Noname manuscript No.
(will be inserted by the editor)

Approximating Min-Cost Chain-Constrained Spanning Trees: A Reduction from Weighted to Unweighted Problems

André Linhares · Chaitanya Swamy

the date of receipt and acceptance should be inserted later

Abstract We study the *min-cost chain-constrained spanning-tree* (abbreviated MCCST) problem: find a min-cost spanning tree in a graph subject to degree constraints on a nested family of node sets. We devise the *first* polytime algorithm that finds a spanning tree that (i) violates the degree constraints by at most a constant factor *and* (ii) whose cost is within a constant factor of the optimum. Previously, only an algorithm for *unweighted* CCST was known [14], which satisfied (i) but did not yield any cost bounds. This also yields the first result that obtains an $O(1)$-factor for *both* the cost approximation and violation of degree constraints for any spanning-tree problem with general degree bounds on node sets, where an edge participates in a super-constant number of degree constraints.

A notable feature of our algorithm is that we *reduce* MCCST to unweighted CCST (and then utilize [14]) via a novel application of *Lagrangian duality* to simplify the *cost structure* of the underlying problem and obtain a decomposition into certain uniform-cost subproblems.

We show that this Lagrangian-relaxation based idea is in fact applicable more generally and, for any cost-minimization problem with packing side-constraints, yields a reduction from the weighted to the unweighted problem. We believe that this reduction is of independent interest. As another application of our technique, we consider the *k-budgeted matroid basis* problem, where we build upon a recent rounding algorithm of [4] to obtain an improved $n^{O(k^{1.5}/\epsilon)}$-time algorithm that returns a solution that satisfies (any) one of the budget constraints exactly and incurs a $(1+\epsilon)$-violation of the other budget constraints.

A. Linhares · C. Swamy
Dept. of Combinatorics and Optimization, University of Waterloo, Waterloo, ON N2L 3G1.
E-mail: {alinhare,cswamy}@uwaterloo.ca
A preliminary version [13] appeared in the Proceedings of the 18th IPCO, 2016.
Research supported partially by NSERC grant 327620-09 and the second author's Discovery Accelerator Supplement Award, and Ontario Early Researcher Award.

1 Introduction

Constrained spanning-tree problems, where one seeks a minimum-cost spanning tree satisfying additional ($\{0,1\}$-coefficient) packing constraints, constitute an important and widely-studied class of problems. In particular, when the packing constraints correspond to node-degree bounds, we obtain the classical *min-cost bounded-degree spanning tree* (MBDST) problem, which has a rich history of study [7,11,12,5,8,16] culminating in the work of [16] that yielded an optimal result for MBDST. Such degree-constrained network-design problems arise in diverse areas including VLSI design, vehicle routing and communication networks (see, e.g., the references in [15]), and their study has led to the development of powerful techniques in approximation algorithms.

Whereas the *iterative rounding and relaxation* technique introduced in [16] (which extends the iterative-rounding framework of [10]) yields a versatile technique for handling node-degree constraints (even for more-general network-design problems), we have a rather limited understanding of spanning-tree problems with more-general degree constraints, such as constraints $|T \cap \delta(S)| \leq b_S$ for sets S in some (structured) family $\mathcal{S}$ of node sets.[1] A fundamental impediment here is our inability to leverage the techniques in [8,16]. The few known results yield: (a) (sub-) optimal cost, but a *super-constant* additive- or multiplicative- factor violation of the degree bounds [3,1,6,2]; or (b) a multiplicative $O(1)$-factor violation of the degree bounds (when $\mathcal{S}$ is a nested family), but *no cost guarantee* [14]. In particular, in stark contrast to the results known for node-degree-bounded network-design problems, there is no known algorithm that yields an $O(1)$-factor cost approximation *and* an (additive or multiplicative) $O(1)$-factor violation of the degree bounds. (Such guarantees are only known when each edge participates in $O(1)$ degree constraints [2]; see however [17] for an exception.)

We consider the *min-cost chain-constrained spanning-tree* (MCCST) problem introduced by [14], which is perhaps the most-basic setting involving general degree bounds where there is a significant gap in our understanding vis-a-vis node-degree bounded problems. In MCCST, we are given an undirected connected graph $G = (V, E)$, nonnegative edge costs $\{c_e\}$, a nested family $\mathcal{S}$ (or *chain*) of node sets $S_1 \subsetneq S_2 \subsetneq \cdots \subsetneq S_\ell \subsetneq V$, and integer degree bounds $\{b_S\}_{S \in \mathcal{S}}$. The goal is to find a minimum-cost spanning tree T such that $|\delta_T(S)| \leq b_S$ for all $S \in \mathcal{S}$, where $\delta_T(S) := T \cap \delta(S)$. Olver and Zenklusen [14] give an algorithm for *unweighted CCST* that returns a tree T such that $|\delta_T(S)| = O(b_S)$ (i.e., there is *no bound on* $c(T)$), and show that, for some $\rho > 0$, it is *NP*-complete to obtain an additive $\rho \cdot \frac{\log |V|}{\log \log |V|}$ violation of the degree bounds. We therefore focus on bicriteria (α, β)-guarantees for MCCST, where the tree T returned satisfies $c(T) \leq \alpha \cdot OPT$ and $|\delta_T(S)| \leq \beta \cdot b_S$ for all $S \in \mathcal{S}$.

Our contributions. Our main result is the *first* $\big(O(1), O(1)\big)$-approximation algorithm for MCCST. Given any $\lambda > 1$, our algorithm returns a tree T with $c(T) \leq \frac{\lambda}{\lambda - 1} \cdot OPT$ *and* $|\delta_T(S)| \leq 9\lambda \cdot b_S$ for all $S \in \mathcal{S}$, using the algorithm of [14] for unweighted CCST, denoted $\mathcal{A}_{\mathrm{OZ}}$, as a black box (Theorem 3.3). As noted above,

[1] Such general degree constraints arise in the context of finding *thin trees* [1], where $\mathcal{S}$ consists of all node sets, which turn out to be a very useful tool in devising approximation algorithms for *asymmetric TSP*.

this is also the *first* algorithm that achieves an $\big(O(1), O(1)\big)$-approximation for any spanning-tree problem with general degree constraints where an edge belongs to a super-constant number of degree constraints.

We show in Section 4 that our techniques are applicable more generally. We give a *reduction* showing that for *any* cost-minimization problem with packing side-constraints, if we have an algorithm for the *unweighted* problem that returns a solution with an $O(1)$-factor violation of the packing constraints and satisfies a certain property, then one can utilize it to obtain an $\big(O(1), O(1)\big)$-approximation for the cost-minimization problem. Furthermore, we show that if the algorithm for the unweighted counterpart satisfies a stronger property, then we can utilize it to obtain a $\big(1, O(1)\big)$-approximation (Theorem 5.1).

We believe that our reductions are of independent interest and will be useful in other settings as well. Demonstrating this, we show an application to the k-*budgeted matroid basis* problem, wherein we seek to find a basis satisfying k budget constraints. Grandoni et al. [9] devised an $n^{O(k^2/\epsilon)}$-time algorithm that returned a $(1, 1+\epsilon, \ldots, 1+\epsilon)$-solution: i.e., the solution satisfies (any) one budget constraint exactly and violates the other budget constraints by a $(1+\epsilon)$-factor (if the problem is feasible). Very recently, Bansal and Nagarajan [4] improved the running time to $n^{O(k^{1.5}/\epsilon)}$ but return only a $(1+\epsilon, \ldots, 1+\epsilon)$-solution. Applying our reduction (to the algorithm in [4]), we obtain the *best of both worlds*: we return a $(1, 1+\epsilon, \ldots, 1+\epsilon)$-solution in $n^{O(k^{1.5}/\epsilon)}$-time (Theorem 5.7).

The chief novelty in our algorithm and analysis, and the key underlying idea, is an unorthodox use of *Lagrangian duality*. Whereas typically Lagrangian relaxation is used to drop complicating constraints and thereby simplify the constraint structure of the underlying problem, in contrast, we use Lagrangian duality to simplify the *cost structure* of the underlying problem by equalizing edge costs in certain subproblems. To elaborate (see Section 3.1), the algorithm in [14] for unweighted CCST can be viewed as taking a solution x to the natural linear-programming (LP) relaxation for MCCST, converting it to another feasible solution x' satisfying a certain structural property, and exploiting this property to round x' to a spanning tree. The main bottleneck here in handling costs (as also noted in [14]) is that $c^\intercal x'$ could be much larger than $c^\intercal x$ since the conversion ignores the c_es and works with an alternate "potential" function.

Our crucial insight is that *we can exploit Lagrangian duality to obtain perturbed edge costs* $\{c_e^{y^*}\}$ *such that there is no change in the perturbed cost due to the conversion process* (see Lemma 3.6). The perturbation also ensures that applying $\mathcal{A}_{\mathrm{OZ}}$ to x' yields a tree whose perturbed cost is equal to $(c^{y^*})^\intercal x' = (c^{y^*})^\intercal x$. Finally, we show that for an optimal LP solution x^*, the "error" $(c^{y^*} - c)^\intercal x^*$ incurred in working with the c^{y^*}-cost is $O(OPT)$; this yields the $\big(O(1), O(1)\big)$-approximation.

We extend the above idea to an arbitrary cost-minimization problem with packing side-constraints as follows. Let x^* be an optimal solution to the LP-relaxation, and $\mathcal{P}$ be the polytope obtained by dropping the packing constraints. We observe that the same Lagrangian-duality based perturbation ensures that all points on the minimal face of $\mathcal{P}$ containing x^* have the same perturbed cost. Therefore, if we have an algorithm for the unweighted problem that rounds x^* to a point $\hat{x}$ on this minimal face, then we again obtain that $(c^{y^*})^\intercal \hat{x} = (c^{y^*})^\intercal x^*$, which then leads to an $\big(O(1), O(1)\big)$-approximation (as in the case of MCCST).

Related work. Whereas node-degree-bounded spanning-tree problems have been widely studied, relatively few results are known for spanning-tree problems with general degree constraints for a family $\mathcal{S}$ of node-sets. With the exception of the result of [14] for unweighted CCST, these other results [3,1,6,2] all yield a tree of cost at most the optimum with an $\omega(1)$ additive- or multiplicative- factor violation of the degree bounds. Both [3] and [2] obtain additive factors via iterative rounding and relaxation. The factor in [3] is $(r-1)$ for an arbitrary $\mathcal{S}$, where r is the maximum number of degree constraints involving an edge (which could be $|V|$ even when $\mathcal{S}$ is a chain), while [2] yields an $O(\log|V|)$ factor when $\mathcal{S}$ is a laminar family (the factor does not improve when $\mathcal{S}$ is a chain). The dependent-rounding techniques in [1,6] yield a tree T satisfying $|\delta_T(S)| \leq \min\{O(\frac{\log|\mathcal{S}|}{\log\log|\mathcal{S}|})b_S, (1+\epsilon)b_S + O(\frac{\log|\mathcal{S}|}{\epsilon})\}$ for all $S \in \mathcal{S}$, for any family $\mathcal{S}$.

For MBDST, Goemans [8] obtained the first $\big(O(1), O(1)\big)$-approximation; his result yields a tree of cost at most the optimum and at most $+2$ violation of the degree bounds. This was subsequently improved to an (optimal) additive $+1$ violation by [16]. Zenklusen [17] considers an orthogonal generalization of MBDST, where there is a matroid-independence constraint on the edges incident to each node, and obtains a tree of cost at most the optimum and "additive" $O(1)$ violation (defined appropriately) of the matroid constraints. To our knowledge, this is the only prior work that obtains an $O(1)$-approximation to both the cost and packing constraints for a constrained spanning-tree problem where an edge participates in $\omega(1)$ packing constraints (albeit this problem is quite different from spanning tree with general degree constraints).

Finally, we note that our Lagrangian-relaxation based technique is somewhat similar to its use in [11]. However, whereas [11] uses this to reduce uniform-degree MBDST to the problem of finding an MST of minimum maximum degree, which is another *weighted* problem, we utilize Lagrangian relaxation in a more refined fashion to reduce the weighted problem to its *unweighted* counterpart.

2 An LP-relaxation for MCCST and preliminaries

We consider the following natural LP-relaxation for MCCST. Throughout, we use e to index the edges of the underlying graph $G = (V, E)$. For a set $S \subseteq V$, let $E(S)$ denote $\{uv \in E : u, v \in S\}$, and $\delta(S)$ denote the edges on the boundary of S. For a vector $z \in \mathbb{R}^E$ and an edge-set F, we use $z(F)$ to denote $\sum_{e \in F} z_e$.

$$\min \quad \sum_e c_e x_e \tag{P}$$

$$\text{s.t.} \quad x\big(E(S)\big) \leq |S| - 1 \qquad \forall \emptyset \neq S \subsetneq V \tag{1}$$

$$x(E) = |V| - 1 \tag{2}$$

$$x\big(\delta(S)\big) \leq b_S \qquad \forall S \in \mathcal{S} \tag{3}$$

$$x \geq 0. \tag{4}$$

For any $x \in \mathbb{R}^E_+$, let $\mathrm{supp}(x) := \{e : x_e > 0\}$ denote the support of x. It is well known that the polytope $\mathrm{P}_{\mathrm{ST}}(G)$ defined by (1), (2), and (4) is the convex hull of spanning trees of G. We call points in $\mathrm{P}_{\mathrm{ST}}(G)$ *fractional spanning trees.* We refer to (1), (2) as the *spanning-tree constraints.* We will also utilize (P_λ), the modified

version of (P) where we replace (3) with $x\big(\delta(S)\big) \le \lambda b_S$ for all $S \in \mathcal{S}$, where $\lambda \ge 1$. Let $OPT(\lambda)$ denote the optimal value of (P_λ), and let $OPT := OPT(1)$.

Preliminaries. A family $\mathcal{L} \subseteq 2^V$ of sets is a *laminar family* if for all $A, B \in \mathcal{L}$, we have $A \subseteq B$ or $B \subseteq A$ or $A \cap B = \emptyset$. As is standard, we say that $A \in \mathcal{L}$ is a child of $L \in \mathcal{L}$ if L is the minimal set of $\mathcal{L}$ such that $A \subsetneq L$. For each $L \in \mathcal{L}$, let $G_L^{\mathcal{L}} = (V_L^{\mathcal{L}}, E_L^{\mathcal{L}})$ be the graph obtained from $\big(L, E(L)\big)$ by contracting the children of L in $\mathcal{L}$; we drop the superscript $\mathcal{L}$ when $\mathcal{L}$ is clear from the context.

Given $x \in \mathrm{P}_{\mathrm{ST}}(G)$, define a *laminar decomposition* $\mathcal{L}$ of x to be a (inclusion-wise) maximal laminar family of sets whose spanning-tree constraints are tight at x, so $x\big(E(A)\big) = |A| - 1$ for all $A \in \mathcal{L}$. Note that $V \in \mathcal{L}$ and $\{v\} \in \mathcal{L}$ for all $v \in V$. A laminar decomposition can be constructed in polytime (using network-flow techniques). For any $L \in \mathcal{L}$, let $x_L^{\mathcal{L}}$, or simply x_L if $\mathcal{L}$ is clear from context, denote x restricted to E_L. Observe that x_L is a fractional spanning tree of G_L. This is trivially true if L has no children; otherwise, for any $\emptyset \ne Q \subseteq V_L$, if R is the subset of V corresponding to Q, and $A_1, \ldots, A_k$ are the children of L whose corresponding contracted nodes lie in Q, we have $x_L\big(E_L(Q)\big) = x\big(E(R)\big) - \sum_{i=1}^k x\big(E(A_i)\big) \le |R \setminus (A_1 \cup \ldots \cup A_k)| + k - 1 = |Q| - 1$ with equality holding when $Q = V_L$.

The following property of laminar decompositions is well known; we include a proof in the Appendix for completeness.

Lemma 2.1. *Let $x \in P_{ST}(G)$ and $\mathcal{L}$ be a laminar decomposition of x. For $F \subseteq E$, let $\chi^{x,F} \in \{0,1\}^{\mathrm{supp}(x)}$ denote the vector with 1s in the coordinates corresponding to F and 0s everywhere else. Let $S \ne \emptyset$ be a set whose spanning-tree constraint is tight at x. Then, $\chi^{x,E(S)}$ is a linear combination of the vectors $\{\chi^{x,E(A)}\}_{A \in \mathcal{L}}$.*

3 An LP-rounding approximation algorithm

3.1 An overview

We first give a high-level overview. Clearly, if (P) is infeasible, there is no spanning tree satisfying the degree constraints, so in the sequel, we assume that (P) is feasible. We seek to obtain a spanning tree T of cost $c(T) = O(OPT)$ such that $|\delta_T(S)| = O(b_S)$ for all $S \in \mathcal{S}$, where $\delta_T(S)$ is the set of edges of T crossing S.

In order to explain the key ideas leading to our algorithm, we first briefly discuss the approach of Olver and Zenklusen [14] for unweighted CCST. Their approach *ignores* the edge costs $\{c_e\}$ and instead starts with a feasible solution x to (P) that minimizes a suitable (linear) potential function. They use this potential function to argue that if $\mathcal{L}$ is a laminar decomposition of x, then $(x, \mathcal{L})$ satisfies a key structural property called *rainbow freeness*. Exploiting this, they give a rounding algorithm, hereby referred to as $\mathcal{A}_{\mathrm{OZ}}$, that for every $L \in \mathcal{L}$, rounds x_L to a spanning tree T_L of G_L such that $|\delta_{T_L}(S)| \approx O\big(x_L(\delta(S))\big)$ for all $S \in \mathcal{S}$, so that concatenating the T_Ls yields a spanning tree T of G satisfying $|\delta_T(S)| = O\big(x(\delta(S))\big) = O(b_S)$ for all $S \in \mathcal{S}$ (Theorem 3.2). However, as already noted in [14], a fundamental obstacle towards generalizing their approach to handle the weighted version (i.e., MCCST) is that in order to achieve rainbow freeness, which is crucial for their rounding algorithm, one needs to *abandon the cost function c and work with an alternate potential function.*

We circumvent this difficulty as follows. First, we note that the algorithm in [14] can be equivalently viewed as rounding an *arbitrary* solution x to (P) as follows.

Let $\mathcal{L}$ be a laminar decomposition of x. Using the same potential-function idea, we can convert x to another solution x' to (P) that admits a laminar decomposition $\mathcal{L}'$ refining $\mathcal{L}$ such that $(x', \mathcal{L}')$ satisfies rainbow freeness (see Lemma 3.1), and then round x' using $\mathcal{A}_{\mathrm{OZ}}$. Of course, the difficulty noted above remains, since the move to rainbow freeness (which again ignores c and uses a potential function) does not yield *any* bounds on the cost $c^{\intercal}x'$ relative to $c^{\intercal}x$.

Now let x^* be an optimal solution to (P) (we will later modify this somewhat) and $\mathcal{L}$ be a laminar decomposition of x^*. The crucial insight, and a key notable aspect of our algorithm and analysis, is that *one can leverage Lagrangian duality to suitably perturb the edge costs so that x^*, the solution x' (satisfying rainbow-freeness) obtained from x^*, and the spanning tree T obtained by rounding x' using $\mathcal{A}_{OZ}$, all have the same perturbed cost.* We achieve this as follows. Let $y^* \in \mathbb{R}_+^{\mathcal{S}}$ denote the optimal values of the dual variables corresponding to constraints (3). If we define the perturbed cost of edge e to be $c_e^{y^*} := c_e + \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S^*$, then using standard duality theory, it follows that x^* *is a fractional spanning tree of minimum c^{y^*}-cost* (Lemma 3.4). Now notice that x' and the tree T satisfy all the spanning-tree constraints for the sets in $\mathcal{L}$ with equality: the laminar-decomposition $\mathcal{L}'$ of x' refines $\mathcal{L}$, and $\mathcal{A}_{\mathrm{OZ}}$ rounds x'_L to a spanning tree of G_L, for every $L \in \mathcal{L}'$. Therefore, by Lemma 2.1, *x', and the characteristic vector of T, lie on the minimal face of $P_{ST}(G)$ containing x^*.* This has the implication that, since the optimal solutions to an LP form a face of its feasible region, x' and T are also min c^{y^*}-cost (fractional) spanning trees, and so $(c^{y^*})^{\intercal}x^* = (c^{y^*})^{\intercal}x' = c^{y^*}(T)$.

Since T is the output of $\mathcal{A}_{\mathrm{OZ}}$ run on x', which is a feasible solution to (P), it immediately satisfies $|\delta_T(S)| = O(b_S)$ for all $S \in \mathcal{S}$. To bound the cost, using complementary slackness, it is easy to argue that $c(T) \leq c^{y^*}(T) = \sum_e c_e^{y^*} x_e^* = c^{\intercal}x^* + \sum_{S \in \mathcal{S}} b_S y_S^*$ (Lemma 3.6). (Note that the perturbed costs are used only in the analysis.) However, $\sum_{S \in \mathcal{S}} b_S y_S^*$ need not be bounded in terms of $c^{\intercal}x^*$. To fix this, we modify our starting solution x^*: we solve (P_λ) (which recall is (P) with inflated degree bounds $\{\lambda b_S\}$), where $\lambda > 1$, to obtain x^*, and use this x^* in our algorithm. Now, letting y^* be the optimal dual values corresponding to the inflated degree constraints, a simple duality argument shows that $\sum_{S \in \mathcal{S}} b_S y_S^* \leq \frac{OPT(1) - OPT(\lambda)}{\lambda - 1}$ (Lemma 3.8), which yields $c(T) = O(OPT)$ (see Theorem 3.3).

A noteworthy feature of our algorithm is the rather unconventional use of Lagrangian relaxation, where we use duality to simplify the *cost structure* (as opposed to the constraint-structure) of the underlying problem by equalizing the costs of certain solutions. This, coupled with the face-preserving nature of $\mathcal{A}_{\mathrm{OZ}}$turn out to be the crucial ingredients that allow us to utilize the algorithm $\mathcal{A}_{\mathrm{OZ}}$for unweighted CCST *as a black box* without worrying about the difficulties posed by (the move to) rainbow freeness. In fact, as we show in Section 4, this idea of exploiting Lagrangian relaxation via a face-preserving rounding algorithm (see Definition 4.1) is applicable more generally, and yields a novel reduction from weighted problems to their unweighted counterparts. We believe that this reduction is of independent interest and will find use in other settings as well.

3.2 Algorithm details and analysis

To specify our algorithm formally, we first define the rainbow-freeness property and state the main result of [14] (which we utilize as a black box) precisely.

For an edge e, define $\mathcal{S}_e := \{S \in \mathcal{S} : e \in \delta(S)\}$. Note that $\mathcal{S}_e$ could be empty. We say that two edges $e, f \in E$ form a *rainbow* if $\mathcal{S}_e \subseteq \mathcal{S}_f$ or $\mathcal{S}_f \subseteq \mathcal{S}_e$. (This definition is slightly different from the one in [14], in that we allow $\mathcal{S}_e = \mathcal{S}_f$.) We say that $(x, \mathcal{L})$ is a *rainbow-free decomposition* if $\mathcal{L}$ is a laminar decomposition of x and for every set $L \in \mathcal{L}$, no two edges of $\mathrm{supp}(x) \cap E_L$ form a rainbow. (Recall that $G_L = (V_L, E_L)$ denotes the graph obtained from $\big(L, E(L)\big)$ by contracting the children of L.) The following lemma shows that one can convert an arbitrary decomposition $(x, \mathcal{L})$ to a rainbow-free one; we defer the proof to the Appendix. (As noted earlier, this lemma is used to equivalently view the algorithm in [14] as a rounding algorithm that rounds an arbitrary solution x to (P).)

Lemma 3.1. *Let $x \in P_{ST}(G)$ and $\mathcal{L}$ be a laminar decomposition of x. We can compute in polytime a fractional spanning tree $x' \in P_{ST}(G)$ and a rainbow-free decomposition $(x', \mathcal{L}')$ such that: (i) $\mathrm{supp}(x') \subseteq \mathrm{supp}(x)$; (ii) $\mathcal{L} \subseteq \mathcal{L}'$; and (iii) $x'(\delta(S)) \leq x(\delta(S))$ for all $S \in \mathcal{S}$.*

Theorem 3.2 ([14])**.** *There is a polytime algorithm, $\mathcal{A}_{OZ}$, that given a fractional spanning tree $x' \in P_{ST}(G)$ and a rainbow-free decomposition $(x', \mathcal{L}')$, returns a spanning tree $T_L \subseteq \mathrm{supp}(x')$ of G_L for every $L \in \mathcal{L}'$ such that the concatenation T of the T_Ls is a spanning tree of G satisfying $|\delta_T(S)| \leq 9x'\big(\delta(S)\big)$ for all $S \in \mathcal{S}$.*

We can now describe our algorithm compactly. Let $\lambda > 1$ be a parameter.

1. Compute an optimal solution x^* to (P_λ), a laminar decomposition $\mathcal{L}$ of x^*.
2. Apply Lemma 3.1 to $(x^*, \mathcal{L})$ to obtain a rainbow-free decomposition $(x', \mathcal{L}')$.
3. Apply $\mathcal{A}_{\mathrm{OZ}}$ to the input $(x', \mathcal{L}')$ to obtain spanning trees $T_L^{\mathcal{L}'}$ of $G_L^{\mathcal{L}'}$ for every $L \in \mathcal{L}'$. Return the concatenation T of all the $T_L^{\mathcal{L}'}$s.

Analysis. We show that the above algorithm satisfies the following guarantee.

Theorem 3.3. *The above algorithm run with parameter $\lambda > 1$ returns a spanning tree T satisfying $c(T) \leq \frac{\lambda}{\lambda-1} \cdot OPT$ and $|\delta_T(S)| \leq 9\lambda b_S$ for all $S \in \mathcal{S}$.*

Let $\chi^T \in \{0,1\}^E$ denote the characteristic vector of T. The bound on $|\delta_T(S)|$ follows immediately from Lemma 3.1 and Theorem 3.2 since x^*, and hence x' obtained in step 2, is a feasible solution to (P_λ). So we focus on bounding $c(T)$. This will follow from three things. First, we define the perturbed c^{y^*}-cost precisely. Next, we argue that x^* has minimum c^{y^*}-cost among all points $x \in \mathrm{P_{ST}}(G)$ (Lemma 3.4). Then, we observe that x' and χ^T lie on the minimal face of $\mathrm{P_{ST}}(G)$ containing x^* (Claim 3.5). Using this it is easy to show that $c(T) \leq c^{y^*}(T) = \sum_e c_e^{y^*} x_e^* = OPT(\lambda) + \lambda \sum_{S \in \mathcal{S}} b_S y_S^*$ (Lemma 3.6). Finally, we show that $\sum_{S \in \mathcal{S}} b_S y_S^* \leq \big(OPT - OPT(\lambda)\big)/(\lambda - 1)$ (Lemma 3.8), which yields the bound stated in Theorem 3.3.

To define the perturbed costs, we consider the Lagrangian dual of (P_λ) obtained by dualizing the (inflated) degree constraints $x\big(\delta(S)\big) \leq \lambda b_S$ for all $S \in \mathcal{S}$:

$$\max_{y \in \mathbb{R}_+^{\mathcal{S}}} \quad \Big(g_\lambda(y) \quad := \quad \min_{x \in \mathrm{P_{ST}}(G)} \Big(\sum_e c_e x_e + \sum_{S \in \mathcal{S}} \big(x(\delta(S)) - \lambda b_S \big) y_S \Big) \Big). \qquad (\mathrm{LD}_\lambda)$$

For $y \in \mathbb{R}^{\mathcal{S}}$, let $\mathcal{G}_{\lambda,y}(x) := \sum_e c_e x_e + \sum_{S\in\mathcal{S}}\big(x(\delta(S)) - \lambda b_S\big)y_S = \sum_e c_e^y x_e - \lambda\sum_{S\in\mathcal{S}} b_S y_S$ denote the objective function of the LP that defines $g_\lambda(y)$, where $c_e^y := c_e + \sum_{S\in\mathcal{S}:e\in\delta(S)} y_S$.

Let y^* be an optimal solution to (LD_λ). Our *perturbed costs* are $\{c_e^{y^*}\}_e$. The following lemma follows from standard results in duality theory. We include a proof for completeness.

Lemma 3.4. *We have $g_\lambda(y^*) = \mathcal{G}_{\lambda,y^*}(x^*) = OPT(\lambda)$.*

Proof. For any $y \in \mathbb{R}^{\mathcal{S}}_+$, we have $g_\lambda(y) + \lambda\sum_{S\in\mathcal{S}} b_S y_S =$

$$\underbrace{\Big(\min \sum_e c_e^y x_e \quad \text{s.t.} \quad x\big(E(S)\big) \le |S|-1 \;\; \forall \emptyset \ne S \subsetneq V, \quad x(E) = |V|-1, \quad x \ge 0\Big)}_{(\mathrm{P}_{\lambda,y})} =$$

$$\underbrace{\Big(\max -\sum_{\emptyset\ne S\subseteq V}(|S|-1)\mu_S \quad \text{s.t.} \quad -\sum_{\substack{\emptyset\ne S\subseteq V:\\ e\in E(S)}} \mu_S \le c_e^y \;\; \forall e\in E, \quad \mu_S \ge 0 \;\; \forall \emptyset\ne S\subsetneq V\Big)}_{(\mathrm{D}_{\lambda,y})}$$

where the second equality follows since $(\mathrm{D}_{\lambda,y})$ is the dual of $(\mathrm{P}_{\lambda,y})$. It follows that

$$\begin{aligned} g_\lambda(y^*) \;=\; \max_{y\in\mathbb{R}^{\mathcal{S}}_+} g_\lambda(y) \;=\; \max \; & -\sum_{\emptyset\ne S\subseteq V}(|S|-1)\mu_S - \lambda\sum_{S\in\mathcal{S}} b_S y_S && (\mathrm{D}_\lambda)\\ \text{s.t.} \quad & -\sum_{\substack{\emptyset\ne S\subseteq V:\\ e\in E(S)}}\mu_S - \sum_{\substack{S\in\mathcal{S}:\\ e\in\delta(S)}} y_S \le c_e && \forall e\in E\\ & y\ge 0, \quad \mu_S \ge 0 && \forall \emptyset\ne S\subsetneq V. \end{aligned}$$

Notice that (D_λ) is the dual of (P_λ), hence, $g_\lambda(y^*) = OPT(\lambda)$. Moreover, it also follows that $\hat{y}$ is an optimal solution to (LD_λ) iff there exists $\hat{\mu} = (\hat{\mu}_S)_{\emptyset\ne S\subseteq V}$ such that $(\hat{\mu},\hat{y})$ is an optimal solution to (D_λ).

So let μ^* be such that (μ^*, y^*) is an optimal solution to (D_λ). It follows that x^* and (μ^*, y^*) satisfy complementary slackness. So we have that if $\mu_S^* > 0$ then $x^*\big(E(S)\big) = |S|-1$, and if $x_e^* > 0$ then $-\sum_{\emptyset\ne S\subseteq V:e\in E(S)}\mu_S^* - \sum_{S\in\mathcal{S}:e\in\delta(S)} y_S^* = c_e$, or equivalently $c_e^{y^*} = -\sum_{\emptyset\ne S\subseteq V:e\in E(S)}\mu_S^*$. Therefore,

$$\begin{aligned} \mathcal{G}_{\lambda,y^*}(x^*) &= \sum_e c_e^{y^*} x_e^* - \lambda\sum_{S\in\mathcal{S}} b_S y_S^* = \sum_e\Big(-\sum_{\emptyset\ne S\subseteq V:e\in E(S)}\mu_S^*\Big)x_e^* - \lambda\sum_{S\in\mathcal{S}} b_S y_S^*\\ &= -\sum_{\emptyset\ne S\subseteq V}\mu_S^* x^*\big(E(S)\big) - \lambda\sum_{S\in\mathcal{S}} b_S y_S^*\\ &= -\sum_{\emptyset\ne S\subseteq V}(|S|-1)\mu_S^* - \lambda\sum_{S\in\mathcal{S}} b_S y_S^* = g_\lambda(y^*). \end{aligned}$$

□

Claim 3.5. *The points x' and χ^T lie on the minimal face of $P_{ST}(G)$ containing x^*.*

Proof. We have $\mathrm{supp}(x'), \mathrm{supp}(\chi^T) \subseteq \mathrm{supp}(x^*)$. We need to show that any spanning-tree constraint that is tight at x^* is also tight at x' and χ^T. By Lemma 2.1, to prove this, it suffices to show this for the (tight) spanning-tree constraints corresponding to the sets in the laminar decomposition $\mathcal{L}$ of x^*. The laminar decomposition $\mathcal{L}'$ of x' refines $\mathcal{L}$, so this is true by definition for x'. The spanning tree T is obtained by concatenating spanning trees $T_L^{\mathcal{L}'}$ for each $L \in \mathcal{L}'$. So, for each $L \in \mathcal{L}'$, $T \cap E(L)$ is a spanning tree of $\big(L, E(L)\big)$ and hence, $\chi^T\big(E(L)\big) = |L| - 1$. □

Lemma 3.6. *We have* $c(T) \le c^{y^*}(T) = (c^{y^*})^\intercal x' = (c^{y^*})^\intercal x^* = \sum_e c_e x_e^* + \lambda \sum_{S \in \mathcal{S}} b_S y_S^*$.

Proof. We have $c(T) \le c^{y^*}(T)$ since $c_e \le c_e^{y^*}$ for all $e \in E$ as $y^* \ge 0$. The set of optimal solutions to the LP $\min_{x \in \mathrm{P_{ST}}(G)} \mathcal{G}_{\lambda, y^*}(x)$ is a face of $\mathrm{P_{ST}}(G)$; thus *all* points on the *minimal* face of $\mathrm{P_{ST}}(G)$ containing x^* are optimal solutions to this LP. Therefore, by Claim 3.5, χ^T and x' are also optimal solutions to this LP, and hence, $c^{y^*}(T) = (c^{y^*})^\intercal x' = (c^{y^*})^\intercal x^*$. Finally,

$$\begin{aligned} c^{y^*}(T) &= \sum_e c_e^{y^*} x_e^* = \sum_e \Big(c_e x_e^* + \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S^* x_e^* \Big) \\ &= \sum_e c_e x_e^* + \sum_{S \in \mathcal{S}} y_S^* x^*\big(\delta(S)\big) = \sum_e c_e x_e^* + \lambda \sum_{S \in \mathcal{S}} b_S y_S^*. \end{aligned}$$ □

Remark 3.7. The property that all points on the the minimal face, say Q, of $\mathrm{P_{ST}}(G)$ containing x^* have the same c^{y^*}-cost is equivalent to the property that for every $L \in \mathcal{L}$, all edges in $\mathrm{supp}(x^*) \cap E_L$ have the same c^{y^*}-cost. If the latter holds, then $(c^{y^*})^\intercal x = (c^{y^*})^\intercal x^*$ for all $x \in Q$, since we have $\sum_e c_e^{y^*} z_e = \sum L \in \mathcal{L} \sum_{e \in E_L} c_e^{y^*} z_e$ for any $z \in \mathrm{P_{ST}}(G)$ with $\mathrm{supp}(z) \subseteq \mathrm{supp}(x^*)$. Conversely, suppose for some $L \in \mathcal{L}$ and $e, f \in \mathrm{supp}(x^*) \cap E_L$, we have $c_e^{y^*} < c_f^{y^*}$. Then, one can obtain a point x'' by shifting some $\epsilon > 0$ weight from x_f^* to x_e^*, so that $(c^{y^*})^\intercal x'' < (c^{y^*})^\intercal x^*$. For a sufficiently small $\epsilon > 0$, we can argue that $x'' \in \mathrm{P_{ST}}(G)$, and hence, $x'' \in Q$ by the same arguments as in the proof of Claim 3.5.

Lemma 3.8. *We have* $\sum_{S \in \mathcal{S}} b_S y_S^* \le \big(OPT(1) - OPT(\lambda)\big)/(\lambda - 1)$.

Proof. By Lemma 3.4, we have that

$$OPT(\lambda) = g_\lambda(y^*) = \mathcal{G}_{\lambda, y^*}(x^*).$$

Using Lemma 3.4 again, now with $\lambda = 1$, and since y^* is a feasible solution to (LD_1), we obtain that $OPT(1) = \max_{y \in \mathbb{R}_+^{\mathcal{S}}} g_1(y) \ge g_1(y^*)$. Note that the objective functions of the LPs defining $g_1(y^*)$ and $g_\lambda(y^*)$ differ by a constant: $\mathcal{G}_{1,y^*}(x) - \mathcal{G}_{\lambda,y^*}(x) = (\lambda - 1) \sum_{S \in \mathcal{S}} b_S y_S^*$ for all $x \in \mathrm{P_{ST}}(G)$. Since x^* is an optimal solution to $\min_{x \in \mathrm{P_{ST}}(G)} \mathcal{G}_{\lambda,y^*}(x)$, it is also an optimal solution to $\min_{x \in \mathrm{P_{ST}}(G)} \mathcal{G}_{1,y^*}(x)$. It follows that

$$OPT(1) \ge g_1(y^*) = \mathcal{G}_{1,y^*}(x^*) \ .$$

Therefore, $OPT(1) - OPT(\lambda) \ge \mathcal{G}_{1,y^*}(x^*) - \mathcal{G}_{\lambda,y^*}(x^*) = (\lambda - 1) \sum_{S \in \mathcal{S}} b_S y_S^*$. □

Proof of Theorem 3.3. As noted earlier, the bounds on $\delta_T(S)$ follow immediately from Lemma 3.1 and Theorem 3.2: for any $S \in \mathcal{S}$, we have $|\delta_T(S)| \le 9x'\big(\delta(S)\big) \le 9x^*\big(\delta(S)\big) \le 9\lambda b_S$. The bound on $c(T)$ follows from Lemmas 3.6 and 3.8 since $\sum_e c_e x_e^* = OPT(\lambda)$. □

4 A reduction from weighted to unweighted problems

We now show that our ideas are applicable more generally, and yield bicriteria approximation algorithms for any cost-minimization problem with packing side-constraints, provided we have a suitable approximation algorithm for the *unweighted* counterpart. Thus, our technique yields a *reduction* from weighted to unweighted problems, which we believe is of independent interest.

To demonstrate this, we first isolate the key properties of the rounding algorithm $\mathcal{B}$ used above for unweighted CCST that enable us to use it as a black box to obtain our result for MCCST. Note that $\mathcal{B}$ is obtained by combining the procedure in Lemma 3.1 and $\mathcal{A}_{\mathrm{OZ}}$ (Theorem 3.2). First, we of course utilize that $\mathcal{B}$ is an approximation algorithm for unweighted CCST, so it returns a spanning tree T such that $|\delta_T(S)| = O\big(x^*(\delta(S))\big)$ for all $S \in \mathcal{S}$. Second, we exploit the fact that $\mathcal{B}$ returns a tree T that *preserves tightness of all spanning-tree constraints that are tight at* x^*. *This is the crucial property that allows us to bound* $c(T)$, since this implies (as shown in Lemma 3.6) that $c^{y^*}(T) = \sum_e c_e^{y^*} x_e^*$, which then yields the bound on $c(T)$. In other words, while $\mathcal{A}_{\mathrm{OZ}}$ proceeds by exploiting the notions of rainbow freeness and laminar decomposition, these notions are not essential to obtaining our result; *any* rounding algorithm for unweighted CCST satisfying the above two properties can be utilized to obtain our guarantee for MCCST.

We now formalize the above two properties for an arbitrary cost-minimization problem with packing side-constraints, and prove that they suffice to yield a bicriteria guarantee. Consider the following abstract problem, where $\mathcal{P} \subseteq \mathbb{R}_+^n$ is a fixed polytope: given $c \in \mathbb{R}_+^n$, $A \in \mathbb{R}_+^{m \times n}$, and $b \in \mathbb{R}_+^m$, find

$$\min \quad c^\intercal x \quad \text{s.t.} \quad x \text{ is an extreme point of } \mathcal{P}, \quad Ax \le b. \tag{$\mathrm{Q}^{\mathcal{P}}$}$$

Observe that we can cast MCCST as a special case of $(\mathrm{Q}^{\mathcal{P}})$, by taking $\mathcal{P} = \mathrm{P}_{\mathrm{ST}}(G)$ (whose extreme points are spanning trees of G), c to be the edge costs, and $Ax \le b$ to be the degree constraints. Moreover, by taking $\mathcal{P}$ to be the convex hull of a bounded set $\{x \in \mathbb{Z}_+^n : Cx \le d\}$ we can use $(\mathrm{Q}^{\mathcal{P}})$ to encode a discrete-optimization problem.

Definition 4.1. We say that $\mathcal{B}$ is a *face-preserving rounding algorithm* (FPRA) for unweighted $(\mathrm{Q}^{\mathcal{P}})$ if given any point $x \in \mathcal{P}$, $\mathcal{B}$ finds in polytime an extreme point $\hat{x}$ of $\mathcal{P}$ such that:
(P1) $\hat{x}$ belongs to the minimal face of $\mathcal{P}$ that contains x.
We say that $\mathcal{B}$ is a *β-approximation FPRA* (where $\beta \ge 1$) if we *also* have:
(P2) $A\hat{x} \le \beta A x$.

Let $(\mathrm{R}_\lambda^{\mathcal{P}})$ denote the LP $\min\big\{c^\intercal x : x \in \mathcal{P},\ Ax \le \lambda b\big\}$; note that $(\mathrm{R}_1^{\mathcal{P}})$ is the LP-relaxation of $(\mathrm{Q}^{\mathcal{P}})$. Let $\mathsf{opt}(\lambda)$ denote the optimal value of $(\mathrm{R}_\lambda^{\mathcal{P}})$, and let $\mathsf{opt} := \mathsf{opt}(1)$. We say that an algorithm is a (ρ_1, ρ_2)-approximation algorithm for $(\mathrm{Q}^{\mathcal{P}})$ if it finds in polytime an extreme point $\hat{x}$ of $\mathcal{P}$ such that $c^\intercal \hat{x} \le \rho_1 \mathsf{opt}$ and $A\hat{x} \le \rho_2 b$.

Theorem 4.2. *Let $\mathcal{B}$ be a β-approximation FPRA for unweighted $(\mathrm{Q}^{\mathcal{P}})$. Then, given any $\lambda > 1$, one can obtain a $\big(\frac{\lambda}{\lambda-1}, \beta\lambda\big)$-approximation algorithm for $(\mathrm{Q}^{\mathcal{P}})$ using a single call to $\mathcal{B}$.*

Proof sketch. We dovetail the algorithm for MCCST and its analysis. We simply compute an optimal solution x^* to $(\mathrm{R}_\lambda^{\mathcal{P}})$ and round it to an extreme point $\hat{x}$ of $\mathcal{P}$ using $\mathcal{B}$. By property (P2), it is clear that $A\hat{x} \le \beta(Ax^*) \le \beta\lambda b$.

For $y \in \mathbb{R}_+^m$, define $c^y := c + A^\intercal y$. To bound the cost, as before, we consider the Lagrangian dual of $(\mathrm{R}_\lambda^{\mathcal{P}})$ obtained by dualizing the side-constraints $Ax \le \lambda b$.

$$\max_{y\in\mathbb{R}_+^m}\Big(h_\lambda(y) \;:=\; \min_{x\in\mathcal{P}} \mathcal{H}_{\lambda,y}(x)\Big), \quad \text{where } \mathcal{H}_{\lambda,y}(x) := (c^y)^\intercal x - \lambda y^\intercal b.$$

Let $y^* = \operatorname{argmax}_{y\in\mathbb{R}_+^m} h_\lambda(y)$. We can mimic the proof of Lemma 3.4 to show that x^* is an optimal solution to $\min_{x\in\mathcal{P}} \mathcal{H}_{\lambda,y^*}(x)$. The set of optimal solutions to this LP is a face of $\mathcal{P}$. So all points on the minimal face of $\mathcal{P}$ containing x^* are optimal solutions to this LP. By property (P1), $\hat{x}$ belongs to this minimal face and so is an optimal solution to this LP. So $(c^{y^*})^\intercal \hat{x} = (c^{y^*})^\intercal x^* = c^\intercal x^* + (y^*)^\intercal A x^* = \mathsf{opt}(\lambda) + \lambda (y^*)^\intercal b$, where the last equality follows by complementary slackness. Also, by the same arguments as in Lemma 3.8, we have $(y^*)^\intercal b \le \frac{\mathsf{opt}(1)-\mathsf{opt}(\lambda)}{\lambda-1}$. Since $c \le c^{y^*}$, we have $c^\intercal \hat{x} \le (c^{y^*})^\intercal \hat{x} \le \frac{\lambda}{\lambda-1}\cdot \mathsf{opt}$. □

5 Towards a $(1, O(1))$-approximation algorithm for $(\mathrm{Q}^{\mathcal{P}})$

A natural question that emerges from Theorems 3.3 and 4.2 is whether one can obtain a $(1, O(1))$-approximation, i.e., obtain a solution of *cost at most* opt that violates the packing side-constraints by an (multiplicative) $O(1)$-factor. Such results are known for degree-bounded spanning tree problems with various kinds of degree constraints [8,16,3,17], so, in particular, it is natural to ask whether such a result also holds for MCCST. (Note that for MCCST, the dependent-rounding techniques in [1,6] yield a tree T with $c(T) \le OPT$ and $|\delta_T(S)| \le \min\{O(\frac{\log|\mathcal{S}|}{\log\log|\mathcal{S}|})b_S, (1+\epsilon)b_S + O(\frac{\log|\mathcal{S}|}{\epsilon})\}$ for all $S \in \mathcal{S}$, while the iterative-rounding algorithm in [2] yields $c(T) \le OPT$ and $|\delta_T(S)| \le b_S + O(\log|V|)$ for all $S \in \mathcal{S}$.) We show that our approach is versatile enough to yield such a guarantee provided we assume a stronger property from the rounding algorithm $\mathcal{B}$ for unweighted $(\mathrm{Q}^{\mathcal{P}})$.

Let A_i denote the i-th row of A, for $i = 1, \ldots, m$. We say that $\mathcal{B}$ is an (α,β)-*approximation FPRA* for unweighted $(\mathrm{Q}^{\mathcal{P}})$ if *in addition* to properties (P1) and (P2), it satisfies:

(P3) it rounds a feasible solution x to $(\mathrm{R}_\alpha^{\mathcal{P}})$ to an extreme point $\hat{x}$ of $\mathcal{P}$ satisfying $A_i^\intercal \hat{x} \ge \frac{A_i^\intercal x}{\alpha}$ for every i such that $A_i^\intercal x = \alpha b_i$.

(For MCCST, property (P3) requires that $|\delta_T(S)| \ge b_S/\alpha$ for every set $S \in \mathcal{S}$ whose degree constraint (in (P_α)) is tight at the fractional spanning tree x.)

Theorem 5.1. *Let $\mathcal{B}$ be an (α,β)-approximation FPRA for unweighted $(\mathrm{Q}^{\mathcal{P}})$. Then, one can obtain a $(1,\alpha\beta)$-approximation algorithm for $(\mathrm{Q}^{\mathcal{P}})$ using a single call to $\mathcal{B}$.*

Proof. We show that applying the algorithm from Theorem 4.2 with $\lambda = \alpha$ yields the claimed result. It is clear that the extreme point $\hat{x}$ returned satisfies $A\hat{x} \le \alpha\beta b$. As in the proof of Theorem 4.2, let y^* be an optimal solution to $\max_{y\in\mathbb{R}_+^m} h_\lambda(y)$ (where $\lambda = \alpha$). In Lemma 3.6 and the proof of Theorem 4.2, we use the weak

bound $c^\intercal \hat{x} \leq (c^{y^*})^\intercal \hat{x}$. We tighten this to obtain the improved bound on $c^\intercal \hat{x}$. We have $c^\intercal \hat{x} = (c^{y^*})^\intercal \hat{x} - (y^*)^\intercal A\hat{x}$, and

$$(y^*)^\intercal A\hat{x} = \sum_{i:A_i^\intercal x^* = \lambda b_i} y_i^*(A_i^\intercal \hat{x}) \geq \sum_{i:A_i^\intercal x^* = \lambda b_i} \frac{y_i^* A_i^\intercal x^*}{\alpha} = \sum_{i:A_i^\intercal x^* = \lambda b_i} y_i^* b_i = (y^*)^\intercal b.$$

The first and last equalities above follow because $y_i^* > 0$ implies that $A_i^\intercal x^* = \lambda b_i$. The inequality follows from property (P3). Thus, following the rest of the arguments in the proof of Theorem 4.2, we obtain that

$$c^\intercal \hat{x} \leq (c^{y^*})^\intercal \hat{x} - (y^*)^\intercal b = c^\intercal x^* + (\lambda - 1)(y^*)^\intercal b \leq \mathsf{opt}(1). \qquad \square$$

5.1 Obtaining an additive approximation for $(\mathrm{Q}^{\mathcal{P}})$ and cost at most opt via an FPRA with two-sided additive guarantees

We now present a variant of Theorem 5.1 that shows that we can achieve cost at most opt and additive approximation for the packing side constraints using an FPRA with two-sided *additive* guarantees. We give an application of this result in Section 5.2, where we utilize it to obtain improved guarantees for the k-budgeted matroid basis problem.

Theorem 5.2. *Let $\mathcal{B}$ be an FPRA for unweighted $(\mathrm{Q}^{\mathcal{P}})$ that given $x \in \mathcal{P}$ returns an extreme point $\hat{x}$ of $\mathcal{P}$ such that $Ax - \Delta \leq A\hat{x} \leq Ax + \Delta$, where $\Delta \in \mathbb{R}_+^m$ may depend on A and c (but not on b). Using a single call to $\mathcal{B}$, we can obtain an extreme point $\tilde{x}$ of $\mathcal{P}$ such that $c^\intercal \tilde{x} \leq \mathsf{opt}$ and $A\tilde{x} \leq b + 2\Delta$.*

The above result is obtained via essentially the same arguments as those in Theorems 4.2 and 5.1. For a vector $\Delta \in \mathbb{R}_+^m$, let $(\mathrm{W}_\Delta^{\mathcal{P}})$ denote the LP $\min\{c^\intercal x : x \in \mathcal{P},\ Ax \leq b + \Delta\}$. Let $\mathbf{0}$ denote the all-zeros vector, and note that $(\mathrm{W}_{\mathbf{0}}^{\mathcal{P}})$ is the LP-relaxation of $(\mathrm{Q}^{\mathcal{P}})$. Let $\mathsf{opt}(\Delta)$ denote the optimum value of $(\mathrm{W}_\Delta^{\mathcal{P}})$, and let $\mathsf{opt} := \mathsf{opt}(\mathbf{0})$. The Lagrangian dual of $(\mathrm{W}_\Delta^{\mathcal{P}})$ obtained by dualizing the side-constraints $Ax \leq b + \Delta$ is

$$\max_{y \in \mathbb{R}_+^m} \Big(\varphi_\Delta(y) \ := \ \min_{x \in \mathcal{P}} \Phi_{\Delta,y}(x) \Big), \tag{LD_Δ}$$

where $\Phi_{\Delta,y}(x) := (c^y)^\intercal x - y^\intercal(b + \Delta)$. (Recall that $c^y := c + A^\intercal y$.) Let x^* be an optimal solution to $(\mathrm{W}_\Delta^{\mathcal{P}})$ and $y^* = \operatorname{argmax}_{y \in \mathbb{R}_+^m} \varphi_\Delta(y)$. We have the following variants of Lemmas 3.4 and 3.8.

Lemma 5.3. *We have $\varphi_\Delta(y^*) = \Phi_{\Delta,y^*}(x^*) = \mathsf{opt}(\Delta)$.*

Proof. This follows from the arguments used in the proof of Lemma 3.4. $\square$

Lemma 5.4. *We have $(y^*)^\intercal \Delta \leq \mathsf{opt}(\mathbf{0}) - \mathsf{opt}(\Delta)$.*

Proof. We mimic the proof of Lemma 3.8. By Lemma 5.3, we have that

$$\mathsf{opt}(\Delta) = \varphi_\Delta(y^*) = \Phi_{\Delta,y^*}(x^*)$$

and $\mathsf{opt}(\mathbf{0}) = \max_{y \in \mathbb{R}_+^S} \varphi_{\mathbf{0}}(y) \geq \varphi_{\mathbf{0}}(y^*)$. Note that $\Phi_{\mathbf{0},y^*}(x) - \Phi_{\Delta,y^*}(x) = (y^*)^\intercal \Delta$, which is independent of x. So since x^* is an optimal solution to $\min_{x \in \mathcal{P}} \Phi_{\Delta,y^*}(x)$, it is also an optimal solution to $\min_{x \in \mathcal{P}} \Phi_{\mathbf{0},y^*}(x)$. It follows that

$$\mathsf{opt}(\mathbf{0}) \geq \varphi_{\mathbf{0}}(y^*) = \Phi_{\mathbf{0},y^*}(x^*).$$

Hence, $\mathsf{opt}(\mathbf{0}) - \mathsf{opt}(\Delta) \geq \Phi_{\mathbf{0},y^*}(x^*) - \Phi_{\Delta,y^*}(x^*) = (y^*)^\intercal \Delta$. □

Proof of Theorem 5.2. The algorithm simply computes an optimal solution x^* to $(\mathrm{W}_\Delta^{\mathcal{P}})$, and rounds it to an extreme point $\tilde{x}$ of $\mathcal{P}$ using algorithm $\mathcal{B}$.

It is clear that $A\tilde{x} \leq Ax^* + \Delta \leq (b + \Delta) + \Delta = b + 2\Delta$. Next we argue that $c^\intercal \tilde{x} \leq \mathsf{opt}$. We have $c^\intercal \tilde{x} = (c^{y^*})^\intercal \tilde{x} - (y^*)^\intercal A\tilde{x}$, and

$$\begin{aligned}(y^*)^\intercal A\tilde{x} &= \sum_{i: A_i^\intercal x^* = b_i + \Delta_i} y_i^* (A_i^\intercal \tilde{x}) \geq \sum_{i: A_i^\intercal x^* = b_i + \Delta_i} y_i^* (A_i^\intercal x^* - \Delta_i) \\ &= \sum_{i: A_i^\intercal x^* = b_i + \Delta_i} y_i^* b_i = (y^*)^\intercal b.\end{aligned}$$

By Lemma 5.3, x^* is an optimal solution to $\min_{x \in \mathcal{P}} \Psi_{\Delta,y^*}(x)$. So all points on the minimal face of $\mathcal{P}$ containing x^* are optimal solutions to this LP. In particular, since $\tilde{x}$ belongs to this minimal face (by property (P1)), $\tilde{x}$ is an optimal solution to this LP. This observation, along with the inequality above, yields $c^\intercal \tilde{x} \leq (c^{y^*})^\intercal x^* - (y^*)^\intercal b = \mathsf{opt}(\Delta) + (y^*)^\intercal \Delta$. Finally, using Lemma 5.4 yields $c^\intercal \tilde{x} \leq \mathsf{opt}(\mathbf{0})$ as required. □

5.2 Application to k-budgeted matroid basis

Here, we seek to find a basis S of a matroid $M = (U, \mathcal{I})$ satisfying k budget constraints $\{d_i(S) \leq B_i\}_{1 \leq i \leq k}$, where $d_i(S) := \sum_{e \in S} d_i(e)$. Note that this can be cast as a special case of $(\mathrm{Q}^{\mathcal{P}})$, where $\mathcal{P} = \mathcal{P}(M)$ is the basis polytope of M, the objective function encodes a chosen budget constraint (say the k-th budget constraint), and $Ax \leq b$ encodes the remaining budget constraints. We show that our techniques, combined with a recent randomized algorithm of [4], yields a (randomized) algorithm that, for any $\epsilon > 0$, returns in $n^{O(k^{1.5}/\epsilon)}$ time a basis that (exactly) satisfies a chosen budget constraint, and violates the other budget constraints by (at most) a $(1 + \epsilon)$-factor, where $n := |U|$ is the size of the ground-set of M. This *matches* the current-best approximation guarantee of [9] (who give a deterministic algorithm) *and* the current-best running time of [4].

Theorem 5.5 ([4]). *For some constant $\nu > 0$, there exists a randomized FPRA, $\mathcal{B}_{BN}$, for unweighted $(Q^{\mathcal{P}(M)})$ that rounds any $x \in \mathcal{P}(M)$ to an extreme point $\hat{x}$ of $\mathcal{P}(M)$ such that $Ax - \nu\sqrt{k}\Delta \leq A\hat{x} \leq Ax + \nu\sqrt{k}\Delta$, where $\Delta = (\max_{1 \leq j \leq n} a_{ij})_{1 \leq i \leq k-1} = (\max_{e \in U} d_i(e))_{1 \leq i \leq k-1}$.*

Lemma 5.6. *There exists a polytime randomized algorithm that finds a basis S of M such that $d_k(S) \leq B_k$, and $d_i(S) \leq B_i + 2\nu\sqrt{k} \max_{e \in U} d_i(e)$ for all $1 \leq i \leq k-1$, or determines that the instance is infeasible.*

Proof. As explained above, we cast the problem as a special case of $(\mathrm{Q}^{\mathcal{P}})$ by using the k-th budget constraint as the objective function, and the remaining budget constraints as packing side-constraints. If the LP-relaxation of $(\mathrm{Q}^{\mathcal{P}})$ is infeasible, then the budgeted-matroid-basis is infeasible. Otherwise, the above guarantee follows by applying Theorem 5.2 with the algorithm $\mathcal{B}=\mathcal{B}_{\mathrm{BN}}$. □

Using ideas from [4], we combine the algorithm from Lemma 5.6 with a partial enumeration step as follows. We say an element $e \in U$ is *heavy* if the inequality $d_i(e) > \frac{\epsilon}{2\nu\sqrt{k}} B_i$ holds for at least one index $i \in \{1, \ldots, k\}$. Let H denote the set of all heavy elements. We state our algorithm below. Let $\epsilon > 0$ be a parameter.

1. For every set $\widetilde{H} \subseteq H$ of size $|\widetilde{H}| \leq \frac{2\nu k^{1.5}}{\epsilon}$, we do the following.
   (a) Let M' be the matroid obtained from M by contracting the elements of $\widetilde{H}$ and deleting the elements of $H \setminus \widetilde{H}$.
   (b) Compute residual budgets $B'_i := B_i - d_i(\widetilde{H})$, for $i \in \{1, \ldots, k\}$.
   (c) Run the algorithm from Lemma 5.6 on matroid M' with budgets $\{B'_i\}_{1 \leq i \leq k}$.
   (d) If the algorithm succeeds (that is, if the LP that it attempts to solve is feasible), then let T be the set of elements returned, and let $S := \widetilde{H} \cup T$. If S is a basis of M, $d_k(S) \leq B_k$, and $d_i(S) \leq (1+\epsilon)B_i$ for all $1 \leq i \leq k-1$, then return S.
2. If step 1 does not return any set S, then return that the instance is infeasible.

Theorem 5.7. *The algorithm above, run with parameter $\epsilon > 0$, finds in $n^{O(k^{1.5}/\epsilon)}$ time a basis S of M such that $d_k(S) \leq B_k$ and $d_i(S) \leq (1+\epsilon)B_i$ for all $1 \leq i \leq k-1$, or determines that the instance is infeasible.*

Proof. Note that the number of iterations is at most $n^{\frac{2\nu k^{1.5}}{\epsilon}} = n^{O(k^{1.5}/\epsilon)}$. Since steps 1(a)–1(d) run in $\mathsf{poly}(n)$ time, the overall running time is $n^{O(k^{1.5}/\epsilon)}$ as claimed.

If the instance is infeasible, then any outcome of the algorithm (infeasible, or a basis S) is correct. (Note that due to the verification done at the end of step 1(d), any set S returned must have the required properties.) So assume that the instance is feasible, and let S^* be a basis of M that exactly satisfies all the budget constraints. We argue that in this case the algorithm does indeed returns a basis with the desired properties. Let $H^* := S^* \cap H$ be the set of heavy elements that S^* contains. Note that since a heavy element uses up at least one budget to an extent greater than $\frac{\epsilon}{2\nu\sqrt{k}}$, and since S^* satisfies all the k budget constraints, we must have $|H^*| \leq \frac{k}{\frac{\epsilon}{2\nu\sqrt{k}}} = \frac{2\nu k^{1.5}}{\epsilon}$. Note that at the iteration corresponding to $\widetilde{H} = H^*$ (if the algorithm reaches it), the set $S^* \setminus H^*$ is feasible for the residual problem (with a matroid M' and residual budgets $\{B'_i\}$ defined in steps 1a and 1b). Further, note that this set also certifies that the resulting set S satisfies $d_k(S) = d_k(H^*) + d_k(T) \leq d_k(H^*) + d_k(S^* \setminus H^*) = d_k(S^*) \leq B_k$. Finally, for every $i \in \{1, \ldots, k-1\}$, we have

$$\begin{aligned} d_i(S) = d_i(H^*) + d_i(T) &\leq d_i(H^*) + B'_i + 2\nu\sqrt{k} \max_{e \in U \setminus H} d_i(e) \\ &\leq B_i + 2\nu\sqrt{k}\frac{\epsilon}{2\nu\sqrt{k}} B_i = (1+\epsilon)B_i, \end{aligned}$$

and so the set S will pass the verification done at step 1(d) and will be returned by the algorithm. □

References

1. A. Asadpour, M. Goemans, A. Madry, S. Oveis Gharan, and A. Saberi. An $O(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the 20th SODA*, pages 379–389, 2010.
2. N. Bansal, R. Khandekar, J. Könemann, V. Nagarajan, and B. Peis. On generalizations of network design problems with degree bounds. *Mathematical Programming*, 141(1-2):479–506, 2013.
3. N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree-bounded directed network design. *SICOMP*, 39(4):1413–1431, 2009.
4. N. Bansal and V. Nagarajan. Approximation-friendly discrepancy rounding. To appear in *Proc., IPCO 2016*. Also appears as arXiv:1512.02254, 2015.
5. K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. What would Edmonds do? Augmenting paths and witnesses for degree-bounded MSTs. *Algorithmica* 55:157–189, 2009.
6. C. Chekuri, J. Vondrak, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51st FOCS*, 2010.
7. M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner Tree to within one of optimal. *Journal of Algorithms* 17(3):409–423, 1994.
8. M. Goemans. Minimum bounded degree spanning trees. In *47th FOCS*, 2006.
9. F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.
10. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21:39–60, 2001.
11. J. Könemann and R. Ravi. A matter of degree: improved approximation algorithms for degree-bounded minimum spanning trees. *SICOMP*, 31:1783–1793, 2002. *SIAM Journal on Computing*, 31:1783–1793, 2002.
12. J. Könemann and R. Ravi. Primal-dual meets local search: approximating MST's with nonuniform degree bounds. In *Proceedings of the 35th STOC*, 389–395 2003.
13. A. Linharés and C. Swamy. Approximating min-cost chain-constrained spanning trees: a reduction from weighted to unweighted problems. In *Proceedings of the 18th IPCO*, pages 38–49, 2016.
14. N. Olver and R. Zenklusen. Chain-constrained spanning trees. In *Proceedings of the 16th IPCO*, pages 324–335, 2013.
15. R. Ravi, M. Marathe, S. Ravi, D. Rosenkrantz, and H. Hunt III. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica* 31(1):58–78, 2001.
16. M. Singh and L. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proceedings of the 39th STOC*, pages 661–670. 2007.
17. R. Zenklusen. Matroidal degree-bounded minimum spanning trees. In *Proceedings of the 23rd SODA*, pages 1512–1521, 2012.

A Proofs omitted from the main body

Proof of Lemma 2.1. Let $\mathrm{span}(x,\mathcal{L})$ denote the vector space spanned by $\{\chi^{x,E(A)}\}_{A\in\mathcal{L}}$. We say that two sets $A,B\subseteq V$ cross, if $A\cap B, A\setminus B, B\setminus A$ are all non-empty. Suppose the statement is false. Among all sets T whose spanning-tree constraints are tight at x and $\chi^{E(T)}\notin\mathrm{span}(\mathcal{L})$, pick one, say S, that crosses the fewest number of sets in $\mathcal{L}$.

Note that S must cross a non-zero number of sets in $\mathcal{L}$, otherwise this would contradict the fact that $\mathcal{L}$ is a *maximal* laminar family of sets whose spanning-tree constraints are tight at x. Let $A\in\mathcal{L}$ be an inclusion-wise minimal set of $\mathcal{L}$ such that S and A cross. We have

$$x\big(E(S\cap A)\big)+x\big(E(S\cap A)\big)\geq x\big(E(S)\big)+x\big(E(A)\big)=|S|+|A|-2=\big(|S\cap A|-1\big)+\big(|S\cup A|-1\big).$$

It follows that the spanning-tree constraints for $S\cap A$ and $S\cup A$ are also tight at x, and the inequality above is in fact an equality. So $x_e=0$ for all edges e with one end in $S\setminus A$ and one end in $A\setminus S$, and hence $\chi^{x,E(S\cap A)}+\chi^{x,E(S\cup A)}=\chi^{x,E(S)}+\chi^{x,E(A)}$. This implies that $\chi^{x,E(S\cap A)}$ and $\chi^{x,E(S\cup A)}$ cannot both lie in $\mathrm{span}(x,\mathcal{L})$ as this would imply that $\chi^{x,S}\in\mathrm{span}(x,\mathcal{L})$. Finally, note that both $S\cap A$ and $S\cup A$ cross fewer sets of $\mathcal{L}$ than S. This yields a contradiction to our choice of S above. □

Proof of Lemma 3.1. This follows from essentially the same potential-function argument as used in [14] to obtain a rainbow-free solution. Sort the edges of $\mathrm{supp}(x)$ in increasing order of $|\mathcal{S}_e|$ breaking ties arbitrarily. Let $e_1, e_2, \ldots, e_k$ denote this ordering. Let $w \in \mathbb{R}^E$ be any weight function such that $w_{e_1} < w_{e_2} < \cdots < w_{e_k}$ (e.g., $w_{e_i} = i$ for all i). Let x' be an optimal solution to the following LP. (Note that the LP has variables $\{z_e\}_{e \in E}$, and that the $\{x_e\}_{e \in E}$ values are fixed.)

$$\begin{aligned} \min \quad & \sum_e w_e z_e & & \text{(P')} \\ \text{s.t.} \quad & z \in \mathrm{P}_{\mathrm{ST}}(G), \quad z_e = 0 & \forall e \notin \mathrm{supp}(x) \\ & z\big(\delta(S)\big) \le x\big(\delta(S)\big) & \forall S \in \mathcal{S} \\ & z\big(E(L)\big) = |L| - 1 & \forall L \in \mathcal{L}. \end{aligned}$$

Properties (i) and (iii) hold by construction. Since we force the spanning-tree constraints corresponding to sets in $\mathcal{L}$ to be tight, we can start with $\mathcal{L}$ and extend it to obtain a laminar decomposition $\mathcal{L}'$ of x' that refines $\mathcal{L}$, so (ii) holds.

It remains to show that $(x', \mathcal{L}')$ is a rainbow-free decomposition. Consider any set $L \in \mathcal{L}'$ and any two edges $e, f \in \mathrm{supp}(x') \cap E_L^{\mathcal{L}'}$, and suppose that e, f form a rainbow. Let $w_e < w_f$, so we must have $\mathcal{S}_e \subseteq \mathcal{S}_f$. Now perturb x' by adding ϵ to x'_e (the argument below will show that $x'_e < 1$) and subtracting ϵ from x'_f, where $\epsilon > 0$ is chosen to be suitably small; let x'' be this perturbed vector. Clearly, $w^T x'' < w^T x'$, so if we show that x'' is feasible to (P'), then we obtain a contradiction. Clearly, $\mathrm{supp}(x'') \subseteq \mathrm{supp}(x)$. Since $\mathcal{S}_e \subseteq \mathcal{S}_f$ it follows that $x''\big(\delta(S)\big) \le x\big(\delta(S)\big)$ for all $S \in \mathcal{S}$. Also, $x''\big(E(L)\big) = x'\big(E(L)\big) = |L| - 1$ for all $L \in \mathcal{L}$.

Finally, we show that $x'' \in \mathrm{P}_{\mathrm{ST}}(G)$ for a sufficiently small $\epsilon > 0$. (Hence, $x'_e < x''_e \le 1$.) For $A \subseteq V$ such that $x'\big(E(A)\big) < |A| - 1$, we obtain $x''\big(E(A)\big) \le |A| - 1$ by taking $\epsilon > 0$ suitably small; for A with $x'\big(E(A)\big) = |A| - 1$, we obtain $x''\big(E(A)\big) = |A| - 1$ since the spanning-tree constraints for all $L \in \mathcal{L}'$ are tight at (x' and) x'' and these span all other spanning-tree constraints that are tight at x' (Lemma 2.1). $\square$