12-2009

Thresholded Covering Algorithms for Robust and Max-Min Optimization

Anupam Gupta
Carnegie Mellon University

Viswanath Nagarajan
IBM Watson Research Center

R. Ravi
Carnegie Mellon University

Thresholded Covering Algorithms for Robust and Max-Min Optimization

Anupam Gupta* Viswanath Nagarajan† R. Ravi‡

Abstract

The general problem of robust optimization is this: one of several possible scenarios will appear tomorrow, but things are more expensive tomorrow than they are today. What should you anticipatorily buy today, so that the worst-case cost (summed over both days) is minimized? For example, in a set cover instance, if any one of the $\binom{n}{k}$ subsets of the universe that have size k may appear tomorrow, what is a good course of action? Feige et al. [FJMM07], and later, Khandekar et al. [KKMS08], considered this k*-robust model* where the possible outcomes tomorrow are given by all demand-subsets of size k, and gave algorithms for the set cover problem, and the Steiner tree and facility location problems in this model, respectively.

In this paper, we give the following simple and intuitive template for k-robust problems: *having built some anticipatory solution, if there exists a single demand whose augmentation cost is larger than some threshold (which is $\approx \mathsf{Opt}/k$), augment the anticipatory solution to cover this demand as well, and repeat.* In this paper we show that this template gives us approximation algorithms for k-robust versions of Steiner tree and set cover (improving on the performance ratios of the previously known algorithms, by a logarithmic factor in the case of set cover), and present the first approximation algorithms for k-robust Steiner forest, minimum-cut and multicut. All our approximation ratios (except for multicut) are almost best possible. In addition to its apparent simplicity and efficacy, a salient feature of our framework is the interesting set of technical ideas—in particular, dual-rounding ideas—needed to show good performance of this heuristic.

As a by-product of our techniques, we get algorithms for max-min problems of the form: "*given a covering problem instance, which k of the elements are costliest to cover?*" If the covering problem does not naturally define a submodular function, very little is known about these problems. For the problems mentioned above (set cover, multicut, Steiner forest), we show that their k-max-min versions have performance guarantees similar to those for the k-robust problems, which are optimal in many cases. E.g., for set cover, we give an $O(\log m + \log n)$ approximation, which improves the previous result by a logarithmic factor, and nearly matches the $\Omega(\frac{\log m}{\log \log m} + \log n)$ hardness.

*Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Supported in part by NSF awards CCF-0448095 and CCF-0729022, and an Alfred P. Sloan Fellowship. Email: anupamg@cs.cmu.edu

†IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA. Email: viswanath@us.ibm.com

‡Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Supported in part by NSF grant CCF-0728841. Email: ravi@cmu.edu

1 Introduction

Consider the following k-*robust* set cover problem: we are given a set system $(U, \mathcal{F} \subseteq 2^U)$. Tomorrow some set of k elements $S \subseteq U$ will want to be covered; however, today we don't know what this set will be. One strategy is to wait until tomorrow and buy an $O(\log n)$-approximate set cover for this set. However, sets are cheaper today: they will cost λ times as much tomorrow as they cost today. Hence, it may make sense to buy some anticipatory partial solution today (i.e. in the first-stage), and then complete it tomorrow (i.e. second-stage) once we know the actual members of the set S. Since we do not know anything about the set S (or maybe we are risk-averse), we want to plan for the worst-case, and minimize:

$$(\text{cost of anticipatory solution}) + \lambda \cdot \max_{S:|S|\leq k} (\text{additional cost to cover } S).$$

Early approximation results for robust problems [DGRS05, GGR06] had assumed that the collection of possible sets S was explicitly given (and the performance guarantee depended logarithmically on the size of this collection). Since this seemed quite restrictive, Feige et al. [FJMM07] proposed this k-robust model where any of the $\binom{n}{k}$ subsets S of size k could arrive. Though this collection of possible sets was potentially exponential sized (for large values of k), the hope was to get results that did not depend polynomially on k. To get such an algorithm, Feige et al. considered the k-max-min set-cover problem ("which subset $S \subseteq U$ of size k requires the largest set cover value?"), and used the online algorithm for set cover to get a greedy-style $O(\log m \log n)$ approximation for this problem; here m and n are the number of sets and elements in the set system. They then used this max-min problem as a separation oracle in an LP-rounding-based algorithm (à la [SS04]) to get the same approximation guarantee for the k-robust problem. They also showed the max-min and k-robust set cover problems to be $\Omega(\frac{\log m}{\log \log m} + \log n)$ hard—which left a logarithmic gap between the upper and lower bounds. However, an online algorithm based approach is unlikely to close this gap, since the online algorithm for set cover is necessarily a log-factor worse that its offline counterparts [AAA$^+$03].

Apart from the obvious goal of improving the result for this particular problem, one may want to develop algorithms for other k-robust problems. E.g., for the k-*robust min-cut* problem, some set S of k sources will want to be separated from the sink vertex tomorrow, and we want to find the best way to cut edges to minimize the total cost incurred (over the two days) for the worst-case k-set S. Similarly, in the k-*robust Steiner forest*, we are given a metric space and a collection of source-sink pairs, and any set S of k source-sink pairs may desire pairwise connection tomorrow; what should we do to minimize the sum of the costs incurred over the two days? One can extend the Feige et al. framework to first solve max-min problems using online algorithms, but in all cases we seem to lose extra logarithmic factors. Moreover, for the above two problems (and others), the LP-rounding-based framework does not seem to extend directly. The latter obstacle was also observed by Khandekar et al. [KKMS08], who gave constant-factor algorithms for k-robust versions of Steiner tree and facility location.

1.1 Main Results

In this paper, we present a general template to design algorithms for k-robust problems. We improve on previous results, by obtaining an $O(\log m + \log n)$ factor for k-robust set cover, and improving the constant in the approximation factor for Steiner tree. We also give the first algorithms for some other standard covering problems, getting constant-factor approximations for both k-robust Steiner forest—which was left open by Khandekar et al.—and for k-robust min-cut, and an $O(\frac{\log^2 n}{\log \log n})$ approximation for k-robust multicut. Our algorithms do not use a max-min subroutine directly: however, our approach ends up giving us approximation algorithms for k-max-min versions of set cover, Steiner forest, min-cut and multicut; all but the one for multicut are best possible under standard assumptions; the ones for Steiner forest and multicut are the first known algorithms for their k-max-min versions.

An important contribution of our work (even more than the new/improved approximation guarantees) is the simplicity of the algorithms, and the ideas in their analysis. The following is our actual algorithm for k-robust set cover.

Suppose we "guess" that the maximum second-stage cost in the optimal solution is T. Let $A \subseteq U$ be all elements for which the cheapest set covering them costs more than $\beta \cdot T/k$, where $\beta = O(\log m + \log n)$. We build a set cover on A as our first stage. (Say this cover costs C_T.)

To remove the guessing, try all values of T and choose the solution that incurs the least total cost $C_T + \lambda\beta T$. Clearly, by design, no matter which k elements arrive tomorrow, it will not cost us more than $\lambda \cdot k \cdot \beta T/k = \lambda\beta T$ to cover them, which is within β of what the optimal solution pays.

The key step of our analysis is to argue why C_T is close to optimum. We briefly describe the intuition; details appear in Section 3. Suppose $C_T \gg \beta\mathsf{Opt}$: then the fractional solution to the LP for set cover for A would cost $\gg \frac{\beta}{\ln n}\mathsf{Opt} \geq \mathsf{Opt}$, and so would its dual. Our key technical contribution is to show how to *"round" this dual LP* to find a "witness" $A' \subseteq A$ with only k elements, and also a corresponding feasible dual of value $\gg \mathsf{Opt}$—i.e., the dual value does not decrease much in the rounding (This rounding uses the fact that the algorithm only put those elements in A that were expensive to cover). Using duality again, this proves that the optimal LP value, and hence the optimal set cover for these k elements A', would cost much more than Opt—a contradiction!

In fact, our algorithms for the other k-robust problems are almost identical to this one; indeed, the only slightly involved algorithm is that for k-robust Steiner forest. Of course, the proofs to bound the cost C_T need different ideas in each case. For example, directly rounding the dual for Steiner forest is difficult, so we give a primal-dual argument to show the existence of such a witness $A' \subseteq A$ of size at most k. For the cut-problems, one has to deal with additional issues because Opt consists of two stages that have to be charged to separately, and this requires a Gomory-Hu-tree-based charging. Even after this, we still have to show that if the cut for a set of sources A is large then there is a witness $A' \subseteq A$ of at most k sources for which the cut is also large—i.e., we have to somehow aggregate the flows (i.e. dual-rounding for cut problems). We prove new flow-aggregation lemmas for single-sink flows using Steiner-tree-packing results, and for multiflows using oblivious routing [Räc08]; both proofs are possibly of independent interest.

To get a quick overview of our basic approach, see the analysis for Steiner tree in Appendix B. While the result is simple and does not require rounding the dual, it is a nice example of our framework in action. In Sections 2 and 2.1 we present the formal framework for k-robust problems, and abstract out the properties that we'd like from our algorithms. Then Section 3 contains such an algorithm for k-robust set cover—Min-cut and Steiner forest appear in Sections 4 and 5. The algorithm for k-robust multicut is in Appendix E. We present a general reduction from robust problems to the corresponding max-min problems in Appendix F. In that section, we also describe extensions of our work to more general uncertainty sets, e.g. incorporating matroid and knapsack type constraints.

1.2 Related Work

Approximation algorithms for robust optimization was initiated by Dhamdhere et al. [DGRS05]: they study the case when the scenarios were explicitly listed, and gave constant-factor approximations for Steiner tree and facility location, and logarithmic approximations to mincut/multicut problems. Golovin et al. [GGR06] improved the mincut result to a constant factor approximation, and also gave an $O(1)$-approximation for robust shortest-paths. The k-robust model was introduced in Feige et al. [FJMM07], where they gave an $O(\log m \log n)$-approximation for set cover. Khandekar et al. [KKMS08] noted that the techniques of [FJMM07] did not give good results for Steiner tree, and developed new constant-factor approximations for k-robust versions of Steiner tree, Steiner forest on trees and facility location. Using our framework, the algorithm we get for Steiner tree can be viewed as a rephrasing of their algorithm—our proof is arguably more transparent and results in a better bound. Our approach can also be used to get a slightly better ratio than [KKMS08] for the Steiner forest problem on trees.

Constrained submodular maximization problems [NWF78, FNW78, Svi04, CCPV07, Von08] appear very relevant at first sight: e.g., the k-max-min version of min-cut ("find the k sources whose separation from the sink costs the most") is precisely submodular maximization under a cardinality constraint, and hence is approximable to within $(1 - 1/e)$. But apart from min-cut, the other problems do not give us submodular functions to maximize, and massaging the functions to make them submodular seems to lose logarithmic factors. E.g., one can use tree embeddings [FRT04] to reduce Steiner tree to a problem on trees and make it submodular; in other cases, one can use online algorithms to get submodular-like properties (we give a general reduction for covering problems that admit good offline and online algorithms in Appendix F). Eventually, it is unclear how to use existing results on submodular maximization in any general way.

Considering the *average* instead of the worst-case performance gives rise to the well-studied model of stochastic optimization [RS04, IKMM04]. Some common generalizations of the robust and stochastic models have been considered (see, e.g., Swamy [Swa08] and Agrawal et al. [ADSY09]).

Feige et al. [FJMM07] also considered the k-max-min set cover—they gave an $O(\log m \log n)$-approximation algorithm for this, and used it in the algorithm for k-robust set cover. They also showed an $\Omega(\frac{\log m}{\log\log m})$ hardness of approximation for k-max-min (and k-robust) set cover. To the best of our knowledge, none of the k-max-min problems other than min-cut have been studied earlier.

The k-*min-min* versions of covering problems (i.e. "which k demands are the *cheapest* to cover?) have been extensively studied for set cover [Sla97, GKS04], Steiner tree [Gar05], Steiner forest [GHNR07], min-cut and multicut [GNS06, Räc08]. However these problems seem to be related to the k-max-min versions only in spirit.

2 Notation and Definitions

Deterministic covering problems. A covering problem Π has a ground-set E of elements with costs $c : E \to \mathbb{R}_+$, and n covering requirements (often called demands or clients), where the solutions to the i-th requirement is specified—possibly implicitly—by a family $\mathcal{R}_i \subseteq 2^E$ which is upwards closed (since this is a covering problem). Requirement i is *satisfied* by solution $S \subseteq E$ iff $S \in \mathcal{R}_i$. The covering problem $\Pi = \langle E, c, \{\mathcal{R}_i\}_{i=1}^n \rangle$ involves computing a solution $S \subseteq E$ satisfying all n requirements and having minimum cost $\sum_{e \in S} c_e$. E.g., in set cover, "requirements" are items to be covered, and "elements" are sets to cover them with. In Steiner tree, requirements are terminals to connect to the root and elements are the edges; in multicut, requirements are terminal pairs to be separated, and elements are edges to be cut.

Robust covering problems. This problem, denoted Robust(Π), is a *two-stage optimization* problem, where elements are possibly bought in the first stage (at the given cost) or the second stage (at cost λ times higher). In the second stage, some subset $\omega \subseteq [n]$ of requirements (also called a *scenario*) materializes, and the elements bought in both stages must satisfy each requirement in ω. Formally, the input to problem Robust(Π) consists of (a) the covering problem $\Pi = \langle E, c, \{\mathcal{R}_i\}_{i=1}^n \rangle$ as above, (b) a set $\Omega \subseteq 2^{[n]}$ of scenarios (possibly implicitly given), and (c) an inflation parameter $\lambda \geq 1$. A feasible solution to Robust(Π) is a set of *first stage elements* $E_0 \subseteq E$ (bought without knowledge of the scenario), along with an *augmentation algorithm* that given any $\omega \in \Omega$ outputs $E_\omega \subseteq E$ such that $E_0 \cup E_\omega$ satisfies all requirements in ω. The objective function is to minimize: $c(E_0) + \lambda \cdot \max_{\omega \in \Omega} c(E_\omega)$. Given such a solution, $c(E_0)$ is called the first-stage cost and $\max_{\omega \in \Omega} c(E_\omega)$ is the second-stage cost.

k**-robust problems.** In this paper (except in Section F), we deal with robust covering problems under *cardinality* uncertainty sets: i.e., $\Omega := \binom{[n]}{k} = \{S \subseteq [n] \mid |S| = k\}$. We denote this problem by $\mathsf{Robust}_k(\Pi)$.

Max-min problems. Given a covering problem Π and a set Ω of scenarios, the *max-min* problem involves finding a scenario $\omega \in \Omega$ for which the cost of the min-cost solution to ω is maximized. Note that by setting $\lambda = 1$ in any robust covering problem, *the optimal value of the robust problem equals that of its corresponding max-min problem*. In a k-**max-min problem** we have $\Omega = \binom{[n]}{k}$.

2.1 The Abstract Properties we want from our Algorithms

Our algorithms for robust and max-min versions of covering problems are based on the following guarantee.

Definition 2.1 *An algorithm is* $(\alpha_1, \alpha_2, \beta)$-discriminating *iff given as input any instance of* $\mathsf{Robust}_k(\Pi)$ *and a threshold* T*, the algorithm outputs (i) a set* $\Phi_T \subseteq E$*, and (ii) an algorithm* $\mathsf{Augment}_T : \binom{[n]}{k} \to 2^E$*, such that:*

A. For every scenario $D \in \binom{[n]}{k}$,

(i) the elements in $\Phi_T \cup \mathsf{Augment}_T(D)$ *satisfy all requirements in* D*, and*

(ii) the resulting augmentation cost $c\left(\mathsf{Augment}_T(D)\right) \leq \beta \cdot T$.

B. Let Φ^* *and* T^* *(respectively) denote the first-stage and second-stage cost of an optimal solution to the* $\mathsf{Robust}_k(\Pi)$ *instance. If the threshold* $T \geq T^*$ *then the first stage cost* $c(\Phi_T) \leq \alpha_1 \cdot \Phi^* + \alpha_2 \cdot T^*$.

The next lemma (whose proof appears in Appendix A.1) shows why having a discriminating algorithm is sufficient to solve the robust problem. The issue to address is that having guessed T for the optimal second stage cost, we have no direct way of verifying the correctness of that guess—hence we choose the best among all possible values of T. For $T \approx T^*$ the guarantees in Definition 2.1 ensure that we pay $\approx \Phi^* + T^*$ in the first stage, and $\approx \lambda T^*$ in the second stage; for guesses $T \ll T^*$, the first-stage cost in guarantee (2) is likely to be large compared to Opt.

Lemma 2.2 *If there is an* $(\alpha_1, \alpha_2, \beta)$*-discriminating algorithm for a robust covering problem* $\mathsf{Robust}_k(\Pi)$*, then for every* $\epsilon > 0$ *there is a* $\left((1+\epsilon) \cdot \max\left\{\alpha_1, \beta + \frac{\alpha_2}{\lambda}\right\}\right)$*-approximation algorithm for* $\mathsf{Robust}_k(\Pi)$.

In the rest of the paper, we focus on providing discriminating algorithms for suitable values of $\alpha_1, \alpha_2, \beta$.

2.2 Additional Property Needed for k-max-min Approximations

As we noted above, a k-max-min problem is a robust problem where the inflation $\lambda = 1$ (which implies that in an optimal solution $\Phi^* = 0$, and T^* is the k-max-min value). Hence a discriminating algorithm immediately gives an approximation to the *value*: for any $D \in \binom{[n]}{k}$, $\Phi_T \cup \mathsf{Augment}_T(D)$ satisfies all demands in D, and for the right guess of $T \approx T^*$, the cost is at most $(\alpha_2 + \beta)T^*$. It remains to output a bad k-set as well, and hence the following definition is useful.

Definition 2.3 *An algorithm for a robust problem is* strongly discriminating *if it satisfies the properties in Definition 2.1, and when the inflation parameter is* $\lambda = 1$ *(and hence* $\Phi^* = 0$*), the algorithm also outputs a set* $Q_T \in \binom{[n]}{k}$ *such that if* $c(\Phi_T) \geq \alpha_2 T$*, the cost of optimally covering the set* Q_T *is* $\geq T$.

Recall that for a covering problem Π, the cost of optimally covering the set of requirements $Q \in \binom{[n]}{k}$ is $\min\{c(E_Q) \mid E_Q \subseteq E \text{ and } E_Q \in \mathcal{R}_i \ \forall i \in Q\}$. We prove the following in Appendix A.2.

Lemma 2.4 *If there is an* $(\alpha_1, \alpha_2, \beta)$-strongly-*discriminating algorithm for a robust covering problem* $\mathsf{Robust}_k(\Pi)$, *then for every* $\epsilon > 0$ *there is an algorithm for* k*-max-min*(Π) *that outputs a set* Q *such that for some* T*, the optimal cost of covering this set* Q *is at least* T*, but every* k*-set can be covered with cost at most* $(1+\epsilon) \cdot (\alpha_2 + \beta)\, T$.

3 Set Cover

Consider the k-robust set cover problem where there is a set system $(U, \mathcal{F})$ with a universe of $|U| = n$ elements, and m sets in $\mathcal{F}$ with each set $R \in \mathcal{F}$ costing c_R, an inflation parameter λ, and an integer k such that each of the sets $\binom{U}{k}$ is a possible scenario for the second-stage. Given Lemma 2.2, it suffices to show a discriminating algorithm as defined in Definition 2.1 for this problem. The algorithm given below is easy: pick all elements which can only be covered by expensive sets, and cover them in the first stage.

Algorithm 1 Algorithm for k-Robust Set Cover

input: k-robust set-cover instance and threshold T.
let $\beta \leftarrow 36 \ln m$, and $S \leftarrow \left\{v \in U \mid \text{ min cost set covering } v \text{ has cost at least } \beta \cdot \frac{T}{k}\right\}$.
output first stage solution Φ_T as the Greedy-Set-Cover(S).
define $\mathsf{Augment}_T(\{i\})$ as the min-cost set covering i, for $i \in U \setminus S$; and $\mathsf{Augment}_T(\{i\}) = \emptyset$ for $i \in S$.
output second stage solution $\mathsf{Augment}_T$ where $\mathsf{Augment}_T(D) := \bigcup_{i \in D} \mathsf{Augment}_T(\{i\})$ for all $D \subseteq U$.

Claim 3.1 (Property A for Set Cover) *For all* $T \geq 0$ *and scenario* $D \in \binom{U}{k}$*, the sets* $\Phi_T \bigcup \mathsf{Augment}_T(D)$ *cover elements in* D*, and have cost* $c(\mathsf{Augment}_T(D)) \leq \beta\, T$.

Proof: The elements in $D \cap S$ are covered by Φ_T; and by definition of $\mathsf{Augment}_T$, each element $i \in D \setminus S$ is covered by set $\mathsf{Augment}_T(\{i\})$. Thus we have the first part of the claim. For the second part, note that by definition of S, the cost of $\mathsf{Augment}_T(\{i\})$ is at most $\beta\, T/k$ for all $i \in U$. ■

Theorem 3.2 (Property B for Set Cover) *Let Φ^* denote the optimal first stage solution (and its cost), and T^* the optimal second stage cost. Let $\beta = 36 \ln m$. If $T \geq T^*$ then $c(\Phi_T) \leq H_n \cdot (\Phi^* + 12 \cdot T^*)$.*

Proof: We claim that there is a *fractional* solution $\bar{x}$ for the set covering instance S with small cost $O(\Phi^* + T^*)$, whence rounding this to an integer solution implies the theorem. For a contradiction, assume not: let every fractional set cover be expensive, and hence there must be a dual solution of large value. We then *round this dual solution* to get a dual solution to a sub-instance with only k elements that costs $> \Phi^* + T^*$, which is impossible (since using the optimal solution we can solve every instance on k elements with that cost).

To this end, let $S' \subseteq S$ denote the elements that are *not* covered by the optimal first stage Φ^*, and let $\mathcal{F}' \subseteq \mathcal{F}$ denote the sets that contain at least one element from S'. By the choice of S, all sets in $\mathcal{F}'$ cost at least $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k}$. Define the "coarse" cost for a set $R \in \mathcal{F}'$ to be $\widehat{c}_R = \lceil \frac{c_R}{6T^*/k} \rceil$. For each set $R \in \mathcal{F}'$, since $c_R \geq \frac{\beta T^*}{k} \geq \frac{6T^*}{k}$, it follows that $\widehat{c}_R \cdot \frac{6T^*}{k} \in [c_R, 2 \cdot c_R)$, and also that $\widehat{c}_R \geq \beta/6$.

Now consider the primal-dual pair of LPs for the set cover instance with elements S' and sets $\mathcal{F}'$ having the coarse costs $\widehat{c}$. Let $\{x_R\}_{R \in \mathcal{F}'}$ be an optimal primal and $\{y_e\}_{e \in S'}$ an optimal dual solution. The following claim bounds the (coarse) cost of these fractional solutions.

Claim 3.3 *If $\beta = 36 \ln m$, then the LP cost is $\sum_{R \in \mathcal{F}'} \widehat{c}_R \cdot x_R = \sum_{e \in S'} y_e \leq 2 \cdot k$.*

Before we prove Claim 3.3, let us assume it and complete the proof of Theorem 3.2. Given the primal LP solution $\{x_R\}_{R \in \mathcal{F}'}$ to cover elements in S', define an LP solution to cover elements in S as follows: define $z_R = 1$ if $R \in \Phi^*$, $z_R = x_R$ if $R \in \mathcal{F}' \setminus \Phi^*$; and $z_R = 0$ otherwise. Since the solution $\bar{z}$ contains Φ^* integrally, it covers elements $S \setminus S'$ (i.e. the portion of S covered by Φ^*); since $z_R \geq x_R$, $\bar{z}$ fractionally covers S'. Finally, the cost of this solution is $\sum_R c_R z_R \leq \Phi^* + \sum_R c_R x_R \leq \Phi^* + \frac{6T^*}{k} \cdot \sum_R \widehat{c}_R x_R$. But Claim 3.3 bounds this by $\Phi^* + 12 \cdot T^*$. Since we have a LP solution of value $\Phi^* + 12T^*$, and the greedy algorithm is an H_n-approximation relative to the LP value for set cover, this completes the proof. ■

Claim 3.1 and Theorem 3.2 show our algorithm for set cover to be an $(H_n, 12H_n, 36 \ln m)$-discriminating algorithm. Applying Lemma 2.2 converts this discriminating algorithm to an algorithm for k-robust set cover, and gives the following improvement to the result of [FJMM07].

Theorem 3.4 *There is an $O(\log m + \log n)$-approximation for k-robust set cover.*

It remains to give the proof for Claim 3.3 above; indeed, that is where the technical heart of the result lies.

Proof of Claim 3.3: Recall that we want to bound the optimal fractional set cover cost for the instance $(S', \mathcal{F}')$ with the coarse (integer) costs; x_R and y_e are the optimal primal and dual solutions. For a contradiction, assume that the LP cost $\sum_{R \in \mathcal{F}'} \widehat{c}_R x_R = \sum_{e \in S'} y_e$ lies in the unit interval $((\gamma - 1)k, \gamma k]$ for some integer $\gamma \geq 3$.

Define integer-valued random variables $\{Y_e\}_{e \in S'}$ by setting, for each $e \in S'$ independently, $Y_e = \lfloor y_e \rfloor + I_e$, where I_e is a Bernoulli($y_e - \lfloor y_e \rfloor$) random variable. We claim that whp the random variables $Y_e/3$ form a feasible dual— i.e., they satisfy all the constraints $\{\sum_{e \in R} (Y_e/3) \leq \widehat{c}_R\}_{R \in \mathcal{F}'}$ with high probability. Indeed, consider a dual constraint corresponding to $R \in \mathcal{F}'$: since we have $\sum_{e \in R} \lfloor y_e \rfloor \leq \widehat{c}_R$, we get that $Pr[\sum_{e \in R} Y_e > 3 \cdot \widehat{c}_R] \leq Pr[\sum_{e \in R} I_e > 2 \cdot \widehat{c}_R]$. But now we use a Chernoff bound [MR95] to bound the probability that the sum of independent 0-1 r.v.s, $\sum_{e \in R} I_e$, exceeds twice its mean (here $\sum_{e \in R} E[I_e] \leq \sum_{e \in R} y_e \leq \widehat{c}_R$) by $\varepsilon^{-\widehat{c}_R/3} \leq e^{-\beta/18} \leq m^{-2}$, since each $\widehat{c}_R \geq \beta/6$ and $\beta = 36 \cdot \ln m$. Finally, a trivial union bound implies that $Y_e/3$ satisfies all the m contraints with probability at least $1 - 1/m$. Moreover, the expected dual objective is $\sum_{e \in S'} y_e \geq (\gamma - 1)k \geq 1$ (since $\gamma \geq 3$ and $k \geq 1$), and by another Chernoff Bound, $\Pr[\sum_{e \in S'} Y_e > \frac{\gamma - 1}{2} \cdot k] \geq a$ (where $a > 0$ is some constant). Putting it all together, with probability at least $a - \frac{1}{m}$, we have a *feasible* dual solution $Y'_e := Y_e/3$ with objective value at least $\frac{\gamma - 1}{6} \cdot k$.

Why is this dual Y'_e any better than the original dual y_e? It is "near-integral"—specifically, each Y'_e is either zero or at least $\frac{1}{3}$. So order the elements of S' in decreasing order of their Y'-value, and let Q be the set of the *first k elements* in this order. The total dual value of elements in Q is at least $\min\{\frac{\gamma-1}{6}k, \frac{k}{3}\} \geq \frac{k}{3}$, since $\gamma \geq 3$, and each non-zero Y' value is $\geq 1/3$. This valid dual for elements in Q shows a lower bound of $\frac{k}{3}$ on minimum (fractional) $\widehat{c}$-cost to cover the k elements in Q. Using $c_R > \frac{3T^*}{k} \cdot \widehat{c}_R$ for each $R \in \mathcal{F}'$, the minimum c-cost to fractionally cover Q is $> \frac{3T^*}{k} \cdot \frac{k}{3} = T^*$. Hence, if Q is the realized scenario, the optimal second stage cost will be $> T^*$ (as no element in Q is covered by Φ^*)—this contradicts the fact that OPT can cover $Q \in \binom{U}{k}$ with cost at most T^*. Thus we must have $\gamma \leq 2$, which completes the proof of Claim 3.3. ∎

The k-Max-Min Set Cover Problem. The proof of Claim 3.3 suggests how to get a $(H_n, 12H_n, 36\ln m)$ strongly discriminating algorithm. When $\lambda = 1$ (and so $\Phi^* = 0$), the proof shows that if $c(\Phi_T) > 12H_n \cdot T$, there is a randomized algorithm that outputs k-set Q with optimal covering cost $> T$ (witnessed by the dual solution having cost $> T$). Now using Lemma 2.4, we get the claimed $O(\log m + \log n)$ algorithm for the k-max-min set cover problem. This nearly matches the hardness of $\Omega(\frac{\log m}{\log\log m} + \log n)$ given by [FJMM07].

Remarks: We note that our k-robust algorithm also extends to the more general setting of (uncapacitated) *Covering Integer Programs*; a CIP (see eg. [Sri99]) is given by $A \in [0,1]^{n\times m}$, $b \in [1,\infty)^n$ and $c \in \mathbb{R}^m_+$, and the goal is to minimize $\{c^T \cdot x \mid Ax \geq b,\, x \in \mathbb{Z}^m_+\}$. The results above (as well as the [FJMM07] result) also hold in the presence of set-dependent inflation factors—details will appear in the full version. Results for the other covering problems do not extend to the case of non-uniform inflation: this is usually inherent, and not just a flaw in our analysis. Eg., [KKMS08] give an $\Omega(\log^{1/2-\epsilon} n)$ hardness for k-robust Steiner forest under just two distinct inflation-factors, whereas we give an $O(1)$-approximation under uniform inflations (in Section 5).

4 Minimum Cut

We now consider the k-robust minimum cut problem, where we are given an undirected graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}_+$, a root $r \in V$, terminals $U \subseteq V$, inflation factor λ. Again, any subset in $\binom{U}{k}$ is a possible second-stage scenario, and again we seek to give a discriminating algorithm. This algorithm, like for set cover, is non-adaptive: we just pick all the "expensive" terminals and cut them in the first stage.

Algorithm 2 Algorithm for k-Robust Min-Cut

1: **input:** k-robust minimum-cut instance and threshold T.
2: **let** $\beta \leftarrow \Theta(1)$, and $S \leftarrow \{v \in U \mid$ min cut separating v from root r has cost at least $\beta \cdot \frac{T}{k}\}$.
3: **output** first stage solution Φ_T as the minimum cut separating S from r.
4: **define** $\mathsf{Augment}_T(\{i\})$ as the min-r-i cut in $G \setminus \Phi_T$, for $i \in U \setminus S$; and $\mathsf{Augment}_T(\{i\}) = \emptyset$ for $i \in S$.
5: **output** second stage solution $\mathsf{Augment}_T$ where $\mathsf{Augment}_T(D) := \bigcup_{i\in D} \mathsf{Augment}_T(\{i\})$ for all $D \subseteq U$.

Claim 4.1 (Property A for Min-Cut) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \bigcup \mathsf{Augment}_T(D)$ separate the terminals D from r; moreover, the cost $c(\mathsf{Augment}_T(D)) \leq \beta T$.*

Theorem 4.2 (Property B for Min-Cut) *Let Φ^* denote the optimal first stage solution (and its cost), and T^* the optimal second stage cost. If $\beta \geq \frac{10e}{e-1}$ and $T \geq T^*$ then $c(\Phi_T) \leq 3 \cdot \Phi^* + \frac{\beta}{2} \cdot T^*$.*

Here's the intuition for this theorem: As in the set cover proof, we claim that if the optimal cost of separating S from the root r is high, then there must be a dual solution (which prescribes flows from vertices in S to r) of large value. We again "round" this dual solution by aggregating these flows to get a set of k terminals that have a large combined flow (of value $> \Phi^* + T^*$) to the root—but this is impossible, since the optimal solution promises us a cut of at most $\Phi^* + T^*$ for any set of k terminals.

However, more work is required. For set-cover, each element was either covered by the first-stage, or it was not; for cut problems, things are not so cut-and-dried, since both stages may help in severing a terminal from the root! So we divide S into two parts differently: the first part contains those nodes whose min-cut in G is large (since they belonged to S) but it fell by a constant factor in the graph $G \setminus \Phi^*$. These we call "low" nodes, and we use a Gomory-Hu tree based analysis to show that all low nodes can be completely separated from r by paying only $O(\Phi^*)$ more (this we show in Claim 4.3). The remaining "high" nodes continue to have a large min-cut in $G \setminus \Phi^*$, and for these we use the dual rounding idea sketched above to show a min-cut of $O(T^*)$ (this is proved in Claim 4.4). Together these claims imply Theorem 4.2.

To begin the proof of Theorem 4.2, let $H := G \setminus \Phi^*$, and let $S_h \subseteq S$ denote the "high" vertices whose min-cut from the root in H is at least $M := \frac{\beta}{2} \cdot \frac{T^*}{k}$. The following claim is essentially from Golovin et al. [GGR06].

Claim 4.3 (Cutting Low Nodes) *If $T \geq T^*$, the minimum cut in H separating $S \setminus S_h$ from r costs at most $2 \cdot \Phi^*$.*

Proof: Let $S' := S \setminus S_h$, and $t := \beta \cdot \frac{T^*}{k}$. For every $v \in S'$, the minimum $r - v$ cut is at least $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k} = 2M$ in G, and at most M in H. Consider the *Gomory-Hu* (cut-equivalent) tree $\mathcal{T}(H)$ on graph H rooted at r [Sch03, Chap. 15]. For each $u \in S'$ let $D_u \subseteq V$ denote the minimum $r - u$ cut in $\mathcal{T}(H)$ where $u \in D_u$ and $r \notin D_u$. Pick a subset $S'' \subseteq S'$ of terminals such that the union of their respective min-cuts in $\mathcal{T}(H)$ separate all of S' from the root and their corresponding sets D_u are disjoint (the set of cuts in tree $\mathcal{T}(H)$ closest to the root r gives such a collection). It follows that (a) $\{D_u \mid u \in S''\}$ are disjoint, and (b) $F := \cup_{u \in S''} \partial_H(D_u)$ is a feasible cut in H separating S' from r. Note that for all $u \in S''$, we have $c(\partial_H(D_u)) \leq M$ (since it is a minimum r-u cut in H), and $c(\partial_G(D_u)) \geq 2M$ (it is a feasible r-u cut in G). Thus $c(\partial_H(D_u)) \leq c(\partial_G(D_u)) - c(\partial_H(D_u)) = c(\partial_{\Phi^*}(D_u))$. Now, $c(F) \leq \sum_{u \in S''} c(\partial_H(D_u)) \leq \sum_{u \in S''} c(\partial_{\Phi^*}(D_u)) \leq 2 \cdot \Phi^*$. The last inequality uses disjointness of $\{D_u\}_{u \in S''}$. Thus the minimum $r - S'$ cut in H is at most $2\Phi^*$. ■

Claim 4.4 (Cutting High Nodes) *If $T \geq T^*$, the minimum r-S_h cut in H costs at most $\frac{\beta}{2} \cdot T^*$, when $\beta \geq \frac{10 \cdot e}{e-1}$.*

Proof: Consider a r-S_h max-flow in the graph $H = G \setminus \Phi^*$, and suppose it sends $\alpha_i \cdot M$ flow to vertex $i \in S_h$. By making copies of terminals, we can assume each $\alpha_i \in (0, 1]$; the k-robust min-cut problem remains unchanged under making copies. Hence if we show that $\sum_{i \in S_h} \alpha_i \leq k$, the total flow (which equals the min r-S_h cut) would be at most $k \cdot M = \frac{\beta}{2} \cdot T^*$, which would prove the claim. For a contradiction, we suppose that $\sum_{i \in S_h} \alpha_i > k$. We will now claim that there exists a subset $W \subseteq S_h$ with $|W| \leq k$ such that the min r-W cut is more than T^*, contradicting the fact that every k-set in H can be separated from r by a cut of value at most T^*. To find this set W, the following redistribution lemma (proved at the end of this theorem) is useful.

Lemma 4.5 (Redistribution Lemma) *Let $N = (V, E)$ be a capacitated undirected graph. Let $X \subseteq V$ be a set of terminals such min-cut$_N(i, j) \geq 1$ for all nodes $i, j \in X$. For each $i \in X$, we are given a value $\epsilon_i \in (0, 1]$. Then for any integer $\ell \leq \sum_{i \in X} \epsilon_i$, there exists a subset $W \subseteq X$ with $|W| \leq \ell$ vertices, and a feasible flow f in N from X to W so that (i) the total f-flow into W is at least $\frac{1-e^{-1}}{4} \cdot \ell$ and (ii) the f-flow out of $i \in X$ is at most $\epsilon_i/4$.*

We apply this lemma to $H = G \setminus \Phi^*$ with terminal set S_h, but with capacities scaled down by M. Since for any cut separating $x, y \in S_h$, the root r lies on one side on this cut (say on y's side), min-cut$_H(x, y) \geq M$—hence the scaled-down capacities satisfy the conditions of the lemma. Now set $\ell = k$, and $\epsilon_i := \alpha_i$ for each terminal $i \in S_h$; by the assumption $\sum_{i \in S_h} \epsilon_i = \sum_{i \in S_h} \alpha_i \geq k = \ell$. Hence Lemma 4.5 finds a subset $W \subseteq S_h$ with k vertices, and a flow f in (unscaled) graph H such that f sends a total of at least $\frac{1-1/e}{4} \cdot kM$ units into W, and at most $\frac{\alpha_i}{4} \cdot M$ units out of each $i \in S_h$. Also, there is a feasible flow g in the network H that simultaneously sends $\alpha_i \cdot M$ flow from the root to each $i \in S_h$, namely the max-flow from r to S_h. Hence the flow $\frac{g+4f}{5}$ is *feasible* in H, and sends $\frac{4}{5} \cdot \frac{1-1/e}{4} \cdot kM = \frac{1-1/e}{5} \cdot kM$ units from r into W. Finally, if $\beta > \frac{10 \cdot e}{e-1}$, we obtain that the min-cut in H separating W from r is greater than T^*: since $|W| \leq k$, this is a contradiction to the assumption that any set with at most k vertices can separated from the root in H at cost at most T^*. ■

From Claim 4.1 and Theorem 4.2, we obtain a $(3, \frac{\beta}{2}, \beta)$-discriminating algorithm for k-robust minimum cut, when $\beta \geq \frac{10e}{e-1}$. We set $\beta = \frac{10e}{e-1}$ and use Lemma 2.2 to infer that the approximation ratio of this algorithm is $\max\{3, \frac{\beta}{2\lambda} + \beta\} = \frac{\beta}{2\lambda} + \beta$. Since picking edges only in the second-stage is a trivial λ-approximation, the better of the two gives an approximation of $\min\{\frac{\beta}{2\lambda} + \beta,\ \lambda\} < 17$. Thus we have,

Theorem 4.6 (Min-cut Theorem) *There is a 17-approximation algorithm for k-robust minimum cut.*

It now remains to prove the redistribution lemma. At a high level, the proof shows that if we add each vertex $i \in X$ to a set W independently with probability $\epsilon_i\, \ell/(\sum_i \epsilon_i)$, then this set W will (almost) satisfy the conditions of the lemma whp. A natural approach to prove this would be to invoke Gale/Hoffman-type theorems [Sch03, Chap. 11]: e.g., it is necessary and sufficient to show that $c(\partial V') \geq |\text{demand}(V') - \text{supply}(V')|$ for all $V' \subseteq V$ for this random choice W. But we need to prove such facts for *all* subsets, and all we know about the network is that the min-cut between any pair of nodes in X is at least 1! Also, such a general approach is likely to fail, since the redistribution lemma is false for directed graphs (see Section D) whereas the Gale-Hoffman theorems hold for digraphs. In our proof, we use undirectedness to fractionally pack Steiner trees into the graph, on which we can do a randomized-rounding-based analysis.

Proof of Lemma 4.5 (Redistribution Lemma): To begin, we assume wlog that the bounds $\epsilon_i = 1/P$ for all $i \in X$ for some integer P. Indeed, let $P \in \mathbb{N}$ be large enough so that $\hat{\epsilon}_i = \epsilon_i P$ is an integer for each $i \in X$. Add, for each $i \in X$, a star with $\hat{\epsilon}_i - 1$ leaves centered at the original vertex i, set all these new vertices to also be terminals, and make edges have unit capacity. Set the new ϵ's to be $1/P$ for all terminals. To avoid excess notation, call this graph N as well; note that the assumptions of the lemma continue to hold, and any solution on this new graph can be mapped back to the original graph.

Let c_e denote the edge capacities in N, and recall the assumption that every cut in N separating X has capacity at least one. Since the natural LP relaxation for Steiner-tree has integrality gap of 2, this implies the existence of Steiner trees $\{T_a\}_{a \in A}$ on the terminal set X that fractionally pack into the edge capacities $\bar{c}$. I.e., there exist positive multipliers $\{\lambda_a\}_{a \in A}$ such that $\sum_a \lambda_a = \frac{1}{2}$, and $\sum_a \lambda_a \cdot \bar{\chi}(T_a) \leq \bar{c}$, where $\bar{\chi}(T_a)$ is the characteristic vector of the tree T_a. Choose $W \subseteq X$ by taking ℓ samples uniformly at random (with replacement) from X. We will construct the flow f from X to W as a sum of flows on these Steiner trees. In the following, let $q := |X|$; note that $\ell \leq |X|\epsilon = q/P$.

Consider any fixed tree T_a in this collection, where we think of the edges as having unit capacities. We claim that in expectation, $\Omega(\ell)$ units of flow can be feasibly routed from X to W in T_a such that each terminal supplies at most ℓ/q. Indeed, let τ_a denote an oriented Euler tour corresponding to T_a. Since the tour uses any tree edge twice, any feasible flow routed in τ_a (with unit-capacity edges) can be scaled by half to obtain a feasible flow in T_a. We call a vertex $v \in X$ *a-close* if there is some W-vertex located at most q/ℓ hops from v on the (oriented) tour τ_a. Construct a flow f_a on τ_a by sending ℓ/q flow from each a-close vertex $v \in X$ to its nearest W-vertex along τ_a. By the definition of a-closeness, the maximum number of flow paths in f_a that traverse an edge on τ_a is q/ℓ; since each flow path carries ℓ/q flow, the flow on any edge in τ_a is at most one, and hence f_a is feasible.

For any vertex $v \in X$ and a tour τ_a, the probability that v is *not* a-close is at most $(1 - \frac{q/\ell}{q})^\ell \leq e^{-1}$; hence $v \in X$ sends flow in f_a with probability at least $1 - e^{-1}$. Thus the expected amount of flow sent in f_a is at least $(1 - e^{-1})|X| \cdot (\ell/q) = (1 - e^{-1}) \cdot \ell$. Now define the flow $f := \frac{1}{2}\sum_a \lambda_a \cdot f_a$ by combining all the flows along all the Steiner trees. It is easily checked that this is a feasible flow in N. Since $\sum_a \lambda_a = \frac{1}{2}$, the expected value of flow f is at least $\frac{1-1/e}{4}\ell$. Finally the amount of flow in f sent out of any terminal is at most $\frac{1}{4} \cdot \ell/q \leq \frac{1}{4P}$. This completes the proof of the redistribution lemma. ■

The k-max-min Min-Cut Problem. When $\lambda = 1$ and $\Phi^* = 0$, the proof of Theorem 4.2 gives a randomized algorithm such that if the minimum r-S cut is greater than $\frac{\beta}{2}T$, it finds a subset W of at most k terminals such that separating W from the root costs more than T (witnessed by the dual value). Using this we get a randomized $(3, \frac{\beta}{2}, \beta)$ *strongly* discriminating algorithm, and hence a randomized $O(1)$-approximation algorithm for k-max-min min cut from Lemma 2.4. We note that for k-max-min min-cut, a $(1 - 1/e)$-approximation algorithm was already known (even for directed graphs) via submodular maximization. However the above approach has the advantage that it also extends to k-robust min-cut.

5 Steiner Forest

In k-robust Steiner forest, we have a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, and a set $U \subseteq V \times V$ of potential terminal pairs; any set in $\binom{U}{k}$ is a valid scenario in the second stage. For a set of pairs $S \subseteq V \times V$, the graph G/S is obtained by identifying each pair in S together; $d_{G/S}(\cdot, \cdot)$ is the distance in this "shrunk" graph. The algorithm is given below. This algorithm is a bit more involved than the previous ones, despite a similar general structure: we maintain a set of "fake" pairs S_f that may not belong to U for this case. The following analysis shows a constant-factor guarantee. (Without lines 6-7, the algorithm is more natural, but for that we can currently only show an $O(\log n)$-approximation; it seems that an $O(1)$-approximation for that version would imply an $O(\log n)$-competitiveness for online greedy Steiner forest.)

Algorithm 3 Algorithm for k-Robust Steiner Forest

1: **input:** k-robust Steiner forest instance and threshold T.
2: **let** $\beta \leftarrow \Theta(1), \gamma \leftarrow \Theta(1)$ such that $\gamma \leq \beta/2$.
3: **let** $S_r, S_f, W \leftarrow \emptyset$
4: **while** there exists a pair $(s, t) \in U$ with $d_{G/(S_r \cup S_f)}(s, t) > \beta \cdot \frac{T}{k}$ **do**
5:     **let** $S_r \leftarrow S_r \cup \{(s, t)\}$
6:     **if** $d_G(s, w) < \gamma \cdot \frac{T}{k}$ for some $w \in W$ **then** $S_f \leftarrow S_f \cup \{(s, w)\}$ **else** $W \leftarrow W \cup \{s\}$
7:     **if** $d_G(t, w') < \gamma \cdot \frac{T}{k}$ for some $w' \in W$ **then** $S_f \leftarrow S_f \cup \{(t, w')\}$ **else** $W \leftarrow W \cup \{t\}$
8: **end while**
9: **output** first stage solution Φ_T to be the 2-approximate Steiner forest [AKR95, GW95] on pairs S_r along with shortest-paths connecting every pair in S_f.
10: **define** $\mathsf{Augment}_T(\{i\})$ to be the edges on the $s_i - t_i$ shortest-path in $G/(S_r \cup S_f)$, for each pair $i \in U$.
11: **output** second stage solution $\mathsf{Augment}_T$ where $\mathsf{Augment}_T(S) := \bigcup_{i \in D} \mathsf{Augment}_T(\{i\})$ for all $D \subseteq U$.

Claim 5.1 (Property A for Steiner forest) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \bigcup \mathsf{Augment}_T(D)$ connect every pair in D, and have cost $c(\mathsf{Augment}_T(D)) \leq \beta T$.*

Proof: The first part is immediate from the definition of $\mathsf{Augment}_T$ and the fact that Φ_T connects every pair in $S_r \cup S_f$. The second part follows from the termination condition $d_{G/(S_r \cup S_f)}(s_i, t_i) \leq \beta \cdot \frac{T}{k}$ for all pairs $i \in U$; this implies $c(\mathsf{Augment}_T(D)) \leq \sum_{i \in D} c(\mathsf{Augment}_T(\{i\})) \leq \sum_{i \in D} d_{G/(S_r \cup S_f)}(s_i, t_i) \leq \frac{|D|}{k} \cdot \beta T$. ■

Lemma 5.2 *The optimal value of the Steiner forest on pairs S_r is at least $|W| \times \frac{\gamma}{2} \frac{T}{k}$.*

Proof: Consider the primal (covering) and dual (packing) LPs corresponding to Steiner forest on S_r. Note that for each pair $i \in S_r$, the distance $d_G(s_i, t_i) \geq \beta \cdot \frac{T}{k} \geq 2\gamma \cdot \frac{T}{k}$; so any ball of radius $\frac{\gamma}{2} \cdot \frac{T}{k}$ around a vertex in S_r may be used in the dual packing problem. Observe that W consists of only vertices from S_r, and each time we add a vertex to W, it is at least $\gamma T/k$ distant from any other vertex in W. Hence we can feasibly pack dual balls of radius $\frac{\gamma}{2} \cdot \frac{T}{k}$ around each W-vertex. This is a feasible dual to the Steiner forest instance on S_r, of value $|W|\gamma/2 \cdot T/k$. ■

Lemma 5.3 *The number of "witnesses" $|W|$ is at least the number of "real" pairs $|S_r|$, and $|S_r|$ is at least the number of "fake" pairs $|S_f|$.*

Proof: Partition the set S_r as follows: S_g are the pairs where both end-points are added to W, S_o are the pairs where exactly one end-point is added to W, and S_b are the pairs where neither end-point is added to W. It follows that $|S_r| = |S_g| + |S_o| + |S_b|$ and $|W| = 2 \cdot |S_g| + |S_o|$.

Consider an auxiliary graph $H = (W, E(W))$ on the vertex set W which is constructed incrementally:

- When a pair $(s, t) \in S_g$ is added, vertices s, t are added to W, and edge (s, t) is added to $E(W)$.
- Suppose a pair $(s, t) \in S_o$ is added, where s is added to W, but t is not because it is "blocked" by $w' \in W$. In this case, vertex s is added, and edge (s, w') is added to $E(W)$.

- Suppose a pair $(s,t) \in S_b$ is added, where s and t are "blocked" by w and w' respectively. In this case, no vertex is added, but an edge (w,w') is added to $E(W)$.

Claim 5.4 *At any point in the algorithm if $x,y \in W$ lie in the same component of H then $d_{G/(S_f \cup S_r)}(x,y) = 0$.*

Proof: By induction on the algorithm, and the construction of the graph H.

- Suppose pair $(s,t) \in S_g$ is added, then the claim is immediate. H has one new connected component $\{s,t\}$ and others are unchanged. Since $(s,t) \in S_r$, $d_{G/(S_f \cup S_r)}(s,t) = 0$ and the invariant holds.
- Suppose pair $(s,t) \in S_o$ is added, with s added to W and t blocked by $w' \in W$. In this case, the component of H containing w' grows to also contain s; other components are unchanged. Furthermore (t,w') is added to S_f and (s,t) to S_r, which implies $d_{G/(S_f \cup S_r)}(s,w') = 0$. So the invariant continues to hold.
- Suppose pair $(s,t) \in S_b$ is added, with s and t blocked by $w,w' \in W$ respectively. In this case, the components containing w and w' get merged; others are unchanged. Also $(s,w),(t,w')$ are added to S_f and (s,t) to S_r; so $d_{G/(S_f \cup S_r)}(w,w') = 0$, and the invariant continues to hold.

Since these are the only three cases, this proves the claim. ■

Claim 5.5 *The auxiliary graph H does not contain a cycle when $\gamma \leq \beta/2$*

Proof: For a contradiction, consider the first edge (x,y) that when added to H by the process above creates a cycle. Let (s,t) be the pair that caused this edge to be added, and consider the situation just before (s,t) is added to S_r. Since (x,y) causes a cycle, x,y belong to the same component of H, and hence $d_{G/(S_f \cup S_r)}(x,y) = 0$ by the claim above. But since x is either s or its "blocker" w, and y is either t or its blocker w', it follows that $d_{G/(S_f \cup S_r)}(s,t) < 2\gamma \cdot \frac{T}{k} \leq \beta \cdot \frac{T}{k}$. But this contradicts the condition which would cause (s,t) to be chosen into S_r by the algorithm. ■

Now for some counting. Consider graph H at the end of the algorithm: W denotes its vertices, and E its edges. From the construction of H, we obtain $|W| = 2 \cdot |S_g| + |S_o|$ and $|E| = |S_g| + |S_o| + |S_b| = |S_r|$. Since H is acyclic, $|S_r| = |E| \leq |W| - 1$. Also note that $|S_f| = 2 \cdot |S_b| + |S_o| = 2 \cdot |S_r| - |W| < |S_r|$. Thus we have $|W| \geq |S_r| \geq |S_f|$ as required in the lemma. ■

Theorem 5.6 (Property B for Steiner forest) *Let Φ^* denote the optimal first stage solution (and its cost), and T^* the optimal second stage cost. If $T \geq T^*$ then $c(\Phi_T) \leq \frac{4\gamma}{\gamma-2} \cdot (\Phi^* + T^*)$.*

Proof: Let $|S_r| = \alpha k$. Using Lemma 5.3, Lemma 5.2 and the optimal solution,

$$\frac{\gamma}{2} \cdot \alpha \cdot T \leq |W| \cdot \frac{\gamma}{2}\frac{T}{k} \leq OPT(S_r) \leq \Phi^* + \left\lceil \frac{|S_r|}{k} \right\rceil T^* \leq \Phi^* + T^* + \alpha \cdot T^* \leq \Phi^* + T^* + \alpha T \tag{5.1}$$

Thus $\alpha \cdot T \leq \frac{2}{\gamma-2} \cdot (\Phi^* + T^*)$ and $OPT(S_r) \leq \frac{\gamma}{\gamma-2} \cdot (\Phi^* + T^*)$. So the 2-approximate Steiner forest on S_r has cost at most $\frac{2\gamma}{\gamma-2} \cdot (\Phi^* + T^*)$. Note that the distance between each pair in S_f is at most $\gamma \cdot \frac{T}{k}$; so the total length of shortest-paths in S_f is at most $|S_f| \cdot \gamma \cdot \frac{T}{k} \leq |S_r| \cdot \gamma \cdot \frac{T}{k}$ (again by Lemma 5.3). Thus the algorithm's first-stage cost is at most $\frac{2\gamma}{\gamma-2} \cdot (\Phi^* + T^*) + \alpha\gamma \cdot T \leq \frac{4\gamma}{\gamma-2} \cdot (\Phi^* + T^*)$. ■

Theorem 5.7 (Steiner Forest Main Theorem) *There is a 10-approximation for k-robust Steiner forest.*

Proof: Using Claim 5.1 and Theorem 5.6, we obtain a $(\frac{4\gamma}{\gamma-2}, \frac{4\gamma}{\gamma-2}, \beta)$-discriminating algorithm (Definition 2.1) for k-robust Steiner forest. Setting $\beta = 2\gamma$ and $\gamma := 2 + 2 \cdot (1 - 1/\lambda)$, Lemma 2.2 implies an approximation ratio of $\max\{\frac{4\gamma}{\gamma-2},\ \frac{4\gamma/\lambda}{\gamma-2} + 2\gamma\} \leq 4 + \frac{4}{1-1/\lambda}$. Again the trivial algorithm that only buys edges in the second-stage achieves a 2λ-approximation. Taking the better of the two, the approximation ratio is $\min\{2\lambda,\ 4 + \frac{4}{1-1/\lambda}\} < 10$. ■

In Appendix C, we show how this algorithm gives us a *strongly* discriminating algorithm, and hence a constant-factor approximation algorithm for k-max-min Steiner forest. Also, in [KKMS08], a 3-approximation for Steiner forest problem was given when the input graph is itself a tree. This bound can be improved to 2.25 via our general approach; we defer details to the full version.

6 Final Remarks

In this paper, we have presented a unified approach to directly solving k-robust covering problems and k-max-min problems. As mentioned in the introduction, one can show that solving the k-max-min problem also leads to a k-robust algorithm—we give a general reduction in Appendix F. While this general reduction leads to poorer approximation guarantees for the cardinality case, it easily extends to more general cases. Indeed, if the uncertainty sets for robust problems are not just defined by cardinality constraints, we can ask: which families of downwards-closed sets can we devise robust algorithms for? The general reduction in the Appendix shows how to incorporate intersections of matroid-type and knapsack-type uncertainty sets as well.

Our work suggests several general directions for research. While the results for the k-robust case are fairly tight, can we improve on our results for general uncertainty sets to match those for the cardinality case? Can we devise algorithms to handle one matroid constraint that are as simple as our algorithms for the cardinality case? An intriguing specific problem is to find a constant factor approximation for the robust Steiner forest problem in the explicit scenarios version.

Acknowledgments. We thank Chandra Chekuri, Ravishankar Krishnaswamy, Danny Segev, and Maxim Sviridenko for invaluable discussions.

References

[AAA+03] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Seffi Naor. The Online Set Cover Problem. In *STOC*, pages 100–105, 2003.

[AAA+04] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Seffi Naor. A General Approach to Online Network Optimization Problems. In *SODA*, pages 570–579, 2004.

[ADSY09] Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation Robust Stochastic Optimization. *http://arxiv.org/abs/0902.1792*, 2009.

[AKR95] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide : An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24(3):445–456, 1995.

[BC97] Piotr Berman and Chris Coulston. On-line algorithms for Steiner tree problems. In *STOC*, pages 344–353, 1997.

[Ber00] Piotr Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. *Nordic J. of Computing*, 7(3):178–184, 2000.

[CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons, 1998.

[CCPV07] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function under a matroid constraint. In *IPCO*, pages 182–196, 2007.

[CK99] Chandra Chekuri and Sanjeev Khanna. On multi-dimensional packing problems. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algo rithms*, pages 185–194, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[CKS04] Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. The all-or-nothing multicommodity flow problem. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 156–165, 2004.

[CKS05] Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 183–192, 2005.

[DGRS05] Kedar Dhamdhere, Vineet Goyal, R. Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *FOCS*, pages 367–378, 2005.

[FJMM07] Uriel Feige, Kamal Jain, Mohammad Mahdian, and Vahab S. Mirrokni. Robust combinatorial optimization with exponential scenarios. In *IPCO*, volume 4513 of *Lecture Notes in Computer Science*, pages 439–453, 2007.

[FNW78] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions II. *Mathematical Programming Study*, 8:73–87, 1978.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004.

[Gar05] Naveen Garg. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 396–402, 2005.

[GGR06] Daniel Golovin, Vineet Goyal, and R. Ravi. Pay today for a rainy day: improved approximation algorithms for demand-robust min-cut and shortest path problems. In *STACS 2006*, volume 3884 of *Lecture Notes in Comput. Sci.*, pages 206–217. Springer, Berlin, 2006.

[GHNR07] A. Gupta, M. T. Hajiaghayi, V. Nagarajan, and R. Ravi. Dial a ride from k-forest. In *Proceedings of the 15th Annual European Symposium on Algorithms*, pages 241–252, 2007.

[GKS04] Rajiv Gandhi, Samir Khuller, and Aravind Srinivasan. Approximation algorithms for partial covering problems. *J. Algorithms*, 53(1):55–84, 2004.

[GNS06] Daniel Golovin, Viswanath Nagarajan, and Mohit Singh. Approximating the k-multicut problem. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 621–630, 2006.

[GVY96] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications. *SIAM J. Comput.*, 25(2):235–251, 1996.

[GW95] Michel X. Goemans and David P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

[HHR03] Chris Harrelson, Kirsten Hildrum, and Satish Rao. A polynomial-time tree decomposition to minimize congestion. In *SPAA*, pages 34–43, 2003.

[HS89] Cor A. J. Hurkens and Alexander Schrijver. On the Size of Systems of Sets Every t of Which Have an SDR, with an Application to the Worst-Case Ratio of Heuristics for Packing Problems. *SIAM J. Discrete Math.*, 2(1):68–72, 1989.

[IKMM04] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *SODA*, pages 691–700, 2004.

[IW91] M. Imase and B.M. Waxman. Dynamic Steiner tree problem. *SIAM J. on Discrete Mathematics*, 4(3):369–384, 1991.

[Jen76] T. A. Jenkyns. The efficiency of the "greedy" algorithm. In *7th South Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 341–350, 1976.

[KKMS08] Rohit Khandekar, Guy Kortsarz, Vahab S. Mirrokni, and Mohammad R. Salavatipour. Two-stage robust network design with exponential scenarios. In *ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 589–600, 2008.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[NWF78] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14:265–294, 1978.

[Räc08] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.

[RS04] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. In *IPCO*, pages 101–115, 2004.

[Sch03] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.

[Sla97] Petr Slavík. Improved performance of the greedy algorithm for partial cover. *Inf. Process. Lett.*, 64(5):251–254, 1997.

[Sri99] Aravind Srinivasan. Improved approximation guarantees for packing and covering integer programs. *SIAM J. Comput.*, 29(2):648–670, 1999.

[SS04] D. Shmoys and C. Swamy. Stochastic Optimization is (almost) as Easy as Deterministic Optimization. In *FOCS*, pages 228–237, 2004.

[Svi04] M. Sviridenko. A note on maximizing a submodular set function subject to knapsack constraint. *Operations Research Letters*, 32:41–43, 2004.

[Swa08] C. Swamy. Algorithms for Probabilistically-Constrained Models of Risk-Averse Stochastic Optimization with Black-Box Distributions. *http://arxiv.org/abs/0805.0389*, 2008.

[Von08] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.

A Proofs of Main Reduction Lemmas

A.1 Proof of Lemma 2.2

Let $\mathcal{A}$ denote an algorithm for $\mathsf{Robust}_k(\Pi)$ such that it is $(\alpha_1, \alpha_2, \beta)$-discriminating. Let ground-set $E = [m]$, and $c_{max} := \max_{e\in[m]} c_e$. By scaling, we may assume WLOG that all costs in the $\mathsf{Robust}_k(\Pi)$ instance are integral. Let $\epsilon > 0$ be any value as given by the lemma (where $\frac{1}{\epsilon}$ is polynomially bounded), and $N := \lceil \log_{1+\epsilon}\ (m\, c_{max})\rceil + 1$; note that N is polynomial in the input size. Define $\mathcal{T} := \left\{(1+\epsilon)^i \mid 0 \le i \le N\right\}$.

The approximation algorithm for $\mathsf{Robust}_k(\Pi)$ runs the $(\alpha_1, \alpha_2, \beta)$-discriminating algorithm $\mathcal{A}$ for every choice of $T \in \mathcal{T}$ (here $|\mathcal{T}|$ is polynomially bounded), and returns the solution corresponding to:

$$\widetilde{T} := \arg\min \left\{c(\Phi_T) + \lambda \cdot \beta\, T \mid T \in \mathcal{T}\right\}.$$

Recall that T^* denotes the optimal second-stage cost, clearly $T^* \le m \cdot c_{max}$. Let $i^* \in \mathbb{Z}_+$ be chosen such that $(1+\epsilon)^{i^*-1} < T^* \le (1+\epsilon)^{i^*}$; also let $T' := (1+\epsilon)^{i^*}$ (note that $T' \in \mathcal{T}$). The objective value of the approximate solution (for $\widetilde{T}$) can be bounded as follows.

$$\begin{aligned} c\big(\Phi_{\widetilde{T}}\big) + \lambda \cdot \max_{\omega\in\Omega}\ c\big(\mathsf{Augment}_{\widetilde{T}}(\omega)\big) &\le c(\Phi_{\widetilde{T}}) + \lambda \cdot \beta\, \widetilde{T} \\ &\le c(\Phi_{T'}) + \lambda \cdot \beta\, T' \\ &\le (\alpha_1 \cdot \Phi^* + \alpha_2 \cdot T^*) + (1+\epsilon)\ \beta\lambda \cdot T^* \\ &\le (1+\epsilon) \cdot \left[\alpha_1 \cdot \Phi^* + \left(\beta + \frac{\alpha_2}{\lambda}\right) \cdot \lambda T^*\right]. \end{aligned}$$

The first inequality follows from Property A(i) in Definition 2.1; the second by the choice of $\widetilde{T}$; the third by Property B (applied with $T = T' \ge T^*$) in Definition 2.1, and using $T' \le (1+\epsilon) \cdot T^*$. Thus this algorithm for $\mathsf{Robust}_k(\Pi)$ outputs a solution that is a $\left((1+\epsilon) \cdot \max\left\{\alpha_1, \beta + \frac{\alpha_2}{\lambda}\right\}\right)$-approximation.

A.2 Proof of Lemma 2.4

The approximation algorithm for $\mathsf{MaxMin}(\Pi)$ is similar to that in Lemma 2.2. Let $\mathcal{A}$ denote an algorithm for the robust problem that is $(\alpha_1, \alpha_2, \beta)$ *strongly discriminating*. Recall that the k-max-min instance corresponds to the $\mathsf{Robust}_k(\Pi)$ instance with $\lambda = 1$, and hence we will run algorithm $\mathcal{A}$ on this robust instance. Also from Definition 2.1, T^* denotes the optimal second-stage cost of $\mathsf{Robust}_k(\Pi)$, and its optimal fist-stage cost $\Phi^* = 0$ (since $\lambda = 1$). Note that the optimal value of the k-max-min instance also equals T^*.

Let ground-set $E = [m]$, and $c_{max} := \max_{e\in[m]} c_e$. By scaling, we may assume WLOG that all costs in the instance are integral. Let $\epsilon > 0$ be any value as given by the lemma (where $\frac{1}{\epsilon}$ is polynomially bounded), and $N := \lceil \log_{1+\epsilon}\ (m\, c_{max})\rceil + 1$; note that N is polynomial in the input size. Consider the integral powers of $(1+\epsilon)$,

$$\mathcal{T} := \{t_i\}_{i=0}^N, \quad \text{where } t_i = (1+\epsilon)^i \text{ for } i = 0, 1, \cdots, N.$$

The approximation algorithm for $\mathsf{MaxMin}(\Pi)$ runs the strongly discriminating algorithm $\mathcal{A}$ for every choice of $T \in \mathcal{T}$, and let $p \in \{1, \cdots, N\}$ be the smallest index such that $c(\Phi(t_p)) \leq \alpha_2 \; t_p$. Observe that there must exist such an index since for all $T \geq T^*$, we have $c(\Phi_T) \leq \alpha_2 \, T^* \leq \alpha_2 \, T$ (property B in Definition 2.1, using $\Phi^* = 0$), and clearly $T^* \leq m \cdot c_{max} \leq t_N$. The algorithm then outputs $Q(t_{p-1})$ as the max-min scenario. Below we prove that it achieves the claimed approximation. We have for all $T \geq 0$,

$$T^* = \max\left\{ \mathsf{Opt}(D) : D \in \binom{[n]}{k} \right\} \leq \max\left\{ c(\Phi_T) + c(\mathsf{Augment}_T(D)) : D \in \binom{[n]}{k} \right\} \leq c(\Phi_T) + \beta \, T.$$

Above, the inequalities are by conditions A(i) and A(ii) of Definition 2.1. Setting $T = t_p$ here, and by choice of p,

$$T^* \leq c(\Phi(t_p)) + \beta \, t_p \leq (\alpha_2 + \beta) \, t_p.$$

Hence t_p is a $(\alpha_2 + \beta)$-approximation to the max-min value T^*. Now applying the condition of Definition 2.3 with $T = t_{p-1}$, since $c(\Phi(t_{p-1})) \geq \alpha_2 \, t_{p-1}$ (by choice of index p), we obtain that the minimum cost to cover requirements $Q(t_{p-1})$ is at least:

$$t_{p-1} = \frac{t_p}{1+\epsilon} \geq \frac{T^*}{(1+\epsilon) \cdot (\alpha_2 + \beta)},$$

which implies the desired approximation guarantee.

B Steiner Tree

In the k-robust Steiner tree, we are given a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, a root vertex r, and a set $U \subseteq V$ of potential terminals. Any set of k terminals from U—i.e., any set in $\binom{U}{k}$—is a valid scenario in the second stage. Let $d(\cdot, \cdot)$ be the shortest-path distance according to the edge costs. For a set $S \subseteq V$ of terminals, define the distance $d(v, S) := \min_{w \in S} d(v, w)$.

By the results in Section 2.1, a discriminating algorithm for this problem immediately gives us an algorithm for the robust version, and this is how we shall proceed. Here is our discriminating algorithm for k-robust Steiner tree: it picks a $\beta T/k$-net S of the terminals in U, and builds a MST on S as the first stage.

Algorithm 4 Algorithm for k-Robust Steiner Tree

1: **input:** instance of k-robust Steiner tree and threshold T.
2: **let** $\beta \leftarrow \Theta(1)$, $S \leftarrow \{r\}$.
3: **while** there exists a terminal $v \in U$ with $d(v, S) > \beta \cdot \frac{T}{k}$ **do**
4: $\quad S \leftarrow S \cup \{v\}$
5: **end while**
6: **output** first-stage solution Φ_T to be a minimum spanning tree on S.
7: **for each** $i \in U$, define $\mathsf{Augment}_T(\{i\})$ to be the edges on a shortest-path from i to S.
8: **output** second-stage solution $\mathsf{Augment}_T$ where $\mathsf{Augment}_T(D) := \bigcup_{i \in D} \mathsf{Augment}_T(\{i\})$ for all $D \subseteq U$.

To show that the algorithm is discriminating, we need to show the two properties in Definition 2.1. The first property is almost immediate from the construction: since every point in $U \setminus S$ is close to some point in the net S, this automatically ensures that the second stage recourse cost is small.

Claim B.1 (Property A for Steiner Tree) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \cup \mathsf{Augment}_T(D)$ connect the terminals in D to the root r, and have cost $c(\mathsf{Augment}_T(D)) \leq \beta \, T$.*

Proof: From the definition of the second-stage solution, $\mathsf{Augment}_T(D)$ contains the edges on shortest paths from each D-vertex to the set S. Moreover, Φ_T is a minimum spanning tree on S (which in turn contains the root r). Hence $\Phi_T \cup \mathsf{Augment}_T(D)$ connects D to the root r. To bound the cost, note that by the termination condition in the **while** loop, every terminal $i \in U$ satisfies $d(i, S) \leq \beta \frac{T}{k}$. Thus,

$$c(\mathsf{Augment}_T(D)) \leq \sum_{i \in D} c(\mathsf{Augment}_T(\{i\})) = \sum_{i \in D} d(i, S) \leq \frac{|D|}{k} \cdot \beta \, T.$$

This completes the proof that the algorithm above satisfies Property A. ■

It now remains to show that the algorithm satisfies Property B as well. Let us show this for a sub-optimal settings of values; we will improve on these values subsequently. The proof is dual-based and shows that if the cost of the MST on S were large, then the optimal first stage solution cost Φ^* must have been large as well!

Theorem B.2 (Property B for Steiner tree) *Let Φ^* denote the optimal first stage solution (and its cost), and T^* the optimal second stage cost. If $T > T^*$ then the first stage cost $c(\Phi_T) \leq 2 \cdot \frac{\beta}{\beta-2} \cdot (\Phi^* + T^*)$.*

Proof: Suppose $|S| = \alpha k$. We can divide up S into $\lceil \alpha \rceil$ sets $S_1, S_2, \ldots, S_{\lceil \alpha \rceil}$ with at most k terminals each, and let $E(S_i)$ denote the second-stage edges bought by the optimal solution under scenario S_i. Hence $\Phi^* \cup (\cup_{i \leq \lceil \alpha \rceil} E(S_i))$ is a feasible solution to the Steiner tree on S of cost at most $\Phi^* + \lceil \alpha \rceil \cdot T^*$. Also, since each of the points in S is at least at distance $\beta T / k$ from each other, we get (below $OPT(S)$ is the length of the minimum Steiner tree on S),

$$\frac{\beta}{2} \cdot \alpha T = |S| \cdot \frac{\beta}{2} \cdot \frac{T}{k} \leq OPT(S) \leq \Phi^* + \lceil \alpha \rceil \cdot T^* \leq \Phi^* + (\alpha + 1) \cdot T^* \leq \Phi^* + T^* + \alpha T.$$

Hence $\alpha T \leq \frac{2}{\beta-2}(\Phi^* + T^*)$ and $OPT(S) \leq \frac{\beta}{\beta-2} \cdot (\Phi^* + T^*)$; since the MST heuristic is a 2-approximation to the optimal Steiner tree, we get the theorem. ■

Combining Claim B.1 and Theorem B.2 shows that our algorithm is a $(\frac{2\beta}{\beta-2}, \frac{2\beta}{\beta-2}, \beta)$-discriminating algorithm for k-robust Steiner tree. Setting, say, $\beta = 4$ and applying Lemma 2.2 gives us an $\max(4, 4+4) = 8$-approximation for k-robust Steiner tree. In the next subsection, we will show how to improve this guarantee.

B.1 Improved Approximation for Steiner Tree

In the previous analysis, we just wanted to show the main ideas and hence were somewhat sloppy with the analysis. Let us now show how to get a tighter bound using a fractional analysis.

Theorem B.3 (Improved Property B for Steiner Tree) *If $T \geq T^*$ then $c(\Phi_T) \leq \frac{2\beta}{\beta-2} \cdot \Phi^* + 2 \cdot T^*$.*

Proof: Firstly suppose $|S| \leq k$: then it is clear that there is a Steiner tree on $\{r\} \cup S$ of cost at most $\Phi^* + T^*$, and the algorithm finds one of cost at most twice that. In the following assume that $|S| > k$.

Let $LP(S)$ denote the minimum length of a *fractional* Steiner tree on terminals $\{r\} \cup S$. Since each of the points in S is at least at distance $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k}$ from each other, we get $LP(S) \geq \frac{\beta}{2k} \cdot |S| \cdot T^*$. We now construct a fractional Steiner tree $x : E \to \mathbb{R}_+$ of small length. Number the terminals in S arbitrarily, and for each $1 \leq j \leq |S|$ let $A_j = \{j, j+1, \cdots, j+k-1\}$ (modulo $|S|$). Let $\Pi_j \subseteq E \setminus \Phi^*$ denote the second-stage edges bought in the optimal solution under scenario A_j: so $\Phi^* \cup \Pi_j$ is a Steiner tree on terminals $\{r\} \cup A_j$, and $c(\Pi_j) \leq T^*$. Define $x := \chi(\Phi^*) + \frac{1}{k} \cdot \sum_{j=1}^{|S|} \chi(\Pi_j)$. We claim that x supports unit flow from r to any $i \in S$: note that there are k sets $A_{i-k+1}, \cdots, A_i$ that contain i, and for each $i-k+1 \leq j \leq i$, we have $\frac{1}{k} \cdot (\chi(\Phi^*) + \chi(\Pi_j))$ supports $\frac{1}{k}$ flow from r to i. Thus x is a feasible fractional Steiner tree on $\{r\} \cup S$, of cost at most $\Phi^* + \frac{|S|}{k} \cdot T^*$. Combined with the lower bound on $LP(S)$,

$$|S| \cdot \frac{\beta}{2} \cdot \frac{T^*}{k} \leq LP(S) \leq \Phi^* + \frac{|S|}{k} \cdot T^*. \tag{B.2}$$

Thus we have $LP(S) \leq \frac{\beta}{\beta-2} \cdot \Phi^*$, which implies the theorem since the minimum spanning tree on $\{r\} \cup S$ costs at most twice $LP(S)$. ■

From Claim B.1 and Theorem B.3, we now get that the algorithm is $(\frac{2\beta}{\beta-2}, 2, \beta)$-discriminating. Thus, setting $\beta = 2 - \frac{1}{\lambda} + \sqrt{4 + 1/\lambda^2}$ and applying Lemma 2.2, we get the following approximation ratio.

$$\max\left\{\frac{2\beta}{\beta-2},\ \frac{2}{\lambda} + \beta\right\} = 2 + \frac{1}{\lambda} + \sqrt{4 + \frac{1}{\lambda^2}}.$$

On the other hand, the trivial algorithm which does nothing in the first stage is a $1.55 \cdot \lambda$ approximation. Hence the better of these two ratios gives an approximation bound better than 4.5.

The k-max-min Steiner Tree Problem. We show that the above algorithm can be extended to be $(\frac{2\beta}{\beta-2}, 2, \beta)$ strongly discriminating. As shown above, it is indeed discriminating. To show that Definition 2.3 holds, consider the proof of Theorem B.3 when $\lambda = 1$ (so $\Phi^* = 0$) and suppose that $c(\Phi_T) \geq 2T$. The algorithm to output the k-set Q proceeds via two cases.

1. If $|S| \leq k$ then $Q := S$. The minimum Steiner tree on Q is at least half its MST, i.e. at least $\frac{1}{2}c(\Phi_T) \geq T$.
2. If $|S| > k$ then $Q \subseteq S$ is *any* k-set; by the construction of S, we can feasibly pack dual balls of radius $\beta\frac{T}{k}$ around each Q-vertex, and so $LP(Q) \geq \beta T \geq T$. Thus the minimum Steiner tree on Q is at least T.

B.2 Unrooted Steiner tree

We note that the k-robust Steiner tree problem studied above differs from [KKMS08] since there is no root in the model of [KKMS08]. In the unrooted version, any subset of k terminals appear in the second stage, and the goal is to connect them *amongst each other*. We show that a small modification in the proof implies that Algorithm 1 (where $r \in U$ is set to an arbitrary terminal) achieves a good approximation in the unrooted case as well. This algorithm is essentially same as the one used by [KKMS08], but with different parameters: hence our framework can be viewed as generalizing their algorithm. Our proof is somewhat shorter and gives a slightly better approximation ratio.

Below, Φ^* and T^* denote the optimal first and second stage costs for the given unrooted instance. It is clear that Claim B.1 continues to hold in this case as well: hence Property A of Definition 2.1 is satisfied. We next bound the first stage cost of the algorithm (i.e. Property B of Definition 2.1).

Theorem B.4 (Property B for Unrooted Steiner Tree) *If $T \geq T^*$ then $c(\Phi_T) \leq \frac{2\beta}{\beta-2} \cdot \Phi^* + 2T^*$.*

Proof: Firstly suppose $|S| \leq k$: then it is clear that there is a Steiner tree on S of cost at most $\Phi^* + T^*$, and the algorithm finds one of cost at most twice that. In the following assume that $|S| > k$.

Let $LP(S)$ denote the minimum length of a *fractional* Steiner tree on terminals S (recall, no root here). Since each of the points in S is at least at distance $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k}$ from each other, we get $LP(S) \geq \frac{\beta}{2k} \cdot |S| \cdot T^*$. We now construct a fractional Steiner tree $x : E \to \mathbb{R}_+$ of small length. Number the terminals in S arbitrarily, and for each $1 \leq j \leq |S|$ let $A_j = \{j, j+1, \cdots, j+k-1\}$ (modulo $|S|$). Let $\Pi_j \subseteq E \setminus \Phi^*$ denote the second-stage edges bought in the optimal solution under scenario A_j: so $\Phi^* \cup \Pi_j$ is a Steiner tree on terminals A_j, and $c(\Pi_j) \leq T^*$. Define $x := \chi(\Phi^*) + \frac{1}{k} \cdot \sum_{j=1}^{|S|} \chi(\Pi_j)$.

Claim B.5 *For any $i \in S$, x supports a unit flow from terminal i to $i+1$ (modulo $|S|$).*

Proof: Note that there are $k-1$ sets $A_{i-k+2}, \cdots, A_i$ that contain both i and $i+1$. Let $J := \{i-k+2, \cdots, i\}$. So for each $j \in J$, we have $\frac{1}{k} \cdot (\chi(\Phi^*) + \chi(\Pi_j))$ supports $\frac{1}{k}$ flow from i to $i+1$. Furthermore, $\left(\cup_{l \in S \setminus J} \Pi_l\right) \cup \Phi^*$ is a Steiner tree connecting terminals $\cup_{l \in S \setminus J} A_l \supseteq \{i, i+1\}$; i.e. $\frac{1}{k} \cdot \left(\chi(\Phi^*) + \sum_{l \in S \setminus J} \chi(\Pi_l)\right)$ also supports $\frac{1}{k}$ flow from i to $i+1$. Thus we obtain the claim. ∎

Thus x is a feasible fractional Steiner tree on terminal S, of cost at most $\Phi^* + \frac{|S|}{k} \cdot T^*$. Combined with the lower bound on $LP(S)$,

$$|S| \cdot \frac{\beta}{2} \cdot \frac{T^*}{k} \leq LP(S) \leq \Phi^* + \frac{|S|}{k} \cdot T^*. \tag{B.3}$$

Thus we have $LP(S) \leq \frac{\beta}{\beta-2} \cdot \Phi^*$, which implies the theorem since the minimum spanning tree on S costs at most twice $LP(S)$. ∎

Thus by the same calculation as in the rooted case, we obtain a result that slightly improves on the constants obtained by [KKMS08] for the same problem.

Theorem B.6 *There is a 4.5-approximation algorithm for (unrooted) k-robust Steiner tree.*

C The k-max-min Algorithm for Steiner Forest

We now extend the k-robust Steiner forest algorithm to be $(\frac{4\gamma}{\gamma-2}, \frac{4\gamma}{\gamma-2}, 2\gamma)$ *strongly* discriminating (when $\gamma = 3$). As shown earlier, it is indeed discriminating. To show that Definition 2.3 holds, consider the proof of Theorem 5.6 when $\lambda = 1$ (so $\Phi^* = 0$) and suppose $c(\Phi_T) \geq \frac{4\gamma}{\gamma-2} T \geq 2\gamma\, T$. The algorithm to output the k-set Q has two cases.

1. If the number of "real" pairs $|S_r| \leq k$ then $Q := S_r$. We have:

$$c(\Phi_T) \leq 2 \cdot OPT(S_r) + \frac{\gamma T}{k}\,|S_f| \leq 2 \cdot OPT(S_r) + \frac{\gamma T}{k}\,|S_r| \leq 2 \cdot OPT(S_r) + \gamma T.$$

   The first inequality is by definition of Φ_T and since distance between each pair in S_f is at most $\gamma \cdot \frac{T}{k}$, the second inequality is by Lemma 5.3, and the last inequality uses $|S_r| \leq k$. Since $c(\Phi_T) \geq 2\gamma T$, it follows that $OPT(S_r) \geq \gamma T/2 \geq T$.

2. If $|S_r| > k$ then the number of "witnesses" $|W| \geq |S_r| > k$, by Lemma 5.3. Let $Q \subseteq S_r$ be *any* k-set of pairs such that for each $i \in Q$ at least one of $\{s_i, t_i\}$ is in W. By the construction of S_r, we can feasibly pack dual balls of radius $\frac{\gamma}{2}\frac{T}{k}$ around each W-vertex, and so $OPT(Q) \geq |Q| \cdot \frac{\gamma}{2}\frac{T}{k} = \frac{\gamma}{2}T \geq T$.

D An Illustrative Example for Min-Cut

Let us show that our theorems for k-robust min-cut have to use the undirectedness of the graph crucially, and that the theorems are in fact false for directed graphs, even for $k = 1$. Consider the digraph G with a root r, a "center" vertex c, and ℓ terminals $v_1, v_2, \ldots, v_\ell$. There is some unit capacity arcs (v_i, r), and arcs of capacity $\sqrt{\ell}$ between (c, r), and between (v_i, c) for all $i \in [\ell]$. The inflation factor is $\lambda = \sqrt{\ell}$. The optimal strategy is to delete the arc (c, r) in the first stage. Since $k = 1$, one of the terminals v_i demands to be separated from the root in the second stage, whence deleting the edge (v_i, r) costs $\lambda \cdot 1 = \sqrt{\ell}$ resulting in a total cost of $2\sqrt{\ell}$. However, any threshold-based algorithm would either choose none of the terminals (resulting in a recourse cost of $\lambda\sqrt{\ell} = \ell$), or all of them (resulting in a first-stage cost of at least ℓ).

A similar bad example shows that the redistribution lemma (Lemma 4.5) is false for digraphs. This graph only has unit capacity edges, and the same set of vertices as the digraph above. The arcs are (c, r), $\{(r, v_i)\}_{i\in[\ell]}$, and $\{(v_i, c)\}_{i\in[\ell]}$. Note that the min-cut between every v_i-v_j pair is 1, but if we give each of the v_i's $\epsilon_i = 1/\sqrt{\ell}$ flow, there is no way to choose $\sqrt{\ell}$ of these vertices and collect a total of $\Omega(\sqrt{\ell})$ flow at these "leaders".

E Multicut

We now consider the multicut problem: we are given an undirected graph $G = (V, E)$ with edge-costs $c : E \to \mathbb{R}_+$, and m vertex-pairs $\{s_i, t_i\}_{i=1}^m$. In the k-robust version, we are also given an inflation parameter λ and bound k on the cardinality of the realized demand-set. Let Φ^* denote the optimal first stage solution (and its cost), and T^* the optimal second stage cost; so $\mathsf{Opt} = \Phi^* + \lambda \cdot T^*$. The algorithm (given below) is essentially the same as for minimum cut, however the analysis requires different arguments.

Claim E.1 (Property A for Multicut) *For all $T \geq 0$ and $\omega \subseteq [m]$, the edges $\Phi_T \bigcup \mathsf{Augment}_T(\omega)$ separate s_i and t_i for all $i \in \omega$; additionally if $|\omega| \leq k$ then the cost $c(\mathsf{Augment}_T(\omega)) \leq \beta\, T$.*

Proof: Pairs in $\omega \cap S$ are separated by Φ_T. By definition of $\mathsf{Augment}_T$, for each pair $i \in \omega \setminus S$ edges $\mathsf{Augment}_T(\{i\})$ form an $s_i - t_i$ cut. Thus we have the first part of the claim. For the second part, note that by definition of S, the cost of $\mathsf{Augment}_T(\{i\})$ is at most $\beta\, T/k$ for all $i \in [m]$. ■

Theorem E.2 (Property B for Multicut) *If $T \geq T^*$ then $c(\Phi_T) \leq O(\log n) \cdot \Phi^* + O(\log^{2+\epsilon} n) \cdot T^*$.*

Algorithm 5 Algorithm for k-Robust MultiCut

input: k-robust multicut instance and threshold T.
let $\rho := O(\log n)$ be the approximation factor in Räcke's oblivious routing scheme [Räc08], $\epsilon \in (0, \frac{1}{2})$ any constant, and $\beta := \rho \cdot \frac{16 \log n}{\epsilon \log \log n}$.
let $S \leftarrow \{i \in [m] \mid \text{ min } s_i\text{-}t_i \text{ cut has cost at least } \beta \cdot \frac{T}{k}\}$.
output first stage solution Φ_T as the $O(\log n)$-approximate multicut [GVY96] for S.
define $\mathsf{Augment}_T(\{i\})$ as edges in the min $s_i - t_i$ cut, for $i \in [m] \setminus S$; and $\mathsf{Augment}_T(\{i\}) = \emptyset$ for $i \in S$.
output second stage solution $\mathsf{Augment}_T$ where $\mathsf{Augment}_T(\omega) := \bigcup_{i \in \omega} \mathsf{Augment}_T(\{i\})$ for all $\omega \subseteq [m]$.

To prove the lemma, the high level approach is similar to that for k-robust min-cut. We first show that the subset of pairs $\widetilde{S} \subseteq S$ whose min-cut fell substantially on deleting the edges in Φ^* can actually be completely separated by paying $O(1)\Phi^*$ (this is proved in Lemma E.3). Then in Lemma E.6 we show that the remaining pairs in $S \setminus \widetilde{S}$ can be *fractionally* separated at cost $O(\log^{1+\epsilon} n)\, T^*$. Since the [GVY96] algorithm for multicut is relative to the LP, this would imply Theorem E.2.

Let us begin by formally defining the cast of characters. Let $H := G \setminus \Phi^*$ and $M := \beta \cdot \frac{T^*}{k}$. Define,

$$\widetilde{S} := \left\{ i \in S \mid \text{ min cost } s_i\text{-}t_i \text{ cut in } H \text{ is less than } \tfrac{M}{4} \right\}$$

to be the set of pairs whose mincut in G was at least M, but has fallen to at most $M/4$ in $H = G \setminus \Phi^*$.

Lemma E.3 *If $T \geq T^*$, there is a multicut separating pairs $\widetilde{S}$ in graph H which has cost at most $2\,\Phi^*$.*

Proof: We work with graph $H = (V, F)$ with edge-costs $c : F \to \mathbb{R}$. A *cluster* refers to any subset of vertices. A *cut equivalent tree* (c.f. [CCPS98]), $P = (\mathcal{N}(P), E(P))$ is an edge-weighted tree on clusters $\mathcal{N}(P) = \{N_j\}_{j=1}^r$ such that:

- the clusters $\{N_j\}_{j=1}^r$ form a partition of V, and
- for any edge $e \in E(P)$, its weight in P equals the c-cost of the cut corresponding to deleting this edge in P. I.e., if (S_e, S_e^c) is the partition of V obtained by unioning the vertices in the clusters belonging to the two connected components of $P \setminus \{e\}$, then e's weight in P equals $c(\delta(S_e)) = c(\delta(S_e^c))$.

The *Gomory-Hu* tree $P_{GH} = (V, E(P_{GH}))$ of H is a cut-equivalent tree where the clusters are singleton vertices, and which has the additional property that for every $u, v \in V$ the minimum u-v cut in P_{GH} equals the minimum u-v cut in H. For any cut-equivalent tree, a cluster $N \subseteq V$ is called *active* if there is some $i \in \widetilde{S}$ such that $|N \cap \{s_i, t_i\}| = 1$; otherwise the cluster N is called *dead*. We obtain a cut-equivalent tree Q from P_{GH} as follows: (1) shrink all edges having weight greater than $\frac{M}{4}$, and (2) repeatedly merge dead clusters with any of its neighbors. Note that in the resulting tree Q, every edge in $E(Q)$ has weight at most $\frac{M}{4}$, and every cluster in $\mathcal{N}(Q)$ is active. Let $\mathcal{D} := \bigcup_{N \in \mathcal{N}(Q)} \partial_H(N)$. In the next two claims we show that $\mathcal{D}$ is a feasible multicut for $\widetilde{S}$ with cost at most $2\,\Phi^*$.

Claim E.4 *$\mathcal{D}$ is a feasible multicut separating pairs $\widetilde{S}$ in H.*

Proof: Clearly for each pair $i \in \widetilde{S}$, vertices s_i and t_i are in distinct active clusters of the Gomory-Hu tree P_{GH}. Additionally there is some edge of weight less that $\frac{M}{4}$ on the $s_i - t_i$ path in P_{GH}: since the minimum $s_i - t_i$ cut in H is less than $\frac{M}{4}$. Observe that in obtaining tree Q from P_{GH}, we never contract two active clusters nor an edge of weight less that $\frac{M}{4}$. Thus s_i and t_i lie in distinct clusters of Q. Since this holds for all $i \in \widetilde{S}$, the claim follows by definition of $\mathcal{D}$. ■

Claim E.5 *The cost $c(\mathcal{D}) = \sum_{e \in \mathcal{D}} c_e \leq 2\,\Phi^*$, if $T \geq T^*$.*

Proof: Consider any cluster $N \in \mathcal{N}(Q)$. Since all clusters in $\mathcal{N}(Q)$ are active, N contains exactly one of $\{s_i, t_i\}$ for some $i \in \widetilde{S}$. Hence the cut $\partial_G(N)$ (in graph G) has cost at least $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k} = M$, by definition of the set $S \supseteq \widetilde{S}$.

Let $\mathcal{N}_2(Q) \subseteq \mathcal{N}(Q)$ denote all clusters in Q having degree at most two in Q. Note that $|\mathcal{N}_2(Q)| \geq \frac{1}{2}|\mathcal{N}(Q)|$. Using the above observation and the fact that clusters in $\mathcal{N}_2(Q)$ are disjoint, we have

$$|\mathcal{N}_2(Q)|\, M \leq \sum_{N \in \mathcal{N}_2(Q)} c(\partial_G(N)) = \sum_{N \in \mathcal{N}_2(Q)} \left(c(\partial_H(N)) + c(\partial_{\Phi^*}(N))\right) \leq \sum_{N \in \mathcal{N}_2(Q)} c(\partial_H(N)) + 2\Phi^*. \quad \text{(E.4)}$$

We now claim that for any $N \in \mathcal{N}_2(Q)$, the cost $c(\partial_H(N)) \leq \frac{M}{2}$. Let e_1 and e_2 denote the two edges incident to cluster N in Q (the case of a single edge is easier). Let $(U_l, V \setminus U_l)$ denote the cut corresponding to edge e_l (for $l = 1, 2$) where $N \subseteq U_l$. Each of these cuts has cost $c(\partial_H(U_l)) \leq \frac{M}{4}$ by property of cut-equivalent tree Q, and their union $\partial_H(U_1) \bigcup \partial_H(U_2)$ is the cut separating N from $V \setminus N$. Hence it follows that $c(\partial_H(N)) \leq 2 \cdot \frac{M}{4} = \frac{M}{2}$. Using this in (E.4) and simplifying, we obtain $|\mathcal{N}(Q)|\, M \leq 2 \cdot |\mathcal{N}_2(Q)|\, M \leq 8\,\Phi^*$.

For each edge $e \in E(Q)$, let $D_e \subseteq F$ denote the edges in graph H that go across the two components of $Q \setminus \{e\}$. By the property of cut-equivalent tree Q, we have $c(D_e) \leq \frac{M}{4}$. Since $\mathcal{D} = \bigcup_{e \in E(Q)} D_e$,

$$c(\mathcal{D}) \leq \sum_{e \in E(Q)} c(D_e) \leq |E(Q)|\, \frac{M}{4} \leq |\mathcal{N}(Q)|\, \frac{M}{4} \leq 2\Phi^*$$

This proves the claim. ■

Combining Claims E.4 and E.5, we obtain the lemma. ■

Now we turn our attention to the remaining pairs $W := S \setminus \widetilde{S}$, and show that there is a cheap cut separating them in H. For this we use a dual-rounding argument, based on Räcke's oblivious routing scheme. Recall that constant $0 < \epsilon < \frac{1}{2}$, $\rho = O(\log n)$ (Räcke's approximation factor), and $\beta = \rho \cdot \frac{16 \log n}{\epsilon \log\log n}$. Define $\alpha := e\rho \cdot \log^{\epsilon} n$.

Lemma E.6 *There exists a* fractional *multicut separating pairs W in the graph H which has cost $8\alpha \cdot T^*$.*

Proof: For any demand vector $d : W \to \mathbb{R}_+$, the optimal *congestion* of routing d in H, denoted $\mathsf{Cong}(d)$, is the smallest $\eta \geq 0$ such that there is a flow routing d_i units of flow between s_i and t_i (for each $i \in W$), using capacity at most $\eta \cdot c_e$ on each edge $e \in H$. Note that for every $i \in W$, the s_i-t_i min-cut in H has cost at least $L := \frac{M}{4} = \frac{\beta}{4} \cdot \frac{T^*}{k}$. Hence for any $i \in W$, the optimal congestion for a *unit* demand between s_i-t_i (and zero between all other pairs) is at most $\frac{1}{L}$.

Now consider Räcke's oblivious routing scheme [Räc08] as applied to graph H. This routing scheme, for each $i \in W$, prescribes a unit flow $\mathcal{F}_i$ between s_i-t_i such that for every demand vector $d : W \to \mathbb{R}_+$,

$$\max_{e \in H} \frac{\sum_{i \in W} d_i \cdot \mathcal{F}_i(e)}{c_e} \leq \rho \cdot \mathsf{Cong}(d), \qquad \text{where } \rho = O(\log n);$$

i.e., the congestion achieved by using these oblivious templates to route the demand d is at most ρ times the best congestion possible for that particular demand d.

Now consider a maximum multicommodity flow in H that sends $y_i \cdot \frac{T^*}{k}$ units between s_i, t_i (for each $i \in W$). For a contradiction, suppose that $\sum_{i \in W} y_i > 8\alpha \cdot k$. (Otherwise the maximum multicommodity flow, and hence its dual, the minimum fractional multicut is at most $8\alpha T^*$, and the lemma holds.) By making copies of vertex-pairs, we may assume that $y_i \in [0, 1]$ for all $i \in W$; this does not change the k-robust multicut instance. Define a (not necessarily feasible) multicommodity flow $\mathcal{G} := \sum_{i \in W} X_i \cdot \frac{T^*}{k} \cdot \mathcal{F}_i$, where each X_i is an independent 0-1 random variable with $Pr[X_i = 1] = \frac{y_i}{\alpha}$, and $\mathcal{F}_i$ is the Räcke oblivious routing template. The flow has expected magnitude at least $\sum_i \frac{y_i}{\alpha} \frac{T^*}{k} \geq 8T^*$, and is the sum of $\{0, \frac{T^*}{k}\}$-valued random variables, hence by a Chernoff bound:

Claim E.7 *With constant probability, the magnitude of flow $\mathcal{G}$ is at least T^*.*

Claim E.8 *The flow $\mathcal{G}$ is feasible with probability* $1 - o(1)$.

Proof: Fix any edge $e \in H$, and let $u_i(e) := \frac{T^*}{k} \cdot \mathcal{F}_i(e)$ for all $i \in W$. Note that the random process gives us a flow of $\sum_i X_i \cdot u_i(e)$ on the edge e. The feasibility of the maximum multicommodity flow says that $\mathsf{Cong}(\{y_i \cdot \frac{T^*}{k}\}_{i\in W}) \leq 1$. Since oblivious routing loses a ρ factor in the congestion,

$$\textstyle\sum_{i\in W} y_i \cdot u_i(e) = \sum_{i\in W} y_i \cdot \frac{T^*}{k} \cdot \mathcal{F}_i(e) \leq \rho \cdot c_e;$$

and the expected flow on edge e sent by the random process above is $\sum_{i\in W} \frac{y_i}{\alpha} \cdot u_i(e) \leq \frac{\rho}{\alpha} c_e$.

Now, since the min s_i-t_i-cut is at least L for any $i \in W$, a unit of flow can (non-obliviously) be sent between s_i, t_i at congestion at most $\frac{1}{L}$. Hence using the oblivious routing template $\mathcal{F}_i$ incurs a congestion at most $\frac{\rho}{L}$. Hence,

$$u_i(e) = \frac{T^*}{k} \cdot \mathcal{F}_i(e) \leq \frac{T^*}{k} \cdot \frac{\rho}{L} \cdot c_e = \frac{4\rho}{\beta} \cdot c_e$$

We divide the individual contributions by the edge capacity and further scale up by $\frac{\beta}{4\rho}$ by defining new $[0, 1]$-random variables $Y_i = \frac{X_i \cdot u_i}{c_e} \cdot \frac{\beta}{4\rho}$. We get that $\mu := E[\sum Y_i] \leq \frac{\beta}{4\alpha}$. Recall the Chernoff bound that says that for independent $[0, 1]$-valued random variables Y_i,

$$Pr\left[\sum Y_i \geq (1 + \delta) \cdot \mu\right] \leq \left(\frac{e}{1 + \delta}\right)^{\mu(1+\delta)}$$

Using this with $\mu(1 + \delta) = \frac{\beta}{4\rho}$ (hence $\delta + 1 \geq \frac{\alpha}{\rho}$) we get that

$$Pr\left[\sum_i X_i \cdot u_i(e) \geq c_e\right] = Pr\left[\sum_i Y_i \geq \frac{\beta}{4\rho}\right] \leq \left(\frac{e\rho}{\alpha}\right)^{\beta/4\rho} = \exp\left(-\epsilon \log\log n \cdot \frac{4\log n}{\epsilon \log\log n}\right) = \frac{1}{n^4},$$

since $\alpha = e\rho \cdot \log^\epsilon n$ and $\beta = \rho \cdot \frac{16\log n}{\epsilon \log\log n}$. Now a trivial union bound over all n^2 edges gives the claim. ■

By another union bound, it follows that there exists a feasible multicommodity flow $\mathcal{G}$ that sends either zero or $\frac{T^*}{k}$ units for each pair $i \in W$, and the total value of $\mathcal{G}$ is at least T^*. Hence there exists some k-set $W' \subseteq W$ such that the maximum multicommodity flow for W' on H is at least T^*. This contradicts the fact that every k-set has a multicut of cost less than T^* in H. Thus we must have $\sum_{i\in W} y_i \leq 8\alpha \cdot k$, which implies Lemma E.6. ■

Combining Lemmas E.3 and E.6, we obtain a *fractional* multicut for pairs S in graph G, having cost $O(1) \cdot \Phi^* + O(\log^{1+\epsilon} n) \cdot T^*$. Since the Garg et al. [GVY96] algorithm for multicut is an $O(\log n)$-approximation relative to the LP, we obtain Theorem E.2.

From Claim E.1 and Theorem E.2, it follows that this algorithm is $O\left(\log n, \log^{2+\epsilon} n, \beta\right)$-discriminating for k-robust multicut. Since $\beta = O(\log^2 n / \log\log n)$, using Lemma 2.2, we obtain an approximation ratio of:

$$\max\left\{\log n, \frac{\log^2 n}{\log\log n} + \frac{\log^{2+\epsilon} n}{\lambda}\right\}.$$

This is an $O\left(\frac{\log^2 n}{\log\log n}\right)$-approximation when $\lambda \geq \log^{2\epsilon} n$. On the other hand, when $\lambda \leq \log^{2\epsilon} n$, we can use the trivial algorithm of buying all edges in the second stage (using the GVY algorithm [GVY96]); this implies an $O(\log^{1+2\epsilon} n)$-approximation. Since $\log^{1+2\epsilon} n = o\left(\frac{\log^2 n}{\log\log n}\right)$, we obtain:

Theorem E.9 *There is an* $O\left(\frac{\log^2 n}{\log\log n}\right)$*-approximation algorithm for* k*-robust multicut.*

The k-max-min Multicut Problem. The above ideas also lead to a $\left(c_1 \cdot \log n,\ c_2 \cdot \log^2 n,\ c_3 \cdot \log^2 n\right)$ strongly discriminating algorithm for multicut, where c_1, c_2, c_3 are large enough constants. The algorithm is exactly Algorithm 5 with parameter $\beta := \Theta(\log n)\cdot\rho$ with an appropriate constant factor; recall that $\rho = O(\log n)$ is the approximation ratio for oblivious routing [Räc08]. Lemma E.6 shows that this algorithm is $\left(c_1 \cdot \log n, c_2 \cdot \log^2 n, c_3 \cdot \log^2 n\right)$ discriminating (the parameters are only slightly different and the analysis still applies). To establish the property in Definition 2.3, consider the case $\lambda = 1$ (i.e. $\Phi^* = 0$) and $c(\Phi_T) \geq (c_2 \log^2 n) \cdot T$. Since the [GVY96] algorithm is $O(\log n)$-approximate relative to the LP, this implies a feasible multicommodity flow on pairs W (since $\Phi^* = 0$ we also have $W = S$) of value at least $(c_4 \log n) \cdot T$ for some constant c_4. Then the randomized rounding (with oblivious routing) can be used to produce a k-set $W' \subseteq W$ and a feasible multicommodity flow on W' of value at least T; by weak duality it follows that the minimum multicut on W' is at least T and so Definition 2.3 holds. Thus by Lemma 2.4 we get a randomized $O(\log^2 n)$-approximation algorithm for k-max-min multicut.

All-or-Nothing Multicommodity Flow. As a possible use of the oblivious routing and randomized-rounding based approach, let us state a result for the all-or-nothing multicommodity flow problem studied by Chekuri et al. [CKS04]: given a capacitated undirected graph $G = (V, E)$ and source-sink pairs $\{s_i, t_i\}$ with demands d_i such that the min-cut$(s_i, t_i) = \Omega(\log^2 n) d_i$, one can approximate the maximum throughput to within an $O(\log n)$ factor without violating the edge-capacities, even with $d_{\max} \geq c_{\min}$—the results of Chekuri et al. [CKS04, CKS05] violated the edge-capacities in this case by an additive $d_{\max}$. This capacity violation in the previous all-or-nothing results is precisely the reason they can not be directly used in our analysis of k-robust multicut.

F General Framework for Robust Covering

In this section we present an abstract framework for robust covering problems under *any uncertainty set* Ω, as long as we are given access to offline, online and max-min algorithms for the base covering problem. Formally, the properties of interest are described below.

Recall that we deal with the robust version of some covering problem $\Pi = \langle E, c, \{\mathcal{R}_i\}_{i=1}^n \rangle$, given by an arbitrary uncertainty set Ω and inflation parameter $\lambda \geq 1$. Given a partial solution $S \subseteq E$ and a set $X \subseteq [n]$ of requirements, any set $E_X \subseteq E$ such that $S \cup E_X \in \mathcal{R}_i\ \forall i \in X$ is called an *augmentation* of S for requirements X. Given X, S, define the min-cost augmentation of S for requirements X as:

$$\mathsf{OptAug}(X \mid S) := \min\{c(E_X) \mid E_X \subseteq E \text{ and } S \cup E_X \in \mathcal{R}_i,\ \forall i \in X\}.$$

An easy consequence of the fact that costs are non-negative is the *monotonicity* property: $\mathsf{OptAug}(X \mid S) \leq \mathsf{OptAug}(X \mid T)$ for all $X \subseteq [n]$ and $T \subseteq S \subseteq E$. Also define $\mathsf{Opt}(X) := \min\{c(E_X) \mid E_X \subseteq E \text{ and } E_X \in \mathcal{R}_i\ \forall i \in X\}$ for any $X \subseteq [n]$.

Property F.1 (Subadditivity) *For any two subsets of requirements $X, Y \subseteq [n]$ and any partial solution $S \subseteq E$, we have* $\mathsf{OptAug}(X \mid S) + \mathsf{OptAug}(Y \mid S) \geq \mathsf{OptAug}(X \cup Y \mid S)$.

Property F.2 (Offline Algorithm) *There is an α_{off}-approximation (offline) algorithm for the covering problem* $\mathsf{OptAug}(X \mid S)$*, for any $S \subseteq E$ and $X \subseteq [n]$.*

Property F.3 (Online Algorithm) *There is a polynomial-time deterministic α_{on}-competitive algorithm for the online version of* $\Pi = \langle E, c, \{\mathcal{R}_i\}_{i=1}^n \rangle$.

Property F.4 (Max-Min Algorithm) *There is an α_{mm}-approximation algorithm for the max-min problem: given input $S \subseteq E$,* $\mathsf{MaxMin}(S) := \max_{X \in \Omega} \min\{c(A) \mid S \cup A \in R_i,\ \forall i \in X\}$.

Theorem F.5 *Under Properties F.1, F.2, F.3 and F.4, there is an $O(\alpha_{\mathsf{off}} \cdot \alpha_{\mathsf{on}} \cdot \alpha_{\mathsf{mm}})$-approximation algorithm for the robust covering problem* $\mathsf{Robust}(\Pi) = \langle E, c, \{R_i\}_{i=1}^n, \Omega, \lambda \rangle$.

Algorithm 6 Algorithm Robust-with-General-Uncertainty-Sets

1: **input:** the $\mathsf{Robust}(\Pi)$ instance and threshold T.
2: **let** counter $t \leftarrow 0$, initial online algorithm's input $\sigma = \langle\rangle$, initial online solution $F_0 \leftarrow \emptyset$.
3: **repeat**
4: **set** $t \leftarrow t + 1$.
5: **let** $E_t \subseteq [n]$ be the scenario returned by the algorithm of Property F.4 on $\mathsf{MaxMin}(F_{t-1})$.
6: **let** $\sigma \leftarrow \sigma \circ E_t$, and $F_t \leftarrow \mathcal{A}_{on}(\sigma)$ be the current online solution.
7: **until** $c(F_t) - c(F_{t-1}) \leq 2\alpha_{\mathsf{on}} \cdot T$
8: **set** $\tau \leftarrow t - 1$.
9: **output** first-stage solution $\Phi_T := F_\tau$.
10: **output** second-stage solution $\mathsf{Augment}_T$ where for any $\omega \subseteq [n]$, $\mathsf{Augment}_T(\omega)$ is the solution of the offline algorithm (Property F.2) for the problem $\mathsf{OptAug}(\omega \mid \Phi_T)$.

Proof: The algorithm proceeds as follows. Note that Definition 2.1 and Lemma 2.2 hold for *any* uncertainty set Ω, although for simplicity it was stated for $\Omega = \binom{[n]}{k}$.

As always, let $\Phi^* \subseteq E$ denote the optimal first stage solution (and its cost), and T^* the optimal second-stage cost; so the optimal value is $\Phi^* + \lambda \cdot T^*$. We prove the performance guarantee using the following claims.

Claim F.6 (General 2nd stage) *For any $T \geq 0$ and $X \in \Omega$, elements $\Phi_T \bigcup \mathsf{Augment}_T(X)$ satisfy all the requirements in X, and $c(\mathsf{Augment}_T(X)) \leq 2\alpha_{\mathsf{off}} \cdot \alpha_{\mathsf{mm}} \cdot \alpha_{\mathsf{on}} \cdot T$.*

Proof: It is clear that $\Phi_T \bigcup \mathsf{Augment}_T(X)$ satisfy all requirements in X. By the choice of set $E_{\tau+1}$ in line 5 of the last iteration, for any $X \in \Omega$ we have:

$$\mathsf{OptAug}(X \mid F_\tau) \leq \alpha_{\mathsf{mm}} \cdot \mathsf{OptAug}(E_{\tau+1} \mid F_\tau) \leq \alpha_{\mathsf{mm}} \cdot (c(F_{\tau+1}) - c(F_\tau)) \leq 2\alpha_{\mathsf{mm}} \cdot \alpha_{\mathsf{on}} \cdot T$$

The first inequality is by Property F.4, the second inequality uses the fact that $F_{\tau+1} \supseteq F_\tau$ (since we use an online algorithm to augment in line 6),[1] and the last inequality follows from the termination condition in line 7. Finally, since $\mathsf{Augment}_T(X)$ is an α_{off}-approximation to $\mathsf{OptAug}(X \mid F_\tau)$, we obtain the claim. ■

Claim F.7 $\mathsf{Opt}(\cup_{t \leq \tau} E_t) \leq \tau \cdot T^* + \Phi^*$.

Proof: Since each $E_t \in \Omega$ (these are solutions to MaxMin), the bound on the second-stage optimal cost gives $\mathsf{OptAug}(E_t \mid \Phi^*) \leq T^*$ for all $t \leq \tau$. By subadditivity (Property F.1) we have $\mathsf{OptAug}(\cup_{t \leq \tau} E_t \mid \Phi^*) \leq \tau \cdot T^*$, which immediately implies the claim. ■

Claim F.8 $\mathsf{Opt}(\cup_{t \leq \tau} E_t) \geq \frac{1}{\alpha_{\mathsf{on}}} \cdot c(F_\tau)$.

Proof: Directly from the competitiveness of the online algorithm in Property F.3. ■

Claim F.9 (General 1st stage) *If $T \geq T^*$ then $c(\Phi_T) = c(F_\tau) \leq 2\,\alpha_{\mathsf{on}} \cdot \Phi^*$.*

Proof: We have $c(F_\tau) = \sum_{t=1}^{\tau} [c(F_t) - c(F_{t-1})] > 2\alpha_{\mathsf{on}}\tau \cdot T \geq 2\alpha_{\mathsf{on}}\tau \cdot T^*$ by the choice in Step (7). Combined with Claim F.8, we have $\mathsf{Opt}(\cup_{t \leq \tau} E_t) \geq 2\tau \cdot T^*$. Now using Claim F.7, we have $\tau \cdot T^* \leq \Phi^*$, and hence $\mathsf{Opt}(\cup_{t \leq \tau} E_t) \leq 2 \cdot \Phi^*$. Finally using Claim F.8, we obtain $c(F_\tau) \leq 2\alpha_{\mathsf{on}} \cdot \Phi^*$. ■

Claim F.6 and Claim F.9 imply that the above algorithm is a $(2\alpha_{\mathsf{on}},\ 0,\ 2\alpha_{\mathsf{mm}}\alpha_{\mathsf{on}}\alpha_{\mathsf{off}})$ approximation for the robust problem $\mathsf{Robust}(\Pi) = \langle E, c, \{R_i\}_{i=1}^n, \Omega, \lambda \rangle$. Now using Lemma 2.2 we obtain the theorem. ■

[1] This is the technical reason we need an online algorithm. If instead we had used an offline algorithm to compute F_t in step 6 then $F_t \not\supseteq F_{t-1}$ and we could not upper bound the augmentation cost $\mathsf{OptAug}(E_t \mid F_{t-1})$ by $c(F_t) - c(F_{t-1})$.

F.1 Explicit uncertainty sets

An easy consequence of Theorem F.5 is for the *explicit scenario* model of robust covering problems [DGRS05, GGR06], where Ω is specified as a list of possible scenarios. In this case, the MaxMin problem can be solved using the α_{off}-approximation algorithm from Property F.2 which implies an $O(\alpha_{\mathsf{off}}^2 \alpha_{\mathsf{on}})$-approximation for the robust version. In fact, we can do slightly better—observing that the algorithm for second-stage augmentation is the same as the Max-Min algorithm, we obtain an $O(\alpha_{\mathsf{off}} \cdot \alpha_{\mathsf{on}})$-approximation algorithm for robust covering with explicit scenarios. As an application of this result, we obtain an $O(\log n)$ approximation for robust Steiner forest with explicit scenarios, which is the best known result for this problem.

F.2 p-Systems and Knapsack uncertainty sets

Notice that the any uncertainty set Ω for a robust covering problem can be assumed WLOG to be *downward-closed*, i.e. $X \in \Omega$ and $Y \subseteq X$ implies $Y \in \Omega$. Eg., in the k-robust model $\Omega = \{S \subseteq [n] : |S| \leq k\}$. Hence it is of interest to obtain good approximation algorithms for robust covering when Ω is specified by means of general models for downward-closed families. We consider the following well-studied models:

Definition F.10 (p-system) *A downward-closed family $\Omega \subseteq 2^{[n]}$ is called a p-system iff:*

$$\frac{\max_{I \in \overline{\Omega}, I \subseteq A} \ |I|}{\min_{J \in \overline{\Omega}, J \subseteq A} \ |J|} \leq p, \quad \text{for each } A \subseteq [n],$$

where $\overline{\Omega} \subseteq \Omega$ denotes the collection of maximal subsets *in Ω. Sets in Ω are called* independent sets. *We assume access to a membership-oracle, that given any subset $I \subseteq [n]$ returns whether or not $I \in \Omega$.*

Definition F.11 (q-knapsack constraints) *Given q non-negative vectors $w^1, \ldots, w^q : [n] \to \mathbb{R}_+$ and capacities $b_1, \ldots, b_q \in \mathbb{R}_+$, the knapsack constrained family is:*

$$\Omega = \left\{ A \subseteq [n] : \sum_{e \in A} w^j(e) \leq b_j, \text{ for all } j \in [q] \right\}.$$

Some interesting special cases of p-systems are p-matroid intersection [Sch03] and p-set packing [HS89, Ber00]; see the appendix in [CCPV07] for more discussion on p-systems. Jenkyns [Jen76] studied the problem of maximizing linear functions over p-systems, and showed that the greedy algorithm is a p-approximation. Maximizing a linear function over q-knapsack constraints is the well-studied class of packing integer programs (PIPs), eg. [Sri99]. It is known that a natural greedy algorithm achieves an $O(q)$-approximation for this problem; when the number of constraints q is constant, there is a PTAS [CK99]. Our main result here is:

Theorem F.12 *Under Properties F.1, F.2 and F.3, the robust covering problem* $\mathsf{Robust}(\Pi) = \langle E, c, \{\mathcal{R}_i\}_{i=1}^n, \Omega, \lambda \rangle$ *admits an $O\left((p+1) \cdot (q+1) \cdot \alpha_{\mathsf{off}} \cdot \alpha_{\mathsf{on}}\right)$-approximation guarantee when Ω is given by the intersection of a p-system and q-knapsack constraints.*

The proof is deferred to the full version. We note that dependence on p and q is necessary due to complexity considerations, even in the special case of maximizing linear functions over such constraints. We now list some specific results for robust covering under this general model.

Problem	Offline ratio	Online ratio	p-system, q-knapsack Robust
Set Cover	$\log n$	$\log m \cdot \log n$ [AAA+03]	$pq \cdot \log m \cdot \log n$
Steiner Tree/Forest	2 [AKR95, GW95]	$\log n$ [IW91, BC97]	$pq \cdot \log n$
Minimum Cut	1	$\log^3 n \cdot \log\log n$ [AAA+04, HHR03]	$pq \cdot \log^3 n \cdot \log\log n$
Multicut	$\log n$ [GVY96]	$\log^3 n \cdot \log\log n$ [AAA+04, HHR03]	$pq \cdot \log^4 n \cdot \log\log n$