azar et al.:

- The algorithm in [5] outputs a *fractional* competitive solution which then needs to be rounded online. For various problems such as Steiner connectivity problems, rounding a solution online is very challenging, even if offline rounding techniques are known. Moreover, the situation becomes hopeless if the integrality gap is unbounded. However, for bounded-frequency IPs, our algorithm directly produces an integral competitive solution. Thus it does not depend on rounding methods, and is applicable to problems with large integrality gap, or the problems for which it is shown that rounding methods do not preserve any approximation guarantee, and as such, the traditional approach fails.
- Azar et al. find the best competitive ratio with respect to the number of packing constraints and the size of constraints. Although these parameters are shown to be bounded in several problems, in many problems such as connectivity problems and flow problems, formulations with exponentially many variables are very natural. Our techniques provide an alternative solution with a tight competitive ratio, for formulations with bounded covering frequency.

1.2 Preliminaries

Let $G = (V, E)$ be an undirected graph of size n ($|V| = n$). Let $w : E \to \mathbb{Z}_{>0}$ be a function denoting the edge weights. For a subgraph $H \subseteq G$, we define $w(H) := \sum_{e \in E(H)} w(e)$. For every vertex $v \in V$, let $b_v \in \mathbb{Z}_{>0}$ denote the degree bound of v. Let $\deg_H(v)$ denote the degree of vertex v in subgraph H. We define the load $l_H(v)$ of vertex v w.r.t. H as

$\deg_H(v)/b_v$. In DB-SF we are given graph G, degree bounds, and k connectivity demands. Let σ_i denote the i-th demand. The i-th demand is a pair of vertices $\sigma_i = (s_i, t_i)$, where $s_i, t_i \in V$. In DB-SF the goal is to find a subgraph $H \subseteq G$ such that for each demand σ_i, s_i is connected to t_i in H, for every vertex $v \in V$, $l_H(v) \leq 1$, and $w(H)$ is minimized. In this paper without loss of generality we assume the demand endpoints are distinct vertices with degree one in G and degree bound infinity.

In the online variant of the problem, we are given graph G and degree bounds in advance. However the sequence of demands are given one by one. At arrival of demand σ_i, we are asked to provide a subgraph H_i, such that $H_{i-1} \subseteq H_i$ and s_i is connected to t_i in H_i.

The following integer program is a natural mixed packing and covering integer program for EW-DB-SF. Let $\mathcal{S}$ denote the collection of subsets of vertices that separate the endpoints of at least one demand. For a set of vertices S, let $\delta(S)$ denote the set of edges with exactly one endpoint in S. In $\mathbb{SF\_IP}$, for an edge e, $x_e = 1$ indicates that we include e in the solution while $x_e = 0$ indicates otherwise. The variable α indicates an upper bound on the violation of the load of every vertex and an upper bound on the violation of the weight. The first set of constraints ensures that the load of a vertex is upper bounded by α. The second constraint ensures that the violation for the weight is upper bounded by α. The third set of constraints ensures that the endpoints of every demand are connected. Here we assume w_{opt} is known to the algorithm, although this can be waived by standard doubling techniques.

$$\text{minimize} \quad \alpha \ . \qquad (\mathbb{SF\_IP})$$

$$\forall v \in V \quad \frac{1}{b_v} \sum_{e \in \delta(\{v\})} x_e \leq \alpha \ . \qquad (1)$$

$$\frac{1}{w_{\text{opt}}} \sum_{e \in E} w(e) x_e \leq \alpha \ . \qquad (2)$$

$$\forall S \subseteq \mathcal{S} \quad \sum_{e \in \delta(S)} x_e \geq 1 \ . \qquad (3)$$

$$x_e \in \{0,1\}, \alpha \in \mathbb{Z}_{>0} \ .$$

1.3 Overview of the Paper

We begin Section 2 by providing a bounded frequency IP for EW-DB-SF. The IP is not a proper formulation of the problem, however, we can show that one can map feasible solutions of EW-DB-SF to feasible solutions of the IP without increasing the cost too much. In Section 3 we provide a deterministic algorithm for online bounded frequency mixed packing/covering IPs. In the full version of the paper, we also provide a matching lower bound for the competitive ratio of any randomized algorithm. Finally, in Section 4 we merge our techniques to obtain online polylogarithmic-competitive algorithms for EW-DB-SF.

2 Finding the Right Integer Program

In this section we design an online mixed packing and covering integer program for EW-DB-SF. We show this formulation is near optimal, i.e. any $f-$approximation for this formulation, implies an $O(f \log^2 n)$-approximation for EW-DB-SF. In Section 4 we show there exists an online algorithm that finds an $O(\log n)$-approximation solution for this IP and violates degree bounds by $O(\log^3 n \log w_{\text{opt}})$, where w_{opt} denotes the optimal weight.

First we define some notations. For a sequence of demands $\sigma = \langle (s_1, t_1), \ldots, (s_k, t_k) \rangle$, we define $R_\sigma(i)$ to be a set of i edges, connecting the endpoints of the first i demands.

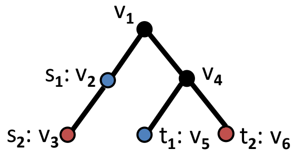

Figure 1 An example where every vertex has degree-bound 3 and every edge has weight 1. The first demand is (v_2, v_5) and the second demand is (v_3, v_6). The optimal solution for $\mathbb{SF\_IP}$ is a subgraph, say H, with the set of all edges and vertices, i.e. $H = G$. However an optimal solution for $\mathbb{PC\_IP}$ is: Two subgraphs H_1 for the first request which has edges $\{e(v_1, v_2), e(v_1, v_4), e(v_4, v_5)\}$ and H_2 for the second request which has edges $\{e(v_2, v_3), e(v_4, v_5), e(v_4, v_6)\}$. Note that $w(H) = 5$ and $w(H_1) + w(H_2) = 6$, since we have edge $e(v_4, v_5)$ in both H_1 and H_2. Moreover the number of edges incident with v_4 in the solution of $\mathbb{PC\_IP}$ is 4, i.e. $\deg_{H_1}(v_4) + \deg_{H_2}(v_4) = 4$.

In particular $R_\sigma(i) := \bigcup_{j=1}^{i} e(s_j, t_j)$, where $e(s_j, t_j)$ denotes a direct edge from s_j to t_j. Moreover, we say subgraph H_i satisfies the connectivity of demand $\sigma_i = (s_i, t_i)$, if s_i and t_i are connected in graph $H_i \cup R_\sigma(i-1)$. Let $\mathcal{H}_i$ denote the set of all subgraphs that satisfy the connectivity of demand σ_i. In $\mathbb{PC\_IP}$ variable α denotes the violation in the packing constraints. Furthermore for every subgraph $H \subseteq G$ and demand σ_i, there exists a variable $x_H^i \in \{0, 1\}$. $x_H^i = 1$ indicates we add the edges of H to the existing solution, at arrival of demand σ_i. The first set of constraints ensure the degree-bounds are not violated more than α. The second constraint ensures the weight is not violated by more than α. The third set of constraints ensure the endpoints of every demand are connected.

$$\text{minimize} \quad \alpha \ . \qquad (\mathbb{PC\_IP})$$

$$\forall v \in V \qquad \frac{1}{b_v} \sum_{i=1}^{k} \sum_{H \subseteq G} \deg_H(v) x_H^i \le \alpha \ . \qquad (4)$$

$$\frac{1}{w_{\text{opt}}} \sum_{i=1}^{k} \sum_{H \subseteq G} w(H) x_H^i \le \alpha \ . \qquad (5)$$

$$\forall \sigma_i \qquad \sum_{H \in \mathcal{H}_i} x_H^i \ge 1 \ . \qquad (6)$$

$$\forall H \subseteq G, 1 \le i \le k \qquad \mathbf{x}_H^i \in \{0, 1\} \ .$$

$$\alpha > 0 \ .$$

We are considering the online variant of the mixed packing and covering program. We are given the packing Constraints (4) and (5) in advance. At arrival of demand σ_i, the corresponding covering Constraint (6) is added to the program. We are looking for an online solution which is feasible at every online stage. Moreover the variables x_H should be monotonic, i.e. once an algorithm sets $x_H = 1$ for some H, the value of x_H is 1 during the rest of the algorithm. Figure 1 illustrates an example which indicates the difference between the solutions of $\mathbb{PC\_IP}$ and $\mathbb{SF\_IP}$.

Let popt and lopt denote the optimal solutions for $\mathbb{PC\_IP}$ and $\mathbb{SF\_IP}$, respectively. Lemma 4 shows that given an online solution for $\mathbb{PC\_IP}$ we can provide a feasible online solution for $\mathbb{SF\_IP}$ of cost popt.

▶ **Lemma 4.** *Given a feasible solution $\{\mathbf{x}, \alpha\}$ for $\mathbb{PC\_IP}$, there exists a feasible solution $\{\mathbf{x}', \alpha\}$ for $\mathbb{SF\_IP}$.*

In the rest of this section, we show that we do not lose much by changing $\mathbb{SF\_IP}$ to $\mathbb{PC\_IP}$. In particular we show $\mathsf{popt} \le O(\log^2 n)\mathsf{lopt}$.

To this end, we first define the *connective* list of subgraphs for a graph G, a forest F, and a list of demands σ. We then prove an existential lemma for such a list of subgraphs with a desirable property for any $\langle G, F, \sigma\rangle$. With that in hand, we prove $\mathsf{popt} \le O(\log^2 n)\mathsf{lopt}$. In what follows, we refer the reader to the full version of the paper for detailed proofs.

Given G, a list of demands $\sigma = \langle (s_1, t_1), \ldots, (s_k, t_k)\rangle$, and a forest $F \subseteq G$:

▶ **Definition 5.** Let $Q = \langle Q_1, Q_2, Q_3, \ldots, Q_k\rangle$ be a list of k subgraphs of F. We say Q is a *connective list* of subgraphs for $\langle G, F, \sigma\rangle$ iff for every $1 \le i \le k$ there exists no cut disjoint from Q_i that separates s_i from t_i, but does not separate any s_j from t_j for $j < i$.

The intuition behind the definition of connective subgraphs is the following: If Q is a connective list of subgraphs for an instance $\langle G, F, \sigma\rangle$ then for every i we are guaranteed that the union of all subgraphs $\cup_{j=1}^{i} Q_i$ connects s_i to t_i. In Lemma 6 we show for every $\langle G, F, \sigma\rangle$, there exists a connective list of subgraphs for $\langle G, F, \sigma\rangle$, such that each edge of F appears in at most $O(\log^2 n)$ subgraphs of Q.

▶ **Lemma 6.** *Let G be a graph and F be a forest in G. If σ is a collection of k demands $\langle (s_1, t_1), \ldots, (s_k, t_k)\rangle$, then there exists a connective list of subgraphs $Q = \langle Q_1, Q_2, \ldots, Q_k\rangle$ for $\langle G, F, \sigma\rangle$ such that every edge of F appears in at most $3\log^2 |V(F)|$ number of Q_i's.*

Proof. Here we give a sketch of the proof of lemma; we refer the reader to the full version for detailed proofs. We first prove a cost-minimization variant of the lemma. Consider an arbitrary weight vector $\hat{w} : F \to \mathbb{R}^{\ge 0}$. We argue that there is a connective list Q, such that $\sum_i \hat{w}(Q_i) \le O(\log^2 n)\hat{w}(F)$. Let $\hat{H}_i = (V, F \cup R_\sigma(i), \hat{w}_i)$ denote a weighted graph for which $\hat{w}_i(e) = \hat{w}(e)$ for $e \in F$, and $\hat{w}_i(e) = 0$ for $e \in R_\sigma(i)$. Now we note that there is no cost-sharing among Q_i's in the goal $\sum_i \hat{w}(Q_i)$. Therefore the optimal choice for Q_i corresponds to the minimum-weight (s_i, t_i)-path in $\hat{H}_{i-1}$. Hence, we need to analyze the cost of these greedy choices.

Awerbuch et al. [4] showed that the greedy algorithm is indeed $O(\log^2 n)$-competitive for the edge-weighted Steiner forest problem. The standard greedy algorithm is slightly different from the greedy process we discussed above. In the greedy algorithm of Awerbuch et al., at time step i we choose a minimum-cost (s_i, t_i)-path in a graph in which there is a zero-cost edge between *any* pair of vertices in the same connected component of the current solution; not just the (s_j, t_j) pairs of the previous demands. However, in their analysis they only use the zero-cost edges among the terminals of a previous demand. This is indeed not surprising since we hardly have any control on the greedy choices other than the fact that they satisfy the demands. Therefore the following claim follows from the result of Awebuch et al.[2]:

▶ **Claim 7** (implicitly proven in Theorem 2.1 of [4]). *For any weight function $\hat{w}$ defined over F, there exists a connective list Q for which*

$$\sum_i \hat{w}(Q_i) \le O(\log^2 n)\hat{w}(F).$$

However, Claim 7 is not enough for us. We need a solution in which every edge is used at most $O(\log^2 n)$ times, not just in an amortized sense. Indeed we can show that since

[2] There is also a lower bound of $\Omega(\log n)$ for the competitive ratio of the greedy algorithm. Closing the gap between this lower bound and the upper bound of $O(\log^2 n)$ for EW Steiner forest is an important open problem.

there is a solution for *every* weight function, we can have a *fractional* connective list Q in which every edge is used (fractionally) at most $O(\log^2 n)$ times. This implies that we have a fractional connective list. Finally, we provide a rounding argument which obtains an integral connective list by losing only a constant factor; which completes the proof of lemma. ◀

Finally, we can leverage Lemma 6 to show $\mathsf{popt} \leq O(\log^2 n)\mathsf{lopt}$. This shows we can use $\mathbb{PC\_IP}$ as an online mixed packing/covering IP to obtain an online solution for ONLINE EDGE-WEIGHTED DEGREE-BOUNDED STEINER FOREST losing a factor of $O(\log^2 n)$. In Section 4 we show this formulation is an online bounded frequency mixed packing/covering IP, thus we leverage our technique for such IPs to obtain a polylogarithmic-competitive algorithm for online EW-DB-SF.

3 Online Bounded Frequency Mixed Packing/Covering IPs

In this section we consider bounded frequency online mixed packing and covering integer programs. For every online mixed packing and covering IP with covering frequency k, we provide an online algorithm that violates each packing constraint by at most a factor of $O(k \log m)$, where m is the number of packing constraints. We note that this bound is independent of the number of variables, the number of covering constraints, and the coefficients of the mixed packing and covering program. Moreover the algorithm is for integer programs, which implies obtaining an integer solution does not rely on (online) rounding.

In particular we prove there exists an online $O(k \log m)$-competitive algorithm for any mixed packing and covering IP such that every variable has covering frequency at most k, where the covering frequency of a variable x_r is the number of covering constraints with a non-zero coefficient for x_r.

We assume that all variables are binary. One can see this is without loss of generality as long as we know every variable $x_r \in \{1, 2, 3, \ldots, 2^l\}$. Since we can replace x_r by l variables $y_r^1, \ldots, y_r^l$ denoting the digits of x_r and adjust coefficients accordingly. Furthermore, for now we assume that the optimal solution for the given mixed packing and covering program is 1. In Theorem 10 we prove that we can use a doubling technique to provide an $O(k \log m)$-competitive solution for online bounded frequency mixed packing and covering programs with any optimal solution. The algorithm is as follows. We maintain a family of subsets $\mathcal{S}$. Initially $\mathcal{S} = \emptyset$. Let $\mathcal{S}(j)$ denote $\mathcal{S}$ at arrival of C_{j+1}. For each covering constraint C_{j+1}, we find a subset of variables S_{j+1} and add S_{j+1} to $\mathcal{S}$. We find S_{j+1} in the following way. For each set of variables S, we define a cost function $\tau_S(\mathcal{S}(j))$ according to our current $\mathcal{S}$ at arrival of C_{j+1}. We find a set S_{j+1} that satisfies C_{j+1} and minimizes $\tau_S(\mathcal{S}(j))$. More precisely we say a set of variables S satisfies C_{j+1} if

- $\sum_{x_r \in S} C_{j+1,r} x_r \geq 1$, where $C_{j+1,r}$ denotes the coefficient of C_{j+1} for x_r.
- For each packing constraint P_i, $\sum_{x_r \in S} \frac{1}{k} P_{ir} \leq 1$.

Now we add S_{j+1} to $\mathcal{S}$ and for every $x_r \in S_{j+1}$, we set $x_r = 1$. We note that there always exists a set S that satisfies C_{j+1}, since we assume there exists an optimal solution with value 1. Setting S to be the set of all variables with value one in an optimal solution which have non-zero coefficient in C_{j+1}, satisfies C_{j+1}. It only remains to define $\tau_S(\mathcal{S}(j))$. But before that we need to define $\Delta_i(S)$ and $F_i(\mathcal{S}(j))$. For packing constraint P_i and subset of variables S, we define $\Delta_i(S)$ as $\Delta_i(S) := \sum_{x_r \in S} \frac{1}{k} P_{ir}$. For packing constraint P_i and $\mathcal{S}(j)$, let

$$F_i(\mathcal{S}(j)) := \sum_{S \in \mathcal{S}(j)} \Delta_i(S) \ . \tag{7}$$

Now let $\tau_S(\mathcal{S}(j)) = \sum_{i=1}^m \rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))}$, where $\rho > 1$ is a constant to be defined later.

Algorithm 1

Input: Packing constraints P, and an online stream of covering constraints $C_1, C_2, \ldots$.
Output: A feasible solution for online bounded frequency mixed packing/covering.
Offline Process:

1: Initialize $\mathcal{S} \leftarrow \emptyset$.

Online Scheme; assuming a covering constraint C_{j+1} is arrived:

1: $S_{j+1} \leftarrow \arg\min_S\{\tau_S(\mathcal{S}(j)) \mid S \text{ satisfies } C_{j+1}\}$.
2: **for all** $x_r \in S_{j+1}$ **do**
3: $\quad x_r \leftarrow 1$.

Let $\mathbf{x}^*$ be an optimal solution, and $\mathbf{x}^*(j)$ denote its values at online stage j. We define $G_i(j)$ as

$$G_i(j) := \sum_{l=1}^{j} \sum_{r:C_{lr}>0} \frac{1}{k} x_r^* P_{ir} \quad . \tag{8}$$

Now we define a potential function Φ_j for online stage j.

$$\Phi_j = \sum_{i=1}^{m} \rho^{F_i(\mathcal{S}(j))}(\gamma - G_i(j)) \quad , \tag{9}$$

where $\rho, \gamma > 1$ are constants to be defined later.

▶ **Lemma 8.** *There exist constants ρ and γ, such that Φ_j is non-increasing.*

Proof. We find ρ and γ such that $\Phi_{j+1} - \Phi_j \leq 0$. By the definition of Φ_j,

$$\Phi_{j+1} - \Phi_j = \sum_{i=1}^{m} \rho^{F_i(\mathcal{S}(j+1))}(\gamma - G_i(j+1)) - \rho^{F_i(\mathcal{S}(j))}(\gamma - G_i(j)) \quad . \tag{10}$$

By Equation (7), $\rho^{F_i(\mathcal{S}(j+1))} - \rho^{F_i(\mathcal{S}(j))} = \rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))}$. Moreover by Equation (8), $(\gamma - G_i(j+1)) - (\gamma - G_i(j)) = -\sum_{r:C_{j+1,r}>0} \frac{1}{k} x_r^* P_{ir}$. For simplicity of notation we define $B_i(j+1) := \sum_{r:C_{j+1,r}>0} \frac{1}{k} x_r^* P_{ir}$. Thus we can write Equation (10) as:

$$\Phi_{j+1} - \Phi_j = \sum_{i=1}^{m} \rho^{F_i(\mathcal{S}(j+1))}(\gamma - G_i(j) - B_i(j+1)) - \rho^{F_i(\mathcal{S}(j))}(\gamma - G_i(j)) \tag{11}$$

$$= \sum_{i=1}^{m} (\gamma - G_i(j))(\rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))}) - \rho^{F_i(\mathcal{S}(j+1))} B_i(j+1) \qquad \text{Since } G_i(j) \geq 0$$

$$\leq \sum_{i=1}^{m} \gamma(\rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))}) - \rho^{F_i(\mathcal{S}(j+1))} B_i(j+1) \qquad F_i(\mathcal{S}(j+1)) \geq F_i(\mathcal{S}(j))$$

$$\leq \sum_{i=1}^{m} \gamma(\rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))}) - \rho^{F_i(\mathcal{S}(j))} B_i(j+1) \quad .$$

Now according to the algorithm for each subset of variables S' such that $\sum_{x_r \in S'} C_{j+1}(x_r) \geq 1$, either $\tau_S(\mathcal{S}(j)) \leq \tau_{S'}(\mathcal{S}(j))$ or there exists a packing constraint P_i such that $\Delta_i(S') > 1$. In

$B_i(j+1)$, we are considering variables x_r such that $x_e^* = 1$, thus for every P_i, $B_i(j+1) \leq 1$. Therefore setting S' to be the set of variables x_r such that $x_r^* = 1$ and $C_{j+1,r} > 0$, we have $\tau_S(\mathcal{S}(j)) \leq \tau_{S'}(\mathcal{S}(j))$. Thus $\sum_{i=1}^m \rho^{F_i(\mathcal{S}(j))+\Delta_i(S)} - \rho^{F_i(\mathcal{S}(j))} \leq \sum_{i=1}^m \rho^{F_i(\mathcal{S}(j))+B_i(j+1)} - \rho^{F_i(\mathcal{S}(j))}$. Therefore we can rewrite Inequality (11) as

$$\begin{aligned}\Phi_{j+1} - \Phi_j &\leq \sum_{i=1}^m \gamma(\rho^{F_i(\mathcal{S}(j))+B_i(j+1)} - \rho^{F_i(\mathcal{S}(j))}) - \rho^{F_i(\mathcal{S}(j))} B_i(j+1) \qquad (12)\\ &= \sum_{i=1}^m \rho^{F_i(\mathcal{S}(j))}(\gamma\rho^{B_i(j+1)} - \gamma - B_i(j+1)) \ .\end{aligned}$$

We would like to find ρ and γ such that Φ_j is non-increasing. We find ρ and γ such that for each packing constraint P_i, $\gamma\rho^{B_i(j+1)} - \gamma - B_i(j+1) \leq 0$. Thus

$$\begin{aligned}\gamma\rho^{B_i(j+1)} - \gamma &\leq B_i(j+1) && \text{Since } 0 \leq B_i(j+1) \leq 1 && (13)\\ \gamma\rho B_i(j+1) - \gamma &\leq B_i(j+1) && \text{By simplifying} && (14)\\ \rho &\leq 1 + 1/\gamma \ . && && (15)\end{aligned}$$

Thus if we set $\rho \leq 1 + 1/\gamma$, Φ_j is non-increasing, as desired. ◀

Now we prove Algorithm 1 obtains a solution of at most $O(k \log m)$.

▶ **Lemma 9.** *Given an online bounded frequency mixed packing covering IP with optimal value 1, there exists a deterministic integral algorithm with competitive ratio $O(k \log m)$, where m is the number of packing constraints and k is the covering frequency of the IP.*

Proof. By Lemma 8 for each stage j, $\Phi_{j+1} \leq \Phi_j$. Therefore $\Phi_j \leq \Phi_0 = \gamma m$. Thus for each packing constraint P_i,

$$\rho^{F_i(\mathcal{S}(j))}(\gamma - G_i(j)) \leq \gamma m \ . \qquad (16)$$

Thus,

$$\rho^{F_i(\mathcal{S}(j))} \leq \frac{\gamma m}{(\gamma - G_i(j))} \leq \frac{\gamma m}{\gamma - 1} \ . \qquad \text{Since } G_i(j) \leq 1 \qquad (17)$$

Thus we can conclude

$$F_i(\mathcal{S}(j)) \in O(\log m) \ . \qquad (18)$$

By definition of $F_i(\mathcal{S}(j))$, $F_i(\mathcal{S}(j)) = \sum_{S \in \mathcal{S}(j)} \Delta_i(S) = \sum_{S \in \mathcal{S}(j)} \sum_{x_r \in S} \frac{1}{k} P_{ir}$. Since each variable x_r is present in at most k sets, $\frac{1}{k} P_i \cdot \mathbf{x}(j) \leq F_i(\mathcal{S}(j))$. Thus by Inequality (18) $P_i \mathbf{x}(j) \in O(k \log m)$, which completes the proof. ◀

Finally we prove there exists an online $O(k \log m)$-competitive algorithm for bounded frequency online mixed packing and covering integer programs with any optimal value.

▶ **Theorem 10.** *Given an instance of the online mixed packing/covering IP, there exists a deterministic integral algorithm with competitive ratio $O(k \log m)$, where m is the number of packing constraints and k is the covering frequency of the IP.*

4 Putting Everything Together

In this section we consider the online mixed packing/covering formulation discussed in Section 2 for ONLINE EDGE-WEIGHTED DEGREE-BOUNDED STEINER FOREST $\mathbb{PC\_IP}$. In this section we show this formulation is an online bounded frequency mixed packing/covering IP. Therefore we our techniques discussed in Section 3 to obtain a polylogarithmic-competitive algorithm for ONLINE EDGE-WEIGHTED DEGREE-BOUNDED STEINER FOREST.

First we assume we are given the optimal weight w_{opt} as well as degree bounds. We can obtain the following theorem.

▶ **Theorem 11.** *Given the optimal weight w_{opt}, there exists an online deterministic algorithm which finds a subgraph with total weight at most $O(\log^3 n)w_{opt}$ while the degree bound of a vertex is violated by at most a factor of $O(\log^3 n)$.*

Proof. By Lemma 4, given a feasible online solution for $\mathbb{PC\_IP}$ with violation α, we can provide an online solution for $\mathbb{SF\_IP}$ with violation α. Moreover, in Section 2 we show that $\mathsf{popt} \leq O(\log^2 n)\mathsf{lopt}$. Thus given an online solution for $\mathbb{PC\_IP}$ with competitive ratio f, there exists an $O(f \log n)$-competitive algorithm for ONLINE DEGREE-BOUNDED STEINER FOREST. We note that in $\mathbb{PC\_IP}$ we know the packing constraints in advance. In addition every variable x^i_H has non-zero coefficient only in the covering constraint corresponding to connectivity of the i-th demand endpoints, i.e. the covering frequency of every variable is 1. Therefore by Theorem 10 there exists an online $O(\log m)$-competitive solution for $\mathbb{PC\_IP}$, where m is the number of packing constraints, which is $n + 1$. Thus there exists an online $O(\log^3 n)$-competitive algorithm for ONLINE DEGREE-BOUNDED STEINER FOREST. This means the violation for both degree bounds and weight is of $O(\log^3 n)$. ◀

Finally if w_{opt} is not given, we show in the full version of the paper that by applying standard doubling techniques one can prove Theorem 1 using the result shown above.

References

1 Ajit Agrawal, Philip Nathan Klein, and R Ravi. How tough is the minimum-degree steiner tree?: A new approximate min-max equality. *Technical Report CS-91-49, Brown University*, 1991.

2 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.

3 James Aspnes, Yossi Azar, Amos Fiat, Serge Plotkin, and Orli Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM (JACM)*, 44(3):486–504, 1997.

4 Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized steiner problem. In *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 68–74, 1996.

5 Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalya Panigrahi. Online mixed packing and covering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 85–100. SIAM, 2013.

6 Fred Bauer and Anujan Varma. Degree-constrained multicasting in point-to-point networks. In *INFOCOM'95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, pages 369–376. IEEE, 1995.

7 Piotr Berman and Chris Coulston. On-line algorithms for steiner tree problems. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 344–353, 1997.

8 Niv Buchbinder and Joseph 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.

9 Deeparnab Chakrabarty, Alina Ene, Ravishankar Krishnaswamy, and Debmalya Panigrahi. Online buy-at-bulk network design. In *FOCS*, 2015.

10 Kamalika Chaudhuri, Satish Rao, Samantha Riesenfeld, and Kunal Talwar. A push-relabel algorithm for approximating degree bounded msts. In *Proceedings of the 33rd international conference on Automata, Languages and Programming-Volume Part I*, pages 191–201. Springer-Verlag, 2006.

11 Kamalika Chaudhuri, Satish Rao, Samantha Riesenfeld, and Kunal Talwar. What would edmonds do? augmenting paths and witnesses for degree-bounded msts. *Algorithmica*, 55(1):157–189, 2009.

12 Sina Dehghani, Soheil Ehsani, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online degree-bounded steiner network design. In *SODA*, 2016. URL: `http://web.stanford.edu/~vliaghat/uploads/notes.pdf`.

13 Alina Ene and Ali Vakilian. Improved approximation algorithms for degree-bounded network design problems with node connectivity requirements. *STOC*, 2014.

14 Takuro Fukunaga and R Ravi. Iterative rounding approximation algorithms for degree-bounded node-connectivity network design. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 263–272. IEEE, 2012.

15 Martin Fürer and Balaji Raghavachari. An NC approximation algorithm for the minimum degree spanning tree problem. In *Allerton Conf. on Communication, Control and Computing*, pages 274–281, 1990.

16 Martin Fürer and Balaji Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *Journal of Algorithms*, 17(3):409–423, 1994.

17 Michel X Goemans. Minimum bounded degree spanning trees. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 273–282. IEEE, 2006.

18 Michel X Goemans. Minimum bounded degree spanning trees. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 273–282, 2006.

19 Mohammad Taghi Hajiaghayi, Vahid Liaghat, and Debmalya Panigrahi. Online node-weighted steiner forest and extensions via disk paintings. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 558–567, 2013.

20 MohammadTaghi Hajiaghayi, Vahid Liaghat, and Debmalya Panigrahi. Near-optimal online algorithms for prize-collecting steiner problems. In *Automata, Languages, and Programming*, pages 576–587. Springer, 2014.

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

22 Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. On some network design problems with degree constraints. *Journal of Computer and System Sciences*, 79(5):725–736, 2013.

23 Philip N Klein, Radha Krishnan, Balaji Raghavachari, and R Ravi. Approximation algorithms for finding low-degree subgraphs. *Networks*, 44(3):203–215, 2004.

24 Jochen Könemann and R Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 537–546. ACM, 2000.

25 Jochen Könemann and R Ravi. Primal-dual meets local search: approximating mst's with nonuniform degree bounds. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 389–395. ACM, 2003.

26 Lap Chi Lau, Joseph Naor, Mohammad R Salavatipour, and Mohit Singh. Survivable network design with degree or order constraints. *SIAM Journal on Computing*, 39(3):1062–1087, 2009.

27 Lap Chi Lau and Mohit Singh. Additive approximation for bounded degree survivable network design. *SIAM Journal on Computing*, 42(6):2217–2242, 2013.

28 Madhav V Marathe, R Ravi, Ravi Sundaram, SS Ravi, Daniel J Rosenkrantz, and Harry B Hunt III. Bicriteria network design problems. *Journal of Algorithms*, 28(1):142–171, 1998.

29 R Garey Michael and S Johnson David. Computers and intractability: a guide to the theory of np-completeness. *WH Freeman & Co., San Francisco*, 1979.

30 Joseph Naor, Debmalya Panigrahi, and Mohit Singh. Online node-weighted steiner tree and related problems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 210–219, 2011.

31 Zeev Nutov. Degree-constrained node-connectivity. In *LATIN 2012: Theoretical Informatics*, pages 582–593. Springer, 2012.

32 Carlos AS Oliveira and Panos M Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers & Operations Research*, 32(8):1953–1981, 2005.

33 Jiawei Qian and David P Williamson. An o (logn)-competitive algorithm for online constrained forest problems. In *Automata, Languages and Programming*, pages 37–48. Springer, 2011.

34 Balaji Raghavachari. Algorithms for finding low degree structures. In *Approximation algorithms for NP-hard problems*, pages 266–295. PWS Publishing Co., 1996.

35 R Ravi, Madhav V Marathe, SS Ravi, Daniel J Rosenkrantz, and Harry B Hunt III. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica*, 31(1):58–78, 2001.

36 R Ravi and Mohit Singh. Delegate and conquer: An lp-based approximation algorithm for minimum degree msts. In *Automata, Languages and Programming*, pages 169–180. Springer, 2006.

37 Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 661–670, 2007.

38 Stefan Voß. Problems with generalized steiner problems. *Algorithmica*, 7(1):333–335, 1992.