Math. Program., Ser. A (2014) 146:583–615
DOI 10.1007/s10107-013-0705-5

FULL LENGTH PAPER

Thresholded covering algorithms for robust and max–min optimization

Anupam Gupta · Viswanath Nagarajan · R. Ravi

Received: 27 June 2012 / Accepted: 8 August 2013 / Published online: 21 August 2013

Abstract In a two-stage robust covering problem, one of several possible scenarios will appear tomorrow and require to be covered, but costs are higher tomorrow than today. What should you anticipatorily buy today, so that the worst-case cost (summed over both days) is minimized? We consider the *k-robust model* where the possible scenarios tomorrow are given by all demand-subsets of size k. In this paper, we give the following simple and intuitive template for k-robust covering problems: *having built some anticipatory solution, if there exists a single demand whose augmentation cost is larger than some threshold, augment the anticipatory solution to cover this demand as well, and repeat.* We show that this template gives good approximation algorithms for k-robust versions of many standard covering problems: set cover, Steiner tree, Steiner forest, minimum-cut and multicut. Our k-robust approximation ratios nearly match the best bounds known for their deterministic counterparts. The main technical contribution lies in proving certain net-type properties for these covering problems,

An extended abstract containing the results of this paper and of [27] appeared jointly in *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP), 2010*.

A. Gupta: Supported in part by NSF awards CCF-0448095 and CCF-0729022, and an Alfred P. Sloan Fellowship.
R. Ravi: Supported in part by NSF Grants CCF-0728841 and CCF-1218382.

A. Gupta
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: anupamg@cs.cmu.edu

V. Nagarajan (✉)
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
e-mail: viswanath@us.ibm.com

R. Ravi
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: ravi@cmu.edu

which are based on dual-rounding and primal–dual ideas; these properties might be of some independent interest. As a by-product of our techniques, we also get algorithms for max–min problems of the form: "*given a covering problem instance, which k of the elements are costliest to cover?*" For the problems mentioned above, we show that their k-max–min versions have performance guarantees similar to those for the k-robust problems.

Mathematics Subject Classification 05C85 · 68W25 · 90C27

1 Introduction

This paper studies *k-robust* covering problems: given a covering problem instance with potential demands U, some set of k demands $S \subseteq U$ will want to be covered tomorrow; however, today we do not know what this set will be. One strategy is to wait until tomorrow and then cover the realized set S. 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. When there is no further information about the demand S (or we are risk-averse), we want to plan for the worst-case, and minimize:

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

This k-robust model was introduced by Feige et al. [18]. Earlier approximation results for robust covering problems [15,24] had assumed that the collection of possible demands S (i.e. the "uncertainty set") is *explicitly* given. Listing the uncertainty set explicitly seems somewhat restrictive, and so [18] proposed the k-robust model as an *implicit* representation of uncertainty: any of the $\binom{n}{k}$ subsets S of size k could appear as demand. Thus this model tries to bridge the gap between explicit-scenario robust optimization and the setting of stochastic optimization, where exponential-sized uncertainty sets can be modeled more easily, but since the goal there is to minimize the expected cost (instead of the worse-case cost), one has to obtain good estimates of the underlying probability distributions. And indeed, when such knowledge of the demand distribution is unknown to the algorithm, one can fall back on the k-robust model to obtain worst-case guarantees, since the only required information is an upper bound k on the size of the demand set. We emphasize that although the size of the k-robust uncertainty set is exponential in k, the goal is to obtain approximation ratios that do not depend polynomially on k.

The k-robust model can be viewed as a discrete analogue of the study of robust optimization for continuous problems [6,9]. The main results in that area involve studying robust counterparts of continuous optimization problems, where the constraint matrix is drawn from an uncertainty set (such as an ellipsoid) and the problem is to find a solution that is optimal over the worst-case choice of constraints from the uncertainty set. The key results above show that the computational tractability of the underlying problems carry over to their robust counterparts for several interesting uncertainty sets. The k-robust model is a specification of discrete uncertainty requirements, and the goal in our study is to preserve the approximation complexity of the underlying NP-hard discrete optimization problems, when augmented with such uncertainty sets.

Closely related to the k-robust model are *k-max–min* problems, where given a covering problem instance, the goal is to determine the k-set of demands that are costliest to cover. The k-max–min objective is also useful as a measure of fault tolerance: it corresponds to the worst-case recovery cost in a fault model where any set of k demands may lose coverage. If the underlying covering problem defines a submodular objective then the k-max–min problem can be solved via constrained submodular optimization, e.g. [11,34]. However, most natural covering problems do not yield submodular functions, and so these results cannot be applied directly.

The k-robust and k-max–min *set cover* problems were studied in [18]. The authors used an online set cover algorithm to obtain an $O(\log m \log n)$-approximation algorithm for k-max–min set cover. Then they used the k-max–min algorithm within an LP-rounding-based algorithm (à la [39]) to obtain the same approximation ratio for k-robust set cover. They also showed k-robust set cover to be $\Omega\left(\frac{\log m}{\log\log m} + \log n\right)$ 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 [5].

Apart from improving these results in context of set cover, one may want to develop algorithms for other k-robust and k-max–min 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-max–min Steiner forest*, we are given a metric space with a collection of source–sink pairs, and seek k source–sink pairs that incur the maximum (Steiner forest) connection cost. Although the online-based-framework [18] can be extended to give algorithms for other k-max–min problems, it does not yield approximation guarantees better than the (deterministic) online competitive ratios. Moreover, for k-robust problems other than set cover, the LP-rounding framework in [18] does not extend directly; this obstacle was also observed by Khandekar et al. [32] who gave combinatorial constant-factor approximation algorithms for k-robust Steiner tree and facility location.

1.1 Main results and techniques

In this paper, we present a general template to design algorithms for k-robust and k-max–min problems. We go beyond the online-based approach and obtain tighter approximation ratios that nearly match the respective offline guarantees; see the table below. 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 approximation algorithms for both k-robust Steiner forest—which was left open by [32]—and for k-robust min-cut, and an $O\left(\frac{\log^2 n}{\log\log n}\right)$-approximation algorithm 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.

An important contribution of our work 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 the set of all elements e for which the cheapest set covering e 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, we 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. This guess-and-verify framework is formalized in Sects. 2.1 and 2.2.

The key step of our analysis is to argue why C_T is close to optimum. We briefly describe the intuition; details appear in Sect. 4. Suppose for a contradiction that $C_T \gg \beta\mathsf{Opt}$: 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 step uses the fact that each element in A is expensive to cover individually. 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. These involve establishing certain *net-type properties* for the respective covering problems (which imply the existence of such a witness $A' \subseteq A$ of size k), and represent our main technical contribution. The proofs for set cover, min-cut and multicut are based on *dual-rounding*.

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 we handle this using a careful Gomory-Hu-tree-based argument. Moreover, we have to show the following property: if the cut for a set of sources A is large (costs $\gg \mathsf{Opt}$) and each source in A has a high individual cut ($\gg \mathsf{Opt}/k$) then there is a witness $A' \subseteq A$ of at most k sources for which the cut is also large ($\gg \mathsf{Opt}$). To this end, we prove new flow-aggregation lemmas for single-sink flows using Steiner-tree-packing results, and for multiflows using oblivious routing [35]; both proofs are possibly of independent interest.

In the case of k-robust Steiner forest, directly rounding the dual to obtain the net-type property does not work, and instead we give a *primal–dual* argument.

The table below summarizes the best-known approximation ratios for various covering problems in the offline, k-robust and online models (results denoted $*$ are in the present paper).

Problem	Offline	k-robust	Deterministic online
Set cover	$\ln n$	$O(\log m \cdot \log n)$ [18] $O(\log m + \log n)$ $(*)$	$O(\log m \cdot \log n)$ [5]
	$(1 - o(1)) \ln n$ [17]	$\Omega\left(\log n + \frac{\log m}{\log\log m}\right)$ [18]	$\Omega\left(\frac{\log m \cdot \log n}{\log\log m + \log\log n}\right)$ [5]
Steiner tree	1.39 [10]	5.33 [32], 4.35 $(*)$	$\Theta(\log n)$ [30]
Steiner forest	2 [2,23]	10 $(*)$	$\Theta(\log n)$ [7]
Minimum cut	1	17 $(*)$	$O(\log^3 n \cdot \log\log n)$ [4,29]
Multicut	$O(\log n)$ [22]	$O\left(\frac{\log^2 n}{\log\log n}\right)$ $(*)$	$O(\log^3 n \cdot \log\log n)$ [4,29]

Paper outline In Sect. 2 we present the formal framework for k-robust and k-max–min problems, and abstract out the properties that we would like from our algorithms. Based on these definitions, the results for the various covering problems can be read independently of each other. Sect. 3 contains the algorithm for k-robust Steiner tree: while this result is simple and does not require rounding the dual, it is a nice example of our framework in action. Then Sect. 4 contains the algorithm for k-robust set cover, which introduces the dual-rounding analysis. The algorithms for k-robust Min-cut, multicut and Steiner forest appear in Sects. 5, 6 and 8, respectively.

1.2 Related work

Robust optimization is well-studied in the operations research literature, see e.g. the surveys [1,8]. Approximation algorithms for robust optimization was initiated by Dhamdhere et al. [15]: they studied the case when the scenarios were explicitly listed, and gave constant-factor approximation algorithms for Steiner tree and facility location, and logarithmic approximation algorithms for mincut and multicut. The algorithms in [15] were based on linear programming. Golovin et al. [24] improved the mincut result to a constant approximation ratio, and also gave an $O(1)$-approximation algorithm for robust shortest-paths. The algorithms in [24] were also "thresholded algorithms" and the algorithms in this paper can be seen as extensions of that idea to more complex uncertainty sets (the uncertainty set in [24] only contained singleton demands) and to a larger class of problems. The algorithms in these papers [15,24] handled inflation parameters that were scenario-dependent (but uniform across elements in each scenario). However, since the k-robust model has an exponential number of scenarios, we assume uniform inflation λ across scenarios.

The k-robust model was introduced in Feige et al. [18], where they gave an $O(\log m \log n)$-approximation algorithm for k-robust set cover; here m and n are the number of sets and elements in the set system. To get such an algorithm [18] first gave an $O(\log m \log n)$-approximation algorithm for k-max–min set-cover problem using the online algorithm for set cover [5]. They then used the k-max–min problem as a separation oracle in an LP-rounding-based algorithm (à la [39]) to get the same approximation guarantee for the k-robust problem. They also showed an $\Omega\left(\frac{\log m}{\log\log m}\right)$ hardness of approximation for k-max–min and k-robust set cover. Khandekar et al. [32]

noted that the LP-based techniques of [18] did not give good results for Steiner tree, and developed new combinatorial constant-factor approximation algorithms 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 [32] for the Steiner forest problem on trees.

Constrained submodular maximization problems [11,19,34,41] 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 a factor $(1 - 1/e)$. 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 at least a logarithmic factor. E.g., one can use tree embeddings [16] 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 and obtain approximation algorithms for the k-max–min problems (as in [18]). Though the LP-based framework [18] for k-robust problems does not seem to extend to problems other than set cover, in the companion paper [27] we give a general algorithm for robust covering using offline and online algorithms. However, since our goal in this paper is to obtain approximation factors better than the online competitive ratios, it is unclear how to use any of these results.

Considering the *average* instead of the worst-case performance gives rise to the well-studied model of two-stage stochastic optimization [31,37]. Here the algorithm assumes access to a probability distribution of the demand, and seeks to minimize the expected covering cost. There are two general techniques known for approximating covering problems in the two-stage stochastic model: LP-based algorithms [39], and combinatorial algorithms using certain cost-sharing properties [28]. Recently, some common generalizations of the robust and stochastic models have also been considered: see, e.g., Swamy [42] and Agrawal et al. [3].

To the best of our knowledge, none of the k-max–min problems other than min-cut and set cover [18] 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 [20,40], Steiner tree [21], Steiner forest [26], min-cut and multicut [25,35]. However these problems seem to be related to the k-max–min versions only in spirit.

2 Notation and definitions

Throughout, for any integer ℓ, we denote by $[\ell]$ the set $\{1, 2, \ldots, \ell\}$.

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 $\mathsf{Robust}(\Pi)$, is a *two-stage optimization* problem, where elements are bought in either 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 combined must satisfy each requirement in ω. Formally, the input to problem $\mathsf{Robust}(\Pi)$ 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 $\mathsf{Robust}(\Pi)$ 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, we deal with robust covering problems under *k-robust* 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 minimum cost of a solution covering 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 $\Omega = \binom{[n]}{k}$.

For all problems considered in this paper, we assume that the costs are polynomially bounded integers: this can be ensured by standard scaling arguments, while only increasing the approximation ratio by $o(1)$.

2.1 Desirable properties for k-robust approximation 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* if given as input any instance of $\mathsf{Robust}_k(\Pi)$ and a threshold T, the algorithm outputs in polynomial time, (i) a set $\Phi_T \subseteq E$, and (ii) the description of an algorithm that for any input $D \in \binom{[n]}{k}$ produces $\mathsf{Aug}(D \mid \Phi_T) \subseteq E$, such that:

A. For every scenario $D \in \binom{[n]}{k}$,
   (i) the elements in $\Phi_T \cup \mathsf{Aug}(D \mid \Phi_T)$ satisfy all requirements in D, and
   (ii) the resulting augmentation cost $c\,(\mathsf{Aug}(D \mid \Phi_T)) \leq \beta \cdot T$.

B. Let F^* 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 $c(\Phi_T) \leq \alpha_1 \cdot F^* + \alpha_2 \cdot T^*$.

It is clear that a solution to $\mathsf{Robust}_k(\Pi)$ need only specify the first stage solution $\Phi \subseteq E$: the natural augmentation algorithm will simply run an approximation algorithm for the deterministic covering problem on the residual instance. However, since we use a "guess and verify" approach, we will need to compare the objective values under different first-stage solutions (in order to verify our algorithm's guess T of the optimal second-stage cost). For an arbitrary first-stage solution Φ, computing its objective value is just as hard as solving a k-max–min instance. To get around this issue, Definition 2.1 requires solutions Φ_T that also come with an upper bound (namely $\beta \cdot T$) of their second-stage cost. Thus, we can easily compute an upper bound $c(\Phi_T) + \beta T$ on the objective value, which can be used to compare different solutions in the following analysis.

The next lemma 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 $\approx F^* + T^*$ in the first stage and $\approx \lambda T^*$ in the second stage; for guesses $T \ll T^*$, the first-stage cost in guarantee (B) 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 there is a* $\max\left\{\alpha_1, \beta + \frac{\alpha_2}{\lambda}\right\}$*-approximation algorithm for* $\mathsf{Robust}_k(\Pi)$.

Proof Let $\mathcal{A}$ denote an algorithm for $\mathsf{Robust}_k(\Pi)$ such that it is $(\alpha_1, \alpha_2, \beta)$-discriminating. Let ground-set $E = [m]$, $c_{max} := \max_{e\in[m]} c_e$ and $N := m \cdot c_{max}$. Since all costs are polynomially bounded integers, so is N.

The approximation algorithm for $\mathsf{Robust}_k(\Pi)$ runs the $(\alpha_1, \alpha_2, \beta)$-discriminating algorithm $\mathcal{A}$ for every choice of $T \in \{0, 1, \ldots, N\}$, and returns the solution corresponding to:

$$\widetilde{T} := \arg\min\left\{c(\Phi_T) + \lambda \cdot \beta\, T \mid 0 \leq T \leq N\right\}.$$

Recall that T^* denotes the optimal second-stage cost, clearly $T^* \leq m \cdot c_{max} = N$. The objective value of the solution from $\mathcal{A}$ for threshold $\widetilde{T}$ can be bounded as follows.

$$\begin{aligned} c\left(\Phi_{\widetilde{T}}\right) + \lambda \cdot \max_{\omega\in\Omega} c\left(\mathsf{Aug}(\omega \mid \Phi_{\widetilde{T}})\right) &\leq c(\Phi_{\widetilde{T}}) + \lambda \cdot \beta\, \widetilde{T} \\ &\leq c(\Phi_{T^*}) + \lambda \cdot \beta\, T^* \\ &\leq \left(\alpha_1 \cdot F^* + \alpha_2 \cdot T^*\right) + \beta\lambda \cdot T^* \\ &= \alpha_1 \cdot F^* + \left(\beta + \frac{\alpha_2}{\lambda}\right) \cdot \lambda T^*. \end{aligned}$$

The first inequality follows from Property A(ii) in Definition 2.1; the second by the choice of $\widetilde{T}$; the third by Property B (applied with threshold T^*) in Definition 2.1.

Thus this algorithm for $\mathsf{Robust}_k(\Pi)$ outputs a solution that is a max $\left\{\alpha_1, \beta + \frac{\alpha_2}{\lambda}\right\}$-approximation. □

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 for k-max–min approximation algorithms

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

Definition 2.3 An algorithm for $\mathsf{Robust}_k(\Pi)$ is $(\alpha_1, \alpha_2, \beta)$-*strongly discriminating* if it satisfies the properties in Definition 2.1, and when the inflation parameter is $\lambda = 1$ (and hence $F^* = 0$), the algorithm also outputs a set $Q_T \in \binom{[n]}{k}$ such that if $c(\Phi_T) > \alpha_2 T$ then the cost of optimally covering the set Q_T is greater than T.

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

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 there is an algorithm that outputs a* $(\alpha_2 + \beta)$-*approximate solution to k-max–min*(Π).

Proof 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 first-stage cost $F^* = 0$ (since $\lambda = 1$). Note that the optimal value of the k-max–min instance also equals T^*.

As in Lemma 2.2, let ground-set $E = [m]$, $c_{max} := \max_{e \in [m]} c_e$ and $N := m \cdot c_{max}$. Since all costs are polynomially bounded integers, so is N.

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

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

Above, the inequalities are by conditions A(i) and A(ii) of Definition 2.1. Setting $T = \widetilde{T}$ here, and by choice of $\widetilde{T}$,

$$T^* \leq c(\Phi_{\widetilde{T}}) + \beta\, \widetilde{T} \leq (\alpha_2 + \beta) \cdot \widetilde{T}.$$

Hence $\widetilde{T}$ is an $(\alpha_2 + \beta)$-approximation to the max–min value T^*. Now applying the condition of Definition 2.3 with $T = \widetilde{T} - 1$, where $c(\Phi_{\widetilde{T}-1}) > \alpha_2 \cdot (\widetilde{T} - 1)$ by choice of index $\widetilde{T}$, we obtain that the minimum cost to cover requirements $Q_{\widetilde{T}-1}$ is greater than $\widetilde{T} - 1$, i.e. at least $\widetilde{T}$. This implies the desired approximation guarantee. □

3 *k*-robust Steiner tree

In the k-robust Steiner tree problem, we are given an undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, a set $U \subseteq V$ of potential terminals, and a root vertex $r \in U$. Any set of k terminals from U—i.e., any set in $\binom{U}{k}$—is a valid scenario in the second stage. There is also an inflation factor $\lambda \geq 1$, where each edge costs λ times more in the second stage. The goal is to select edges in the first stage (without knowledge of the scenario) and second stage (knowing the realized scenario) so that the realized terminals are connected to r, and the worst case cost is minimized. Let $d : V \times V \to \mathbb{R}_+$ denote the shortest-path distances according to the edge costs c. For any vertex $v \in V$ and subset $S \subseteq V$, define the distance $d(v, S) := \min_{w \in S} d(v, w)$ to be the minimum distance between v and some vertex in S.

By the results in Sect. 2.1, a discriminating algorithm immediately gives us an algorithm for k-robust Steiner tree, and this is how we shall proceed. The algorithm picks a "net" $S \subseteq U$ at distance separation of $\beta T/k$, and builds a minimum spanning tree (MST) on S as the first stage solution.

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 $U \backslash S$ is close to some point in the net S, this automatically ensures that the second stage recourse cost is small. Formally,

Algorithm 1 Algorithm for k-robust Steiner tree

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

Claim 3.1 (Property A for Steiner tree) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \cup \mathsf{Aug}(D \mid \Phi_T)$ connect the terminals in D to the root r, and have cost $c(\mathsf{Aug}(D \mid \Phi_T)) \leq \beta T$.*

Proof From the definition of the second-stage solution, $\mathsf{Aug}(D \mid \Phi_T)$ 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 $\Phi_T \cup \mathsf{Aug}(D \mid \Phi_T)$ 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 $d(i, S) \le \beta \frac{T}{k}$. Thus,

$$c(\mathsf{Aug}(D \mid \Phi_T)) \le \sum_{i \in D} c(\mathsf{Aug}(\{i\} \mid \Phi_T)) = \sum_{i \in D} d(i, S) \le \frac{|D|}{k} \cdot \beta T = \beta T.$$

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

It now remains to show that the algorithm satisfies Property B as well. 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 F^* must have been large as well.

Theorem 3.2 (Property B for Steiner tree) *Let F^* and T^* denote the optimal first and second stage costs. If $\beta \ge 2$ and $T > T^*$ then $c(\Phi_T) \le \frac{2\beta}{\beta-2} \cdot F^* + 2 \cdot T^*$.*

Proof First suppose $|S| \le k$: then it is clear that there is a Steiner tree on $\{r\} \cup S$ of cost at most $F^* + T^*$. The algorithm finds one of cost at most twice that, since the length of an MST is at most two times that of an optimal Steiner tree. In the following assume that $|S| > k$.

Let $LP(S)$ denote the minimum length of a *fractional* Steiner tree on terminals $\{r\} \cup S$, i.e. the optimal value of the natural cut-based LP relaxation for Steiner tree. Since vertices of S are at least distance $\beta \cdot \frac{T}{k} \ge \beta \cdot \frac{T^*}{k}$ from each other, we get $LP(S) \ge \frac{\beta}{2k} \cdot |S| \cdot T^*$. We now construct a fractional Steiner tree $x : E \to \mathbb{R}_+$ of small length. Number the terminals in S from $\{0, 1, \ldots, |S|-1\}$ arbitrarily, and for each $0 \le j \le |S| - 1$ let $A_j = \{j, j+1, \ldots, j+k-1\}$ (modulo $|S|$). Let $\Phi^* \subseteq E$ denote the edges of the optimal first stage solution, and $\Pi_j \subseteq E$ be the second-stage edges in the optimal solution under scenario A_j. So for all $0 \le j \le |S|-1$, $\Phi^* \cup \Pi_j$ is a Steiner tree on terminals $\{r\} \cup A_j$, and $c(\Pi_j) \le T^*$. Define $x := \chi(\Phi^*) + \frac{1}{k} \cdot \sum_{j=0}^{|S|-1} \chi(\Pi_j)$, where $\chi(P) \in \{0, 1\}^E$ denotes the characteristic vector for any $P \subseteq E$. We claim that x supports unit flow from r to any $i \in S$. Note that there are k sets $A_{i-k+1}, \ldots, A_i$ (indices are modulo $|S|$) that contain i, and for each $i - k + 1 \le j \le i$, it is clear that $\frac{1}{k} \cdot \big(\chi(\Phi^*) + \chi(\Pi_j)\big)$ supports $\frac{1}{k}$ flow from r to i. So $\chi(\Phi^*) + \frac{1}{k} \cdot \sum_{j=i-k+1}^{i} \chi(\Pi_j)$, which is dominated by x, supports unit flow from r to i. Thus x is a feasible fractional Steiner tree on $\{r\} \cup S$, of cost at most $F^* + \frac{|S|}{k} \cdot T^*$. Combined with the lower bound on $LP(S)$,

$$|S| \cdot \frac{\beta}{2} \cdot \frac{T^*}{k} \le LP(S) \le F^* + \frac{|S|}{k} \cdot T^*. \tag{3.1}$$

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

From Claim 3.1 and Theorem 3.2, 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 yields a $1.39\cdot\lambda$ approximation, using the 1.39-approximation algorithm for deterministic Steiner tree [10]. Hence the better of these two ratios gives an approximation bound better than 4.35.

The k-max–min Steiner tree problem We show that the above algorithm can be extended to be $\left(\frac{2\beta}{\beta-2}, 2, \beta\right)$ strongly discriminating. As shown above, it is indeed discriminating. To show that Definition 2.3 holds, consider the proof of Theorem 3.2 when $\lambda = 1$ (so $F^* = 0$) and suppose that $c(\Phi_T) > 2\,T$. The algorithm to output the k-set Q proceeds via two cases.

1. If $|S| \le k$ then $Q := S$. The minimum Steiner tree on Q is at least half its MST, i.e. at least $\frac{1}{2}c(\Phi_T) > T$.
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 $\beta\frac{T}{2k}$ around each Q-vertex, and so $LP(Q) > \frac{\beta}{2}T \ge T$. Thus the minimum Steiner tree on Q is more than T.

3.1 Unrooted Steiner tree

Here we show that our algorithm extends to the setting of k-robust *unrooted* Steiner tree, which was previously studied in [32]. 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 will show that setting $r \in U$ to an arbitrary terminal in Algorithm 1 achieves the same approximation ratio for the unrooted case as well. This algorithm is essentially same as the one used by [32], but with different parameters: hence our framework can be viewed as generalizing their algorithm. Our proof is shorter and gives a slightly better approximation ratio.

It is clear that Claim 3.1 continues to hold in this unrooted 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 3.3 (Property B for unrooted Steiner tree) *Let F^* and T^* denote the optimal first and second stage costs. If $\beta \ge 2$ and $T \ge T^*$ then $c(\Phi_T) \le \frac{2\beta}{\beta-2}\cdot F^* + 2T^*$.*

Proof Firstly suppose $|S| \le k$: then it is clear that there is a Steiner tree on S of cost at most $F^* + T^*$, 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). As in Theorem 3.2, we have $LP(S) \ge \frac{\beta}{2k}\cdot|S|\cdot T^*$. Again, we will construct a fractional Steiner tree $x : E \to \mathbb{R}_+$ of small length. Recall the notation from Theorem 3.2: terminals in S are numbered arbitrarily from $\{0, 1, \ldots, |S|-1\}$; for each $0 \le j \le |S|-1$, $A_j := \{j, j+1, \ldots, j+k-1\}$; $\Phi^* \subseteq E$ are the edges of the optimal first stage solution; and $\Pi_j \subseteq E$ are the optimal second-stage edges under scenario A_j. Note that for all $0 \le j \le |S| - 1$, $\Phi^* \cup \Pi_j$ is a Steiner tree on terminals A_j, and $c(\Pi_j) \le T^*$. Define $x := \chi(\Phi^*) + \frac{1}{k} \cdot \sum_{j=0}^{|S|-1} \chi(\Pi_j)$.

Claim 3.4 *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 $A_{i-k+2}, \ldots, A_i$ that contain both i and $i + 1$. Let $J := \{i - k + 2, \ldots, i\}$. So for each $j \in J$, $\frac{1}{k} \cdot \left(\chi(\Phi^*) + \chi(\Pi_j)\right)$ supports $\frac{1}{k}$ flow from i to $i + 1$. Furthermore, $\left(\cup_{l \in S \setminus J} \Pi_l\right) \cup F^*$ is a Steiner tree connecting terminals $\cup_{l \in S \setminus J} A_l \supseteq \{i, i + 1\}$; i.e. $\frac{1}{k} \cdot \left(\chi(F^*) + \sum_{l \in S \setminus J} \chi(\Pi_l)\right)$ also supports $\frac{1}{k}$ flow from i to $i + 1$. This proves the claim. □

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

$$|S| \cdot \frac{\beta}{2} \cdot \frac{T^*}{k} \le LP(S) \le F^* + \frac{|S|}{k} \cdot T^*. \tag{3.2}$$

Thus we have $LP(S) \le \frac{\beta}{\beta - 2} \cdot F^*$, 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 improves on the constant obtained by [32] for the same problem.

Theorem 3.5 *There is a 4.35-approximation algorithm for (unrooted) k-robust Steiner tree.*

4 *k*-robust set cover

In this section we consider the k-robust set cover problem. There is a set system $(U, \mathcal{F})$ with a universe U of n elements, and m sets in $\mathcal{F}$ with each set $R \in \mathcal{F}$ having cost c_R in the first stage and $\lambda \cdot c_R$ in the second stage (where $\lambda \ge 1$ is the inflation factor). Any subset of k elements is a possible scenario in the second stage. The goal is to select sets from $\mathcal{F}$ in both stages so that all realized elements are covered, and the worst-case cost is minimized. Given Lemma 2.2, it suffices to show a discriminating algorithm as defined in Definition 2.1 for this problem. The algorithm given below simply picks all elements which can only be covered by expensive sets, and covers them in the first stage using the greedy algorithm for set cover.

Algorithm 2 Algorithm for k-robust set cover

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

Claim 4.1 (Property A for set cover) *For all $T \ge 0$ and scenario $D \in \binom{U}{k}$, the sets $\Phi_T \bigcup \mathsf{Aug}(D \mid \Phi_T)$ cover elements in D, and have cost $c(\mathsf{Aug}(D \mid \Phi_T)) \le \beta T$.*

Proof The elements in $D \cap S$ are covered by Φ_T. Each element $i \in D \backslash S$ is covered by set $\mathsf{Aug}(\{i\} \mid \Phi_T)$. So by the definition of $\mathsf{Aug}(D \mid \Phi_T)$ we have the first part of the claim. For the second part, note that by definition of S, the cost of set $\mathsf{Aug}(\{i\} \mid \Phi_T)$ is at most $\beta\, T/k$ for all $i \in U$. □

Below $H_n := \sum_{i=1}^{n} \frac{1}{i} \approx \ln n$; recall that the greedy algorithm for set cover has an H_n approximation ratio, where n is the number of elements in the given instance. Moreover, this guarantee holds with respect to the natural linear programming relaxation.

Theorem 4.2 (Property B for set cover) *Let Φ^* denote the optimal first stage solution, and T^* the optimal second stage cost. Let $\beta = 36 \ln m$. If $T \geq T^*$ then $c(\Phi_T) \leq H_n \cdot (c(\Phi^*) + 12 \cdot T^*)$.*

Proof We claim that there is a *fractional* solution $\mathbf{x}$ for the set covering instance $(S, \mathcal{F})$ with cost at most $c(\Phi^*) + 12 \cdot T^*$, whence rounding this to an integer solution implies the theorem. For a contradiction, we will assume that the minimum cost fractional set cover is at least $c(\Phi^*) + 12 \cdot T^*$. So there must be a dual LP solution of at least this value. We then *round this dual solution* to get another dual solution to a sub-instance with only k elements, that costs more than $c(\Phi^*) + T^*$, which would contradict with the definition of the optimal cost.

To this end, let $S' \subseteq S$ denote the elements that are *not* covered by the optimal first stage solution Φ^*, and let $\mathcal{F}' \subseteq \mathcal{F}$ denote the sets that contain at least one element from S'. By the choice of S, each set in $\mathcal{F}'$ costs at least $\beta \cdot \frac{T}{k} \geq \beta \cdot \frac{T^*}{k}$. Define the “coarse” cost for each 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 following primal–dual pair of LPs for the set cover instance with elements S' and sets $\mathcal{F}'$ having the coarse costs $\widehat{c}$.

$$\begin{aligned} \min \sum_{R \in \mathcal{F}'} \widehat{c}_R \cdot x_R & \\ \sum_{R \ni e} x_R \geq 1, &\quad \forall e \in S', \\ x_R \geq 0, &\quad \forall R \in \mathcal{F}'. \end{aligned} \qquad\qquad \begin{aligned} \max \sum_{e \in S'} y_e & \\ \sum_{e \in R} y_e \leq \widehat{c}_R, &\quad \forall R \in \mathcal{F}', \\ y_e \geq 0, &\quad \forall e \in S'. \end{aligned}$$

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 4.3 *If $\beta = 36 \ln m$, then the optimal 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 4.3, let us assume it and complete the proof of Theorem 4.2. Given the primal LP solution $\{x_R\}_{R \in \mathcal{F}'}$ that covers elements in S', define an LP solution to cover elements in S as follows: set $z_R := 1$ if $R \in \Phi^*$, $z_R := x_R$ if $R \in \mathcal{F}' \backslash \Phi^*$; and $z_R = 0$ otherwise. Since the solution $\mathbf{z}$ contains Φ^* integrally, it clearly covers elements $S \backslash S'$ (i.e. the portion of S covered by Φ^*). And since $z_R \geq x_R$ for all R, solution $\mathbf{z}$ also covers S' fractionally. Finally, the cost of this solution is

$\sum_R c_R z_R \leq c(\Phi^*) + \sum_R c_R \cdot x_R \leq c(\Phi^*) + \frac{6T^*}{k} \cdot \sum_R \widehat{c}_R \cdot x_R$. Now Claim 4.3 bounds this by $c(\Phi^*) + 12 \cdot T^*$. Since we have an LP solution of value $c(\Phi^*) + 12T^*$, and the greedy algorithm is an H_n-approximation relative to the LP value for set cover, this would complete the proof of Theorem 4.2.

It remains to give the proof for Claim 4.3 above; indeed, that is where the technical heart of the result lies.

Proof of Claim 4.3 Recall that we want to bound the optimal fractional set cover cost for the instance $(S', \mathcal{F}')$ with the coarse (integer) costs; $\{x_R\}_{R\in\mathcal{F}'}$ and $\{y_e\}_{e\in S'}$ are the optimal primal and dual solutions. For a contradiction, assume that the LP cost $\sum_{R\in\mathcal{F}'} \widehat{c}_R \cdot x_R = \sum_{e\in S'} y_e$ 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 the random variables $\{Y_e/3\}_{e\in S'}$ form a feasible dual solution with high probability, i.e., they satisfy $\sum_{e\in R}(Y_e/3) \leq \widehat{c}_R$ for all $R \in \mathcal{F}'$. 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 $\Pr[\sum_{e\in R} Y_e > 3 \cdot \widehat{c}_R] \leq \Pr[\sum_{e\in R} I_e > 2 \cdot \widehat{c}_R]$. Note also that $\sum_{e\in R} E[I_e] \leq \sum_{e\in R} y_e \leq \widehat{c}_R$. Now we use a Chernoff bound [33] to bound the probability that the sum of independent 0–1 random variables, $\sum_{e\in R} I_e$, exceeds twice its mean. This gives $\Pr[\sum_{e\in R} I_e > 2 \cdot \widehat{c}_R] \leq e^{-\widehat{c}_R/3} \leq e^{-\beta/18} \leq m^{-2}$, since each $\widehat{c}_R \geq \beta/6$ and $\beta = 36 \cdot \ln m$. Finally, a trivial union bound implies that $\{Y_e/3\}_{e\in S'}$ satisfies all the m dual 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 $k \geq 1$). By another Chernoff Bound, $\Pr[\sum_{e\in S'} Y_e > \frac{\gamma-1}{2} \cdot k] \geq a$, where $a > 0$ is some fixed 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 we 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 at least $1/3$. This dual solution $\{Y'_e\}_{e\in Q}$ 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 were the realized scenario, the optimal second stage cost would exceed T^* (recall, no element in Q is covered by F^*). This contradicts the definition of the optimal solution. Thus we must have $\gamma \leq 2$ and so the LP optimum $\sum_{R\in\mathcal{F}'} \widehat{c}_R \cdot x_R \leq 2k$, which completes the proof of Claim 4.3. □

This finishes the proof of Theorem 4.2. □

Claim 4.1 and Theorem 4.2 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 [18].

Theorem 4.4 *There is an $O(\log m + \log n)$-approximation algorithm for k-robust set cover.*

The k-max–min set cover problem The proof of Claim 4.3 suggests how to get a $(H_n, 12H_n, 36 \ln m)$ strongly discriminating algorithm. When $\lambda = 1$ (and so $c(\Phi^*) = 0$), 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 more than T (witnessed by the dual solution having cost $> T$). This step can be derandomized using conditional expectation and pessimistic estimators [36]. Now using Lemma 2.4, we get the claimed $O(\log m + \log n)$-approximation algorithm for the k-max–min set cover problem. This nearly matches the hardness of $\Omega\left(\frac{\log m}{\log \log m} + \log n\right)$ given by [18].

4.1 Set cover with non-uniform inflation

Here we consider the k-robust set cover problem under *set-dependent inflation*, and show that our algorithm extends to this setting as well. Recall that there is a set system $(U, \{R_j\}_{j=1}^m)$ with a universe U of n elements and m sets. Moreover, there are two different cost-vectors $\mathbf{b}, \mathbf{c} \in \mathbb{R}_+^m$ denoting the costs in the first and second stage, respectively. Again, any subset of k elements is a possible scenario, and the goal is to minimize the worst-case cost (summed over both stages). The usual k-robust set cover (studied above) is the special case when $\mathbf{c} = \lambda\, \mathbf{b}$ for some uniform inflation factor $\lambda \geq 1$.

We may assume, without loss of generality, that the first-stage cost for each set is at most its second-stage cost, i.e. $\mathbf{b} \leq \mathbf{c}$. (If some set R has $c_R < b_R$, then we pretend that its first-stage cost is c_R; and if R is chosen into the first-stage solution it can always be bought in the second stage). Under non-uniform inflations, the definition of an $(\alpha_1, \alpha_2, \beta)$-discriminating algorithm is the same as Definition 2.1 where Condition B is replaced by:

B'. Let Φ^* denote the optimal first stage solution, and T^* the optimal second stage c-cost (hence the optimal value $\mathsf{Opt} = b(\Phi^*) + T^*$). If the threshold $T \geq T^*$ then the first stage cost $b(\Phi_T) \leq \alpha_1 \cdot b(\Phi^*) + \alpha_2 \cdot T^*$.

It can be shown exactly as in Lemma 2.2, that any such algorithm is a $\max\{\alpha_1,\ \alpha_2 + \beta\}$-approximation algorithm for k-robust set cover. Note that the factor α_2 was scaled down by λ in the uniform inflation case (Lemma 2.2). The algorithm and analysis here are very similar to that for k-robust set-cover under uniform inflation.

Algorithm 3 Algorithm for k-robust set cover with non-uniform inflation

```
input: non-uniform $k$-robust set-cover instance and bound $T$.
let $\beta \leftarrow 36 \cdot \ln m$, and $S \leftarrow \left\{ v \in U \mid \text{ minimum } c\textbf{-cost} \text{ set in } \mathcal{F} \text{ covering } v \text{ has cost at least } \beta \cdot \frac{T}{k} \right\}$.
output first stage solution $\Phi_T$ as the Greedy-Set-Cover($S$) under $b$-costs.
define $\mathsf{Aug}(\{i\} \mid \Phi_T)$ as the minimum $c$-cost set covering $i$ if $i \in U \setminus S$, and $\emptyset$ otherwise.
output second stage solution where $\mathsf{Aug}(\omega \mid \Phi_T) := \bigcup_{i \in \omega} \mathsf{Aug}(\{i\} \mid \Phi_T)$ for all $\omega \subseteq U$.
```

We will show that this algorithm is $(H_n,\ 12H_n,\ 36 \ln m)$-discriminating. The following claim is immediate.

Claim 4.5 (Property A) *For all $T \geq 0$ and $\omega \subseteq U$, the sets $\Phi_T \bigcup \mathsf{Aug}(\omega \mid \Phi_T)$ cover elements ω. Additionally, if $|\omega| \leq k$ then the cost $c(\mathsf{Aug}(\omega \mid \Phi_T)) \leq \beta T$.*

Theorem 4.6 (Property B') *Assume $\beta \geq 36 \cdot \ln m$. If $T \geq T^*$ then $b(\Phi_T) \leq H_n \cdot (b(\Phi^*) + 12 \cdot T^*)$.*

Proof We will show that there is a *fractional* solution $\mathbf{x}$ for covering S with small b-**cost**, at most $b(\Phi^*) + 12 \cdot T^*$, whence rounding this to an integer solution implies the theorem.

Let $S' \subseteq S$ denote the elements that are *not* covered by the optimal first stage solution Φ^*, and let $\mathcal{F}' \subseteq \mathcal{F}$ denote the sets that contain at least one element from S'. By the choice of S, each set in $\mathcal{F}'$ has c-**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 LP 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 fractional solution; then Claim 4.3 applies directly to yield:

$$\sum_{R\in\mathcal{F}'} \widehat{c}_R \cdot x_R \leq 2 \cdot k \tag{4.1}$$

Given the primal LP solution $\{x_R\}_{R\in\mathcal{F}'}$ covering elements in S', define a fractional solution $\mathbf{z}$ covering elements S as follows: define $z_R = 1$ if $R \in \Phi^*$, $z_R = x_R$ if $R \in \mathcal{F}' \backslash \Phi^*$, and $z_R = 0$ otherwise. Since the solution $\mathbf{z}$ contains F^* integrally, it covers elements $S \backslash S'$ (i.e. the portion of S covered by F^*); since $z_R \geq x_R$ for all $R \in \mathcal{F}'$, it follows that $\mathbf{z}$ fractionally covers S'. Finally, the b-**cost** of this solution is:

$$\begin{aligned}\sum_{R\in\mathcal{F}} b_R \cdot z_R, &= b(\Phi^*) + \mathbf{b} \cdot \mathbf{x} \ \leq \ b(\Phi^*) + \mathbf{c} \cdot \mathbf{x} \leq b(\Phi^*) + \frac{6T^*}{k} \cdot (\widehat{\mathbf{c}} \cdot \mathbf{x}) \\ &\leq b(\Phi^*) + 12 \cdot T^*,\end{aligned}$$

where the second inequality uses $\mathbf{b} \leq \mathbf{c}$, the next one is by definition of $\widehat{\mathbf{c}}$ and the last inequality is from (4.1). Thus we have an LP solution of b-**cost** at most $b(\Phi^*) + 12T^*$, and since the greedy algorithm has an H_n-approximation ratio relative to the LP value, this completes the proof. □

Thus we obtain:

Theorem 4.7 *There is an $O(\log m + \log n)$-approximation algorithm for k-robust set cover with set-dependent inflation factors.*

Remark For the other covering problems, our algorithms do not extend to the case of non-uniform inflation: this is usually inherent, and not just a shortcoming of our analysis. E.g., [32] gave an $\Omega(\log^{1/2-\epsilon} n)$-hardness of approximation for k-robust Steiner forest under just two distinct inflation-factors, whereas we give an $O(1)$-approximation algorithm under uniform inflations (in Sect. 8).

5 k-robust minimum cut

We now consider the k-robust minimum cut problem, where we are given an undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, a root $r \in V$, terminals $U \subseteq V$, and an inflation factor $\lambda \geq 1$. Again, any subset in $\binom{U}{k}$ is a possible second-stage scenario. The goal is to select edges in both stages so that all terminals in the realized scenario are separated from the root r, and the worst-case cost is minimized. As before, it suffices to give a a discriminating algorithm (Definition 2.1). This algorithm is similar to the one for set cover: we just pick all the "expensive" terminals and separate them from the root in the first stage.

Algorithm 4 Algorithm for k-robust min-cut

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

Claim 5.1 (Property A for min-cut) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \bigcup \mathsf{Aug}(D \mid \Phi_T)$ separate the terminals D from r; moreover, the cost $c(\mathsf{Aug}(D \mid \Phi_T)) \leq \beta T$.*

Theorem 5.2 (Property B for min-cut) *Let Φ^* denote the optimal first stage solution and T^* the optimal second stage cost. If $\beta \geq \frac{10e}{e-1}$ and $T \geq T^*$ then $c(\Phi_T) \leq 3 \cdot c(\Phi^*) + \frac{\beta}{2} \cdot T^*$.*

Before the formal proof, we provide some 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 will "round" this dual solution by aggregating these flows to get a set of k terminals that have a large combined flow (of value $> c(\Phi^*) + T^*$) to the root—but this is impossible, since the optimal solution promises us a cut of at most $c(\Phi^*) + T^*$ for any set of k terminals. However, more work is required in formalizing this. For set-cover, each element was covered entirely in either the first-stage or the second-stage; whereas for cut problems, both stages may partially help in separating a terminal from the root. So we divide the set S into two different parts. The first part contains those terminals (called "low" nodes) whose min-cut in graph $G \backslash \Phi^*$ is smaller than that in G by some constant factor; we use a Gomory–Hu tree based analysis to show that all low nodes can be completely separated from r at cost $O(1) \cdot c(\Phi^*)$ (this is shown in Claim 5.3). The second part consists of the remaining terminals (called "high" nodes) that continue to have a large min-cut in $G \backslash \Phi^*$, and for these we use the dual rounding idea sketched above to show a min-cut of cost $O(T^*)$ (this is proved in Claim 5.4). Together, these claims imply Theorem 5.2.

To begin the proof of Theorem 5.2, let $H := G \backslash \Phi^*$, and let $S_h \subseteq S$ denote the "high" nodes whose min-cut from the root in H is at least $M := \frac{\beta}{2} \cdot \frac{T^*}{k}$. For any undirected graph $J = (V, F)$ and subset $U \subseteq V$ of vertices, we use the standard notation $\partial_J(U) := \{(u, v) \in F : u \in U, v \notin U\}$ for edges having exactly one end-point in the set U. The following claim is essentially from Golovin et al. [24].

Claim 5.3 (Cutting low nodes) *If $T \geq T^*$, the minimum cut in H separating $S \backslash S_h$ from r costs at most* $2 \cdot c(\Phi^*)$.

Proof Let $S' := S \backslash S_h$, 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 [38, Chap. 15] $\mathcal{T}(H)$ of graph H. 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 $r \notin D_u$: note that D_u corresponds to the cheapest edge on the r–u path in $\mathcal{T}(H)$, with vertices D_u on one side of this edge and $V \backslash D_u$ on the other side. 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 r and their corresponding sets $\{D_u : u \in S''\}$ are disjoint: the set of min r–u cuts (for $u \in S'$) in tree $\mathcal{T}(H)$ that are closest to 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 $c(\partial_G(D_u)) \geq 2M$ since 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, $c(F) \leq \sum_{u \in S''} c(\partial_H(D_u)) \leq \sum_{u \in S''} c(\partial_{F^*}(D_u)) \leq 2 \cdot c(\Phi^*)$. The last inequality uses disjointness of $\{D_u\}_{u \in S''}$. Thus the minimum $r - S'$ cut in H costs at most $2 \cdot c(\Phi^*)$. □

Claim 5.4 (Cutting high nodes) *If $T \geq T^*$ and $\beta \geq \frac{10 \cdot e}{e-1}$, the minimum cut in H separating S_h from r costs at most $\frac{\beta}{2} \cdot T^*$.*

Proof Consider an r–S_h maximum flow in the graph $H = G \backslash \Phi^*$, and suppose that it sends $\alpha_i \cdot M$ flow to each vertex $i \in S_h$. Note that the k-robust min-cut problem remains unchanged upon making copies of terminals: so we can assume without loss of generality that each $\alpha_i \in (0, 1]$. Hence if we show that $\sum_{i \in S_h} \alpha_i \leq k$, the total flow (which by flow-cut duality, 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 then show that there exists a subset $W \subseteq S_h$ with $|W| \leq k$ such that the minimum r–W cut in graph H costs 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 below) is useful.

Lemma 5.5 (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 flow into vertices of W is at least $\frac{1-e^{-1}}{4} \cdot \ell$ and (ii) the flow out of each $i \in X$ is at most $\epsilon_i/4$.*

We apply this lemma to the graph $H = G \backslash \Phi^*$ with terminal set S_h and edge-capacities equal to the costs $\mathbf{c}$ scaled down by M. Since for any cut separating $x, y \in S_h$, the root r lies on one side of this cut (say on y's side), min-cut$_H(x, y) \geq M$—hence the capacities satisfy the conditions of the lemma. Now set $\ell = k$, and $\epsilon_i := \alpha_i$ for each terminal $i \in S_h$. By our assumption, $\sum_{i \in S_h} \epsilon_i = \sum_{i \in S_h} \alpha_i \geq k = \ell$. Hence Lemma 5.5 finds a subset $W \subseteq S_h$ with k vertices, and a flow f in graph H (with capacities $\mathbf{c}$) such that f sends a total of at least $\frac{1-1/e}{4} \cdot kM$ units into vertices of W, and at most $\frac{\alpha_i}{4} \cdot M$ units out of each $i \in S_h$. Moreover, there is a feasible flow g in H (namely the max-flow from r to S_h) that simultaneously sends $\alpha_i \cdot M$ flow from r to each $i \in S_h$. Hence the flow $\frac{g+4f}{5}$ is *feasible* in H with capacities $\mathbf{c}$, which sends a total of at least $\frac{4}{5} \cdot \frac{1-1/e}{4} \cdot kM = \frac{1-1/e}{5} \cdot kM$ 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^*. This completes the proof of Claim 5.4. □

From Claim 5.1 and Theorem 5.2, 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 gives a trivial λ-approximate solution, the better of the two gives an approximation ratio of $\min\{\frac{\beta}{2\lambda} + \beta,\ \lambda\} < 17$. Thus we have,

Theorem 5.6 *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 with high probability. A natural approach to prove this would be to invoke Gale/Hoffman-type theorems [38, Chap. 11] using which it is necessary and sufficient to show for this random choice W that $c(\partial V') \geq |\text{demand}(V') - \text{supply}(V')|$ for all $V' \subseteq V$. But we would need to prove this condition 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. Furthermore, such an approach is likely to fail, since the redistribution lemma is false for directed graphs (see remark at the end of this section) whereas the Gale–Hoffman theorems hold for digraphs as well. 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 5.5 (Redistribution lemma) To begin, we assume without loss of generality, 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 let all new 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 W on this new graph can be mapped back to the original graph.

Let c_e denote the edge capacities in N. By the assumption that every cut in N separating X has capacity at least one, it follows that $\mathbf{c}$ is a feasible solution to the natural cut-based LP relaxation for Steiner tree on terminals X. Since this LP relaxation has an integrality gap of two, there exist Steiner trees $\{T_a\}_{a \in A}$ on the terminal set X that fractionally pack into the twice the edge capacities (see e.g. [12] for a formal argument). I.e., there exist positive multipliers $\{\lambda_a\}_{a\in A}$ such that $\sum_a \lambda_a = \frac{1}{2}$, and $\sum_a \lambda_a \cdot \chi(T_a) \le \mathbf{c}$, where $\chi(T_a)$ is the characteristic vector of the edge-set of 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 $\ell \le |X|\epsilon = q/P$.

Consider any fixed Steiner tree T_a in this collection, where its edges have 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 each 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 of 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 always feasible.

For any vertex $v \in X$ and a tour τ_a, the probability that v is *not* a-close is at most $(1 - \frac{q/\ell}{q})^\ell \le e^{-1}$ by the random choice of W; 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 the flows from all the Steiner trees. It can be easily checked that this is a feasible flow in N with probability one. 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 $\frac{1}{4} \cdot \ell/q \le \frac{1}{4P}$. This completes the proof of the redistribution lemma. □

The k-max–min min-cut problem When $\lambda = 1$ and $\Phi^* = \emptyset$, the proof of Theorem 5.2 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.

Bad example for directed graphs We note here that our results for k-robust min-cut do not hold for directed graphs. Consider the digraph G with a root r, a "center" vertex u, and m terminals $v_1, v_2, \ldots, v_m$. This graph has arcs (u, r), $\{(r, v_i)\}_{i\in[m]}$ and $\{(v_i, u)\}_{i\in[m]}$; each having unit capacity. Note that the min-cut between every v_i–v_j pair is one, but if we set each $\epsilon_i = 1/\sqrt{m}$ flow, there is no way to choose $\sqrt{m}$ of these vertices and collect a total of $\Omega(\sqrt{m})$ flow at these "leaders". This shows that the redistribution lemma (Lemma 5.5) is false for digraphs.

A similar example shows that that thresholded algorithms perform poorly for k-robust directed min-cut, even for $k = 1$. Consider graph D with vertices r, u and $\{v_i\}_{i\in[m]}$ as above. Graph D has unit capacity arcs $\{(v_i, r)\}_{i\in[\ell]}$, and $\sqrt{m}$ capacity arcs (u, r) and $\{(v_i, u)\}_{i\in[m]}$. The inflation factor is $\lambda = \sqrt{m}$. The optimal strategy is to delete the arc (u, 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 $\lambda \cdot 1 = \sqrt{m}$ resulting in a total cost of $2\sqrt{m}$. However, any threshold-based algorithm would either choose none of the terminals (resulting in a recourse cost of $\lambda\sqrt{m} = m$), or all of them (resulting in a first-stage cost of at least m).

6 *k*-robust multicut

Here we consider the k-robust multicut problem. The input is an undirected graph $G = (V, E)$ with edge-costs $c : E \to \mathbb{R}_+$, m source–sink pairs $\{s_i, t_i\}_{i=1}^{m}$, an inflation factor $\lambda \geq 1$ and bound k. Any subset of k source–sink pairs is a possible scenario. The goal is to select edges in the first and second stages so that all pairs in the realized scenario are separated, and the worst-case cost is minimized. The algorithm (given below) is essentially the same as for minimum cut, however the analysis requires different arguments.

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 [35], $\epsilon \in (0, \frac{1}{2})$ any constant, and $\beta := \rho \cdot \frac{24 \log n}{\epsilon \log\log n}$.
let $S \leftarrow \{i \in [m] \mid$ min $s_i$–$t_i$ cut in $G$ has cost at least $\beta \cdot \frac{T}{k}\}$.
output first stage solution $\Phi_T$ as the $O(\log n)$-approximate multicut [22] for $S$.
define $\mathsf{Aug}(\{i\} \mid \Phi_T)$ as edges in the min $s_i - t_i$ cut, for $i \in [m] \setminus S$; and $\mathsf{Aug}(\{i\} \mid \Phi_T) = \emptyset$ for $i \in S$.
output second stage solution where $\mathsf{Aug}(\omega \mid \Phi_T) := \bigcup_{i \in \omega} \mathsf{Aug}(\{i\} \mid \Phi_T)$ for all $\omega \subseteq [m]$.
```

Claim 6.1 (Property A for multicut) *For all $T \geq 0$ and $\omega \subseteq [m]$, the edges $\Phi_T \bigcup \mathsf{Aug}(\omega \mid \Phi_T)$ separate s_i and t_i for all $i \in \omega$; additionally if $|\omega| \leq k$ then the cost $c(\mathsf{Aug}(\omega \mid \Phi_T)) \leq \beta T$.*

Proof Pairs in $\omega \cap S$ are separated by Φ_T. By definition of $\mathsf{Aug}(, \mid \Phi_T)$ for each pair $i \in \omega \backslash S$, the edges $\mathsf{Aug}(\{i\} \mid \Phi_T)$ 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 $\mathsf{Aug}(\{i\} \mid \Phi_T)$ is at most $\beta T/k$ for all $i \in [m]$. □

Theorem 6.2 (Property B for multicut) *Let Φ^* denote the optimal first stage solution and T^* the optimal second stage cost. If $T \geq T^*$ then $c(\Phi_T) \leq O(\log n) \cdot c(\Phi^*) + O(\log^{2+\epsilon} n) \cdot T^*$.*

To prove the theorem, the high level approach is similar to that for k-robust min-cut. We first show in Lemma 6.3 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 at cost $O(1) \cdot c(\Phi^*)$. This is based on a careful charging argument on the Gomory–Hu tree and generalizes Claim 5.3 from min-cut to multicut. Then in Lemma 6.6 we show that the remaining pairs $S \backslash \widetilde{S}$ can be *fractionally* separated at cost $O(\log^{1+\epsilon} n)\, T^*$. This uses the dual-rounding approach combined with Räcke's oblivious routing scheme [35]. Finally, since the algorithm for multicut [22] is relative to the LP, this implies Theorem 6.2.

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

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

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

Lemma 6.3 *If $T \geq T^*$, there is a multicut separating pairs $\widetilde{S}$ in graph H which has cost at most* $2\,c(\Phi^*)$.

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* [38] $P = (\mathcal{N}(P), E(P))$ 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 $e \in E(P)$, if V_e and $V \backslash V_e$ are the vertex-sets on either side of e in the tree P then the weight of edge e equals $c(\partial_H(V_e))$, the cost of cut $(V_e, V \backslash V_e)$.

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. We call a cluster $N \subseteq V$ *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 perform the following two modifications on the Gomory–Hu tree P_{GH}:

1. While there is an edge of weight greater than $\frac{M}{4}$, merge the clusters corresponding to its end points. Let Q' denote the resulting cut-equivalent tree.
2. While there is a dead cluster, merge it with one of its neighboring clusters. Let Q denote the final cut-equivalent tree.

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 $2\,c(\Phi^*)$, which suffices to prove the lemma.

Claim 6.4 *$\mathcal{D}$ is a feasible multicut separating pairs $\widetilde{S}$ in H.*

Proof Fix any pair $i \in \widetilde{S}$. Clearly, vertices s_i and t_i are in distinct active clusters of the Gomory–Hu tree P_{GH}. By definition of $\widetilde{S}$, the minimum $s_i - t_i$ cut in H is less than $\frac{M}{4}$ and so there is some edge of weight less that $\frac{M}{4}$ on the $s_i - t_i$ path in P_{GH}. Note that in obtaining tree Q' from P_{GH}, we never contract an edge of weight less that $\frac{M}{4}$. Hence s_i and t_i lie in distinct active clusters of the tree Q'. In the second modification (from Q' to Q), we never merge two active clusters. So s_i and t_i lie in distinct active clusters of tree Q as well. Since this holds for all $i \in \widetilde{S}$, the claim follows by definition of $\mathcal{D}$. □

Claim 6.5 *The cost $c(\mathcal{D}) = \sum_{e \in \mathcal{D}} c_e \leq 2\,c(\Phi^*)$, 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 $i \in \widetilde{S}$. So 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

$$\begin{aligned}|\mathcal{N}_2(Q)|\, M &\leq \sum_{N\in\mathcal{N}_2(Q)} c(\partial_G(N)) = \sum_{N\in\mathcal{N}_2(Q)} (c(\partial_H(N)) + c(\partial_{\Phi^*}(N)))\\ &\leq \sum_{N\in\mathcal{N}_2(Q)} c(\partial_H(N)) + 2\cdot c(\Phi^*).\end{aligned} \tag{6.1}$$

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 $(U_l, V\backslash U_l)$ 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 the cut-equivalent tree Q, and their union $\partial_H(U_1) \bigcup \partial_H(U_2)$ contains the cut $\partial_H(N)$. Hence it follows that $c(\partial_H(N)) \leq 2\cdot\frac{M}{4} = \frac{M}{2}$. Using this in (6.1) and simplifying, we obtain $|\mathcal{N}(Q)|\, M \leq 2\cdot|\mathcal{N}_2(Q)|\, M \leq 8\, c(\Phi^*)$.

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 $Q\backslash\{e\}$. By the property of cut-equivalent tree Q, we have $c(D_e) \leq \frac{M}{4}$. Since $\mathcal{D} = \bigcup_{e\in E(Q)} D_e$,

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

This proves the claim. □

Combining Claims 6.4 and 6.5, we complete the proof of Lemma 6.3. □

Now we turn our attention to the remaining pairs $S' := S\backslash\widetilde{S}$, and show that there is a low cost multicut separating them in H. For this we use a dual-rounding argument, based on Räcke's oblivious routing scheme. Recall the parameters $0 < \epsilon < \frac{1}{2}$, $\rho = O(\log n)$ (Räcke's approximation factor), and $\beta = \rho\cdot\frac{24\log n}{\epsilon\log\log n}$. Define $\alpha := e\rho\cdot\log^{\epsilon} n$.

Lemma 6.6 *There exists a* fractional *multicut separating pairs S' in the graph H which has cost* $8\alpha\cdot T^*$.

Proof For any demand vector $d : S' \to \mathbb{R}_+$, 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 $i \in S'$), using capacity at most $\eta\cdot c_e$ on each edge $e \in H$. Note that for every $i \in S'$, 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 $i \in S'$, 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 [35] as applied to graph H. This routing scheme, for each $i \in S'$, prescribes a unit flow $\mathcal{F}_i$ between s_i–t_i such that for every demand vector $d : S' \to \mathbb{R}_+$,

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

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 with source–sink pairs S'; suppose that it sends $y_i \cdot \frac{2T^*}{k}$ units between s_i and t_i, for each $i \in S'$. For a contradiction, suppose that $\sum_{i\in S'} y_i > 4\alpha \cdot k$. (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 source–sink pairs, we may assume, without loss of generality, that $y_i \in [0, 1]$ for all $i \in S'$; this modification does not change the k-robust multicut instance. Define a (not necessarily feasible) multicommodity flow $\mathcal{G} := \sum_{i\in S'} X_i \cdot \frac{2T^*}{k} \cdot \mathcal{F}_i$, 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 oblivious routing template between s_i and t_i. The magnitude of flow $\mathcal{G}$ is the sum of $\{0, \frac{2T^*}{k}\}$-valued random variables, with expectation at least $\sum_i \frac{y_i}{\alpha} \frac{2T^*}{k} \geq 8T^*$. So by a Chernoff bound,

Claim 6.7 *With constant probability, the magnitude of flow $\mathcal{G}$ is at least* $2 \cdot T^*$.

Claim 6.8 *The flow $\mathcal{G}$ is feasible with probability* $1 - o(1)$.

Proof Fix any edge $e \in H$, and let $u_i(e) := \frac{2T^*}{k} \cdot \mathcal{F}_i(e)$ for all $i \in S'$. Note that the random choice of $\mathcal{G}$ above gives us a flow of $\sum_i X_i \cdot u_i(e)$ on the edge e. The feasibility of the maximum multicommodity flow implies that $\mathsf{Cong}(\{y_i \cdot \frac{2T^*}{k}\}_{i\in S'}) \leq 1$. Since oblivious routing loses only a ρ factor in the congestion,

$$\sum_{i\in S'} y_i \cdot u_i(e) = \sum_{i\in S'} y_i \cdot \frac{2T^*}{k} \cdot \mathcal{F}_i(e) \leq \rho \cdot c_e;$$

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

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

$$u_i(e) = \frac{2T^*}{k} \cdot \mathcal{F}_i(e) \leq \frac{2T^*}{k} \cdot \frac{\rho}{L} \cdot c_e = \frac{8\rho}{\beta} \cdot c_e$$

We divide the individual contributions by the edge capacity and further scale up by $\frac{\beta}{8\rho}$ by defining new $[0, 1]$-random variables $Y_i = \frac{X_i \cdot u_i(e)}{c_e} \cdot \frac{\beta}{8\rho}$. We get that $\mu := E[\sum Y_i] \leq \frac{\beta}{8\alpha}$. By a Chernoff bound for the sum of independent $[0, 1]$-valued random variables Y_i,

$$\Pr\left[\sum Y_i \geq (1+\delta)\cdot\mu\right] \leq \left(\frac{e}{1+\delta}\right)^{\mu(1+\delta)}, \qquad \forall \delta \geq 0.$$

Using this with $\mu(1+\delta) = \frac{\beta}{8\rho}$ (hence $\delta + 1 \geq \frac{\alpha}{\rho}$) we get that

$$\Pr\left[\sum_i X_i \cdot u_i(e) \geq c_e\right] = \Pr\left[\sum_i Y_i \geq \frac{\beta}{8\rho}\right] \leq \left(\frac{e\rho}{\alpha}\right)^{\beta/8\rho}$$
$$= \exp\left(-\epsilon \log\log n \cdot \frac{3\log n}{\epsilon \log\log n}\right) = \frac{1}{n^3},$$

since $\alpha = e\rho \cdot \log^{\epsilon} n$ and $\beta = \rho \cdot \frac{24\log n}{\epsilon \log\log n}$. Now a trivial union bound over all n^2 edges gives the claim. □

Using Claims 6.7 and 6.8 and union bound, it follows that there exists a feasible multicommodity flow $\mathcal{G}$ that sends either zero or $\frac{2T^*}{k}$ units for each pair $i \in S'$ so that the total flow in $\mathcal{G}$ is at least $2 \cdot T^*$. Hence there exists some k-set $\omega \subseteq S'$ such that the maximum multicommodity flow for ω on H is at least $2 \cdot T^*$. This contradicts the fact that every k-set has a multicut of cost at most T^* in H. Thus we must have $\sum_{i \in S'} y_i \leq 4\alpha \cdot k$, which implies Lemma 6.6. □

Combining Lemmas 6.3 and 6.6, we obtain a *fractional* multicut for pairs $S = \widetilde{S} \cup S'$ in graph G, having cost $O(1) \cdot c(\Phi^*) + O(\log^{1+\epsilon} n) \cdot T^*$. Since the Garg et al. [22] algorithm for multicut achieves an $O(\log n)$-approximation relative to the natural LP relaxation, we obtain Theorem 6.2.

From Claim 6.1 and Theorem 6.2, 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 algorithm 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 choosing an entire multicut in the second stage (using the approximation algorithm [22]); this implies an $O(\log^{1+2\epsilon} n)$-approximation algorithm. Since $\log^{1+2\epsilon} n = o\left(\frac{\log^2 n}{\log\log n}\right)$, we obtain:

Theorem 6.9 *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 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 [35]. Lemma 6.6 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 an identical analysis applies). To establish the property in Definition 2.3, consider the case $\lambda = 1$ (i.e. $\Phi^* = \emptyset$) and $c(\Phi_T) > (c_2 \log^2 n) \cdot T$. Since the [22] algorithm is $O(\log n)$-approximate relative to the LP, this implies a feasible multicommodity flow on pairs S' (since $\Phi^* = \emptyset$ we also have $S' = S$) 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 $\omega \subseteq S'$ and a feasible multicommodity flow on ω of value more than T; by weak duality it follows that the minimum multicut on ω is greater than 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. This algorithm can also be derandomized using conditional expectation and pessimistic estimators [36].

All-or-nothing multicommodity flow As a possible use of the oblivious routing and randomized-rounding based approach, we state a result for the all-or-nothing multicommodity flow problem studied by Chekuri et al. [13]. In this problem, we are given a capacitated undirected graph and source–sink pairs $\{s_i, t_i\}$ with demands d_i, and the goal is to route the maximum number of source–sink pairs (if the ith pair is chosen then all d_i units of flow must be sent between s_i and t_i). The results of this section imply that if the min-cut$(s_i, t_i) = \Omega(\log^2 n)d_i$ for all pairs, then we can approximate the maximum throughput to within an $O(\log n)$ factor without violating the edge-capacities, even when $d_{\max} \geq c_{\min}$. The results of Chekuri et al. [13,14] violated the edge-capacities in this case by an additive $d_{\max}$ term. 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.

Remark A possible approach for the k-max–min and k-robust Multicut problems, is to directly use Räcke decomposition [29,35]. For k-max–min Multicut, we can indeed use *deterministic* Räcke decomposition [29] that reduces to trees losing a factor of $O(\log^2 n \cdot \log \log n)$; the problem on trees is simply an instance of k-max–min set cover, for which we can use our $O(\log n)$-approximation algorithm (Sect. 4); altogether this implies an $O(\log^3 n \cdot \log \log n)$-approximation algorithm for k-max–min Multicut.

- However, this reduction to trees is not valid in case of *k-robust Multicut*, since the optimal first-stage solution in the graph may *not* correspond to a "multicut" (i.e. minimal edge-set separating some vertex-pairs), and hence it is not approximately preserved in the Räcke tree.
- Moreover, even for k-max–min Multicut, the randomized Räcke decomposition [35] does not apply directly since the final objective is maximization. This is unlike k-Multicut [25,35] where the goal was to find k pairs of *minimum* multicut value.

Our dual-rounding based approach uses randomized Räcke decomposition (in a less direct manner), and obtains an approximation algorithm for k-robust Multicut, and a better approximation ratio for k-max–min Multicut.

7 Summarizing properties from dual rounding

The proofs for k-robust set cover, minimum cut and multicut relied on certain dual rounding arguments. We now summarize the resulting net-type properties in a self-contained form.

Theorem 7.1 *Consider any instance of set cover; let $B \in \mathbb{R}_+$ and $k \in \mathbb{Z}_+$ be values such that*

- *every set costs at least $36 \ln m \cdot \frac{B}{k}$, and*
- *the minimum cost of covering any k-subset of elements is at most B.*

Then the minimum cost of covering **all** *elements is at most $O(\log n) \cdot B$.*

Theorem 7.2 *Consider any instance of minimum cut in an undirected graph with root r and terminals X; let $B \in \mathbb{R}_+$ and $k \in \mathbb{Z}_+$ be values such that*

- *the minimum cut separating r and u costs at least $10 \cdot \frac{B}{k}$, for each terminal $u \in X$.*
- *the minimum cut separating r and S costs at most B, for every k-set $S \in \binom{X}{k}$.*

Then the minimum cut separating r and all terminals X costs at most $O(1) \cdot B$.

Theorem 7.3 *Consider any instance of multicut in an n-vertex undirected graph with source–sink pairs $\{s_i, t_i\}_{i \in [m]}$; let $B \in \mathbb{R}_+$ and $k \in \mathbb{Z}_+$ be values such that*

- *the minimum $s_i - t_i$ cut costs at least $c \cdot \log^2 n \cdot \frac{B}{k}$, for each pair $i \in [m]$.*
- *the minimum multicut separating pairs in P costs at most B, for every k-set $P \in \binom{[m]}{k}$.*

Then the minimum multicut separating all pairs $[m]$ costs at most $O(\log^2 n) \cdot B$. Here c is a universal constant that is independent of the multicut instance.

Such properties rely crucially on the specific problem structure, and do not hold for general covering problems. Consider for example, the Steiner-tree cost function on a tree metric (which is in fact submodular). Consider a tree on vertices $\{r, u\} \bigcup \{v_i\}_{i=1}^n$ with root r and terminals $\{v_i\}_{i=1}^n$. The edges are: (r, u) with cost k, and for each $i \in [n]$, edge (u, v_i) with cost one. For parameter $B = 2k$, the cost for connecting any single terminal to the root is $k + 1 > \frac{B}{2}$, whereas the cost for connecting *any* k-set of terminals is $2k = B$. If a theorem like the ones above was true for Steiner tree then the cost to connect all the n terminals would be $\tilde{O}(B)$; instead the Steiner tree cost is $n + k > \frac{n}{2k} \cdot B \gg B$. This is also the reason why the algorithms for Steiner tree (Sect. 3) and Steiner forest (next section) are slightly more involved, and their proofs rely on a primal–dual argument instead of dual rounding.

8 *k*-robust Steiner forest

In the k-robust Steiner forest problem, we have a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, a set $U \subseteq V \times V$ of potential source–sink pairs, and an inflation factor $\lambda \geq 1$. Any set in $\binom{U}{k}$ is a valid scenario in the second stage. The goal is to choose edges in both stages so that each source–sink pair in the realized scenario is connected, and the worst-case cost is minimized.

For a set of pairs $S \subseteq V \times V$, the graph G/S is obtained by identifying each pair in S together; and $d_{G/S} : V \times V \to \mathbb{R}_+$ denotes the shortest path distances in this "shrunk" graph. The algorithm (given below) is a bit more involved than the previous ones, despite a similar general structure.

The source–sink pairs in $S_r \subseteq U$ are those with a high incremental connection cost. For technical reasons, the algorithm also maintains a set S_f of "fake" pairs, which may not belong to U. The set W consists of well-separated vertices from S_r, and is used to lower bound the Steiner forest length of S_r (see Lemma 8.2). 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 ratio; it seems that proving an $O(1)$-approximation ratio for that version would imply an $O(\log n)$-competitiveness for online greedy Steiner forest.

Algorithm 6 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 $d_G(s,w) \leq \gamma \cdot T/k$ for some $w \in W$ then $S_f \leftarrow S_f \cup \{(s,w)\}$ else $W \leftarrow W \cup \{s\}$
7:     if $d_G(t,w') \leq \gamma \cdot T/k$ 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 $\Phi_T$ to be the 2-approximate Steiner forest [2,23] on pairs $S_r$ along with
   shortest-paths connecting every pair in $S_f$.
10: define $\mathsf{Aug}(\{i\} \mid \Phi_T)$ 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 where $\mathsf{Aug}(D \mid \Phi_T) := \bigcup_{i \in D} \mathsf{Aug}(\{i\} \mid \Phi_T)$ for all $D \subseteq U$.
```

Claim 8.1 (Property A for Steiner forest) *For all $T \geq 0$ and $D \in \binom{U}{k}$, the edges $\Phi_T \bigcup \mathsf{Aug}(D \mid \Phi_T)$ connect every pair in D, and have cost $c(\mathsf{Aug}(D \mid \Phi_T)) \leq \beta T$.*

Proof The first part is immediate from the definition of $\mathsf{Aug}(D \mid \Phi_T)$ 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 $c(\mathsf{Aug}(D \mid \Phi_T)) \leq \sum_{i \in D} c(\mathsf{Aug}(\{i\} \mid \Phi_T)) \leq \sum_{i \in D} d_{G/(S_r \cup S_f)}(s_i,t_i) \leq \frac{|D|}{k} \cdot \beta T$. □

Lemma 8.2 *The optimal value of the Steiner forest on pairs S_r is at least $|W| \cdot \frac{\gamma}{2} \frac{T}{k}$.*

Proof Consider the natural cut-based LP relaxation for Steiner forest on source–sink pairs S_r, and its dual which is a packing LP. 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 every ball of radius at most $\frac{\gamma}{2} \cdot \frac{T}{k}$ around a vertex in S_r separates some pair in S_r, and appears as a variable in the dual packing problem. Observe that W only contains 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 $|W|\gamma/2 \cdot T/k$. The lemma now follows by weak duality. □

The next lemma relates the sets S_r, S_f and W that are constructed by the algorithm, and plays a key role in the subsequent analysis.

Lemma 8.3 *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_2 are the pairs where both end-points are added to W, S_1 are the pairs where exactly one end-point is added to W, and S_0 are the pairs where neither end-point is added to W. It follows that $|S_r| = |S_2| + |S_1| + |S_0|$ and $|W| = 2 \cdot |S_2| + |S_1|$.

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

- When a pair $(s, t) \in S_2$ is added, vertices s, t are added to W, and edge (s, t) is added to $E(W)$.
- Suppose a pair $(s, t) \in S_1$ 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 $(s, t) \in S_0$ 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 8.4 *At any point in the algorithm, if $x, y \in W$ lie in the same connected component of the auxiliary graph 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 $(s, t) \in S_2$ 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 $(s, t) \in S_1$ 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 $(s, t) \in S_0$ 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 8.5 *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 $d_{G/(S_f \cup S_r)}(s, t) < 2\gamma \cdot \frac{T}{k} \leq \beta \cdot \frac{T}{k}$. However, this contradicts the condition which would cause (s, t) to be chosen into S_r by the algorithm. □

Now is time 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 $|W| = 2 \cdot |S_2| + |S_1|$ and $|E| = |S_2| + |S_1| + |S_0| = |S_r|$. Since H is acyclic, $|S_r| = |E| \leq |W| - 1$. Also note that $|S_f| = 2 \cdot |S_0| + |S_1| = 2 \cdot |S_r| - |W| < |S_r|$. Thus we have $|W| \geq |S_r| \geq |S_f|$ as required in the lemma. This completes the proof of Lemma 8.3. □

Theorem 8.6 (Property B for Steiner forest) *Let F^* and T^* denote the optimal first and second stage costs, respectively. If $T \geq T^*$ then $c(\Phi_T) \leq \frac{4\gamma}{\gamma-2} \cdot (F^* + T^*)$.*

Proof Let $|S_r| = \alpha k$. Using Lemma 8.3, Lemma 8.2 and the property of the optimal solution,

$$\begin{aligned} \tfrac{\gamma}{2} \cdot \alpha \cdot T \;&\leq\; |W| \cdot \frac{\gamma}{2}\frac{T}{k} \;\leq\; \mathsf{Opt}(S_r) \;\leq\; F^* + \left\lceil \frac{|S_r|}{k} \right\rceil T^* \;\leq\; F^* + T^* + \alpha \cdot T^* \\ &\leq\; F^* + T^* + \alpha\, T \end{aligned} \tag{8.1}$$

Thus $\alpha \cdot T \leq \frac{2}{\gamma-2} \cdot (F^* + T^*)$ and $\mathsf{Opt}(S_r) \leq \frac{\gamma}{\gamma-2} \cdot (F^* + T^*)$. So the 2-approximate Steiner forest on S_r has cost at most $\frac{2\gamma}{\gamma-2} \cdot (F^* + T^*)$. Note that the distance (in G) between each pair of 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 8.3). Thus the algorithm's first-stage cost is at most $\frac{2\gamma}{\gamma-2} \cdot (F^* + T^*) + \alpha\gamma \cdot T \leq \frac{4\gamma}{\gamma-2} \cdot (F^* + T^*)$. □

Theorem 8.7 *There is a 10-approximation algorithm for k-robust Steiner forest.*

Proof Using Claim 8.1 and Theorem 8.6, 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 $\max\{\frac{4\gamma}{\gamma-2}, \frac{4\gamma/\lambda}{\gamma-2} + 2\gamma\} \leq 4 + \frac{4}{1-1/\lambda}$. The trivial algorithm that just selects a 2-approximate Steiner forest in the second stage, achieves a 2λ-approximation. Taking the better of the two, the approximation ratio is $\min\{2\lambda,\ 4 + \frac{4}{1-1/\lambda}\} < 10$. □

The k-max–min Steiner forest problem 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 8.6 when $\lambda = 1$ (so $F^* = 0$) and suppose $c(\Phi_T) > \frac{4\gamma}{\gamma-2}T \geq 2\gamma\, T$. 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:

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

   The first inequality is by definition of Φ_T and since the distance between each pair in S_f is at most $\gamma \cdot \frac{T}{k}$, the second inequality is by Lemma 8.3, and the last inequality uses $|S_r| \leq k$. Since $c(\Phi_T) > 2\gamma T$, it follows that $\mathsf{Opt}(S_r) > \gamma T/2 \geq T$.
2. If $|S_r| > k$ then the number of "witnesses" $|W| \geq |S_r| > k$, by Lemma 8.3. 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 $\mathsf{Opt}(Q) > |Q| \cdot \frac{\gamma}{2}\frac{T}{k} = \frac{\gamma}{2}T \geq T$.

Thus we obtain a constant-factor approximation algorithm for k-max–min Steiner forest.

9 Final remarks

In this paper, we presented a unified approach to directly solving k-robust covering problems and k-max–min problems. The results for all problems except multicut are fairly tight (and nearly match the best-possible for the offline versions). It would be interesting to obtain an $O(\log n)$-approximation algorithm for k-robust and k-max–min multicut.

As mentioned earlier, approximating the value of any max–min problem reduces to the corresponding robust problem, for *any* uncertainty set. We show in the companion paper [27] that there is also a relation in the reverse direction—for any covering problem that admits good offline and online approximation algorithms, an algorithm for the max–min problem implies one for the robust version. This reduction can be used to give algorithms for robust covering under matroid and knapsack constrained uncertainty sets [27].

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

References

1. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton University Press, Princeton (2009)
2. Agrawal, A., Klein, P., Ravi, R.: When trees collide: an approximation algorithm for the generalized Steiner problem on networks. SIAM J. Comput. **24**(3), 445–456 (1995)
3. Agrawal, S., Ding, Y., Saberi, A., Ye, Y.: Price of correlations in stochastic optimization. Oper. Res. **60**(1), 150–162 (2012)
4. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: A general approach to online network optimization problems. ACM Trans. Algorithms **2**(4), 640–660 (2006)
5. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: The online set cover problem. SIAM J. Comput. **39**(2), 361–370 (2009)
6. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. Math. Oper. Res. **23**(4), 769–805 (1998)
7. Berman, P., Coulston, C.: On-line algorithms for Steiner tree problems. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 344–353 (1997)
8. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. SIAM Rev. **53**(3), 464–501 (2011)
9. Bertsimas, D., Sim, M.: Tractable approximations to robust conic optimization problems. Math. Program. **107**(1–2), 5–36 (2006)
10. Byrka, J., Grandoni, F., Rothvoß, T., Laura, S.: An improved LP-based approximation for Steiner tree. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 583–592 (2010)
11. Călinescu, G., Chekuri, C., Pál, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM J. Comput. **40**(6), 1740–1766 (2011)
12. Carr, B., Vempala, S.: Randomized meta-rounding. Random Struct. Algorithms **20**(3), 343–352 (2002)
13. Chekuri, C., Khanna, S., Shepherd, F.B.: The all-or-nothing multicommodity flow problem. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 156–165 (2004)
14. Chekuri, C., Khanna, S., Shepherd, F.B.: Multicommodity flow, well-linked terminals, and routing problems. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 183–192 (2005)
15. Dhamdhere, K., Goyal, V., Ravi. R., Singh, M.: How to pay, come what may: approximation algorithms for demand-robust covering problems. In: Proceedings of the Symposium on Foundations of Computer Science (FOCS), pp. 367–378 (2005)
16. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. Syst. Sci. **69**(3), 485–497 (2004)
17. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (1998)

18. Feige, U., Jain, K., Mahdian, M., Mirrokni, V.S.: Robust combinatorial optimization with exponential scenarios. In: Proceedings of the Integer Programming and Combinatorial Optimization Conference (IPCO), pp. 439–453 (2007)
19. Fisher, M.L., Nemhauser, G.L., Wolsey, L.A.: An analysis of approximations for maximizing submodular set functions II. Math. Program. Study **8**, 73–87 (1978)
20. Gandhi, R., Khuller, S., Srinivasan, A.: Approximation algorithms for partial covering problems. J. Algorithms **53**(1), 55–84 (2004)
21. Garg, N.: Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 396–402 (2005)
22. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput. **25**(2), 235–251 (1996)
23. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM J. Comput. **24**(2), 296–317 (1995)
24. Golovin, D., Goyal, V., Ravi, R.: Pay today for a rainy day: improved approximation algorithms for demand-robust min-cut and shortest path problems. In: Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS), pp. 206–217 (2006)
25. Golovin, D., Nagarajan, V., Singh, M.: Approximating the k-multicut problem. In: Proceedings of the Symposium on Discrete Algorithms (SODA), pp. 621–630 (2006)
26. Gupta, A., Hajiaghayi, Mohammad T., Nagarajan, V., Ravi, R.: Dial a Ride from k-forest. ACM Trans. Algorithms **6**(2), 1–21 (2010)
27. Gupta, A., Nagarajan, V., Ravi, R.: Robust and max–min optimization under matroid and knapsack uncertainty sets. http://arxiv.org/abs/1012.4962
28. Gupta, A., Pál, M., Ravi, R., Amitabh, S.: Sampling and cost-sharing: approximation algorithms for stochastic optimization problems. SIAM J. Comput. **40**(5), 1361–1401 (2011)
29. Harrelson, C., Hildrum, K., Rao, S.: A polynomial-time tree decomposition to minimize congestion. In: Proceedings of the Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 34–43 (2003)
30. Imase, M., Waxman, B.M.: Dynamic Steiner tree problem. SIAM J. Discret. Math. **4**(3), 369–384 (1991)
31. Immorlica, N., Karger, D., Minkoff, M., Mirrokni, V.S.: On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In: Proceedings of the Symposium on Discrete Algorithms (SODA), pp. 691–700 (2004)
32. Khandekar, R., Kortsarz, G., Mirrokni, V.S., Salavatipour, M.R.: Two-stage robust network design with exponential scenarios. Algorithmica **65**(2), 391–408 (2013)
33. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
34. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions I. Math. Program. **14**, 265–294 (1978)
35. Räcke, H.: Optimal hierarchical decompositions for congestion minimization in networks. In: Proceedings of the Symposium on Theory of Computing (STOC), pp. 255–264 (2008)
36. Raghavan, P.: Probabilistic construction of deterministic algorithms: approximating packing integer programs. J. Comput. Syst. Sci. **37**(2), 130–143 (1988)
37. Ravi, R., Sinha, A.: Approximation algorithms for stochastic optimization problems. Math. Program. **108**(1), 97–114 (2006)
38. Schrijver, A.: Combinatorial Optimization. Springer, Berlin (2003)
39. Shmoys, D.B., Swamy, C.: An approximation scheme for stochastic linear programming and its application to stochastic integer programs. J. ACM **53**(6), 978–1012 (2006)
40. Slavík, P.: Improved performance of the greedy algorithm for partial cover. Inf. Process. Lett. **64**(5), 251–254 (1997)
41. Sviridenko, M.: A note on maximizing a submodular set function subject to knapsack constraint. Oper. Res. Lett. **32**, 41–43 (2004)
42. Swamy, C.: Risk-averse stochastic optimization: probabilistically-constrained models and algorithms for black-box distributions. In: SODA, pp. 1627–1646 (2011)