Improved Algorithms for MST and Metric-TSP Interdiction[*]

André Linhares[†]   Chaitanya Swamy[†]

Abstract

We consider the *MST-interdiction* problem: given a multigraph $G = (V, E)$, edge weights $\{w_e \geq 0\}_{e \in E}$, interdiction costs $\{c_e \geq 0\}_{e \in E}$, and an interdiction budget $B \geq 0$, the goal is to remove a set $R \subseteq E$ of edges of total interdiction cost at most B so as to maximize the w-weight of an MST of $G - R := (V, E \setminus R)$.

Our main result is a 4-approximation algorithm for this problem. This improves upon the previous-best 14-approximation [30]. Notably, our analysis is also significantly simpler and cleaner than the one in [30]. Whereas [30] uses a greedy algorithm with an involved analysis to extract a good interdiction set from an over-budget set, we utilize a generalization of knapsack called the *tree knapsack problem* that nicely captures the key combinatorial aspects of this "extraction problem." We prove a simple, yet strong, LP-relative approximation bound for tree knapsack, which leads to our improved guarantees for MST interdiction. Our algorithm and analysis are nearly tight, as we show that one cannot achieve an approximation ratio better than 3 relative to the upper bound used in our analysis (and the one in [30]).

Our guarantee for MST-interdiction yields an 8-approximation for *metric-TSP interdiction* (improving over the 28-approximation in [30]). We also show that the *maximum-spanning-tree interdiction* problem is at least as hard to approximate as the minimization version of densest-k-subgraph.

1 Introduction

Interdiction problems are a broad class of optimization problems with a wide range of applications. They model the problem faced by an attacker, who given an underlying, say, minimization, problem, aims to destroy or *interdict* the elements involved in the optimization problem (e.g., nodes or edges in a network-optimization problem) without exceeding a given interdiction budget, so as to maximize the optimal value of the residual optimization problem (where one cannot use the interdicted elements). A classical example is the *minimum-spanning-tree (MST) interdiction* problem [23, 10, 30], which is the focus of this work: we are given a multigraph $G = (V, E)$, edge weights $\{w_e \geq 0\}_{e \in E}$, interdiction costs $\{c_e \geq 0\}_{e \in E}$, and an interdiction budget $B \geq 0$; the goal is to interdict (i.e., remove) a set $R \subseteq E$ of edges of total interdiction cost at most B so as to maximize the w-weight of an MST of the multigraph $G - R := (V, E \setminus R)$. Note that G may have parallel edges, which can be useful in modeling partial-interdiction effects, wherein interdicting an edge causes an increase in its weight that depends on the interdiction cost incurred for the edge.

At a high level, interdiction problems can be seen as investigating the sensitivity of an underlying optimization problem with respect to the removal of a limited set of underlying elements. This type of sensitivity analysis may be utilized to identify vulnerable spots (e.g., regions in a network) either: (a) for possible reinforcement, or, (b) if the optimization problem models an undesirable process (e.g., the spread of infection, or nuclear-arms smuggling), for disruption, so as to maximally impair the underlying process. A variety of applications of interdiction problems ensue from these two perspectives, including infrastructure protection [5, 28], hospital-infection control [1], prevention of nuclear-arms smuggling [25],

---

and military planning [11] (see also the references in [30]). Consequently, interdiction problems have been extensively studied, especially in the Operations Research literature; besides MST-interdiction, some well-studied interdiction problems include network-flow interdiction [26, 29, 2, 32, 13, 3], shortest s-t path interdiction [9, 14, 17, 20], and maximum-matching interdiction [31, 6]. All these problems, as well as MST-interdiction, are *NP*-hard.

Our results. Our main result is a 4-approximation algorithm for MST interdiction (Theorem 4.1), i.e., we compute in polytime a solution of value at least (optimum)/4. This constitutes a substantial improvement over the previous-best approximation ratio of 14 obtained by Zenklusen [30].

Notably, and perhaps more importantly, our algorithm is simple, and its analysis is significantly simpler and cleaner than the one in [30]. The key ingredient (see also "Our techniques") of both our algorithm and the one in [30] is a procedure for extracting a good interdiction set from one that exceeds the interdiction budget. Whereas [30] uses a greedy algorithm with a rather involved analysis to achieve this, our simple and more-effective procedure is based on two chief insights. First, we discern that the key combinatorial aspects of this "extraction problem" can be captured quite nicely via a clean generalization of the knapsack problem called the *tree knapsack problem* [15] (Section 3). In particular, we argue that approximation guarantees for tree knapsack *relative to the natural LP for this problem* translate directly to guarantees for MST interdiction. Second, complementing the above insight, we show that the tree knapsack problem admits a simple *iterative-rounding* based algorithm that achieves a strong LP-relative guarantee (Theorem 3.1, Corollary 3.3). Our improved guarantee for MST interdiction then readily follows by combining these two ideas.

We also show a lower bound of 3 (Theorem 4.10) on the approximation ratio achievable relative to the upper bound used in our analysis (and the analysis in [30]), thereby showing that our algorithm and analysis are nearly tight.

Our MST-interdiction result also yields an improved guarantee for the *metric-TSP interdiction* problem (Section 5), wherein we have *metric* edge weights $\{w_e\}$, and we seek an interdiction set R with $\sum_{e \in R} c_e \leq B$ so as to maximize the minimum w-weight of a closed walk in $G - R$ that visits all nodes at least once. Since an α-approximation for MST interdiction yields a 2α-approximation for metric-TSP interdiction [30], we obtain an approximation factor of 8 for metric-TSP interdiction, which improves upon the previous-best factor of 28 [30].

In Section 6, we consider the *maximum-spanning-tree interdiction* problem, where the goal is to *minimize the maximum w-weight of a spanning tree of $G - R$*. We show that this problem is at least as hard to approximate as the minimization version of the *densest-k-subgraph* problem (MinDkS). MinDkS does not admit any constant-factor approximation under certain less-standard complexity assumptions [27] (and is believed to have a larger inapproximability threshold), so this highlights a stark contrast with the MST-interdiction problem.

Our techniques. We give an overview of our algorithm for MST interdiction. Let val(R) be the w-weight of an MST of $G - R$. Using standard arguments, we can reduce the problem to the following setting (see Section 2 and Theorem 2.3): we are given interdiction sets $R_1 \subseteq R_2$ with $c(R_1) < B < c(R_2)$ such that $a \cdot \text{val}(R_1) + b \cdot \text{val}(R_2) \geq OPT$, where OPT is the optimal value and $a, b \geq 0$ are such that $a + b = 1$ and $a \cdot c(R_1) + b \cdot c(R_2) = B$. These arguments resemble the ones in [30], but we do not need to assume that the w_e weights are powers of 2. (We emphasize however that this by itself is not the chief source of our improvement.) The technical meat of the algorithm, and where we diverge significantly from [30] to obtain our improved guarantee, is to show how to extract a good interdiction set from R_1, R_2. As mentioned earlier, we replace the greedy algorithm of [30] for extracting a good interdiction set from R_2, and its associated intricate analysis, by considering the tree-knapsack problem to capture the key aspects of this extraction problem, and devise a simple iterative-rounding algorithm that yields a strong LP-relative guarantee for tree knapsack. This conveniently translates to a much-improved 5-approximation algorithm for MST interdiction

(Theorem 4.7). The further improvement to a 4-approximation arises by also leveraging R_1 to find a good interdiction set: instead of focusing solely on R_2 (as done in [30]), we return an interdiction set R such that $R_1 \subseteq R \subseteq R_2$ (see Section 4.1).

To arrive at the tree knapsack problem, observe that $\mathrm{val}(R)$ can be conveniently expressed as a weighted sum of the number of components of $(V, \{e \in E \setminus R : w_e \leq t\})$, where t ranges over some distinct edge weights, say, $0 \leq w_1 < \cdots < w_k$ (Lemma 2.2). Let $\mathcal{A}_0$ denote the components of $(V, E_{\leq 0} := \emptyset)$, and $\mathcal{A}_i$ denote the components of $(V, E_{\leq i} := \{e \in E \setminus R_2 : w_e \leq w_i\})$ for $i = 1, \ldots, k$. The multiset $\bigcup_{i=0}^{k} \mathcal{A}_i$ forms a laminar family, which can be viewed as a rooted tree. We seek to build our interdiction set R by selecting a suitable collection of sets from this laminar family, ensuring that if we pick a component $A \in \mathcal{A}_i$, then $\delta(A) \cap E_{\leq i}$ is included in R (so that A is indeed a component of $(V, E_{\leq i} \setminus R)$). Whereas $\mathrm{val}(R)$ is nicely *decoupled* across the selected components, it is harder to decouple the interdiction cost incurred and account for it. For instance, summing $c\big(\delta(A) \cap E_{\leq i}\big)$ for each selected $A \in \mathcal{A}_i$ may grossly overestimate the interdiction cost, whereas summing $c\big(\delta(A) \cap \{e : w_e = w_i\}\big)$ for each selected $A \in \mathcal{A}_i$ *underestimates* the interdiction cost. A crucial insight is that, if we ensure that *whenever we pick $A \in \mathcal{A}_i$, we also pick its children in the laminar family*, then *summing $c\big(\delta(A) \cap \{e : w_e = w_i\}\big)$ for each selected $A \in \mathcal{A}_i$ is a good proxy for the interdiction cost incurred.*

This motivates the definition of the tree knapsack problem: given a rooted tree Γ with node values $\{\alpha_v\}$, node weights $\{\beta_v\}$, and budget B, we want to pick a maximum-value *downwards-closed* set of nodes (not containing the root) whose weight is at most B, where downwards-closed means that if we pick a node, then we also pick all its children. The standard knapsack problem is thus the special case where Γ is a star (rooted at its center). We consider the natural LP (TK-P) for tree knapsack, and generalizing a well-known result for knapsack, show that we can efficiently compute a solution of value at least $OPT_{\text{TK-P}} - \max_{\text{chains } C} \sum_{v \in C} \alpha_v$ (Theorem 3.1), where a chain is a subset of a root-leaf path.

Finally, we show that for the tree-knapsack instance derived (as above) from R_2, $OPT_{\text{TK-P}}$ is "large" (Lemma 4.4), and combining this with the above bound yields our approximation guarantee.

Related work. MST interdiction in its full generality seems to have been first considered by [23], who showed that the problem is *NP*-hard. The approximation question for MST interdiction was first investigated by [10]. They focused on the setting with unit interdiction costs, often called the *B-most-vital-edges* problem, showed that this special case remains *NP*-hard, and obtained an $O(\log B)$-approximation (which also yields an $O(\log |E|)$-approximation with general interdiction costs). This guarantee was improved only recently by Zenklusen [30], who gave the first (and previous-best) $O(1)$-approximation algorithm for (general) MST interdiction, achieving an approximation ratio of 14. The B-most-vital edges problem has been well studied for $B = 1$ and for $B = O(1)$, where it can be solved optimally; see, e.g., [21] and the references therein. The special case of MST interdiction where we have only two distinct edge weights captures the *budgeted graph disconnection* (BGD) problem [7] for which a 2-approximation is known [7]. As noted by [30], MST interdiction can be viewed as multilevel-BGD, which makes it much more challenging as it is difficult to control the interactions at the different levels. It is noteworthy that our approximation ratio of 4 for MST interdiction is quite close to the approximation ratio of 2 for BGD.

As with MST interdiction, until recently, there were wide gaps in our understanding of the approximability of the other classic *NP*-hard interdiction problems mentioned earlier. Maximum s-t flow interdiction, even on undirected graphs with unit interdiction costs, is now known to be at least as hard as MinDkS on λ-uniform hypergraphs. This follows from a recent hardness result for k-route s-t cut in [13], which turns out to be an equivalent problem.[1] This hardness result has been rediscovered (in a slightly weaker form)

---

[1] In k-route s-t cut, the goal is to remove a min w-cost set of edges so as to reduce the s-t edge connectivity to at most $k - 1$. This corresponds to taking all but the $k - 1$ most-expensive edges of some cut. So we can rephrase this problem as follows: remove at most $k - 1$ edges to minimize the (min-s-t-cut value = max-s-t-flow value) with capacities $\{w_e\}$; this is precisely the maximum s-t flow interdiction problem with unit interdiction costs and budget $k - 1$.

by [3], who also gave an $O(n)$-approximation algorithm. For shortest s-t path interdiction, very recently, Lee [20] proved a super-constant hardness result. For maximum-matching interdiction, [6] devised the first $O(1)$-approximation algorithm. Despite this recent progress, interdiction variants of common optimization problems are generally not well understood, especially from the viewpoint of approximability.

The tree knapsack problem was introduced by [15], and is a special case of the *partially-ordered knapsack* (POK) problem [18]. While an FPTAS can be obtained for tree knapsack and some special cases of POK via dynamic programming [15, 18], and the natural LP for POK has been investigated [18], our LP-relative guarantee and rounding algorithm for tree knapsack are new.

2 Preliminaries

For any vector $d \in \mathbb{R}^E$ and any subset $F \subseteq E$ of edges, we use $d(F)$ to denote $\sum_{e \in F} d_e$. Given a subset $R \subseteq E$ of edges, we use $\mathrm{val}(R)$, which we call the *value* of R, to denote the w-weight of an MST in the multigraph $G - R$, i.e., $\mathrm{val}(R) := \min_{\text{spanning trees } T \text{ of } G - R} w(T)$. The *minimum-spanning-tree interdiction* problem can thus be restated as $\max \{ \mathrm{val}(R) : R \subseteq E, \ c(R) \leq B \}$.

If there is an interdiction set R with $c(R) \leq B$ such that $G - R$ is disconnected, then $\mathrm{val}(R) = \infty$, and so the MST-interdiction problem is unbounded. Note that this happens iff a min-cut $\delta(S)$ of G satisfies $c(\delta(S)) \leq B$, and we can efficiently detect this. So in the sequel, we assume that this is not the case. Let OPT denote the optimal value of the MST-interdiction problem (which is now finite). For $F \subseteq E$, let $\sigma(F)$ denote the number of connected components of (V, F).

Let $w_1, w_2, \ldots, w_M$ be the distinct weights in $\{w_e : e \in E\}$, where $0 \leq w_1 < w_2 < \cdots < w_M$. For $i = 1, \ldots, M$, define $E_i := \{e \in E : w_e = w_i\}$ and $E_{\leq i} := \{e \in E : w_e \leq w_i\}$. For notational convenience, we define $w_0 := 0$ and $E_0 = E_{\leq 0} := \emptyset$. (Note that E_0 is *not* necessarily $\{e \in E : w_e = w_0\}$, and $E_{\leq 0}$ is not necessarily $\{e \in E : w_e \leq w_0\}$.)

Let $k \in \{1, \ldots, M\}$ be the *smallest* index such that $c(\delta(S) \cap E_{\leq k}) > B$ for every $\emptyset \neq S \subsetneq V$; that is, the multigraph $(V, E_{\leq k} \setminus R)$ is connected for all R such that $c(R) \leq B$. Note that k is well defined due to our earlier assumption. This implies the following properties, as also observed in [30]: (i) $OPT \geq w_k$ (since, by definition of k, there is a feasible interdiction set R whose removal disconnects $(V, E_{\leq k-1})$); (ii) for any R with $c(R) \leq B$, we have $\mathrm{val}(R) = \mathrm{val}(R \cap E_{\leq k-1})$, and hence, there is an optimal solution that only interdicts edges from $E_{\leq k-1}$; and (iii) given (ii), we may add additional edges of weight w_k without impacting the optimal value, so we may assume that (V, E_k) is connected. We summarize these properties and assumptions below.

Claim 2.1. *Let $k \in \{1, \ldots, M\}$ be the smallest index such that $(V, E_{\leq k} \setminus R)$ is connected for every $R \subseteq E$ with $c(R) \leq B$. Assume that such a k exists. Then, (i) $OPT \geq w_k$, and (ii) there is an optimal solution R^* such that $R^* \subseteq E_{\leq k-1}$. Moreover, we may assume that (iii) the multigraph (V, E_k) is connected.*

Lemma 2.2. *Let $R \subseteq E$ be an edge-set such that $(V, E_{\leq k} \setminus R)$ is connected. Then $\mathrm{val}(R) = -w_k + \sum_{i=0}^{k-1} \sigma(E_{\leq i} \setminus R)(w_{i+1} - w_i)$.*

Proof. Consider, for example, running Kruskal's algorithm to obtain an MST of $G - R$. We include exactly $\sigma(E_{\leq j-1} \setminus R) - \sigma(E_{\leq j} \setminus R)$ edges of weight w_j for every $1 \leq j \leq M$, and this quantity is 0 for all $j > k$.

4

It follows that

$$
\begin{aligned}
\mathrm{val}(R) &= \sum_{j=1}^{M}\Big(\sigma\left(E_{\leq j-1}\setminus R\right)-\sigma(E_{\leq j}\setminus R)\Big)w_j = \sum_{j=1}^{k}\Big(\sigma(E_{\leq j-1}\setminus R)-\sigma(E_{\leq j}\setminus R)\Big)w_j \\
&= \sum_{j=1}^{k}\Big(\sigma\left(E_{\leq j-1}\setminus R\right)-\sigma(E_{\leq j}\setminus R)\Big)\sum_{i=0}^{j-1}(w_{i+1}-w_i) \\
&= \sum_{i=0}^{k-1}(w_{i+1}-w_i)\sum_{j=i+1}^{k}\Big(\sigma(E_{\leq j-1}\setminus R)-\sigma(E_{\leq j}\setminus R)\Big) \\
&= \sum_{i=0}^{k-1}\big(\sigma(E_{\leq i}\setminus R)-1\big)(w_{i+1}-w_i) = -w_k+\sum_{i=0}^{k-1}\sigma(E_{\leq i}\setminus R)(w_{i+1}-w_i). \qquad \square
\end{aligned}
$$

Given Claim 2.1, we focus on interdiction sets $R\subseteq E_{\leq k-1}$ and recast the MST-interdiction problem as: $\max\big\{\mathrm{val}(R):R\subseteq E_{\leq k-1},\ c(R)\leq B\big\}$. As is common in the study of constrained optimization problems (see, e.g., [19, 12] and the references therein), we Lagrangify the budget constraint $c(R)\leq B$, and consider the following Lagrangian problem (offset by $-\lambda B$), where $\lambda\geq 0$ is a parameter:

$$
\max_{R\subseteq E_{\leq k-1}}\quad f_\lambda(R):=\mathrm{val}(R)-\lambda c(R). \tag{P_λ}
$$

The expression for $\mathrm{val}(R)$ in Lemma 2.2 holds for all $R\subseteq E_{\leq k-1}$ as (V,E_k) is connected. Since $\sigma(E_{\leq i}\setminus R)$ is a supermodular function of R, this implies that $\mathrm{val}(\cdot)$, and hence the objective function $f_\lambda(\cdot)$ of (P_λ), is *supermodular* over the domain $2^{E_{\leq k-1}}$: for any $A_1,A_2\subseteq E_{\leq k-1}$, we have $f_\lambda(A_1)+f_\lambda(A_2)\leq f_\lambda(A_1\cap A_2)+f_\lambda(A_1\cup A_2)$. Hence, (P_λ) can be solved exactly, which we crucially exploit.

Let $\mathcal{O}_\lambda^*$ denote the set of optimal solutions to (P_λ). Observe that for any $\lambda\geq 0$ and any $R\in\mathcal{O}_\lambda^*$, we have $\mathrm{val}(R)-\lambda c(R)\geq OPT-\lambda B$. So if we find some $\lambda\geq 0$ and $R\in\mathcal{O}_\lambda^*$ such that $c(R)=B$, we have $\mathrm{val}(R)\geq OPT$, so R is an optimal solution. In general, such a pair (λ,R) need not exist, or can be hard to find. However, by doing a binary search for λ, or alternatively, as noted in [30], via parametric submodular-function minimization [8, 24], we can obtain the following result; we include a self-contained proof in Appendix A.

Theorem 2.3 ([30])**.** *One can find in polytime: either (i) an optimal solution to the MST-interdiction problem, or (ii) a parameter $\lambda\geq 0$ and two optimal solutions R_1,R_2 to (P_λ) such that $R_1\subseteq R_2$ and $c(R_1)<B<c(R_2)$.*

3 The tree knapsack problem

We now define the *tree knapsack problem*, and devise a simple, clean LP-based approximation algorithm for this problem (Theorem 3.1, Corollary 3.3). As we show in Section 4, the tree knapsack problem nicely abstracts the key combinatorial problem encountered in extracting a good interdiction set from an over-budget set R_2 in case (ii) of Theorem 2.3, and our LP-relative guarantees for tree knapsack readily yield improved approximation guarantees for MST interdiction.

In the tree knapsack problem [15], we have a tree $\Gamma=(\{r\}\cup N,A)$ rooted at node r. Each node $v\in N$ has a *value* $\alpha_v\geq 0$ and a *weight* $\beta_v\geq 0$, and we have a budget B. We say that a subset $S\subseteq N$ of nodes is *downwards-closed* if for every $v\in S$, all children of v are also in S. The goal is to find a maximum-value downwards-closed set $S\subseteq N$ (so $r\notin S$) such that $\sum_{v\in S}\beta_v\leq B$. Observe that the (standard) knapsack problem is precisely the special case of tree knapsack where the underlying tree is a star (rooted

at its center). Throughout, we use v to index nodes in N. For $S \subseteq N$ and a vector $\rho \in \mathbb{R}^N$, we use $\rho(S)$ to denote $\sum_{v \in S} \rho_v$.

The following is a natural LP-relaxation for the tree knapsack problem involving variables x_v for all v. Let $\mathsf{ch}(v)$ denote the set of children of node v.

$$\max \quad \sum_v \alpha_v x_v \qquad\qquad \text{(TK-P)}$$

$$\text{s.t.} \quad x_v \leq x_u \qquad \text{for all } v, \text{ for all } u \in \mathsf{ch}(v) \qquad (1)$$

$$\sum_v \beta_v x_v \leq B, \qquad 0 \leq x_v \leq 1 \quad \text{for all } v.$$

Tree knapsack was first defined by [15] who devised an FPTAS for this problem via dynamic programming. However, for our purposes, we need an approximation guarantee relative to the above LP, which was not known previously.

The main result of this section is as follows. We say that $C \subseteq N$ is a *chain* if for every two distinct nodes in C, one is a descendant of the other.

Theorem 3.1. *We can compute in polytime an integer solution to* (TK-P) *of value at least* $OPT_{TK\text{-}P} - \max_{chains\ C \subseteq N} \alpha(C)$.

Theorem 3.1 nicely generalizes a well-known result about the standard knapsack problem, namely, that we can always obtain a solution of value at least (LP-optimum)$- \max_v \alpha_v$. Notice that when Γ is a star (i.e., we have a knapsack instance), this is *precisely* the guarantee that we obtain from the theorem. The proof of Theorem 3.1 relies on the following structural result (which extends a similar result known for knapsack). Let $\Gamma(v)$ denote the subtree of Γ rooted at v.

Lemma 3.2. *Let $\bar{x}$ be an extreme-point solution to the linear program* (TK-P). *Then there is at most one child v of r for which the subtree $\Gamma(v)$ contains a fractional node, i.e., some node w with $0 < \bar{x}_w < 1$.*

Proof. Suppose for a contradiction that the root r has two children v_1 and v_2 such that the subtrees $\Gamma(v_1)$ and $\Gamma(v_2)$ both contain at least one fractional node. We show that for some nonzero vector $d \in \mathbb{R}^N$, the solutions $\bar{x} \pm d$ are feasible to (TK-P), which contradicts that $\bar{x}$ is an extreme point.

For $j = 1, 2$, let N_j be a maximal set of nodes in the subtree $\Gamma(v_j)$ such that: (a) N_j induces a connected subgraph of $\Gamma(v_j)$; and (b) all nodes in N_j have the same x_w value, which is fractional. We will always set $d_v = 0$ for all $v \notin N_1 \cup N_2$. Note that for any $\mu_1, \mu_2 \in \mathbb{R}$ with sufficiently small absolute value, if we set $d_v = \mu_1$ for all $v \in N_1$ and $d_v = \mu_2$ for all $v \in N_2$, then the vectors $x \pm d$ satisfy constraints (1) (due to the maximality of N_1, N_2), and $0 \leq (x \pm d)_v \leq 1$ for all $v \in N$.

We argue that we can choose suitably small μ_1, μ_2 (not both equal to zero) so that $\sum_v \beta_v d_v = 0$, and so $x \pm d$ also satisfy the budget constraint, and hence are feasible to (TK-P). If $\beta(N_1) = 0$, if we take a sufficiently small $\mu_1 > 0$ and $\mu_2 = 0$, then clearly $\sum_v \beta_v d_v = 0$. Otherwise, for $\epsilon > 0$ and suitably small, we take $\mu_1 = \epsilon\beta(N_2)$ and $\mu_2 = -\epsilon\beta(N_1)$. Then again, $\sum_v \beta_v d_v = \epsilon\beta(N_1)\beta(N_2) - \epsilon\beta(N_2)\beta(N_1) = 0$ (so $x \pm d$ is feasible to (TK-P)). $\square$

Proof of Theorem 3.1. We use *iterative rounding*, and the proof is by induction on the depth d of Γ, which is the maximum number of edges on a root-leaf path.

If $d = 0$, then $N = \emptyset$, and (TK-P) has no variables and constraints, so the statement is vacuously true. So suppose $d \geq 1$. Let x^* be an extreme-point optimal solution of (TK-P). If x^* is integral, then we obtain value $OPT_{TK\text{-}P}$, completing the induction step. Otherwise, by Lemma 3.2, there is exactly one child v of r such that the subtree $\Gamma(v)$ contains a fractional node.

6

Set $\tilde{x}' = x^*\big|_{N\setminus\Gamma(v)}$, i.e., x^* restricted to $N \setminus \Gamma(v)$, which is integral. We have $\sum_{u\in N\setminus\Gamma(v)} \alpha_u \tilde{x}'_u = OPT_{\text{TK-P}} - \sum_{w\in\Gamma(v)} \alpha_w x^*_w$. Now consider the tree knapsack instance defined by the tree $\Gamma(v)$ with root v, and budget $B - \sum_{u\in N\setminus\Gamma(v)} \beta_u \tilde{x}'_u$ (and values α_w and weights β_w for all $w \in \Gamma(v) \setminus \{v\}$). Observe that $x^*\big|_{\Gamma(v)\setminus\{v\}}$ is a fractional solution to the LP-relaxation (TK-P) corresponding to this tree knapsack problem, so the optimal value of this LP is at least $\sum_{w\in\Gamma(v)\setminus\{v\}} \alpha_w x^*_w$. (These objects are null if $\Gamma(v) = \{v\}$.) Thus, since $\Gamma(v)$ has depth at most $d - 1$, by our induction hypothesis, our rounding procedure applied to this tree knapsack instance yields an integer solution $\tilde{x}'' \in \{0,1\}^{\Gamma(v)\setminus\{v\}}$ of value at least $\sum_{w\in\Gamma(v)\setminus\{v\}} \alpha_w x^*_w - \max_{\text{chains } C \subseteq \Gamma(v)\setminus\{v\}} \alpha(C)$. Thus, taking $\tilde{x} = (\tilde{x}', \tilde{x}_v = 0, \tilde{x}'')$, we obtain a feasible integer solution to (TK-P) having value at least

$$OPT_{\text{TK-P}} - \sum_{w\in\Gamma(v)} \alpha_w x^*_w + \sum_{w\in\Gamma(v)\setminus\{v\}} \alpha_w x^*_w - \max_{\text{chains } C \subseteq \Gamma(v)\setminus\{v\}} \alpha(C)$$

$$\geq OPT_{\text{TK-P}} - \alpha_v - \max_{\text{chains } C \subseteq \Gamma(v)\setminus\{v\}} \alpha(C)$$

$$\geq OPT_{\text{TK-P}} - \max_{\text{chains } C \subseteq N} \alpha(C) .$$

This completes the induction step, and hence the proof of the theorem. $\qquad\square$

We remark that (as is standard) the iterative-rounding procedure in Theorem 3.1 is in fact combinatorial, since when we move to the subtree $\Gamma(v)$, we only need to move from $x^*\big|_{\Gamma(v)\setminus\{v\}}$ to an extreme-point of the LP of the smaller tree-knapsack instance of no smaller value (instead of obtaining an optimal LP solution), which can be done combinatorially (as in the proof of Lemma 3.2).

We now prove a somewhat stronger version of Theorem 3.1 that will be useful in Section 4, where we utilize tree knapsack to solve the MST-interdiction problem. The depth of a node v is the number of edges on the (unique) r-v path of Γ. Let $L_i(\Gamma)$ be the set of nodes of Γ at depth i; we drop Γ if it is clear from the context. For a chain C of Γ, let C_i denote $C \cap L_i(\Gamma)$; note that $|C_i| \leq 1$.

Corollary 3.3. *We can obtain in polytime an integer solution $\tilde{x}$ to* (TK-P) *of value at least $OPT_{TK-P} - \max_{chains\ C \subseteq N}\big\{\sum_{i\geq 1:\tilde{x}(L_i)<|L_i|} \alpha(C_i)\big\}$.*

Proof. The result follows from the proof of Theorem 3.1 via a more-careful accounting. Recall that we use induction on the depth d of Γ. The base case when $d = 0$ is again vacuously true. So suppose $d \geq 1$, and let x^* be an extreme-point optimal solution to (TK-P). If x^* is integral, we are done, so suppose that there is a child v of r such that $\Gamma(v)$ contains a fractional node. In the sequel, L_i denotes $L_i(\Gamma)$.

As before, let $\tilde{x}' = x^*\big|_{N\setminus\Gamma(v)}$, $\tilde{x}_v = 0$, and let $\tilde{x}''$ be the integer solution obtained by induction for the tree knapsack instance defined by the tree $\Gamma(v)$ with root v and budget $B - \sum_{u\in N\setminus\Gamma(v)} \beta_u \tilde{x}'_u$. Therefore, $\tilde{x}''$ has value at least

$$\sum_{w\in\Gamma(v)\setminus\{v\}} \alpha_w x^*_w - \max_{\text{chains } C \subseteq \Gamma(v)\setminus\{v\}} \sum_{\substack{i\geq 1:\\ \tilde{x}''(L_i(\Gamma_v))<|L_i(\Gamma_v)|}} \alpha(C_i). \qquad (2)$$

Thus, $\tilde{x} = (\tilde{x}', \tilde{x}_v, \tilde{x}'')$ is a feasible integer solution to (TK-P) of value at least

$$OPT_{\text{TK-P}} - \sum_{w \in \Gamma(v)} \alpha_w x_w^* + \sum_{w \in \Gamma(v) \setminus \{v\}} \alpha_w x_w^* - \max_{\text{chains } C \subseteq \Gamma(v) \setminus \{v\}} \sum_{\substack{i \geq 1: \\ \tilde{x}''(L_i(\Gamma_v)) < |L_i(\Gamma_v)|}} \alpha(C_i)$$

$$\geq OPT_{\text{TK-P}} - \alpha_v - \max_{\text{chains } C \subseteq \Gamma(v) \setminus \{v\}} \sum_{\substack{i \geq 1: \\ \tilde{x}''(L_i(\Gamma_v)) < |L_i(\Gamma_v)|}} \alpha(C_i)$$

$$\geq OPT_{\text{TK-P}} - \max_{\text{chains } C \subseteq N} \sum_{\substack{i \geq 1: \\ \tilde{x}(L_i) < |L_i|}} \alpha(C_i).$$

The last inequality above follows by noting that for any chain $C \subseteq \Gamma(v) \setminus \{v\}$, letting $C' := C \cup \{v\}$ (which is also a chain), we have

$$\sum_{\substack{i \geq 1: \\ \tilde{x}(L_i) < |L_i|}} \alpha(C_i') = \alpha_v + \sum_{\substack{i \geq 1: \\ \tilde{x}''(L_i(\Gamma_v)) < |L_i(\Gamma_v)|}} \alpha(C_i).$$

This completes the induction step, and hence the proof. $\qquad\square$

4 MST interdiction

Our main technical result is the following theorem.

Theorem 4.1. *There is a 4-approximation algorithm for MST interdiction.*

The above guarantee substantially improves the previous-best approximation ratio of 14 obtained by [30]. Also, notably and significantly, our algorithm and analysis, which are based on the tree knapsack problem introduced in Section 3, are noticeably simpler and cleaner than the one in [30]. Improved guarantees for MST interdiction readily follow from (Theorem 3.1 and) Corollary 3.3 and Lemma 4.8, yielding approximation ratios of 5 and 4 respectively for MST interdiction (see Theorem 4.7 and Section 4.1). The proof below shows a slightly worse guarantee of 5 but introduces the main underlying ideas. Section 4.1 discusses the refinement needed to obtain the 4-approximation.

Our algorithm follows the same high-level outline as the one in [30]. As mentioned earlier, we consider the Lagrangian problem (P_λ), $\max_{R \subseteq E_{\leq k-1}} f_\lambda(R) := \text{val}(R) - \lambda c(R)$, obtained by dualizing the budget constraint $c(R) \leq B$. We then utilize Theorem 2.3. If this returns an optimal solution, then we are done. So assume in the sequel that Theorem 2.3 returns $\lambda \geq 0$ and two optimal solutions R_1 and R_2 to (P_λ) such that $R_1 \subseteq R_2$ and $c(R_1) < B < c(R_2)$.

For $R \subseteq E_{\leq k-1}$, define $h(R) := \sum_{i=0}^{k-1} \sigma(E_{\leq i} \setminus R)(w_{i+1} - w_i) = \text{val}(R) + w_k$. Let $R^* \subseteq E_{\leq k-1}$ denote an optimal solution to the MST-interdiction problem, so $OPT = h(R^*) - w_k$. Let $a, b \geq 0$ such that $a + b = 1$ and $ac(R_1) + bc(R_2) = B$. Then, since $\text{val}(R_1) - \lambda c(R_1) = \text{val}(R_2) - \lambda c(R_2) \geq OPT - \lambda B$, we have $ah(R_1) + bh(R_2) \geq h(R^*)$. We establish our approximation guarantee by comparing the value of our solution against the upper bound $ah(R_1) + bh(R_2) - w_k$. The following claim shows that this upper bound is precisely the optimal value of the Lagrangian relaxation of the MST interdiction problem, which is $UB := \min_{\lambda' \geq 0}(\lambda' B + \max_{R \subseteq E_{\leq k-1}} f_{\lambda'}(R))$. Complementing our 4-approximation, in Section 4.2, we prove a lower bound of 3 on the approximation ratio achievable relative to UB.

Claim 4.2. *We have $ah(R_1) + bh(R_2) - w_k = UB$.*

Proof. Let $OPT(\mathrm{P}_{\lambda'})$ denote the optimal value of the subproblem

$$\max_{R \subseteq E_{\leq k-1}} \quad f_{\lambda'}(R) := \mathrm{val}(R) - \lambda'c(R). \tag{$\mathrm{P}_{\lambda'}$}$$

Define $\eta(\lambda') := OPT(\mathrm{P}_{\lambda'}) + \lambda'B$. So we have $UB = \min_{\lambda' \geq 0} \eta(\lambda')$. We have

$$ah(R_1) + bh(R_2) - w_k = a \cdot \mathrm{val}(R_1) + b \cdot \mathrm{val}(R_2) = a(\mathrm{val}(R_1) - \lambda c(R_1)) + b(\mathrm{val}(R_2) - \lambda c(R_2)) + \lambda B$$
$$= (a + b)OPT(\mathrm{P}_\lambda) + \lambda B = \eta(\lambda) \geq UB.$$

We now argue that $UB \geq ah(R_1) + bh(R_2) - w_k$ by showing that $\eta(\lambda') \geq \eta(\lambda)$ for every $\lambda' \geq 0$. We have

$$\eta(\lambda') = OPT(\mathrm{P}_{\lambda'}) + \lambda'B \geq \max\{\mathrm{val}(R_1) - \lambda'c(R_1), \mathrm{val}(R_2) - \lambda'c(R_2)\} + \lambda'B$$
$$\geq a(\mathrm{val}(R_1) - \lambda'c(R_1)) + b(\mathrm{val}(R_2) - \lambda'c(R_2)) + \lambda'B$$
$$= a \cdot \mathrm{val}(R_1) + b \cdot \mathrm{val}(R_2) = \eta(\lambda). \qquad \square$$

Translation to tree knapsack. We now describe how the problem of combining R_1 and R_2 to extract a good, feasible interdiction set can be captured by a suitable instance of the tree knapsack problem defined in Section 3.

For $i = 0, \ldots, k$, let $\mathcal{A}_i \subseteq 2^V$ be the partition of V induced by the connected components of the multigraph $(V, E_{\leq i} \setminus R_2)$. Thus, $\mathcal{A}_k = \{V\}$ and $\mathcal{A}_0 = \{\{v\} : v \in V\}$. The multiset $\bigcup_{i=0}^k \mathcal{A}_i$, where we include $S \subseteq V$ multiple times if it lies in multiple $\mathcal{A}_i$s, is a *laminar family* (i.e., any two sets in the collection are either disjoint or one is contained in the other). This laminar family can naturally be viewed as a rooted tree, which defines the tree Γ in the tree knapsack problem. Taking a cue from Lemma 2.2, we build our interdiction set R by selecting a suitable collection of sets from this laminar family, ensuring that if we pick some $A \in \mathcal{A}_i$, then we include all edges of $\delta(A) \cap E_{\leq i}$ in R and create A as a component of $(V, E_{\leq i} \setminus R)$ (and hence contribute $w_{i+1} - w_i$ to $h(R)$). Formally, the tree Γ has a node $v^{A,i}$ for every component $A \in \mathcal{A}_i$ and all $i = 0, \ldots, k$. For $i > 0$, the children of $v^{A,i}$ are the nodes $\{v^{S,i-1} : S \in \mathcal{A}_{i-1}, S \subseteq A\}$. Thus, Γ has depth k and root $r = v^{V,k}$. Recall that $L_i := L_i(\Gamma)$ denotes the set of nodes of Γ at depth i, which correspond to the components in $\mathcal{A}_{k-i}$ here. Let N be the set of non-root nodes of Γ.

For a node $v^{A,i} \in N$ (so $0 \leq i < k$), define its value $\alpha_{v^{A,i}} := w_{i+1} - w_i$. Let $R(v^{A,i}) := \delta(A) \cap E_i$ (which is $\emptyset$ for every leaf $v^{A,0}$). Define the weight of $v^{A,i}$ to be $\beta_{v^{A,i}} := c(R(v^{A,i}))$. For $N' \subseteq N$, let $R(N') := \bigcup_{q \in N'} R(q)$. Observe that $R(N) \subseteq R_2$. We set the budget of the tree-knapsack instance to B, the budget for MST interdiction.

The intuition is that we want to encode that picking node $v^{A,i}$ corresponds to creating component A in the multigraph $(V, E_{\leq i} \setminus R)$, where R is our interdiction set, in which case $\alpha_{v^{A,i}}$ gives the contribution from A to $h(R)$. However, in order to pay for the interdiction cost $c(\delta(A))$ incurred, we need to take the β_q weights of all nodes q in the subtree rooted at $v^{A,i}$. Therefore, we insist that if we pick $v^{A,i}$ then we pick all its descendants (i.e., we pick a downwards-closed set of nodes), and then $\sum_{q \in \Gamma(v^{A,i})} \alpha_q$ gives the contribution from the components created to $h(R)$. Lemma 4.3 formalizes this intuition, and shows that if $N' \subseteq N$ is a downwards-closed set of nodes, then $\beta(N')$ and $\alpha(N')$ are good proxies (roughly speaking) for the interdiction cost $c(R(N'))$ incurred and $h(R(N'))$ respectively.

Lemma 4.3. *Let $N' \subseteq N$ be downwards closed, and $R = R(N')$. Then*
(i) $\beta(N')/2 \leq c(R) \leq \beta(N')$; *and*

(ii) $h(R) = \mathrm{val}(R) + w_k \geq \alpha(N') + \sum_{0 \leq i \leq k-1 : L_{k-i} \setminus N' \neq \emptyset}(w_{i+i} - w_i)$.

Proof. Each edge in R appears in at least one, and at most two, of the sets $\{R(q)\}_{q \in N'}$, so we obtain $\frac{1}{2}\sum_{q \in N'} c(R(q)) \le c(R) \le \sum_{q \in N'} c(R(q))$. Part (i) follows by noting that $\beta(N') = \sum_{q \in N'} c(R(q))$.

For part (ii), consider an index $0 \le i \le k-1$. Since N' is downwards closed, for every node $v^{A,i} \in N'$, all descendants of $v^{A,i}$ are in N'; so $R \supseteq \delta(A) \cap E_{\le i}$ and A is a connected component of the multigraph $(V, E_{\le i} \setminus R)$. Further, note that if $L_{k-i} \setminus N' \neq \emptyset$, then the sets $\{A : v^{A,i} \in N'\}$ do not cover V entirely, and so $(V, E_{\le i} \setminus R)$ must have at least one additional connected component. It follows that $(V, E_{\le i} \setminus R)$ always has at least $\min\{|N' \cap L_{k-i}| + 1, |L_{k-i}|\}$ connected components. Plugging this in Lemma 2.2 yields the result. $\qquad\square$

Lemma 4.4. *The vector* $\hat{x} := \left(\hat{x}_q = \frac{b}{2}\right)_{q \in N}$ *is a feasible solution to* (TK-P) *for the above tree-knapsack instance* $(\Gamma, \{\alpha_q\}, \{\beta_q\}, B)$. *Hence,* $OPT_{TK\text{-}P} \ge \frac{b}{2} \cdot h(R_2)$.

Proof. It is clear that $\hat{x}$ satisfies (1), and $0 \le \hat{x}_q \le 1$ for all $q \in N$. Applying Lemma 4.3 to $N' = N$ (which is indeed downwards-closed), we obtain $\beta(N) \le 2c(R(N)) \le 2c(R_2)$. So $\sum_{q \in N} \beta_q \hat{x}_q \le b \cdot c(R_2) \le a \cdot c(R_1) + b \cdot c(R_2) = B$. Finally, $OPT_{TK\text{-}P}$ is at least the objective value of $\hat{x}$, which is $\frac{b}{2} \cdot \alpha(N) = \frac{b}{2} \cdot h(R_2)$. $\qquad\square$

Given this translation between the tree-knapsack and MST-interdiction problems, it is easy to see that Corollary 3.3 (coupled with Lemmas 4.3 and 4.4) yields the following guarantee, which directly leads to an improved approximation guarantee of 5 for MST interdiction (see Claim 4.6).

Lemma 4.5. *We can obtain a feasible interdiction set R such that* $h(R) \ge \frac{b}{2} \cdot h(R_2)$.

Proof. This is consequence of Corollary 3.3, Lemma 4.3, and Lemma 4.4. Let $N' \subseteq N$ be the downwards-closed set corresponding to the integer solution returned by Corollary 3.3. Let $R = R(N')$. We have $\alpha(N') \ge \frac{b}{2} \cdot h(R_2) - \max_{\text{chains } C \subseteq N} \left\{\sum_{i \ge 1 : \tilde{x}(L_i) < |L_i|} \alpha(C_i)\right\}$ by Corollary 3.3 and Lemma 4.4. To complete the proof, we apply part (ii) of Lemma 4.3, noting that

$$\sum_{0 \le i \le k-1 : L_{k-i} \setminus N' \neq \emptyset} (w_{i+i} - w_i) = \max_{\text{chains } C \subseteq N} \left\{\sum_{i \ge 1 : \tilde{x}(L_i) < |L_i|} \alpha(C_i)\right\} \qquad\square$$

Claim 4.6. *We have* $\max\{w_k, h(R_1) - w_k, \frac{b}{2} \cdot h(R_2) - w_k\} \ge UB/5 \ge OPT/5$.

Proof. We have

$$\max\left\{w_k, h(R_1) - w_k, \tfrac{b}{2} \cdot h(R_2) - w_k\right\} \ge \tfrac{2-b}{5-2b} \cdot w_k + \tfrac{1-b}{5-2b} \cdot \left(h(R_1) - w_k\right)$$
$$+ \tfrac{2}{5-2b} \cdot \left(\tfrac{b}{2} \cdot h(R_2) - w_k\right)$$
$$= \frac{1}{5-2b}\left(ah(R_1) + bh(R_2) - w_k\right)$$
$$= \frac{UB}{5-2b} \ge UB/5 \ge OPT/5. \qquad\square$$

Theorem 4.7. *There is a 5-approximation algorithm for MST interdiction.*

Proof. If Theorem 2.3 returns an optimal solution, we are done. Otherwise, we return the best among a min-cut of $(V, E_{\le k-1})$ (which has value at least w_k), the set R_1, and the interdiction set returned by Lemma 4.5. The proof now follows from Claim 4.6. $\qquad\square$

4.1 Improvement to the guarantee stated in Theorem 4.1

The improved approximation guarantee of 4 comes from the fact that instead of focusing only on R_2, we now *interpolate* between R_1 and R_2 to obtain our interdiction set R, i.e., we return R such that $R_1 \subseteq R \subseteq R_2$. Since we always include R_1, we change the definition of the tree-knapsack instance that we create accordingly. The tree Γ and the node weights $\{\alpha_q\}$ are unchanged; the weight of $v^{A,i}$ is now $\beta^{\mathsf{new}}_{v^{A,i}} := c\big(R^{\mathsf{new}}(v^{A,i})\big)$, where $R^{\mathsf{new}}(v^{A,i}) := R(v^{A,i}) \setminus R_1 = \big(\delta(A) \setminus R_1\big) \cap E_i$, and our budget is $B^{\mathsf{new}} := B - c(R_1)$. For $N' \subseteq N$, define $R^{\mathsf{new}}(N') := R_1 \cup \bigcup_{q \in N'} R^{\mathsf{new}}(q)$. Observe that $R^{\mathsf{new}}(N) \subseteq R_2$.

Since $R_1 \subseteq R_2$, each component U of $(V, E_{\leq i} \setminus R_1)$ is a union of components of $(V, E_{\leq i} \setminus R_2)$, and hence, maps to a subset S of the nodes of Γ at depth $k - i$. We exploit the fact that since we include R_1 in our interdiction set, if we pick ℓ nodes from S, then we create $\min\{\ell + 1, |S|\}$ components within U; this $+1$ term that we accrue (roughly speaking) from all components of $(V, E_{\leq j} \setminus R)$ over all $j = 0, \ldots, k - 1$ is the source of our improvement.

The following variant of Corollary 3.3 exploits the structure of the tree-knapsack instance obtained from the MST-interdiction problem, which we then utilize to obtain an interdiction set with an improved bound on $h(R)$ (Lemma 4.9).

Lemma 4.8. *Let* $\big(\Gamma, \{\alpha_v\}, \{\beta_v\}, B\big)$ *be an instance of the tree knapsack problem such that* $\alpha_v = \alpha^{(i)}$ *for all* $v \in L_i(\Gamma)$ *and all* $i \geq 1$. *Let* $\mathcal{S}_i$ *be a partition of* $L_i(\Gamma)$ *for all* $i \geq 1$. *Let* $\theta \in [0, 1]$ *be such that* $(\hat{x}_q = \theta)_{q \in N}$ *is a feasible solution to* (TK-P). *We can obtain in polytime an integer solution* $\tilde{x}$ *to* (TK-P) *such that*

$$\sum_{i \geq 1} \sum_{S \in \mathcal{S}_i} \alpha^{(i)} \min\{\tilde{x}(S) + 1, |S|\} \geq \sum_{i \geq 1} \alpha^{(i)} |L_i| \theta + \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} \Big((1 - \theta) |\mathcal{S}_i| - 1 \Big).$$

Proof. Define $g(x) := \sum_{i \geq 1} \sum_{S \in \mathcal{S}_i} \alpha^{(i)} \min\{x(S) + 1, |S|\}$. As usual L_i denotes $L_i(\Gamma)$. Define $\alpha' \in \mathbb{R}^N_+$ as follows. For each i such that $|\mathcal{S}_i| = 1$, set $\alpha'_v = \alpha_v$ for all $v \in L_i$. For each i with $|\mathcal{S}_i| > 1$ and each $S \in \mathcal{S}_i$, pick some node $v_S \in S$; set $\alpha'_{v_S} = 0$ and $\alpha'_v = \alpha_v$ for all $v \in S \setminus \{v_S\}$. We claim that for $\tilde{x} \in \{0, 1\}^N$, we have

$$g(\tilde{x}) \geq \sum_{v \in N} \alpha'_v \tilde{x}_v + \sum_{\substack{i \geq 1 : |\mathcal{S}_i| = 1, \\ \tilde{x}(L_i) < |L_i|}} \alpha^{(i)} + \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} |\mathcal{S}_i| . \tag{3}$$

To see this, consider any level $i \geq 1$. If $|\mathcal{S}_i| = 1$, the total contribution from this level to $g(\tilde{x})$ is $\alpha^{(i)} \min\{\tilde{x}(L_i) + 1, |L_i|\}$, which is the same as the contribution from this level to the RHS of (3). If $|\mathcal{S}_i| > 1$, consider each set $S \in \mathcal{S}_i$. The contribution from S to $g(\tilde{x})$ is $\alpha^{(i)} \min\{\tilde{x}(S) + 1, |S|\}$, and the contribution from S to the RHS of (3) is $\alpha^{(i)} \big(\tilde{x}(S \setminus \{v_S\}) + 1\big)$, which is no larger.

To complete the proof, note that by Corollary 3.3, we can obtain an integer solution $\tilde{x}$ to (TK-P) such that

$$\sum_{v \in N} \alpha'_v \tilde{x}_v \geq \Big(\max_{\text{feasible solutions } x \text{ to (TK-P)}} \sum_{v \in N} \alpha'_v x_v \Big) - \max_{\text{chains } C \subseteq N} \sum_{i \geq 1 : \tilde{x}(L_i) < |L_i|} \alpha'(C_i)$$

$$\geq \sum_{v \in N} \alpha'_v \hat{x}_v - \sum_{i \geq 1 : \tilde{x}(L_i) < |L_i|} \alpha^{(i)}$$

$$= \sum_{i \geq 1} \alpha^{(i)} |L_i| \theta - \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} |\mathcal{S}_i| \theta - \sum_{i \geq 1 : \tilde{x}(L_i) < |L_i|} \alpha^{(i)}.$$

11

Therefore,

$$
\begin{aligned}
g(\tilde{x}) &\geq \sum_{v \in N} \alpha'_v \tilde{x}_v + \sum_{\substack{i \geq 1 : |\mathcal{S}_i| = 1, \\ \tilde{x}(L_i) < |L_i|}} \alpha^{(i)} + \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} |\mathcal{S}_i| \\
&\geq \sum_{i \geq 1} \alpha^{(i)} |L_i| \theta + \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} |\mathcal{S}_i| (1 - \theta) - \sum_{\substack{i \geq 1 : |\mathcal{S}_i| > 1, \\ \tilde{x}(L_i) < |L_i|}} \alpha^{(i)} \\
&\geq \sum_{i \geq 1} \alpha^{(i)} |L_i| \theta + \sum_{i \geq 1 : |\mathcal{S}_i| > 1} \alpha^{(i)} \Big((1 - \theta) |\mathcal{S}_i| - 1 \Big). \qquad \square
\end{aligned}
$$

Lemma 4.9. *We can obtain a feasible interdiction set R such that $h(R) \geq \frac{a}{2} \cdot h(R_1) + \frac{b}{2} \cdot h(R_2) - \frac{a}{2} \cdot w_k$.*

Proof. We apply Lemma 4.8 to the tree-knapsack instance given by $\big(\Gamma, \{\alpha_q\}, \{\beta_q^{\text{new}}\}, B^{\text{new}}\big)$, but we need to specify the partitions $\mathcal{S}_i$ for all levels $i \geq 1$, and the value θ.

For $i = 0, \ldots, k$, let $\mathcal{B}_i$ denote the partition of V induced by the connected components of the multigraph $(V, E_{\leq i} \setminus R_1)$. Since $R_1 \subseteq R_2$, the partition $\mathcal{A}_i$ refines (not necessarily strictly) the partition $\mathcal{B}_i$ for all $i = 0, \ldots, k$. The components in $\mathcal{B}_{k-i}$ therefore naturally induce a partition $\mathcal{S}_i$ of the nodes of Γ at depth i, consisting of the sets $\{v^{A, k-i} : A \in \mathcal{A}_{k-i}, A \subseteq S\}_{S \in \mathcal{B}_{k-i}}$.

We apply Lemma 4.8 to the tree-knapsack instance $\big(\Gamma, \{\alpha_q\}, \{\beta_q^{\text{new}}\}, B^{\text{new}}\big)$, taking $\alpha^{(i)} = w_{k-i+1} - w_{k-i}$ and $\mathcal{S}_i$ to be the partition defined above, for all $i = 1, \ldots, k$, and $\theta = \frac{b}{2}$. We need to show that $\hat{x} := (\hat{x}_q = \theta)_{q \in N}$ is a feasible solution to (TK-P) for this tree-knapsack instance. This follows because $\beta^{\text{new}}(N) = 2c\big(\bigcup_{q \in N} R^{\text{new}}(q)\big) \leq 2\big(c(R_2) - c(R_1)\big)$ and we have $(1 - b) \cdot c(R_1) + b \cdot c(R_2) = B$, so

$$
\sum_q \beta_q^{\text{new}} \hat{x}_q = \theta \beta^{\text{new}}(N) \leq b\big(c(R_2) - c(R_1)\big) = B - c(R_1) = B^{\text{new}}.
$$

Let $\tilde{x}$ be the integer solution returned by Lemma 4.8, which specifies a downwards-closed set $N' \subseteq N$. Let $R = R^{\text{new}}(N')$. We first show that, analogous to Lemma 4.3, R is feasible, and $h(R) \geq g(\tilde{x}) := \sum_{i \geq 1} \sum_{S \in \mathcal{S}_i} \alpha^{(i)} \min\{\tilde{x}(S) + 1, |S|\}$. We have

$$
\begin{aligned}
c(R) = c(R_1) + c\Big(\bigcup_{q \in N'} R^{\text{new}}(q) \Big) &\leq c(R_1) + \sum_{q \in N'} c\big(R^{\text{new}}(q) \big) \\
&= c(R_1) + \beta^{\text{new}}(N') \leq c(R_1) + B^{\text{new}} = B.
\end{aligned}
$$

Consider any index $0 \leq i \leq k - 1$. As in the proof of part (ii) of Lemma 4.3, for every node $v^{A,i} \in N'$, we know that A is a component of $(V, E_{\leq i} \setminus R)$. Consider any $S \in \mathcal{S}_{k-i}$, and let $U = \bigcup_{v^{A,i} \in S} A$. Note that if $S \setminus N' \neq \emptyset$, then $\bigcup_{v^{A,i} \in S \setminus N'} A$ is non-empty. So there are always at least $\min\{|N' \cap S| + 1, |S|\}$ components of $(V, E_{\leq i} \setminus R)$ contained in U. Therefore, by Lemma 2.2 (and since $\mathcal{S}_i$ is a partition of L_i for each i), we obtain

$$
h(R) = \text{val}(R) + w_k \geq \sum_{i=0}^{k-1} \sum_{S \in \mathcal{S}_{k-i}} (w_{i+1} - w_i) \min\{|N' \cap S| + 1, |S|\} = g(\tilde{x}).
$$

The guarantee in Lemma 4.8 then yields the following. Recall that $a = 1 - b$.

$$h(R) \geq \sum_{i=0}^{k-1} (w_{i+1} - w_i)\sigma(E_{\leq i} \setminus R_2) \cdot \tfrac{b}{2} + \sum_{\substack{i=0,\ldots,k-1: \\ \sigma(E_{\leq i} \setminus R_1) > 1}} (w_{i+1} - w_i)\left[\left(1 - \tfrac{b}{2}\right)\sigma(E_{\leq i} \setminus R_1) - 1\right]$$

$$\geq \tfrac{b}{2} \cdot h(R_2) + \sum_{\substack{i=0,\ldots,k-1: \\ \sigma(E_{\leq i} \setminus R_1) > 1}} (w_{i+1} - w_i)\sigma(E_{\leq i} \setminus R_1) \cdot \tfrac{a}{2} \tag{4}$$

$$= \tfrac{b}{2} \cdot h(R_2) + \sum_{i=0}^{k-1} (w_{i+1} - w_i)\sigma(E_{\leq i} \setminus R_1) \cdot \tfrac{a}{2} - \sum_{\substack{i=0,\ldots,k-1: \\ \sigma(E_{\leq i} \setminus R_1) = 1}} (w_{i+1} - w_i) \cdot \tfrac{a}{2}$$

$$\geq \tfrac{a}{2} \cdot h(R_1) + \tfrac{b}{2} \cdot h(R_2) - \tfrac{a}{2} \cdot w_k.$$

Inequality (4) follows since $t\left(1 - \tfrac{b}{2}\right) - 1 \geq t(1 - b)/2$ for all $t \geq 2$. $\qquad\square$

Proof of Theorem 4.1. We either return an optimal solution found by Theorem 2.3, or return the better of a min-cut of $(V, E_{\leq k-1})$ and the interdiction set returned by Lemma 4.9. We obtain a solution of value

$$\max\left\{w_k, \tfrac{a}{2} \cdot h(R_1) + \tfrac{b}{2} \cdot h(R_2) - \left(1 + \tfrac{a}{2}\right)w_k\right\}$$

$$\geq \frac{1+a}{3+a} \cdot w_k + \frac{2}{3+a} \cdot \left(\tfrac{a}{2} \cdot h(R_1) + \tfrac{b}{2} \cdot h(R_2) - \left(1 + \tfrac{a}{2}\right)w_k\right)$$

$$= \frac{ah(R_1) + bh(R_2) - w_k}{3+a} = \frac{UB}{3+a} \geq UB/4 \geq OPT/4. \qquad\square$$

4.2 Lower bound on the approximation ratio achievable relative to UB

We show that for every $\epsilon > 0$, there exist MST-interdiction instances, where $UB/OPT \geq 3 - \epsilon$. This implies that one cannot achieve an approximation ratio better than 3 when comparing against the upper bound UB used in our analysis (and the one in [30]).

Theorem 4.10. *For any fixed $\epsilon > 0$, there exists an instance of MST interdiction where $UB/OPT \geq 3 - \epsilon$.*

Proof. Our instance is a graph $G = (V, E)$, where $V := \{v_1, \ldots, v_n\}$ with $n \geq \min\{4, 4/\epsilon\}$. The edge set is $E = E_1 \cup E_2$, where $E_1 := \{v_1v_2, v_2v_3, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_1\}$ is a simple cycle on $v_1, \ldots, v_{n-1}$, and $E_2 := \{v_1v_n, v_2v_n, \ldots, v_{n-1}v_n\}$ is a star rooted at v_n with leaves $v_1, \ldots, v_{n-1}$. The edges in E_1 have weight $w_1 = 0$ and interdiction cost n, while the edges in E_2 have weight $w_2 = 1$ and interdiction cost $2n$. The interdiction budget is $B = 2n - 2$.

Observe that the quantity k, as defined in Claim 2.1, is equal to 2. Taking $R = \emptyset$, the graph $(V, E_{\leq 1} \setminus R)$ is disconnected, so $k \geq 2$. Any feasible interdiction set R contains at most one edge from E_1 and no edges from E_2, so $(V, E_{\leq 2} \setminus R)$ is connected, and therefore $k = 2$.

This also implies that $\text{val}(R) \leq 1$ for any feasible interdiction set R: since $R \subseteq E_1$ and $|R \cap E_1| \leq 1$, we can construct a spanning tree of $G - R$ by taking $n - 2$ edges from $E_1 \setminus R$ and any edge from E_2. So $OPT = 1$.

Now we proceed to compute the upper bound UB. For $R \subseteq E_{\leq 1}$, we have $f_\lambda(R) = 1$ if $R = \emptyset$, and $|R|(1 - n\lambda)$ otherwise. Therefore,

$$\eta(\lambda) := \lambda B + \max_{R \subseteq E_{\leq 1}} f_\lambda(R) = \lambda B + \max\{1, (n-1)(1-n\lambda)\} = \max\left\{\lambda B + 1, (n-1) - \lambda\big(n(n-1) - B\big)\right\},$$

which is minimized at $\lambda = \frac{n-2}{n(n-1)}$. Therefore $UB := \min_{\lambda \geq 0} \eta(\lambda) = \frac{2(n-2)}{n} + 1 = 3 - \frac{4}{n} \geq (3 - \epsilon)OPT$. $\qquad\square$

13

5 Extension to metric-TSP interdiction

In the *metric-TSP interdiction* problem, we are given a complete graph $G = (V, E)$ with metric edge weights $\{w_e\}_{e \in E}$ and nonnegative interdiction costs $\{c_e\}_{e \in E}$, along with a nonnegative budget B. The goal is to find a set of edges $R \subseteq E$ such that $c(R) \leq B$ so as to maximize the minimum w-weight of a closed walk in the graph $G - R$ that visits each vertex at least once. Zenklusen [30] observed that an α-approximation algorithm for the MST interdiction problem yields a 2α-approximation algorithm for the metric-TSP interdiction problem. As a corollary to our Theorem 4.1, we therefore obtain the following result.

Theorem 5.1. *There is an 8-approximation algorithm for the metric-TSP interdiction problem.*

6 Maximum-spanning-tree interdiction

We now consider the *maximum-spanning-tree interdiction* problem, wherein the input $\big(G = (V, E), \{w_e \geq 0\}_{e \in E}, \{c_e \geq 0\}_{e \in E}, B\big)$ is the same as in the MST interdiction problem, but the goal is to remove a set $R \subseteq E$ of edges with $c(R) \leq B$ so as to *minimize* the w-weight of a *maximum spanning tree* of $G - R$. We show that this problem is at least as hard as the minimization version of the *densest-k-subgraph* problem (MinDkS), wherein we seek a minimum-size set S of nodes in a given graph such that at least k edges have both endpoints in S. This shows a stark contrast between MST interdiction and maximum-spanning-tree (MaxST) interdiction.

Theorem 6.1. *An $\alpha(m, n)$-approximation algorithm for the maximum-spanning-tree interdiction problem for instances with m edges, n nodes, yields a $2\alpha(m + n - 1, n)$-approximation algorithm for MinDkS for instances with m edges and n nodes.*

Proof. Let $\mathcal{I} = \big(H = (N, F), k\big)$ be a MinDkS instance, with $|N| = n$, $|F| = m$. We may assume that $|F| \geq k$ as otherwise the instance is infeasible. We construct the following instance $\mathcal{I}'$ of the maximum-spanning-tree interdiction problem. The underlying multigraph is $G = (N, E := E' \cup F)$, where E' is an arbitrary tree spanning N. Define $w_e = 0$ for all $e \in E'$, and $w_e = 1$ for all $e \in F$. The interdiction costs are $c_e = 1$ for all $e \in F$, and $c_e = m - k + 1$ for all $e \in E'$. Finally, we set the budget to $B = m - k$. Thus, if R is a feasible interdiction set, we must have $R \subseteq F$, and so $G - R$ is connected and the interdiction problem has a finite optimal value.

We show that: (1) if $R \subseteq F$ is a feasible interdiction set, then the set S of non-isolated nodes of $(N, F \setminus R)$ is a feasible MinDkS solution of value at most $2 \cdot \mathsf{MaxST}(G - R)$, where $\mathsf{MaxST}(G - R)$ is the weight of a maximum spanning tree of $G - R$; (2) conversely, if $S \subseteq N$ is a feasible MinDkS solution, then $F \setminus F(S)$ is a feasible interdiction set with objective value at most $|S|$, where $F(S)$ is the set of edges in F having both endpoints in S.

These two statements imply the theorem as follows. Let $\mathcal{A}$ be the stated $\alpha = \alpha(m + n - 1, n)$-approximation algorithm for maximum-spanning-tree interdiction. We run $\mathcal{A}$ to obtain a feasible interdiction set R, which yields a corresponding MinDkS solution S. Then,

$$|S| \leq 2 \cdot \mathsf{MaxST}(G - R) \leq 2\alpha\, OPT(\mathcal{I}') \leq 2\alpha\, OPT(\mathcal{I})\,,$$

where the first and last inequalities follow from statements (1) and (2) above.

We now prove statements (1) and (2). Let $R \subseteq F$ be such that $c(R) = |R| \leq B$. Let S denote the set of non-isolated vertices in the graph $(N, F \setminus R)$, so every node in S has at least one edge of $F \setminus R$ incident to it. First, we argue that S is a feasible MinDkS-solution. Since each vertex of $N \setminus S$ is isolated in the graph $(N, F \setminus R)$, it follows that $R \supseteq F \setminus F(S)$. Therefore, $|F| - |F(S)| \leq |R| \leq B = m - k$, and so

14

$|F(S)| \geq k$. The weight of a maximum spanning tree in $G - R$ is equal to $|S| - \sigma$, where σ is the number of connected components of the graph $(S, F \setminus R)$. By the definition of S, this multigraph has no isolated vertices. So $\sigma \leq |S|/2$, and therefore $\mathsf{MaxST}(G - R) = |S| - \sigma \geq |S|/2$. This proves (1).

Conversely, suppose $S \subseteq N$ is such that $|F(S)| \geq k$. Then $R = F \setminus F(S)$ satisfies $c(R) = m - |F(S)| \leq B$, so is a feasible interdiction set. We have $\mathsf{MaxST}(G - R) = |S| - \sigma \leq |S|$, where σ is the number of connected components of $(S, F \setminus R)$. This proves (2). $\qquad\square$

The above hardness result continues to hold with unit interdiction costs, since we can replace each edge e with $c_e = m - k + 1$ in the above reduction with $m - k + 1$ parallel unit-cost edges (of weight 0). Our reduction creates a MaxST-interdiction instance with only two distinct edge weights $w_1 < w_2$. This interdiction problem can be seen as a special case of the following *matroid interdiction* problem (involving the graphic matroid on the ground set $\{e \in E : w_e = w_2\}$): given a matroid with ground set U and rank function rk, interdiction costs $c : U \mapsto \mathbb{R}_+$, and budget B, minimize $\mathrm{rk}(U \setminus R)$ subject to $c(R) \leq B$. Our hardness result for MaxST interdiction thus also implies that matroid interdiction is $\mathsf{MinD}k\mathsf{S}$-hard. A related rank-reduction problem—minimize $c(R)$ subject to $\mathrm{rk}(U \setminus R) \leq \mathrm{rk}(U) - k$—was considered by [16] and shown to be $\mathsf{MinD}k\mathsf{S}$-hard for transversal matroids (but not for graphic matroids, wherein this is essentially the min k-cut problem).

We remark that it is possible to achieve *bicriteria* approximation guarantees for MaxST interdiction: we can obtain a solution of weight $W \leq (1 + \epsilon)OPT$ while violating the budget by a $\left(1 + \frac{1}{\epsilon}\right)$ factor (and $W > OPT$ implies no budget violation). This follows by taking $\lambda = \epsilon OPT/B$ in the Lagrangian problem $\min_R \big(\mathsf{MaxST}(G - R) + \lambda c(R) \big)$, which is a submodular minimization problem that can be solved exactly; it also follows from the work of [4].

References

[1] N. Assimakopoulos. A network interdiction model for hospital infection control. *Computers in Biology and Medicine*, 17(6):413–422, 1987.

[2] C. Burch, R. Carr, S. Krumke, M. Marathe, C. Phillips, and E. Sundberg. A decomposition-based pseudoapproximation algorithm for network flow inhibition. In *Network Interdiction and Stochastic Integer Programming*, Chapter 3, pages 51–68. Springer, 2003.

[3] S. Chestnut and R. Zenklusen. Hardness and approximation for network flow interdiction. *CS arXiv*, November 2015.

[4] S. Chestnut and R. Zenklusen. Interdicting structured combinatorial optimization problems with $\{0, 1\}$-objectives. *CS arXiv*, November 2015.

[5] R. L. Church, M. P. Scaparra, and R. S. Middleton. Identifying critical infrastructure: the median and covering facility interdiction problems. *Annals of the Association of American Geographers*, 94(3):491–502, 2004.

[6] M. Dinitz and A. Gupta. Packing interdiction and partial covering problems. In *Proceedings of 16th IPCO*, pages 157–168, 2013.

[7] A. Engelberg, J. Könemann, S. Leonardi, and J. Naor. Cut problems in graphs with a budget constraint. *Journal of Discrete Algorithms*, 5:262–279, 2007.

[8] L. Fleischer and S. Iwata. A push-relabel framework for submodular function minimization and applications to parametric optimization. Discrete Applied Mathematics, 131(2):311–322, 2003.

[9] D. R. Fulkerson and G. C. Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Math. Programming*,13:116–118, 1977.

[10] G. N. Frederickson and R. Solis-Oba. Increasing the weight of minimum spanning trees. *Journal of Algorithms*, 33:244–266, 1999.

[11] P. M. Ghare, D. C. Montgomery, and W. C. Turner. Optimal interdiction policy for a flow network. *Naval Research Logistics Quarterly*, 18:37–45, 1971.

[12] F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.

[13] G. Guruganesh, L. Sanità, and C. Swamy. Improved region-growing and combinatorial algorithms for k-route cut problems. In *SODA*, pages 676–695, 2015.

[14] E. Israeli and R. K. Wood. Shortest-path network interdiction, *Networks*, 40:97–111, 2002.

[15] D. S. Johnson and K. A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Math. of Oper. Research*, 8(1):1–14, 1983.

[16] G. Joret and A. Vetta. Reducing the rank of a matroid. *Discrete Mathematics & Theoretical Computer Science*, 17(2):143–156, 2015.

[17] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf, and J. Zhao. On short paths interdiction problems: total and node-wise limited interdiction. *Theoretical Computer Science*, 43(2):204–233, 2008.

[18] S. Kolliopoulos and G. Steiner. Partially-ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, 155(8):889–897, 2007.

[19] J. Könemann, O. Parekh, and D. Segev. A unified approach to approximating partial covering problems. Algorithmica 59(4):489–509 2011.

[20] Euiwoong Lee. Improved hardness for cut, interdiction, and firefighter problems. *CS arXiv*, July 2016.

[21] W. Liang. Finding the k most vital edges with respect to minimum spanning trees for fixed k. *Discrete Applied Mathematics*, 113(2-3):319–327, 2001.

[22] A. Linhares and C. Swamy. Improved algorithms for MST and metric-TSP interdiction. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2017.

[23] K. Liri and M. Chern. The most vital edges in the minimum spanning tree problem. *Information Processing Letters*, 45:25–31, 1993.

[24] K. Nagano. A faster parametric submodular function minimization algorithm and applications. Technical report, University of Tokyo, 2007. METR 2007-43.

[25] F. Pan, W. Charlton, and D. P. Morton. *Stochastic network interdiction of nuclear material smuggling*. In D.L. Woodruff, editor, *Network Interdiction and Stochastic Integer Programming*, pages 1–19, 2002.

[26] C. A. Phillips. The network inhibition problem. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 776–785, 1993.

[27] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd STOC*, pages 755–764, 2010.

[28] J. Salmeron, K. Wood, and R. Baldick. Worst-case interdiction analysis of large-scale electric power grids. *IEEE Trans. on Power Systems*, 24(1):96–104, 2009.

[29] R.K. Wood. Deterministic network interdiction. *Mathematical and Computer Modeling*, 17(2):1–18, 1993.

[30] R. Zenklusen. An $O(1)$-Approximation for Minimum Spanning Tree Interdiction. In *Proceedings of the 56th FOCS*, pages 709–728, 2015.

[31] R. Zenklusen. Matching interdiction. *Discrete App. Math.*, 158:1676–1690, 2010.

[32] R. Zenklusen. Network flow interdiction on planar graphs. *Discrete Applied Mathematics*, 158(13):1441–1455, 2010.

A Proof of Theorem 2.3

For any $\lambda \geq 0$, by supermodularity of f_λ, there is a unique minimal set in $\mathcal{O}_\lambda^*$, which we denote by R_λ^*. We remark that R_λ^* can be computed in polytime using an algorithm for submodular-function minimization (see, e.g., [8]). Let $g(\lambda)$ denote the optimal value of (P$_\lambda$). Then $g(\lambda)$ is the maximum of a finite collection of nonincreasing linear functions, so it is a continuous, piecewise-linear, nonincreasing, convex function. Further, $\lambda \geq 0$ is a *breakpoint* of $g(.)$ iff there are at least two distinct values in $\{c(R) : R \in \mathcal{O}_\lambda^*\}$. Also, if λ is not a breakpoint, then the slope of g at λ is $-c(R_\lambda^*)$.

We first prove the following:
$$\text{if } \lambda > \lambda', \quad \text{then} \quad R_\lambda^* \subseteq R_{\lambda'}^*. \tag{*}$$

Let $R = R_\lambda^*$ and $R' = R_{\lambda'}^*$. Suppose that $R \not\subseteq R'$. We have $f_\lambda(R \cap R') + f_\lambda(R \cup R') \geq f_\lambda(R) + f_\lambda(R')$. Since $R \cap R' \subsetneq R$ (since we assume that R' is not a superset of R), and $R = R_\lambda^*$ is the minimal set in $\mathcal{O}_\lambda^*$, we have $f_\lambda(R \cap R') < f_\lambda(R)$. So we must have $f_\lambda(R \cup R') > f_\lambda(R')$. But then

$$f_{\lambda'}(R \cup R') = f_\lambda(R \cup R') + (\lambda - \lambda')c(R \cup R') > f_\lambda(R') + (\lambda - \lambda')c(R') = f_{\lambda'}(R'),$$

which contradicts that $R' \in \mathcal{O}_{\lambda'}^*$.

Next we present the two approaches for proving the theorem. The first one utilizes binary search and yields a more elementary, but weakly polytime algorithm. The second one utilizes the fact that $g(\lambda)$ has polynomially many breakpoints and slopes, and uses results on parametric submodular-function minimization to obtain $g(\lambda)$ in strongly polynomial time, which then yields the theorem.

If $c(R_0^*) \leq B$, then $\text{val}(R_0^*) \geq OPT$ and we are done, so assume this is not the case in the sequel. Let M be the smallest integer such that all the w_es, c_es, and B are multiples of $\frac{1}{M}$; note that $\log M$ is polynomially bounded. For $\lambda = K := 2w_k(n-1)M$, we must have $c(R_\lambda^*) \leq B$ as otherwise $c(R_\lambda^*) \geq B + \frac{1}{M}$, and so $\text{val}(R_\lambda^*) - \lambda c(R_\lambda^*) \leq -w_k(n-1) - \lambda B < OPT - \lambda B$.

Binary search. As noted above $c(R_K^*) \leq B$. Let $\epsilon := \frac{1}{M^2 c(E)^2}$. We perform binary search in the interval $[0, K]$ to find $\lambda_1, \lambda_2 \in [0, K]$ with $0 < \lambda_1 - \lambda_2 \leq \epsilon/2$ such that $c(R_{\lambda_1}^*) \leq B < c(R_{\lambda_2}^*)$. If $c(R_{\lambda_1}^*) = B$, then we have $\text{val}(R_{\lambda_1}^*) \geq OPT$ and we are done; so assume this does not happen. Let $R_1 := R_{\lambda_1}^*$ and $R_2 := R_{\lambda_2}^*$. Note that since $\lambda_1 > \lambda_2$, we have $R_1 \subseteq R_2$ by (*). The only thing left to prove is that $R_1, R_2 \in \mathcal{O}_\lambda^*$ for some $\lambda \geq 0$.

We claim that any two breakpoints of g are separated by at least ϵ. This is because if λ is a breakpoint and $A, B \in \mathcal{O}_\lambda^*$ are such that $c(A) > c(B)$, then we have $\lambda = \frac{\text{val}(A) - \text{val}(B)}{c(A) - c(B)}$, which can be written as a fraction with integer numerator and positive integer denominator bounded by $Mc(E)$. So the interval

$[\lambda_2, \lambda_1]$ contains at most one breakpoint, and hence, exactly one breakpoint (since $c(R_1) \neq c(R_2)$). Let λ be this breakpoint. Then, g is linear in (λ_2, λ) and (λ, λ_1) with slopes $-c(R_2)$ and $-c(R_1)$ respectively, so $g(\lambda) = \mathrm{val}(R_1) - \lambda c(R_1) = \mathrm{val}(R_2) - \lambda c(R_2)$. Thus, $R_1, R_2 \in \mathcal{O}_\lambda^*$.

Parametric submodular-function minimization. By (*), we know that $\{R_\lambda^* : \lambda \geq 0\}$ is a nested family, and hence consists of at most $|E| + 1$ sets. Thus, g consists of at most $|E| + 1$ linear segments. Since $-f_\lambda$ is submodular, one can use algorithms for parametric submodular-function minimization [8, 24] to obtain the slopes of all these segments (and the corresponding sets in $\mathcal{O}_\lambda^*$) in strongly polynomial time. Now if some slope is equal to $-B$, then the corresponding interdiction set is an optimal solution to the MST-interdiction problem. Otherwise, we have that the slope of g at $\lambda = 0$ is less than $-B$, and the slope at $\lambda = K$ is more than $-B$, so there is some breakpoint λ where g has slope less than $-B$ at λ^- (a value infinitesimally smaller than λ), and more than $-B$ at λ^+ (a value infinitesimally larger than λ). Then, the interdiction sets R_1 and R_2 corresponding to the slopes at λ^+ and λ^- respectively satisfy the theorem. $\square$