On some network design problems with degree constraints[☆]

Rohit Khandekar[1]

IBM T.J.Watson Research Center

Guy Kortsarz[1]

Rutgers University, Camden. Partially supported by NSF grant number 434923

Zeev Nutov[1]

The Open University of Israel

Abstract

We study several network design problems with degree constraints. For the minimum-cost Degree-Constrained 2-Node-Connected Subgraph problem, we obtain the first non-trivial bicriteria approximation algorithm, with $5b(v)+3$ violation for the degrees and a 4-approximation for the cost. This improves upon the logarithmic degree violation and no cost guarantee obtained by Feder, Motwani, and Zhu (2006). Then we consider the problem of finding an arborescence with at least k terminals and with minimum maximum outdegree. We show that the natural LP-relaxation has a gap of $\Omega(\sqrt{k})$ or $\Omega(n^{1/4})$ with respect to the multiplicative degree bound violation. We overcome this hurdle by a combinatorial $O(\sqrt{(k \log k)/\Delta^*})$-approximation algorithm, where Δ^* denotes the maximum degree in the optimum solution. We also give an $\Omega(\log n)$ lower bound on approximating this problem. Then we consider the undirected version of this problem, however, with an extra diameter constraint, and give an $\Omega(\log n)$ lower bound on the approximability of this version. We also consider a closely related Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network problem, and obtain several

[☆]A preliminary version of this paper is [18].

Email addresses: `rohitk@us.ibm.com` (Rohit Khandekar), `guyk@camden.rutgers.edu` (Guy Kortsarz), `nutov@openu.ac.il` (Zeev Nutov)

results in this direction by reducing the prize-collecting variant to the regular one. Finally, we consider the Terminal Steiner Tree problem, which is a simple variant of the Degree-Constrained Steiner Tree problem, when some terminals are required to be leaves. We show that this seemingly simple problem is equivalent to the Group Steiner Tree problem.

Keywords: Network design; Degree-constraints; Approximation algorithms

---

1. Introduction

1.1. Problems considered

In network design problems one seeks a cheap subgraph H of a given graph G that satisfies some given properties. In the b-Matching problem H should satisfy prescribed degree constraints, while in the Survivable Network problem H should satisfy prescribed connectivity requirements. The Degree-Constrained Survivable Network problems is a combination of these two fundamental problems, where H should satisfy both degree constraints and connectivity requirements. For most of these problems, even checking whether there exists a feasible solution is NP-hard, hence one considers a bicriteria approximation when the degree constraints are relaxed. Namely, the goal is to compute a cheap solution that satisfies the connectivity requirements and has small degree violation.

Many recent papers considered *edge-connectivity* Degree-Constrained Survivable Network problems, see a recent survey in [23]. Our first problem is the simplest *node-connectivity* problem. A graph H is k-(node-)connected if it contains k internally disjoint paths between every pair of its nodes. In the k-Connected Subgraph problem we are given a graph $G = (V, E)$ with edge-costs and an integer k. The goal is to find a minimum-cost k-connected spanning subgraph H of G. In the Degree-Constrained k-Connected Subgraph problem, we are also given degree bounds $\{b(v) : v \in B \subseteq V\}$. The goal is to find a minimum-cost k-connected spanning subgraph H of G such that in H, the degree of every node $v \in B$ is at most $b(v)$. We consider the case $k = 2$.

Degree-Constrained 2-Connected Subgraph
Instance: An undirected graph $G = (V, E)$ with non-negative edge-costs $\{c_e : e \in E\}$, and degree bounds $\{b(v) : v \in B \subseteq V\}$. *Objective:* Find a minimum cost 2-connected spanning subgraph H of G that satisfies the *degree constraints* $\deg_H(v) \leq b(v)$ for all $v \in B$.

In the Steiner k-Tree problem one seeks a minimum-cost tree that contains at least k-terminals (when every node is a terminal we get the k-MST problem). Our next problem is the minimum-degree directed version of this problem. Given a directed graph G, a set S of terminals, and an integer $k \leq |S|$, a *k-arborescence* is an arborescence in G that contains k terminals; in the case of undirected graphs we have a *k-tree*. For a directed/undirected graph or edge-set H let $\Delta(H)$ denote the maximum outdegree/degree of a node in H.

Minimum Degree k-Arborescence
Instance: A directed graph $G = (V, E)$, a root $s \in V$, a subset $S \subseteq V \setminus \{s\}$ of terminals, and an integer $k \leq |S|$.
Objective: Find in G a k-arborescence T rooted at s that minimizes $\Delta(T)$.

The origin of this problem is in peer-to-peer networking, when one wants to bound the maximum load (degree) of a node, while connecting the root to the maximum number of terminals. It is also of interest to bound the height of such a tree, to limit the time for sending messages from the root. This motivates our next problem, for which we only show a lower bound. Hence we show it for the *less* general case of undirected graphs.

Degree and Diameter Bounded k-Tree
Instance: An undirected graph $G = (V, E)$, a subset $S \subseteq V$ of terminals, an integer $k \leq |S|$, and a diameter bound D.
Objective: Find a k-tree T with diameter bounded by D that minimizes $\Delta(T)$.

Let $\lambda_H(u, v)$ denote the the maximum number of edge-disjoint uv-paths in H. In the Edge-Connectivity Survivable Network problem we are given a graph $G = (V, E)$ with edge-costs, a collection $\mathcal{P} = \{\{u_1, v_1\}, \ldots, \{u_k, v_k\}\}$ of node pairs, and connectivity requirements $\mathcal{R} = \{r_1, \ldots, r_k\}$. The goal is to find a minimum-cost subgraph H of G that satisfies the connectivity requirements $\lambda_H(u_i, v_i) \geq r_i$ for all i.

We consider a combination of the following two generalizations of this problem. In Degree-Constrained Edge-Connectivity Survivable Network, we are given degree bounds $\{b(v) : v \in B\}$. The goal is to find a minimum-cost subgraph H of G that satisfies the connectivity requirements and the degree constraints $\deg_H(v) \leq b(v)$ for all $v \in B$. In the Prize-Collecting Edge-Connectivity Survivable Network we are given a submodular monotone non-decreasing penalty function $\pi : 2^{\{1,\ldots,k\}} \to \mathbb{R}_+$ (π is given by an evaluation oracle). The goal is to find a subgraph H of G that minimizes the *value* $\texttt{val}(H) = c(H) + \pi(\texttt{unsat}(H))$ of H, where $\texttt{unsat}(H) = \{i \mid \lambda_H^S(u_i, v_i) < r_i\}$ is the set of requirements *not* (completely) satisfied by H. Formally, the problem we consider is as follows.

Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network
Instance: An undirected graph $G = (V, E)$ with non-negative edge-costs $\{c_e : e \in E\}$, a collection $\mathcal{P} = \{\{u_1, v_1\}, \ldots, \{u_k, v_k\}\}$ of node pairs, connectivity requirements $\mathcal{R} = \{r_1, \ldots, r_k\}$, a submodular monotone non-decreasing penalty function $\pi : 2^{\{1,\ldots,k\}} \to \mathbb{R}_+$ given by an evaluation oracle, and degree bounds $\{b(v) : v \in B \subseteq V\}$.   *Objective:* Find a subgraph H of G that satisfies the *degree constraints* $\deg_H(v) \leq b(v)$ for all $v \in B$, and minimizes the *value* $$\texttt{val}(H) = c(H) + \pi(\texttt{unsat}(H))$$ of H, where $\texttt{unsat}(H) = \{i \mid \lambda_H^S(u_i, v_i) < r_i\}$ is the set of requirements *not* satisfied by H.

The Steiner Tree problem is a particular case of this problem, when we seek a minimum-cost subtree T of G that contains a specified subset S of terminals. In the degree constrained version of Steiner Tree, we are also given degree bounds on nodes in S and need to satisfy the degree constraints. We consider the case of $\{0, 1\}$-constraints, namely, we require that certain nodes in S should be leaves of T, and do not allow to relax this condition, as was done in previous papers [22, 24, 26, 2]. Namely, the degree bounds here are of the "hard capacity" type, and cannot be violated. Formally, our problem can be casted as follows.

Terminal Steiner Tree
Instance: An undirected graph $G = (V, E)$ with non-negative edge-costs $\{c_e : e \in E\}$ and node subsets $L \subseteq S \subseteq V$.   *Objective:* Find a minimum-cost tree T in G that contains S such that every $v \in L$ is a leaf of T.

1.2. Previous and related work

Fürer and Raghavachari [10] were the first to consider degree-constrained connectivity problems. They gave a "plus 1" approximation for the Minimum Degree Steiner Tree problem. Namely, if the lowest maximum degree possible is Δ^*, their algorithm returns a Steiner tree with maximum degree $\Delta^* + 1$. This result is the best possible, as computing an optimal solution is NP-hard even in Minimum Degree Spanning Tree case.

The first result for the min-cost case is due to Ravi et al. [27]; they obtained an $(O(\log n) \cdot b(v), O(\log n))$-approximation for Degree-Constrained MST, namely, the degree of every node v in the output tree is $O(\log n) \cdot b(v)$ while its cost is $O(\log n)$ times the optimal cost. A major breakthrough was obtained by Goemans [12]; his algorithm computes a minimum cost spanning tree with degree at most $\Delta + 2$, with Δ the minimum possible degree.

In [19] and [32] is given an $O(n^\delta)$-approximation algorithm for the Minimum Degree k-Edge-Connected Subgraph problem, for any fixed $\delta > 0$.

It turned out that an extension of the iterative rounding method of Jain [17] may be the leading technique for degree-constrained problems. Singh and Lau [34] were the first to extend this method to achieve the best possible result for Min-Cost Minimum Degree MST; their tree has optimal cost while the maximum degree is at most $\Delta + 1$. Lau et al. [22] obtained a $(2b(v) + 3, 2)$-approximation for the *edge-connectivity* Degree-Constrained Survivable Network problem, which was recently improved to $(2b(v) + 2, 2)$ in [26]. Lau and Singh [24] further obtained a $(b(v) + O(r_{\max}), 2)$-approximation, where $r_{\max}$ denotes the maximum connectivity requirement.

For directed graphs, Bansal et al. [2] gave an $(\lceil \frac{b(v)}{1-\epsilon} \rceil + 4, \frac{1}{\epsilon})$-approximation scheme for the Degree-Constrained k-Edge-Outconnected Subgraph problem; the case $k = 1$ is the Degree-Constrained Arborescence problem, for which [2] gave a $b(v) + 2$-approximation, without bounding the cost. Some extensions and slight improvements can be found in [29].

Note that all the above results are for *edge-connectivity* Survivable Network problems. The only known result for *node-connectivity* degree-constrained problems is by Feder, Motwani, and Zhu [9] who gave an algorithm that computes in $n^{O(k)}$ time a k-connected spanning subgraph H of G such that $\deg_H(v) = O(\log n) \cdot b(v)$. Their algorithm cannot handle costs.

The special case $k = |S|$ of the Minimum Degree k-Arborescence problem was already studied in [8], where a $\tilde{O}(\sqrt{k})$ *additive* approximation was given. Their technique does not seem to extend to the case $k < |S|$. Even for the easier undirected case, if we ask for a tree containing k nodes and want to minimize the maximum degree (this is the Degree Bounded k-MST problem), the above techniques of [8] seem to fail.

Hajiaghayi and Nasri [14] obtained a constant ratio for a very special case of Degree-Constrained Prize-Collecting Edge-Connectivity Survivable Network problem when the penalty function π is modular.

A particular case of the Terminal Steiner Tree problem, when the costs are metric and $S = L$, admits a constant ratio algorithm [25, 5, 7].

1.3. Our results and techniques

Recall that for the Degree-Constrained k-Connected subgraph problem, Feder, Motwani, and Zhu [9] gave an algorithm that computes in $n^{O(k)}$ time a k-connected spanning subgraph H of G such that $\deg_H(v) = O(\log n) \cdot b(v)$, and that their algorithm cannot handle costs. Our first result significantly improves their result for $k = 2$, from logarithmic factor degree violation to constant factor violation. Furthermore, we are also able to bound the cost.

Theorem 1.1. *The* Degree-Constrained 2-Connected Subgraph *problem admits a bicriteria* $(5b(v) + 3, 4)$*-approximation algorithm; namely, a polynomial time algorithm that computes a* 2*-connected spanning subgraph* H *of* G *in which the degree of every node* v *is at most* $5b(v) + 3$*, and the cost of* H *is at most* 4 *times the optimal cost.*

To prove Theorem 1.1 we first compute a degree-constrained spanning tree J with $+1$ degree violation using the algorithm of [34]. Then we compute an augmenting edge-set I such that $J \cup I$ is 2-connected, using the iterative rounding method. To apply this method for degree constrained problems, one proves that any basic LP-solution $x > 0$ has an edge e with hight x_e value, or there exists a node $v \in B$ such that $\deg_E(v)$ is close to $b(v)$. Otherwise, one shows a contradiction using the so called "token assignment argument". Here one shows that there exists a laminar family $\mathcal{L}$ of "violated sets" and a set T of nodes, such that x is the unique solution to the equation system defined by cut-constraints of sets in $\mathcal{L}$ and degree constraints of nodes in T. The contradiction is obtained by showing that the number of entries in x is strictly larger than $|\mathcal{L}| + |T|$. All previous "token assignment arguments" associated every node with a *unique* set in the laminar family $\mathcal{L}$. However, even in the simplest node-connectivity problem of augmenting a tree to be 2-connected, this is not possible, as the cut-nodes of the tree are associated with many sets in $\mathcal{L}$. We will allow for a node to be "shared" by many members of $\mathcal{L}$, and still will be able to distribute the tokens to obtain the desired contradiction.

Our second result gives the first approximation algorithm for the Minimum Degree k-Arborescence problem.

Theorem 1.2. *The* Minimum Degree k-Arborescence *problem admits an approximation algorithm with ratio $O(\sqrt{(k \log k)/\Delta^*})$, where Δ^* is the optimal solution value, namely, the minimal maximum outdegree possible. Furthermore, the problem admits no $o(\log n)$-approximation, unless* NP=Quasi(P).

Our algorithm for the Minimum Degree k-Arborescence problem uses a new method, which might be useful for related problems. We show that any k-arborescence with maximum degree Δ^* admits a "balanced partition" into roughly $\sqrt{k} \cdot \sqrt{\Delta^*}$ edge-disjoint arborescence, each containing at most $\sqrt{k} \cdot \sqrt{\Delta^*}$ terminals. We find iteratively, via max-flow computations, trees that contain $\sqrt{k} \cdot \Delta^*$ terminals. This will create many separate trees, that should be connected to the root. Thus, we have to show that there will be not too many such trees. We prove this by using the fact that the flow computation problem can be casted as a submodular covering problem, and thus admits a $O(\log n)$-approximation [35].

Integrality gap of the natural LP relaxation for Minimum Degree k-MST*.* To get some indication that the problem might be hard even on undirected graphs, consider the following natural LP-relaxation for Minimum Degree k-MST. The intended integral solution has $y_v = 1$ for nodes picked in the optimum tree T^*, $x_e = 1$ for $e \in T^*$, and d equal to the maximum degree of T^*.

$$\begin{array}{llcll}
\text{Minimize} & d & & & \\
\text{Subject to} & \sum_{v \neq r} y_v & \geq & k & \\
& \sum_{e \in \delta(S)} x_e & \geq & y_v & \forall v \in V \setminus \{r\} \ \ \forall S \subset V, r \in S, v \notin S \\
& \sum_{e \in \delta(v)} x_e & \leq & d & \forall v \in V \\
& x_e, y_v & \in & [0,1] & \forall e \in E \ \ \forall v \in V
\end{array} \tag{1}$$

We show that this LP-relaxation has integrality gap $\Omega(\sqrt{k})$ or $\Omega\left(n^{1/4}\right)$ where $n = |V|$. This holds even for the undirected case. Consider a rooted at r complete Δ-ary tree T of height h and let $k = \lfloor(\Delta+\Delta^2+\cdots+\Delta^h)/(\Delta+1)\rfloor$. It is easy to see that giving $x_e = 1/(\Delta+1)$ to all the edges $e \in T$ and $y_v = 1/(\Delta+1)$ to all nodes $v \neq r$ satisfies all the constraints with fractional objective value $d = 1$. In order to cover k nodes, any integral tree however has to have a maximum degree of at least δ where $\delta + \delta(\delta-1) + \delta(\delta-1)^2 + \cdots + \delta(\delta-1)^{h-1} \geq k$. Such δ satisfies $\delta = \Omega(k^{1/h})$. Thus the optimum integral tree must have maximum degree $\Omega\left(k^{1/h}\right)$ giving an integrality gap of $\Omega\left(k^{1/h}\right)$. If we let $h = 2$, we get that $k = \Delta$ and $n = 1 + \Delta + \Delta^2$ and the integrality gap is $\Omega(\sqrt{\Delta})$ which is $\Omega(\sqrt{k})$ or $\Omega(n^{1/4})$.

Proposition 1.3. *The* Degree and Diameter Bounded k-Tree *problem admits no $o(\log n)$-approximation algorithm, unless* NP=Quasi(P). *In the undirected case this holds true only if we add a diameter bound of 4 on the k-tree.*

Let $\delta_F(S)$ denote the set of edges in F going from S to $V \setminus S$. For $i \in K$ let $S \odot i$ denote that $|S \cap \{u_i, v_i\}| = 1$. Menger's Theorem for edge-connectivity (see [20]) states that for a node pair u_i, v_i of a graph $H = (V, F)$ we have $\lambda_H(u_i, v_i) = \min_{S \odot i} |\delta_F(S)|$. Hence if $\lambda_H(u_i, v_i) \geq r_i$ for a graph $H = (V, F)$, then for any S with $S \odot i$ we must have $|\delta_F(S)| \geq r_i$. A standard "cut-type" LP-relaxation for Degree-Constrained Edge-Connectivity Survivable Network problem is as follows.

$$
\begin{array}{llll}
\text{Minimize} & \sum_{e \in E} c_e x_e & & \\
\text{Subject to} & \sum_{e \in \delta_E(S)} x_e \geq r_i(S) & \forall i \in K, S \subseteq V, S \odot i & \\
& \sum_{e \in \delta_E(v)} x_e \leq b(v) & \forall v \in B & (2) \\
& x_e \in [0,1] & \forall e \in E &
\end{array}
$$

Theorem 1.4. *Suppose that for a* Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network *instance the following holds. For any $\mathcal{P}' \subseteq \mathcal{P}$, the* Degree-Constrained Edge-Connectivity Survivable Network *instance defined by $\mathcal{P}'$ admits a polynomial-time algorithm that computes a solution H' of cost at most ρ times the optimal value of LP (2) such that $\deg_{H'}(v) \leq \alpha b(v) + \beta$ for all $v \in B$. Then the* Prize-Collecting Degree-Constrained Edge-Connectivity

Survivable Network *instance admits a polynomial time algorithm that for any* $\mu \in (0,1)$ *computes a subgraph* H *of* G *such that* $\mathtt{val}(H) \leq \frac{\rho}{1-\mu}c^* + \frac{1}{\mu}\pi^*$ *and* $\deg_H(v) \leq \frac{\alpha}{1-\mu}b(v) + \beta$ *for all* $v \in V$, *where* c^*, π^* *satisfy* $c^* + \pi^* \leq \mathsf{opt}$.

The above theorem can be used along with the following known results. Louis and Vishnoi [26] obtain $\rho = 2, \alpha = 2, \beta = 2$ for Degree-Constrained Edge-Connectivity Survivable Network. Lau and Singh [24] obtain $\rho = 2, \alpha = 1, \beta = 3$ for Degree-Constrained Steiner Forest and $\rho = 2, \alpha = 1, \beta = 6r_{\max} + 3$ for Degree-Constrained Edge-Connectivity Survivable Network where $r_{\max}$ is the maximum requirement.

In the Group Steiner Tree problem we are given a collection $\mathcal{S}$ of node-subsets (groups), and seek a minimum-cost subtree T of G that contains at least one node from each group. The Group Steiner Tree problem admits ratio $O(\log n \log |\mathcal{S}| \log \mathcal{S}_{\max})$ [11], where $\mathcal{S}_{\max} = \max_{S \in \mathcal{S}} |S|$. Group Steiner Tree with G being a tree admits no $\Omega(\log^{2-\epsilon} n)$ ratio, unless NP has quasi-polynomial Las-Vegas algorithms [15]. Our last result shows that Terminal Steiner Tree (with arbitrary costs), and Group Steiner Tree are equivalent w.r.t. approximation.

Theorem 1.5.

(i) *If* Group Steiner Tree *admits approximation ratio* $\rho(|V|, |\mathcal{S}|, \mathcal{S}_{\max})$ *then* Terminal Steiner Tree *admits ratio* $\rho(|L||V|, |S|, |V|)$.

(ii) *If* Terminal Steiner Tree *admits ratio* $\rho(|V|, |S|)$ *then* Group Steiner Tree *admits ratio* $\rho(|V| + |\mathcal{S}|, |\mathcal{S}|)$.

Consequently, Terminal Steiner Tree *admits ratio* $O(\log^2 n \log |S|)$, *and admits no* $\Omega(\log^{2-\epsilon} n)$ *ratio, unless NP has quasi-polynomial Las-Vegas algorithms.*

Note that in the Terminal Steiner Tree problem the degree bounds are 1, and that in the case of degree bounds 2, even checking whether the problem admits a feasible solution is NP-hard (by a reduction to the Hamiltonian Path problem).

Theorems 1.1, 1.2, 1.4, and 1.5, are proved in Sections 2, 3, 4, and 5, respectively.

2. Degree-Constrained 2-Connected Subgraph (Theorem 1.1)

We start by considering the problem of augmenting a connected graph $J = (V, E_J)$ by a minimum-cost edge-set $I \subseteq E$ such that $\deg_I(v) \leq b(v)$ for all $v \in V$ and such that $J \cup I$ is 2-connected.

Definition 2.1. *For a node v of J let $\mu_J(v)$ be the number of connected components of $J \setminus \{v\}$; v is a* cut-node *of J if $\mu_J(v) \geq 2$.*

Note that since J is connected, any node v has a neighbor in every connected component of $J \setminus \{v\}$. This implies $\mu_J(v) \leq \deg_J(v)$ for every $v \in V$. Let r be a non-cut-node of J; it is known that such r always exists. A set $S \subseteq V \setminus \{r\}$ is *violated* if it has a unique neighbor which we denote by a_S, and a_S is distinct from r. Let $\mathcal{S}_J$ denote the set of violated sets of J. Recall that $\delta_F(S)$ denotes the set of edges in F between S and $V \setminus S$. For $S \in \mathcal{S}_J$ let $\zeta_F(S)$ denote the set of edges in F with one endnode in S and the other in $V \setminus (S \cup \{a_S\})$. By Menger's Theorem, $J \cup I$ is 2-connected if, and only if, $|\zeta_I(S)| \geq 1$ for every $S \in \mathcal{S}_J$. Thus a natural LP-relaxation for our augmentation problem is $\tau = \min\{c \cdot x : x \in P(J, b)\}$, where $P(J, b)$ is the polytope defined by the following constraints:

$$
\begin{array}{rcll}
x(\zeta_E(S)) & \geq & 1 & \text{for all } S \in \mathcal{S}_J \\
x(\delta_E(v)) & \leq & b(v) & \text{for all } v \in B \\
x_e & \in & [0,1] & \text{for all } e \in E
\end{array}
$$

Theorem 2.1. *There exists a polynomial time algorithm that given an instance of* Degree-Constrained 2-Connected Subgraph *and a connected spanning subgraph (V, J) of G computes an edge set $I \subseteq E \setminus J$ such that $c(I) \leq 3\tau$ and such that $\deg_I(v) \leq 3b(v) + \max\{\mu_J(v), 3\} + 1$ for all $v \in B$.*

Theorem 2.1 will be proved later. Now we show how to deduce the promised approximation ratio from it. Consider the following two phase algorithm.

Phase 1: With degree bounds $b(v)$, use the $(b(v) + 1, 1)$-approximation algorithm of Singh and Lau [34] for the Degree Constrained Spanning Tree problem to compute a spanning tree J in G.
Phase 2: Use the algorithm from Theorem 2.1 to compute an augmenting edge set I such that $H = J \cup I$ is 2-connected.

We prove the approximation ratio. We have $c(J) \leq \mathsf{opt}$ and $c(I) \leq 3\tau$, hence $c(H) = c(J) + c(I) \leq 4\mathsf{opt}$. We now prove the approximability of the degrees. Let $v \in V$. Note that $\mu_J(v) \leq \deg_J(v) \leq b(v) + 1$. Thus we have

$$\deg_I(v) \leq 3b(v) + \max\{\mu_J(v), 3\} + 1 \leq 3b(v) + \max\{b(v) + 1, 3\} + 1 \ .$$

Since we must have $b(v) \geq 2$ for all $v \in V$, this implies

$$\deg_H(v) \leq \deg_J(v) + \deg_I(v) \leq 4b(v) + \max\{b(v), 2\} + 3 \leq 5b(v) + 3 \ .$$

In the rest of this section we will prove the following statement, that implies Theorem 2.1.

Lemma 2.2. *Let x be an extreme point of the polytope $P(J, b)$ such that $0 < x_e < 1/3$ for every $e \in E$. Then there is a node $v \in B$ such that $\deg_E(v) \leq \max\{\mu_J(v), 3\} + 2$.*

Lemma 2.2 implies Theorem 2.1 as follows. Given a partial solution I and a parameter $\alpha \geq 1$, the residual degree bounds are $b_I^\alpha(v) = b(v) - \deg_I(v)/\alpha$. The following algorithm starts with $I = \emptyset$ and performs iterations. In every iteration, we work with the residual polytope $P(\mathcal{S}_{J \cup I}, b_I^\alpha)$, and remove some edges from E and/or some nodes from B, until E becomes empty. Let $\alpha = 3$ and $\beta(v) = \max\{\mu_J(v), 3\} + 2$ for all $v \in B$. It is easy to see that for any edge-set $I \subseteq E$ we have $\mu_{J \cup I}(v) \leq \mu_J(v)$ for every $v \in V$.

Algorithm as in Theorem 2.1
Input: A connected graph (V, J), a set of edges E on V with costs $\{c_e : e \in E\}$, integral degree bounds $\{b(v) : v \in V\}$, and non-negative integers $\{\beta(v) : v \in V\}$.
Initialization: $I \leftarrow \emptyset$.
If $P(J, b) = \emptyset$, then return 'UNFEASIBLE' and STOP.
While $E \neq \emptyset$ do:

1. Find a basic solution $x \in P(\mathcal{S}_{J \cup I}, b_I^\alpha)$.
2. Remove from E all edges with $x_e = 0$.
3. Add to I and remove from E all edges with $x_e \geq 1/\alpha$.
4. Remove from B every $v \in B$ with $\deg_E(v) \leq \beta(v)$.

EndWhile
Return I.

It is a routine to prove the following statement, c.f. [29].

Lemma 2.3. *If the above algorithm terminates and does not return 'UN-FEASIBLE', then it computes an edge set I such that $J \cup I$ is 2-connected, $c(I) \leq \alpha\tau$, and $\deg_J(v) \leq \alpha b(v) + \beta(v) - 1$ for all $v \in B$.*

It remains to prove Lemma 2.2. The following statement is well known and can be easily proved using the tree structure of the cut-nodes.

Lemma 2.4. *For any $X, Y \in \mathcal{S}_J$ exactly one of the following holds:*

- *$X \subseteq Y$ and $a_X \in Y \cup \{a_Y\}$, or $Y \subseteq X$ and $a_Y \in X \cup \{a_X\}$;*
- *$X \cap Y = \emptyset$ and $a_X \notin Y$, $a_Y \notin X$.*

Recall that a set-family $\mathcal{L}$ is *laminar* if for any distinct sets $X, Y \in \mathcal{L}$ either $X \subset Y$, or $Y \subset X$, or $X \cap Y = \emptyset$. Note that the family $\mathcal{S}_J$ is laminar. Any laminar family $\mathcal{L}$ defines a partial order on its members by inclusion; we use the usual notion of children, descendants, and leaves of laminar set families. The following statement follows from polyhedral theory.

Lemma 2.5. *For any basic solution $x \in P(J, b)$ with $0 < x(e) < 1$ for all $e \in E$, there exists $\mathcal{L} \subseteq \mathcal{S}_J$ and $T \subseteq B$, such that x is the unique solution to the linear equation system:*

$$\begin{aligned} x(\zeta_E(S)) &= 1 && \text{for all } S \in \mathcal{L} \\ x(\delta_E(v)) &= b(v) && \text{for all } v \in T \end{aligned}$$

Thus $|\mathcal{L}| + |T| = |E|$ and the characteristic vectors of $\{\zeta_E(S) : S \in \mathcal{L}\}$ are linearly independent.

Let x be an extreme point of $P(J, b)$ with $0 < x_e < 1/3$ for every $e \in E$. Let $\mathcal{L}$ and T be as in Lemma 2.5. To prove Lemma 2.2, we will assume that $\deg_J(v) \geq \max\{\mu_J(v) + 3, 6\}$ for all $v \in T$ and obtain a contradiction. For that we will assign two tokens to every edge $e = uv \in E$, placing 1 token at u and 1 token at v. We will show that these tokens can be redistributed such that every member of $\mathcal{L} \cup T$ gets 2 tokens and some spare tokens remain. We need some definitions and simple statement to continue.

Definition 2.2. *Let $S \in \mathcal{L}$. We say that* an edge $e \in E$ covers S *if $e \in \zeta_E(S)$. Let E_S denote the set of edges in E that cover S or a child of S, but not both. Let us say that S* owns *a node v if S is an inclusion minimal set in $\mathcal{L}$ that contains v, and S* shares *v if S is an inclusion minimal set in $\mathcal{L}$ with $a_S = v$.*

From the definition and Lemma 2.4 we have the following.

Claim 2.6. *For any $v \in V$ the following holds.*

(i) *At most one set in $\mathcal{L}$ owns v.*

(ii) *If two distinct sets in $\mathcal{L}$ share v then they are disjoint.*

(iii) *If X owns v and Y shares v then Y is a descendant of X.*

Corollary 2.7. *For any $v \in V$ the following holds.*

(i) *If each of $X, Y \in \mathcal{L}$ owns or shares v then $\delta_{E_X}(v) \cap \delta_{E_Y}(v) = \emptyset$.*

(ii) *At most $\mu_J(v) - 1 \leq \deg_E(v) - 1$ sets in $\mathcal{L}$ share v.*

Proof: We prove (i). It is not possible that both X, Y own v, by Claim 2.6 (i), so assume that one of X, Y, say Y, shares v. Let $e \in \delta_{E_Y}(v)$. Then e goes from v to a child of Y. We will show that $e \notin \delta_{E_X}(v)$. If X, Y share v, then X, Y are disjoint, by Claim 2.6 (ii). This implies that e cannot cover a child of X. Consequently, $e \notin \delta_{E_X}(v)$. The remaining case is when X owns v. Then Y is a descendant of X, by Claim 2.6 (iii). It is easy to see that since e goes from v to a child of Y, then e cannot cover X or a child of X.

We prove (ii). By Claim 2.6 (ii), the sets that share v are pairwise disjoint. Also, every set that shares v is a union of some connected components of $J \setminus \{v\}$ that do not contains r. Consequently, the number of sets that share v is at most the number of such components, which is at most $\mu_J(v) - 1$. $\square$

We start with an intermediate assignment of tokens to the members of $\mathcal{L}$, using the following rules for each $S \in \mathcal{L}$.

1. If S owns or shares $v \notin T$, then v gives $\deg_{E_S}(v)$ tokens to S.
2. If S owns $v \in T$ then v gives to S: 4 tokens if S is a leaf set in $\mathcal{L}$ (namely, if S has no children in $\mathcal{L}$), and 2 tokens otherwise.
3. If S shares $v \in T$, then v gives to S: 2 tokens if S has a unique child C and at least 4 edges incident to v enter C, and $\min\{\deg_{E_S}(v), 1\}$ tokens otherwise.

Lemma 2.8. *For every $v \in V$, the amount of tokens v gives to the members of $\mathcal{L}$ is at most $\deg_E(v)$ if $v \in V \setminus T$, and at most $\deg_E(v) - 2$ otherwise. Thus at most $2|E| - 2|T|$ tokens are assigned to the members of $\mathcal{L}$.*

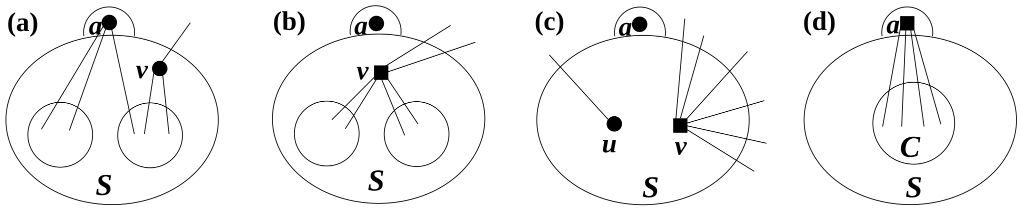

Figure 1: Illustration for Rules 1,2,3. Nodes in T are shown by squares. (a) S gets 3 tokens from each of a, v by Rule 1. (b) S gets 2 tokens from v by Rule 2. (c) S gets 4 tokens from v by Rule 2 and 1 token from u by Rule 1. (d) S gets 2 tokens from a by Rule 3.

Proof: Suppose that $v \in V \setminus T$. Then the tokens at v are assigned by Rule 1. Hence from Corollary 2.7 (i) it follows that v gives to the members of $\mathcal{L}$ at most $\sum_{S \in \mathcal{L}} \deg_{E_S}(v) \leq \deg_E(v)$ tokens, and the statement holds in this case.

If $v \in T$, then $\deg_E(v) - 2 \geq \max\{\mu_J(v) + 1, 4\}$ and the tokens at v are assigned by Rules 2 and 3. Suppose that there is $S \in \mathcal{L}$ that owns v. Then Rule 2 applies and no other set owns v, by Claim 2.6 (i). If S is a leaf set of $\mathcal{L}$ then v gives 4 tokens to S, and v is not shared by any other biset, by Claim 2.6 (iii); hence no other member of $\mathcal{L}$ gets tokens from v, and the statement holds in this case. Otherwise, if S is not a leaf set, then v gives 2 tokens to S, and has $\deg_E(v) - 4 \geq \mu_J(v) - 1$ tokens to give, using Rule 3, to the sets in $\mathcal{L}$ that share v. Let q be the number of sets in $\mathcal{L}$ that share v and get 2 tokens from v. For each such set S, $\deg_{E_S}(v) \geq 4$. Hence $\deg_E(v) - 4 \geq 3q + \mu_J(v) - 1$, by Corollary 2.7 (i). On the other hand, the number of tokens v gives to sets that share v is $q + \mu_J(v) - 1 \leq 3q + \mu_J(v) - 1$. The statement follows. $\square$

Lemma 2.9. *For every $S \in \mathcal{L}$, it is possible to redistributed the tokens assigned to S and its descendants such that every descendant of S gets at least 2 tokens, and S gets at least 4 tokens.*

Proof: The proof is by induction on the number of descendant of S. The induction base is when S has no descendants. Then at least 4 edges cover S. If S owns no node in T then S gets 4 tokens from these edges, by Rule 1. If S owns $v \in T$ then S gets 4 tokens from v, by Rule 2.

Now suppose that S has at least 1 descendant, and hence S has at least one child. By the induction hypothesis, S can can get 2 tokens from every child, hence if S has at least 2 children then we are done. Suppose therefore that S has a unique child C. Then S can get 2 tokens from C and needs 2 more tokens. Note that $|E_S| \geq 2$, by the linear independence and the integrality of cuts in Lemma 2.5.

Let V_S be the set of those endnodes of the edges in E_S that are owned or shared by S. By our assignment rules, if $|V_S| \geq 2$, of if $V_S \cap T = \emptyset$, or if S owns a node $v \in V_S \cap T$, then S gets 2 tokens from nodes in V_S. We are left with the case $V_S = \{a_S\}$, $a_S \in T$, and S shares a_S. Then the assumption $x_e < 1/3$ implies that at least 4 edges go from a_S to R, and S gets 2 tokens from a_S, by Rule 3. □

Lemmas 2.8 and 2.9 imply the contradiction $2|E| - 2|T| < 2|\mathcal{L}|$. Consequently, the proof of Lemma 2.2, and thus also of Theorem 1.1, is complete.

3. Minimum Degree k-Arborescence (Theorem 1.2)

3.1. Hardness

We prove Proposition 1.3 which is basically a corollary of the work of [16]. In [16] an arbitrary NPC (yes or no) instance is reduced to a Set Cover instance in which the elements are divided to ground sets. In fact a ground set $M(A, B)$ is associated to a subset $A \cup B$ of the sets, and elements of $M(A, B)$ may belong to $A \cup B$ only. Note that a set A and a set B may be assigned other ground sets (say $M(A, B')$). The size of the ground sets can be chosen by the reduction and is typically equal to the number of sets and the size of the input which is $O(n^{\log\log n})$.

The following is achieved in [16]. Starting with a yes instance of the NPC problem, the resulting Set-Cover one could always pick two sets $a \in A, b \in B$ so that these sets contain all of the elements of $M(A, B)$.

In the case of a Set Cover derived from a no instance of the NPC problem, every set in $A \cup B$ is essentially joined to a random half of $M(A, B)$ (even though the construction can be derandomized [28]). Intuitively, this implies a gap (hence inapproximability) of $\log_2 |M|/2$ for Set-Cover.

Changes: We later change the size of ground sets but first we describe an elementary reduction to the Minimum Degree k-Arborescence. Make the Set Cover instance a directed bipartite graph $G(S, M, E)$ with S all the sets and M the union of ground sets. A set s is joined by a directed edge to an element x if $x \in s$. The edges of this graph given cost 1. Add a node r and add directed edges from r to all the sets each edge, of cost 1. The reduction to Minimum Degree k-Arborescence complete by declaring all elements M to be terminals and setting $k = |M|$. Hence, in fact we give the lower bound for the special case that k equals the number of terminals. It may well be that for general k a much better hardness can be proven.

Note that in order for a tree rooted at r to span all the terminals, it must be connected in S to a Set-Cover of M.

The degrees, an important detail:. We want to make the maximum degree in $G(S, M, E)$ negligible compared to the size of the set cover. From [16] it follows that the maximum degree in M is $O(\log^c n)$ hence negligible compared to the size of a set cover. To make the degrees in S negligible compared to the size of a set cover we make every groundset of size $\sqrt{n}$ instead of the usual size which is roughly n. It follows from [16] that the maximum degree of a vertex in S is now $O(\log^{c_1} n \cdot \sqrt{n})$ for some constant c_1.

The size of the Minimum Set Cover in [16] is $n/O(\log^{c_2} n)$ for some constant c_2. Hence, clearly the maximum degree in the graph will be the one of r, that, since the graph is directed, and r must be joined to a Set Cover in S. The only difference is that the gap is now $\log_2(\sqrt{n})/2$ which still is a logarithmic gap.

For the undirected case, the only way we find to force r to be connected to a Set Cover in S, is to bound the diameter of the resulting graph by 4. This ends the proof of Proposition 1.3.

3.2. The approximation

We may assume that in the input graph G every node is reachable from the root s, that every terminal has indegree 1 and outdegree 0, and that the set of terminal of every arborescence T coincides with the set of leaves of T. Let $U = V \setminus S$. Before describing the algorithm, we need some definitions.

Definition 3.1. *For $W \subseteq U$ and an integer parameter $\alpha \geq 1$ the network $F_\alpha(W)$ with source s' and sink t' is obtained from G as follows.*

1. *Assign infinite capacity to every edge of G and capacity α to every node in U.*
2. *Add a new node s' and add new edges of capacity α each from s' to every node in W.*
3. *Add two new nodes t, t', add an edge of capacity 1 from every terminal to t, and add an edge of capacity k from t to t'.*

Our algorithm runs with an integer parameter α set eventually to

$$\alpha = \left\lceil \sqrt{k \cdot \Delta^* \cdot (\ln k + 1)} \right\rceil \; . \tag{3}$$

Although Δ^* is not known, $\Delta^* \leq k$, and our algorithm applies exhaustive search in the range $1, \ldots, k$.

Recall that a set-function ν defined on subsets of a ground-set U is *submodular* if $\nu(A) + \nu(B) \geq \nu(A \cup B) + \nu(A \cap B)$ for all $A, B \subseteq U$. Consider the following well known generic problem (for our purposes we state only the unweighted version).

> Submodular Cover
> *Instance:* A finite set U and a non-decreasing submodular function $\nu : 2^U \mapsto \mathbb{Z}$.
> *Objective:* A minimum-size subset $W \subseteq U$ such that $\nu(W) = \nu(U)$.

The Submodular Cover Greedy Algorithm (for the unweighted version) starts with $W = \emptyset$ and while $\nu(W) < \nu(U)$ repeatedly adds to W an element $u \in U \setminus W$ that maximizes $\nu(W \cup \{u\}) - \nu(W)$. At the end, W is output. It is proved in [35] that the Greedy Algorithm for Submodular Cover has approximation ratio $\ln \max\limits_{u \in U} \nu(\{u\}) + 1$.

A generalization of the following statement is proved in [3].

Lemma 3.1 ([3]). *For $W \subseteq U$ let $\nu_\alpha(W)$ be the maximum $s't'$-flow value in the network $F_\alpha(W)$. Then ν_α is non-decreasing and submodular, and $\nu_\alpha(U) \leq k$.*

The algorithm is as follows.

1. Execute the Submodular Cover Greedy Algorithm with $U = V \setminus S$ and with $\nu = \nu_\alpha$; let $W \subseteq U$ be the node-set computed.
2. Let f be a maximum integral flow in $F_\alpha(W)$ and let $J_W = \{e \in E : f(e) > 0\}$ be the set of those edges in E that carry a positive flow in $F_\alpha(W)$.
   Let T_W be an inclusion-minimal arborescence in G rooted at s containing W.
3. Return any k-arborescence contained in the graph $(V, J_W) \cup T_W$.

In the rest of this section we prove that the graph $(V, J_W) \cup T_W$ indeed contains a k-arborescence, and that for any integer $\alpha \geq 1$ it has maximum outdegree at most $\alpha + (\ln k + 1) \cdot k\Delta^*/\alpha$.

For α given by (3), this implies the approximation ratio

$$\alpha/\Delta^* + (\ln k + 1) \cdot k/\alpha = O(\sqrt{(k \log k)/\Delta^*}) .$$

Definition 3.2. *A collection $\mathcal{T}$ of sub-arborescence of an arborescence T is an α-*leaf-covering decomposition *of T if the arborescence in $\mathcal{T}$ are pairwise node-disjoint, every leaf of T belongs to exactly one of them, and each of them has at most α leaves.*

Lemma 3.2. *Suppose that G contains a k-arborescence T that admits an α-leaf-covering decomposition $\mathcal{T}$. Let R be the set of roots of the arborescence in $\mathcal{T}$. Then $\nu_\alpha(R) = k$, and for the set W computed by the algorithm the following holds:*

(i) *$\nu_\alpha(W) = k$ and thus the graph $J_W \cup T_W$ contains a k-arborescence.*

(ii) *The graph $(V, J_W) \cup T_W$ has maximum outdegree $\leq \alpha + |\mathcal{T}| \cdot (\ln k + 1)$.*

Proof: We prove that $\nu_\alpha(R) = k$. For a terminal v in T, let $r_v \in R$ be the root of the (unique) arborescence $T_v \in \mathcal{T}$ that contains v, and let P_v be the path in $F_\alpha(R)$ that consists of: the edge $s'r_v$, the unique path from r_v to v in T_v, and the edges vt' and $t't$. Let f be the flow obtained by sending for every terminal v of T one flow unit along P_v. Then f has value k, since T has k terminals. We verify that f obeys the capacity constraints in $F_\alpha(R)$. For every $r \in R$, the arborescence $T_r \in \mathcal{T}$ which root is r, has at most α terminals; hence the edge $s'r$ carries at most α flow units, which does not exceed its capacity α. This also implies that the capacity α on all nodes in U is met. For every terminal v of T, the edge vt carries one flow unit and has capacity 1. The edge $t't$ carries k flow units and has capacity k. Other edges have infinite capacity.

We prove (i). By Lemma 3.1, ν_α is non-decreasing and $\nu_\alpha(U) \leq k$. As $\nu_\alpha(U) \geq \nu_\alpha(R) = k$ and $\nu_\alpha(W) = \nu_\alpha(U)$, we conclude that $\nu(W) = k$. This implies that in the graph (V, J_W), k terminals are reachable from W, and (i) follows.

We prove (ii). In the graph (V, J_W), the outdegree of any node is at most α. This follows from the capacity α on any node in U. We have $|W| \leq$

$|R| \cdot (\ln k+1) = |\mathcal{T}| \cdot (\ln k+1)$, by Lemma 3.2 (i) and the approximation ratio of the Submodular Cover Greedy Algorithm. Since T_W is an arborescence with leaf-set W, the maximum outdegree of T_W is at most $|W| \leq |\mathcal{T}| \cdot (\ln k + 1)$. The statement follows. □

The following lemma implies that the optimal tree T^* admits an α-leaf-covering decomposition $\mathcal{T}$ of size $|\mathcal{T}| \leq k\Delta^*/\alpha$ for any $\alpha \geq 1$. This together with Lemma 3.2 concludes the proof of Theorem 1.2.

Lemma 3.3. *Any arborescence T with k leaves and maximum outdegree Δ admits an α-leaf-covering decomposition $\mathcal{T}$ of size $|\mathcal{T}| \leq \Delta \cdot \lfloor k/(\alpha+1) \rfloor + 1$, for any integer $\alpha \geq 1$.*

Proof: For a node r of an arborescence T with root s let us use the following notation: T_r is the sub-arborescence of T with root r that contains all descendants of r in T, and P_r is the set of internal nodes in the ar-path in T, where a is the closest to r ancestor of r that has outdegree at least 2. Let us say that a node $u \in U$ of T is *α-critical* if T_u has more than α leaves, but no child of u has this property. It is easy to see that T has an α-critical node if, and only if, T has more than α leaves.

Consider the following algorithm. Start with $\mathcal{T} = \emptyset$. While T has an α-critical node u do the following: add T_r to $\mathcal{T}$ for every child r of u, and remove T_u and P_u from T (note that since we remove P_u no new leaves are created). When the while loop ends, if T is nonempty, add the remaining arborescence $T = T_s$ (which now has at most α leaves) to $\mathcal{T}$.

By the definition, the arborescence in $\mathcal{T}$ are pairwise node-disjoint, every leaf of T belongs to exactly one of them, and each of them has at most α leaves. It remains to prove the bound on $\mathcal{T}$. In the loop, when we consider an α-critical node u, at least $\alpha + 1$ leaves are removed from T and at most Δ arborescence are added to $\mathcal{T}$. Hence $|\mathcal{T}| \leq \Delta \cdot \lfloor k/(\alpha + 1) \rfloor$ at the end of the loop. At most one additional arborescence is added to $\mathcal{T}$ after the loop. The statement follows. □

4. Prize-Collecting Degree-Constrained Survivable Network (Theorem 1.4)

Our LP-relaxation for Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network is:

$$
\begin{array}{llcll}
\text{Minimize} & \sum_{e\in E} c_e x_e + \sum_{I\subseteq K} \pi(I) z_I & & & \\
\text{Subject to} & \sum_{e\in\delta(T)} f_{i,e} & \geq & (1-\sum_{I:i\in I} z_I) r_i(T) & \forall i\ \forall T \odot i \\
& f_{i,e} & \leq & 1-\sum_{I:i\in I} z_I & \forall i\ \forall e \\
& x_e & \geq & f_{i,e} & \forall i\ \forall e \\
& \sum_{I\subseteq K} z_I & = & 1 & \\
& \sum_{e\in\delta(v)} x_e & \leq & b(v) & \forall v \\
& x_e, f_{i,e}, z_I & \in & [0,1] & \forall i\ \forall e\ \forall I
\end{array}
\tag{4}
$$

Without the degree constraints, this LP-relaxation was used in [13] for Prize-Collecting Edge-Connectivity Survivable Network. In the intended integral solution H, the variables are supposed to take the following values: $x_e = 1$ if $e \in H$, $f_{i,e} = 1$ if $i \notin \texttt{unsat}(H)$ and e appears on a chosen set of r_i edge-disjoint $\{u_i, v_i\}$-paths in H and $z_I = 1$ if $I = \texttt{unsat}(H)$. We prove the following refinement of Theorem 1.4.

Theorem 4.1. *Suppose that for a* Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network *instance the following holds. For any $\mathcal{P}' \subseteq \mathcal{P}$, the* Degree-Constrained Edge-Connectivity Survivable Network *instance defined by $\mathcal{P}'$ admits a polynomial-time algorithm that computes a solution H' of cost at most ρ times the optimal value of LP (2) such that $\deg_{H'}(v) \leq \alpha b(v) + \beta$ for all $v \in B$. Then the* Prize-Collecting Degree-Constrained Edge-Connectivity Survivable Network *instance admits a polynomial time algorithm that for any $\mu \in (0,1)$ computes a subgraph H of G such that $c(H) \leq \frac{\rho}{1-\mu} \sum_{e\in E} c_e x_e^*$, $\pi(\texttt{unsat}(H)) \leq \frac{1}{\mu} \sum_{I\subseteq K} \pi(I) z(I)$, and $\deg_H(v) \leq \frac{\alpha}{1-\mu} b(v) + \beta$ for all $v \in B$.*

We prove Theorem 4.1. Let $\{x_e^*, f_{i,e}^*, z_I^*\}$ be a feasible solution to LP (4) and let $\mu \in (0,1)$. We partition the requirements into two classes: we call a requirement r_i *good* if $\sum_{I:i\in I} z_I^* \leq \mu$ and *bad* otherwise. Let $\mathcal{R}_g$ denote the set of good requirements. The following statement shows how to satisfy the good requirements.

Lemma 4.2. *There exists a polynomial-time algorithm that computes a subgraph H of G of cost $c(H) \leq \frac{\rho}{1-\mu} \cdot \sum_e c_e x_e^*$ that satisfies all good requirements such that $\deg_H(v) \leq \frac{\alpha}{1-\mu} b(v) + \beta$ for all $v \in V$.*

Proof: Consider the LP-relaxation (2) of the Degree-Constrained Edge-Connectivity Survivable Network problem with good requirements only, with K replaced by K_g; namely, we seek a minimum cost subgraph H of G that satisfies the set K_g of good requirements and the degree constraints. We claim that $x_e^{**} = \min\{1, x_e^*/(1-\mu)\}$ for each $e \in E$ is a feasible solution to LP (2) with degree bounds $\frac{b(v)}{1-\mu}$. Thus the optimum value of LP (2) is at most $\sum_{e\in E} c_e x_e^{**}$. Consequently, using the algorithm that computes an integral solution to LP (2) of cost at most ρ times the optimal value of LP (2) and with degrees at most $\alpha b(v)+\beta$, we can construct a subgraph H that satisfies all good requirements and has cost at most $c(H) \le \rho \sum_{e\in E} c_e x_e^{**} \le \frac{\rho}{1-\mu}\sum_e c_e x_e^*$, and degrees at most $\deg_H(v) \le \frac{\alpha}{1-\mu} b(v)+\beta$, as desired.

We now show that $\{x_e^{**}\}$ is a feasible solution to LP (2), namely, that $\sum_{e\in\delta(A)} x_e^{**} \ge r_i(A)$ for any $i \in K_g$ and any $A \odot i$. Let $i \in K_g$ and let $\zeta_i = 1 - \sum_{I:i\in I} z_I^*$. Note that $\zeta_i \ge 1-\mu$, by the definition of K_g. By the second and the third sets of constraints in LP (4), for every $e \in E$ we have $\min\{\zeta_i, x_e^*\} \ge f_{i,e}^*$. Thus we obtain: $x_e^{**} = \min\left\{1, \frac{x_e^*}{1-\mu}\right\} = \frac{1}{\zeta_i}\min\left\{\zeta_i, \frac{\zeta_i}{1-\mu}x_e^*\right\} \ge \frac{1}{\zeta_i}\min\{\zeta_i, x_e^*\} \ge \frac{f_{i,e}^*}{\zeta_i} = \frac{f_{i,e}^*}{1-\sum_{I:i\in I} z_I^*}$. Consequently, combining with the first set of constraints in LP (4), for any $A \odot i$ we obtain that $\sum_{e\in\delta(A)} x_e^{**} \ge \frac{\sum_{e\in\delta(A)} f_{i,e}^*}{1-\sum_{I:i\in I} z_I^*} \ge r_i(A)$. □

Let H be as in Lemma 4.2, and recall that $\texttt{unsat}(H)$ denotes the set of requirements not satisfied by H. Clearly each requirement $i \in \texttt{unsat}(H)$ is bad. The following lemma bounds the total penalty we pay for $\texttt{unsat}(H)$.

Lemma 4.3. $\pi(\texttt{unsat}(H)) \le \frac{1}{\mu}\cdot\sum_I \pi(I) z_I^*$.

Proof: This lemma was proved in [13] for the case when there are no degree bounds, and the proof of the case with degree bounds is identical. □

The proof of Theorem 4.1 and thus also of Theorem 1.4 is now complete.

5. Terminal Steiner Tree (Theorem 1.5)

We start by proving Part (i) of Theorem 1.5, namely, that if Group Steiner Tree admits approximation ratio $\rho(|V|, |\mathcal{S}|, \mathcal{S}_{\max})$ then Terminal Steiner Tree admits ratio $\rho(|L|\cdot|V|, |S|, |V|)$. Given an instance $G=(V,E), c, S, L$ of Leaf-Constrained Steiner Tree construct an instance $G'=(V',E'), c', \mathcal{S}'$ of Group Steiner Tree as follows.

- The pair G', c' is obtained from G, c as follows. For every $v \in L$ do the following. For every $u \in \Gamma_G(v) \setminus L$ add a new node v_u, and replace the edge $e = uv$ by the new edge $e' = uv_u$, of the same cost as e. Then remove v and all the edges incident to it from the graph.

- The set of groups is as follows. Every $v \in L$ defines the group $S(v) = \{v_u : u \in \Gamma_G(v) \setminus L\}$. The collection of groups is $\mathcal{S}' = \{\{S(v)\} : v \in L\} \cup \{\{s\} : s \in S \setminus L\}$.

By the construction, $|V'| \leq |V| \cdot |L|$, $|S| = |\mathcal{S}|$, and $\mathcal{S}_{\max} \leq |V|$. Note that to every edge-set $F' \subseteq E'$ corresponds the edge-set $F \subseteq E$, where to every edge $e' = uv_u$ corresponds the edge uv, and the other edges appear in both F and F'. Note that if F corresponds to F', then F, F' have the same cost, namely, $c(F) = c'(F')$, and that if F' is a tree then so is F. Now we prove the following.

Lemma 5.1. *If T' is an inclusion-minimal solution to the obtained* Group Steiner Tree *instance then the edge set T that corresponds to T' is a feasible solution to the* Terminal Steiner Tree *instance. Furthermore, to every inclusion-minimal solution T to the* Terminal Steiner Tree *instance there exists a feasible solution T' to the* Group Steiner Tree *instance, such that T corresponds to T'.*

Proof: Let T' be an inclusion-minimal solution to the obtained Group Steiner Tree instance. Let $T \subseteq E$ the edge set that corresponds to T'. From the construction it is clear that T satisfies the connectivity requirements. We show that T satisfies the degree constraints. Since T' is an inclusion minimal solution, for every $v \in L$ there is a unique node $v_u \in S(v)$ included in the tree T'. This implies $\deg_T(v) = 1$.

Let T be an inclusion-minimal solution to the Terminal Steiner Tree instance. If $|S| = 2$ then the statement is easily verified, so assume that $|S| \geq 3$. There is no edge in T between two nodes in L. Hence every $v \in L$ has its unique neighbor in $V \setminus L$. The tree T' is obtained from T by replacing for every $v \in L$ the unique edge uv incident to v in T by the edge uv_u. Clearly, T corresponds to T', and it is easy to see that T' is a feasible solution to the obtained Group Steiner Tree instance. □

Now we prove Part (ii) of Theorem 1.5, namely that if Terminal Steiner Tree admits ratio $\rho(|V|, |S|)$ then Group Steiner Tree admits ratio $\rho(|V| +$

$|\mathcal{S}|, |\mathcal{S}|)$. Given an instance $G = (V, E), c, \mathcal{S}$ of Group Steiner Tree construct an instance $G' = (V', E'), c', S', L'$ of Terminal Steiner Tree as follows.

- The pair G', c' is obtained from G, c by adding for every group S_i a new node v_i and connecting v_i to every node in S_i by an edge of cost zero.
- The node sets S', L' are defined by $S' = L' = \{v_1, \ldots, v_{|\mathcal{S}|}\}$.

By the construction, $|V'| = |V| + |\mathcal{S}|$ and $|S'| = |L'| = |\mathcal{S}|$. Now it is easy to see the following.

Lemma 5.2. *If T' is a feasible solution to the obtained* Terminal Steiner Tree *instance then the tree $T = T' \setminus L$ is a feasible solution to the original* Group Steiner Tree *instance and $c(T) = c'(T')$. Furthermore, to every feasible solution T to the* Group Steiner Tree *instance there exists a feasible solution T' to the* Group Steiner Tree *instance, such that $T = T' \setminus L$.* □

The proof of Theorem 1.5 is complete.

6. Discussion and open problems

In this paper we gave the first constant degree and cost approximation for the Degree-Constrained 2-Connected Subgraph problem. Recently, in [30], the method here was generalized to obtain constant ratios for several other node-connectivity degree-constrained problems. For the Degree-constrained k-Connected Subgraph problem the algorithm in [30] has ratio $O(2^k)$ for the degrees and $O(\log k)$ for the cost.

Now let us focus on the *undirected* Minimum Degree k-MST problem. We do not know a lower bound for this problem beyond the standard APX-hardness result. However, it may be a bad sign that the natural LP already has a large integrality gap. Hence a polylogarithmic approximation for this problem may be hard or not possible to obtain. An easier task might be to get an n^ϵ-approximation scheme, similar to the one that exists for the Directed Steiner Tree problem (see [4]).

Finally, we note that for the Minimum Degree k-MST problem (the undirected variant) it is not hard to design an *"iterative merging"* algorithm with ratio $O(n/k)$ (c.f., [27]). Combined with our result in Theorem 1.2 this implies ratio $O(n^{1/3})$, which in terms of n might be better than the one in Theorem 1.2. We do not know if this holds also for directed graphs. The main open problem here is either to achieve a polylogarithmic ratio, or to give a strong evidence that a polylogarithmic ratio is unlikely.

References

[1] N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree bounded directed network design. *SIAM J. Computing*, 39(4):1413–1431, 2009.

[2] J. Bar-Ilan, G. Kortsarz, and D. Peleg. Generalized submodular cover problems and applications. *Theoretical Computer Science*, 250:179–200, 2001.

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

[4] Y. H. Chen, C. L. Lu, and C. Y. Tang. On the full and bottleneck full steiner tree problems. In *COCOON*, pages 122–129, 2003.

[5] D. Drake and S. Hougardy. On approximation algorithms for the terminal Steiner tree problem. *Information Processing Letters*, 89(1):15–18, 2004.

[6] M. Elkin and G. Kortsarz. An approximation algorithm for the directed telephone multicast problem. *Algorithmica*, 45(4):569–583, 2006.

[7] T. Feder, R. Motwani, and A. Zhu. k-connected spanning subgraphs of low degree. *Electronic Colloquium on Computational Complexity (ECCC)*, 041, 2006.

[8] M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *J. Algorithms*, 17(3):409–423, 1994.

[9] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000.

[10] M. Goemans. Bounded degree minimum spanning trees. In *FOCS*, pages 273–282, 2006.

[11] M. Hajiaghayi, R. Khandekar, G. Kortsarz, and Z. Nutov. Prize-Collecting Steiner Network Problems. In *IPCO*, pages 71–84, 2010.

[12] M. Hajiaghayi and A. Nasri. Prize-collecting steiner networks via iterative rounding. In *LATIN*, pages 515–526, 2010.

[13] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *STOC*, pages 585–594, 2003.

[14] D. Hochbaum, editor. *Approximation algorithms for NP-hard problems.* PWS Publishing Co., Boston, MA, USA, 1997.

[15] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[16] R. Khandekar, G. Kortsarz, and Z. Nutov. Network-design with degree constraints. In *APPROX-RANDOM*, pages 289–301, 2011.

[17] P. Klein, R. Krishnan, B. Raghavachari, and R. Ravi. Approximation through local optimality: designing networks with small degree. *Networks*, 44:203–215, 2004.

[18] G. Kortsarz and Z. Nutov. *Approximating minimum cost connectivity problems,* in *Approximation Algorithms and Metahueristics,* T. F. Gonzalez ed. Chapman and Hall/CRC, 2007. Chapter 58.

[19] Lap Chi Lau, J. Naor, M. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. *SIAM J. Computing*, 39(3):1062–1087, 2009.

[20] Lap Chi Lau, R. Ravi, and M. Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.

[21] Lap Chi Lau and M. Singh. Additive approximation for bounded degree survivable network design. In *STOC*, pages 759–768, 2008.

[22] G.-H. Lin and G. Xue. On the terminal steiner tree problem. *Information Processing Letters*, 84:103–107, 2002.

[23] A. Louis and N. Vishnoi. Improved algorithm for degree bounded survivable network design problem. In *SWAT*, pages 408–419, 2010.

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

[25] M. Naor, L. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *FOCS*, pages 182–191, 1995.

[26] Z. Nutov. Approximating directed weighted-degree constrained networks. *Theoretical Computer Science*, 412(8-10):901–912, 2011.

[27] Z. Nutov. Degree-constrained node-connectivity. Manuscript, 2011.

[28] R. Ravi, B. Raghavachari, and P. Klein. Approximation through local optimality: Designing networks with small degree. In *FSTTCS*, pages 279–290, 1992.

[29] M. Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC*, pages 661–670, 2007.

[30] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.