

How to Pay, Come What May: Approximation Algorithms for Demand-Robust Covering Problems*

Kedar Dhamdhere[†]
Google Inc.
Mountain View, CA 94043
kedar@cs.cmu.edu

Vineet Goyal
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA 15213
vgoyal@andrew.cmu.edu

R. Ravi
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA 15213
ravi@andrew.cmu.edu

Mohit Singh
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA 15213
mohits@andrew.cmu.edu

Abstract

Robust optimization has traditionally focused on uncertainty in data and costs in optimization problems to formulate models whose solutions will be optimal in the worst-case among the various uncertain scenarios in the model. While these approaches may be thought of defining data- or cost-robust problems, we formulate a new “demand-robust” model motivated by recent work on two-stage stochastic optimization problems. We propose this in the framework of general covering problems and prove a general structural lemma about special types of first-stage solutions for such problems: there exists a first-stage solution that is a minimal feasible solution for the union of the demands for some subset of the scenarios and its objective function value is no more than twice the optimal. We then provide approximation algorithms for a variety of standard discrete covering problems in this setting, including minimum cut, minimum multi-cut, shortest paths, Steiner trees, vertex cover and un-capacitated facility location. While many of our results draw from rounding approaches recently developed for stochastic programming problems, we also show new applications of old metric rounding techniques for cut problems in this demand-robust setting.

*Supported in part by NSF grants CCF-0430751 and ITR grant CCR-0122581 (The ALADDIN project).

[†]Work was done while the author was at Computer Science Department, Carnegie Mellon University, Pittsburgh, USA.

1 Introduction

Robust optimization has roots in both Decision Theory [14, 16] as well as Mathematical Programming [4]. While minmax regret approaches were advanced in the former field as conservative decision rules, robust optimization was discussed along with stochastic programming [3] as an alternate way of tackling data uncertainty.

More recent attempts at capturing the concept of robust solutions in optimization problems include the work of Rosenblatt and Lee [21] in facility design problem, Mulvey et al. [17] in mathematical programming, and most recently, Kouvelis and Yu [12] in combinatorial optimization; here robust means “good in all or most versions”, a version being a plausible set of values for the data in the model. Even more recent work along similar lines is advocated by Bertsimas et al. [1, 2]. A recent annotated bibliography available online summarizes this line of work [18].

1.1 A new model of “demand-robustness”

In this paper, we take a different approach in our model of uncertainty. We do not address uncertainty in the form of inaccuracy in the data itself; rather we address the uncertainty in a subset of the constraints that the problem is required to satisfy. As a simple example, consider the two alternate formulations of the shortest path problem from a root node r under the data-robust and the demand-robust formulations. In the more traditional data-robust formulation, the other terminal node t to which the shortest path from r must be built is specified in advance. However,

the costs of the edges in the graph may change as stipulated either in a set of discrete scenarios, or by intervals within which each edge cost lies. The data-robust formulation models the problem of finding a path P from r to t such that over all possible settings of the data (edge-costs) among the scenarios, the maximum value of the cost of P is minimized by taking this path. In the demand-robust model we consider, the costs of edges are specified in advance. Each scenario now specifies which terminal t_k must be connected to r via the shortest path. Furthermore, in the scenario k specified by terminal t_k , all the edge costs are costlier by a specified factor σ_k . The problem is now modelled as one of choosing a few edges to buy today at the current specified (non-inflated) cost and then, for each scenario k , completing the current solution by adding more edges (at costs inflated by σ_k) to form a path from r to t_k . The objective is to minimize the maximum value of the first stage costs plus the second stage completion costs over all possible scenarios k .

1.2 Relation to Stochastic Programming

The roots of our new model have strong links to the class of two-stage stochastic programming problems with recourse, for which some approximable versions were studied in recent work [9, 5, 11, 19, 23]. These two-stage models (e.g., from [9]) have a very similar structure: costs are specified today and the demands occurring tomorrow (along with their respective inflation factors) are specified by a probability distribution. The goal is to purchase some anticipatory part of the solution in the first stage so that the expected cost of the solution over all possible scenarios is minimized. While the expected value minimization is reasonable in a repeated decision-making framework, one shortcoming of this approach is that it does not sufficiently guard against the worst case over all the possible scenarios. Our demand-robust model for such problems is a natural way to overcome this shortcoming by postulating a model that minimizes this worst-case cost.

1.3 Model and Notation

We define an abstract covering problem in the demand-robust two-stage model with finite number of scenarios. Let U be the universe of *clients* (or demand requirements), and let X be the set of *elements* that we can purchase. Every scenario is a subset of clients and is explicitly specified. Let $S_1, S_2, \dots, S_m \subset U$ be all the scenarios. For every scenario S_k , let $sol(S_k)$ denote the sets in 2^X which are feasible to cover S_k : the covering formulation require that $A \subseteq B$ and $A \in sol(S_k) \Rightarrow B \in sol(S_k)$. The cost of an element $x \in X$ in the first stage is $c_0(x)$. In the k^{th} scenario, it becomes costlier by a factor σ_k i.e. $c_k(x) = \sigma_k c_0(x)$. In the second stage, one of the scenar-

ios is realized i.e. one of the subsets S_i materializes and the corresponding requirements need to be satisfied. Now, a feasible solution specifies the elements X_0 to be bought in the first stage, and for each k , a set of elements X_k to be bought in the recourse stage if scenario k is realized, such that $X_0 \cup X_k$ contains a feasible solution for client set S_k . The cost of covering scenario k is $c_0(X_0) + c_k(X_k)$. In the demand-robust two-stage problem, the objective is to minimize the maximum cost over all scenarios. Note that we pay for all the elements in X_0 even though some of them may not be required in the solution for any one fixed scenario.

As an example, the demand-robust “rooted” min-cut problem has $X =$ the edge set of an undirected graph, a specified root and each S_k specified by a terminal t_k . $sol(S_k)$ is the set of all edge sets that separate t_k from r . As another example, in the demand-robust “rooted” Steiner tree problem, we have $X =$ the edges of an undirected graph, a specified root r and each scenario S_k specified by a set of terminals $\{t_1^k, t_2^k, \dots\}$. $sol(S_k)$ is the set of all edge sets that connect all terminals $\{t_1^k, t_2^k, \dots\}$ to the root r .

1.4 Results

In this paper, we formulate demand-robust versions of commonly studied covering problems in optimization including minimum cut, minimum multi-cut, shortest paths, Steiner trees, vertex cover and un-capacitated facility location, and provide approximation algorithms for these problems. Our results are summarized in Figure 1. While many of our results draw from rounding approaches recently developed for stochastic programming problems, we also show new applications of old metric rounding techniques for cut problems in this demand-robust setting.

1.5 Contributions

One of the main contributions of this paper is to frame the demand-robust problems and show how this leads to interesting versions of well-studied problems in combinatorial optimization. In Section 2, we show how a natural LP formulation of the demand-robust version of the minimum-cut problem can be rounded within a logarithmic factor using ideas for rounding multi-cut problems [8, 13]. In Section 3, we also show how a demand-robust version of the multi-cut problem can also be approximated using further ideas by taking care of a constant fraction of the demands per scenario in each iteration of an iterative method (also used in [13] for the feedback arc set problem). One of the unanticipated outcomes of these new algorithms for the demand-robust versions of the cut problems is that we get the same guarantees for the two-stage stochastic versions of these problems, thus giving first poly-logarithmic approximations for them as well (See Section 3.2).

Problem	Deterministic Approximation	Stochastic Approximation	Demand-robust Approximation
Steiner Tree	1.55 [20]	3.55 [9], 30 [10]	30*
Vertex Cover	2 (Primal-dual)	2 [19]	4*
Facility Location	1.52 [15]	5 [19], 3.04 [23]	5*
Min Cut	1	$O(\log m)^*$	$O(\log m)^*$
Min Multi-Cut	$O(\log r)$ [8]	$O(\log rm \log \log rm)^*$	$O(\log rm \log \log rm)^*$

Figure 1. Result Summary . * denotes results in this paper. In the table, m, n and r denote the number of scenarios, number of nodes and maximum number of pairs per scenario respectively.

In Section 4, we prove a simple structural lemma about special classes of first-stage solutions to robust covering problems: Informally, this states that there is a first-stage solution that is a minimal feasible solution for the union of demands for a subset of the scenarios in the specification of the problem whose total cost is no more than twice that of the optimal. This result holds for a large class of covering problems including vertex cover, minimum (multi)cut, Steiner trees and facility location. However, in that section we mainly apply it to the robust Steiner tree problem to formulate a more structured LP relaxation which is the starting point for applying the methods in [10], finally giving us the constant-factor approximation result for robust Steiner trees.

In Section 5, we point out how techniques previously developed for two-stage stochastic problems that work by charging the first-stage and second-stage parts of the solution independently to the corresponding lower bounds in the relaxation to arrive at the final performance guarantee, can be used to derive analogous results for the robust versions of such problems. This remark applies to all covering problems addressed by Shmoys and Swamy [23] such as vertex cover and the rounding methods of Ravi and Sinha [19] for facility location.

Our paper opens up the investigation on a new class of robust models, and leaves much to be done in this area.

2 Robust Min-cut Problem

Problem Definition We are given an undirected graph $G = (V, E)$ with a root r . The k^{th} scenario consists of a single terminal t_k . Edge costs $c_0(e)$ in the first stage and $\sigma_k c_0(e)$ in the recourse stage if the k^{th} scenario is realized. Here σ_k is the inflation factor for the k^{th} scenario.

The objective is to find a set of edges E_0 to be bought in the first stage and for each k , a set E_k to be bought in the recourse stage if scenario k is realized, such that removing $E_0 \cup E_k$ from the graph G disconnects r from the terminal t_k . The objective is to minimize the maximum cost over all scenarios. The complexity of robust min-cut (as formulated above) is still open.

Integer Program Formulation We formulate an integer linear program for the problem as follows.

$$\begin{aligned}
 \min \quad & z \\
 z \geq \quad & \sum_e c_0(e)(x_e^0 + \sigma_k x_e^k) && \forall k \\
 (x^0 + x^k)(P) \geq \quad & 1 && \forall r-t_k \text{ path } P, \forall k \\
 x_e^k \in \quad & \{0, 1\} && \forall e, k
 \end{aligned}$$

Relaxing the integrality constraints to $x_e^k \geq 0$ gives us the linear programming relaxation. While the LP formulation given here has an exponential number of constraints, it can be solved efficiently by the ellipsoid algorithm where the separation oracle is just a shortest path computation.

2.1 Algorithm

We start by solving the LP relaxation. Let \tilde{x}_e^0 and \tilde{x}_e^k denote the values of the variables in the fractional optimal solution. Let LP_{opt} denote the optimum value of the LP relaxation. To round the fractional LP solution, we use the region growing technique of Garg et al. [8]. We would like to stress that the notion of *volume* used here is different from the LP volume used in [8]. Moreover, in our problem the LP gives a different metric on the graph for each scenario.

We start by making a copy of the graph $G = (V, E)$ for the first stage and all the second stage scenarios. We denote the copies by G_0, G_1, \dots, G_m . Edge e costs $c_0(e)$ in the graph G_0 and $\sigma_k c_0(e)$ in the graph G_k . First we give some notation to use in our algorithm description. Let $dist_k$ be the shortest path metric defined by the following lengths on the edges: $l_k(e) = \tilde{x}_e^0 + \tilde{x}_e^k, \forall e \in E$. Let $B_k(t_k, \rho)$ denote a ball of radius ρ around the terminal t_k in the metric $dist_k$. We define the volume $V_k(t_k, \rho)$ of the ball as

$$\begin{aligned}
 V_k(t_k, \rho) := \quad & \frac{LP_{opt}}{m} + \sum_{e \in B_k(t_k, \rho)} c_0(e)(\tilde{x}_e^0 + \tilde{x}_e^k) \\
 & + \sum_{e \in \delta(B_k(t_k, \rho))} c_0(e)(\rho - dist_k(t_k, e))
 \end{aligned}$$

Here $dist_k(t_k, e)$ denotes the metric distance between t_k and the closer endpoint of edge e . Note that the volume $V_k(t_k, \rho)$, for any ρ , is not same as the LP volume.

Algorithm Robust-Min-cut

1. Let G_0, G_1, \dots, G_m be copies of $G = (V, E)$. Initialize $E_0, E_1, \dots, E_k \leftarrow \phi$.
2. Repeat the following:
 - (a) Find a terminal t_k that is connected to r in the graph $G_k \cap G_0 = (V, E \setminus (E_0 \cup E_k))$.
 - (b) Find a radius $\rho < 1/2$ for which $V_k(t_k, \rho)/C_k(t_k, \rho)$ is minimum.
 - (c) If $V_k^0(t_k, \rho) \geq \frac{1}{2}V_k(t_k, \rho)$,
 - $E_0 \leftarrow E_0 \cup \delta(B_k(t_k, \rho))$
 - Remove $\delta(B_k(t_k, \rho))$ from G_0 .
 - Else $V_k^1(t_k, \rho) > \frac{1}{2}V_k(t_k, \rho)$,
 - $E_k \leftarrow E_k \cup \delta(B_k(t_k, \rho))$
 - Remove $\delta(B_k(t_k, \rho))$ from G_k .

Until all the terminals are separated from r .

However, it is bounded above by LP_{opt} , which facilitates Claim 2.1. We split the volume among first and recourse stage contributions as being the part of the volume contributed by first-stage and second-stage variables respectively.

$$V_k^0(t_k, \rho) = \frac{\text{LP}_{\text{opt}}}{m} + \sum_{e \in B_k(t_k, \rho)} c_0(e) \tilde{x}_e^0 + \sum_{e \in \delta(B_k(t_k, \rho))} c_0(e) (\min\{\rho - \text{dist}_k(t_k, e), \tilde{x}_e^0\})$$

and

$$V_k^1(t_k, \rho) = \sum_{e \in B_k(t_k, \rho)} c_0(e) \tilde{x}_e^k + \sum_{e \in \delta(B_k(t_k, \rho))} c_0(e) (\max\{0, \rho - \text{dist}_k(t_k, e) - \tilde{x}_e^k\})$$

Observe that $V_k^0(t_k, \rho) + V_k^1(t_k, \rho) = V_k(t_k, \rho)$. We define the cost of the edges crossing the boundary of the ball as $C_k(t_k, \rho) := \sum_{e \in \delta(B_k(t_k, \rho))} c_0(e)$.

Claim 2.1 *The analysis technique of Garg et. al [8] can be used to show that there exists a radius $\rho < 1/2$ such that the following holds in the step 2b of the above algorithm:*

$$C_k(t_k, \rho) \leq 2 \log m \cdot V_k(t_k, \rho). \quad (2.1)$$

We will show that the total cost paid in any scenario is at most $4 \log m \cdot \text{LP}_{\text{opt}}$. We argue about the cost of the first stage solution and the cost in the recourse stage respectively in the next two lemmas.

Lemma 2.2 *Cost of the edges E_0 is at most $4 \log m \cdot (\text{LP}_{\text{opt}} + \sum_e c_0(e) \tilde{x}_e^0)$.*

Proof: In the algorithm, we include the edges $\delta(B_k(t_k, \rho))$ in E_0 when $2V_k^0(t_k, \rho) \geq V_k(t_k, \rho)$. Therefore, the cost of the edges of $\delta(B_k(t_k, \rho))$ is bounded above by $4 \log m \cdot V_k^0(t_k, \rho)$. In other words, each unit of volume inside $B_k(t_k, \rho)$ gets a charge of $4 \log m$. Since we remove the ball $B_k(t_k, \rho)$ from graph G_0 , each edge in G_0 is charged at most once. Therefore the total cost of edges in E_0 is bounded by

$$c(E_0) \leq 4 \log m \sum_k V_k^0(t_k, \rho) \leq 4 \log m (\text{LP}_{\text{opt}} + \sum_e c_0(e) \tilde{x}_e^0). \quad \blacksquare$$

Lemma 2.3 *Cost of the edges E_k is at most $4 \log m \cdot \sum_e \sigma_k c_0(e) \tilde{x}_e^k$.*

Proof: Note that the only time we include edges in E_k is when $V_k^1(t_k, \rho) > \frac{1}{2} V_k(t_k, \rho)$. Buying edge e in G_k costs σ_k times higher. Therefore the costs of the edges in E_k can be bounded as follows:

$$c(E_k) \leq \sigma_k C_k(t_k, \rho) \leq 4 \log m \cdot \sigma_k V_k^1(t_k, \rho) \leq 4 \log m \sum_e \sigma_k c_0(e) \tilde{x}_e^k. \quad \blacksquare$$

Theorem 2.4 *The Algorithm Robust-Min-Cut produces an $O(\log m)$ -approximate solution to the robust min-cut problem.*

Proof: Using Lemmas 2.2 and 2.3 the total cost of any scenario k can be bounded as follows:

$$c(E_0) + c(E_k) \leq 4 \log m \left(\text{LP}_{\text{opt}} + \sum_e c_0(e) (\tilde{x}_e^0 + \sigma_k \tilde{x}_e^k) \right) \leq 8 \log m \cdot \text{LP}_{\text{opt}}$$

Therefore the maximum cost over all scenarios is $O(\log m) \text{LP}_{\text{opt}}$ as well. \blacksquare

2.2 Robust Min-cut in Trees

In the special case when the input graph G is a tree, we give a polynomial time exact algorithm for the robust min-cut problem. The algorithm uses the following fact crucially: if a terminal t_k is not separated from the root r by the first stage solution, then we need to buy only one edge in the k^{th} scenario in the recourse stage.

Theorem 2.5 *There is a polynomial-time exact algorithm for the robust min-cut problem on a tree.*

Proof: The algorithm for robust min-cut on trees is as follows. “Guess” c to be the maximum second-stage cost of an edge to be cut in recourse stage. Since for each terminal, we need to remove only a single edge to separate it from the root, there are m choices for this maximum cost (m is the number of scenarios). All terminals t_k , that have first-stage min-cut cost less than $\frac{c}{\sigma_k}$ are cut in the recourse stage. The rest of the terminals are separated from the root by a minimum cost cut in the first stage.

One of the guesses of c is the correct one, for which we will find a solution that pays at most c in the recourse stage. Furthermore, the first stage min-cut cost for every terminal t_k that is cut in the first stage by this solution is greater than $\frac{c}{\sigma_k}$. Thus, any optimal solution separates t_k from the root in the first stage. Hence, the algorithm returns an optimal solution. ■

3 Robust Multi-cut

The robust multi-cut problem is a generalization of the robust min-cut problem. The problem is defined on a graph $G = (V, E)$. Here the k^{th} scenario consists of pairs of terminals $\{(s_1^k, t_1^k), (s_2^k, t_2^k), \dots\}$. We want to find a set of edges E_0 to buy in the first stage and E_k to buy in the recourse stage if scenario k is materialized such that $E^0 \cup E^k$ separates each of the pairs $\{(s_1^k, t_1^k), (s_2^k, t_2^k), \dots\}$. Edge e costs $c_0(e)$ in the first stage and $\sigma_k c_0(k)$ in the scenario k of the recourse stage. The objective is to minimize the maximum cost over all scenarios.

We first describe an $O(\log^2 rm)$ algorithm for robust multi-cut problem, where r is the maximum number of pairs in any scenario. The algorithm is similar to the one for robust min-cut.

We formulate an integer linear program for the robust multi-cut problem as follows.

$$\begin{aligned} \min \quad & z \\ z \geq \quad & \sum_e (c_0(e)x_e^0 + \sigma_k c_0(e)x_e^k) \quad \forall k \\ (x^0 + x^k)(P) \geq \quad & 1 \quad \forall s_i^k - t_i^k \text{ paths } P, \forall k, i \\ x_e^k \in \quad & \{0, 1\} \quad \forall e, k \end{aligned}$$

Relaxing the integrality constraints to $x_e^k \geq 0$ gives us the LP relaxation. Let \tilde{x}_e^0 and \tilde{x}_e^k denote the optimal fractional solution. The rounding procedure is similar to the rounding procedure for robust min-cut. As before, we maintain m graphs G_1, G_2, \dots, G_m , one for each scenario. We also maintain G_0 for the first stage solution. Each of these graphs are initialized as copies of $G = (V, E)$. However, we need to modify the ball growing procedure. In robust min-cut problem, when a boundary of a ball $B_k(t_k, \rho)$ is removed from the graph G_0 , there are no terminal pairs left inside the ball. This property no longer holds for the robust multi-cut problem. Therefore, we recursively apply the algorithm inside each component of the graph formed after

removing the boundary. We give a sketch of the algorithm here.

We find disjoint balls $B_k(s_i^k, \rho)$ and $B_k(t_i^k, \rho')$ around s_i^k and t_i^k respectively. The radii $\rho, \rho' \leq 1/4$ are chosen such that the cost of the edges crossing the boundary of a ball is within $O(\log rm)$ factor of the volume inside the ball. If $V_k^1(s_i^k, \rho) \geq \frac{1}{2}V_k(s_i^k, \rho)$ then we include $\delta(B_k(s_i^k, \rho))$ in the edge set E_k and remove $\delta(B_k(s_i^k, \rho))$ from the graph G_k . Else if $V_k^1(t_i^k, \rho') \geq \frac{1}{2}V_k(t_i^k, \rho')$ include $\delta(B_k(t_i^k, \rho'))$ in E_k and remove $\delta(B_k(t_i^k, \rho'))$ from the graph G_k .

Otherwise, consider the number of unseparated terminal pairs (from all scenarios) in $B_k(s_i^k, \rho)$ and $B_k(t_i^k, \rho')$. Assume without loss of generality, $B_k(s_i^k, \rho)$ is the ball with smaller number of unseparated terminal pairs. We include the edges $\delta(B_k(s_i^k, \rho))$ in E_0 and remove $\delta(B_k(s_i^k, \rho))$ from G_0 . We run the algorithm recursively inside each of the components formed.

This algorithm is similar to the divide-and-conquer algorithm for Feedback Arc Set problem due to Leighton and Rao [13]. It divides the graph G_0 in various components and recurses inside each component. In order to bound the approximation factor of the algorithm, we need to prove that the depth of the recursion tree is small and the algorithm pays only a small cost at each level of the recursion.

Lemma 3.1 *Depth of the recursion of the above algorithm is bounded by $\log(rm)$.*

Proof: Each time our algorithm makes a recursive call, the number of terminal pairs inside the ball is at most half as many as the total number of terminal pairs in all scenarios. Since the total number of terminal pairs we started with is bounded by rm , the recursion depth is at most $\log_2 rm$. ■

Using an argument similar to that of Lemma 2.2 we can bound the cost of the algorithm paid for edges in G_0 as follows.

Lemma 3.2 *In each level of recursion, each unit of volume in the graph G_0 gets a charge of $O(\log rm)$.*

Theorem 3.3 *There is a polynomial-time $O(\log^2 rm)$ -approximation algorithm for the robust multi-cut problem.*

Proof: Each unit of volume in the graph G_k is charged at most once and receives a charge of $O(\log rm)$. On the other hand, each unit of volume in the graph G_0 gets a charge of $O(\log rm)$ for $O(\log rm)$ levels of recursion. Therefore the total cost paid by the algorithm for edges in G_0 is $O(\log^2 rm \cdot OPT)$, where OPT is the optimum value of the LP relaxation. Hence, the total cost paid in any scenario is $O(\log^2 rm \cdot OPT) + O(\log rm \cdot OPT) = O(\log^2 rm \cdot OPT)$. ■

3.1 Improved approximation

We now show how to improve the approximation factor to $O(\log rm \log \log rm)$ using the ideas from [6, 7, 22]. We modify our divide-and-conquer algorithm as follows. For a terminal s_i^k , we find a ball $B_k(s_i^k, \rho)$ such that $C_k(s_i^k, \rho) \leq V_k(s_i^k, \rho) \cdot 4 \log(V_k/V_k(s_i^k, \rho)) \log \log V_k$, where $V_k = \sum_e c_e^0(x_e^0 + x_e^k)$ is the total volume. The analysis technique from [6] shows that such a radius ρ exists.

To bound the total cost of the algorithm, we note that each unit of volume in the recourse stage graph G_k gets a charge of $O(\log rm \log \log rm)$ at most once. On the other hand each unit of volume in graph G_0 gets charged multiple times. Given a graph G and solution to robust multi-cut problem (x^0, x^1, \dots, x^m) , let the total first stage volume of graph G , $V_0 = \sum_e c_e^0 \cdot x_e^0$. Also, let $\text{cost}(V_0)$ denote the cost of the first stage solution constructed by the above recursive solution with total first stage volume V_0 . We bound the cost paid using the following recurrence relation:

$$\text{cost}(V_0) \leq \text{cost}(V_k^0(s_i^k, \rho)) + \text{cost}(V_0 - V_k^0(s_i^k, \rho)) + 4 \log(V_k/V_k(s_i^k, \rho)) \log \log V_k \cdot V_k^0(s_i^k, \rho).$$

Solving this recurrence, we get that the cost paid for the first stage edges is $O(V_0 \cdot \log rm \log \log rm)$. Hence the total cost paid by the algorithm is bounded by $O(\log rm \log \log rm) \cdot \text{LP}_{\text{opt}}$ which proves the following theorem.

Theorem 3.4 *There exists a $O(\log rm \log \log rm)$ -approximation to the robust multi-cut problem.*

3.2 Stochastic Min-Cut and Multi-Cut

The stochastic min-cut problem is defined as follows: We are given a graph $G = (V, E)$ with a cost function c_0 on the edges and a root node r . We are also given a collection S_1, \dots, S_m of m scenarios with p_k being the probability of occurrence of scenario S_k . For each scenario S_k , there exists a node t_k and we demand that r and t_k must be separated if scenario S_k appears in the recourse stage. Edge e costs $c_0(e)$ in the first stage and $\sigma_k c_0(e)$ if scenario S_k appears in the recourse stage. The objective function to minimize the sum of the first-stage cost and the expected recourse stage cost. Stochastic multi-cut is similarly defined to be the stochastic counterpart of robust multi-cut problem.

We show that a simple modification to the approximation algorithms for robust min-cut and multi-cut yields approximation algorithms for the stochastic version of the problems with same performance guarantees.

The region growing argument is not directly applicable to the stochastic min-cut problem for the following reason: the ‘‘volume’’ V of a ball defined in the proof of robust min-cut is different from the cost of the LP solution in the ball unlike the algorithm of Garg et al. [8] for the deterministic

multi-cut. In the case of robust min-cut or multi-cut, the volume is bounded by cost of the LP solution. Thus, we could claim that there exists a radius $\rho \leq \frac{1}{2}$ such that the cost of the cut $C_k(t_k, \rho)$ is at most $O(\log m) \cdot V_k(t_k, \rho)$ in the robust min-cut problem. But this is not true in the case of stochastic min-cut problem. Here, we do some preprocessing before applying the region growing argument.

For all scenarios in $S := \{i \mid \sigma_i p_i \leq \frac{1}{m^2}\}$, we introduce the constraint in the LP that cut for these scenarios will be completely a recourse stage solution. We claim that this transformation does not affect the optimum solution by a large factor: in an optimum solution if we buy all the first stage edges helping scenarios in S during the recourse stage as well, the extra edges bought incur a cost of at most $\sum_{i \in S} \sigma_i p_i \cdot \text{OPT} \leq \frac{|S|}{m^2} \text{OPT} \leq \frac{\text{OPT}}{m}$. Hence, we can ignore these scenarios while constructing our first stage solution.

Now, when we apply the region growing algorithm for the rest of the scenarios. The total volume in the graph is at most $V = \sum_k \sum_e c_0(e)(x_e^0 + x_e^k)$. The cost of the LP solution is at least

$$\begin{aligned} \sum_{k,e} c_0(e)(x_e^0 + \sigma_k p_k x_e^k) &\geq \sum_{k,e} c_0(e)(x_e^0 + \frac{1}{m^2} x_e^k) \\ &\geq \frac{1}{m^2} \sum_{k,e} c_0(e)(x_e^0 + x_e^k) \end{aligned}$$

Hence, $V \leq m^2 \cdot c(LP)$. Now, we can show using the techniques of Garg et al. [8] that there exists a radius $\rho \leq \frac{1}{2}$ such that $C_k(t_k, \rho) \leq 4 \log m \cdot V_k(t_k, \rho)$. Hence, by running the same algorithm described above for the robust min-cut losing an extra factor of 2, we obtain the following theorem.

Theorem 3.5 *There exists a polynomial time algorithm which returns an $O(\log m)$ approximate solution to the stochastic min-cut problem.*

A similar transformation for the stochastic multi-cut problem will yield the following theorem.

Theorem 3.6 *There exists a polynomial time algorithm which returns an $O(\log rm \log \log rm)$ approximate solution to the stochastic multi-cut problem.*

4 Special types of first-stage solutions and Steiner trees

In this section, we prove that for any robust two-stage problem there is an approximate first stage solution with a special structure: it is a minimal feasible solution for a subset of scenarios and can be without much cost overhead extended to a complete solution in the second stage. We use this structural result to obtain a constant factor approximation for the robust Steiner tree problem.

4.1 A Structural Lemma for the First Stage Solution

Lemma 4.1 Given any problem Π in the robust two-stage model, there exists a first stage solution \tilde{X}_0 and a subset $\mathcal{S} \subseteq \{S_1, \dots, S_m\}$ of scenarios, such that \tilde{X}_0 is a minimal feasible solution for scenarios in \mathcal{S} . Furthermore, it can be extended to a solution for the remaining scenarios in the second stage and the cost of the final solution is at most $2 \cdot OPT$.

Proof: Consider an optimal integral solution to the robust problem: let X_0^* be the first stage solution and X_i^* be the recourse stage solution in scenario i . Also, let X_{0i}^* be the part of first stage solution used in scenario i i.e. it is a minimal subset of X_0^* such that $X_{0i}^* \cup X_i^*$ is a feasible solution for scenario i . We construct an alternate first stage solution \tilde{X}_0 , such that it is a union of feasible solutions for a subset of scenarios. \tilde{X}_0 will contain elements from the optimal first stage solution X_0^* , and also from the optimal recourse stage solutions X_1^*, \dots, X_m^* . Let A denote the elements of X_0^* in \tilde{X}_0 . We construct \tilde{X}_0 as follows.

1. Initialize $A \leftarrow \phi$ and $B \leftarrow \phi$.
2. For each scenario $i = 1, 2, \dots, m$, repeat the following
 - (a) $X'_{0i} = X_{0i}^* \setminus A$.
 - (b) If $c_0(X'_{0i}) \geq c_0(X_i^*)$, then $A \leftarrow A \cup X'_{0i}$ and $B \leftarrow B \cup X_i^*$.
3. $\tilde{X}_0 \leftarrow A \cup B$.

Our new first stage solution $\tilde{X}_0 = A \cup B$. Note that $A \subseteq X_0^*$. Therefore, $c_0(A) \leq c_0(X_0^*)$. Also, all elements in B are charged to disjoint parts of A . Thus, by construction $c_0(B) \leq c_0(A)$ which implies $c_0(\tilde{X}_0) \leq 2 \cdot c_0(X_0^*)$. Clearly, \tilde{X}_0 is a feasible solution for a subset of scenarios and it is minimal due to optimality of $X_0^*, X_1^*, \dots, X_m^*$ and the minimality of X_{0i}^* for each i . Furthermore, we claim that \tilde{X}_0 can be extended to a feasible solution for all scenarios in the second stage such that the cost of final solution is at most $2 \cdot OPT$.

Consider some scenario which is not covered in the first stage by \tilde{X}_0 , say i . This implies that when scenario i was considered in the above sequence, $c_0(X'_{0i}) < c_0(X_i^*)$. Thus, we can buy X'_{0i} in the recourse stage and charge it to the cost of X_i^* . Let the new recourse stage solution be $\tilde{X}_i = X_i^* \cup (X'_{0i} \setminus A)$. Hence, $c_i(\tilde{X}_i) \leq 2 \cdot c_i(X_i^*)$ as $c_i(x) = \sigma_i \cdot c_0(x)$. Thus, the final cost of the new solution is

$$\max_i \{c_0(\tilde{X}_0) + c_i(\tilde{X}_i)\} \leq \max_i 2 \cdot (c_0(X_0^*) + c_i(X_i^*)) \leq 2 \cdot OPT$$

■

The above structural result about the first-stage solution of a covering problem in the robust two-stage model also holds for the problem in the stochastic two-stage model. Starting with an integral optimum solution to the stochastic version of the problem (say $X_0^*, X_1^*, \dots, X_m^*$), the special solution can be constructed as in the procedure described above. Let the constructed solution be X_0, X_1, \dots, X_m . From the proof of Lemma 4.1, we have that $c_i(X_i) \leq 2 \cdot c_i(X_i^*)$, $i = 0, 1, \dots, m$. Thus, the stochastic objective for the new solution is,

$$\begin{aligned} c_0(X_0) + \sum_{i=1}^m p_i c_i(X_i) &= c_0(X_0) + \sum_{i=1}^m p_i \sigma_i c_0(X_i) \\ &\leq 2(c_0(X_0^*) + \sum_{i=1}^m p_i \sigma_i c_0(X_i^*)) \end{aligned}$$

Thus, the above lemma gives an alternate proof for a similar lemma in [10] that proves that there is a connected first-stage solution for the stochastic Steiner tree problem which costs at most thrice the optimal.

4.2 Robust Steiner Tree

We use the structural lemma proved above to give a constant factor approximation for the robust Steiner tree problem. The problem is defined on a graph $G = (V, E)$ with a root vertex r and a cost function c on the edges. In the second stage one of the m scenarios materializes. The k^{th} scenario consists of a set $S_k \subseteq V$ of terminals and an inflation factor σ_k . An edge e costs $c_0(e)$ in the first stage and $c_k(e) = \sigma_k c_0(e)$ in the k^{th} scenario of the second stage. A solution to the problem is a set of edges E_0 to be bought in the first stage and a set E_k in the recourse stage for each scenario k . The solution is feasible if $E_0 \cup E_k$ contains a Steiner tree connecting $S_k \cup \{r\}$. The cost paid in the k^{th} scenario is $c_0(E_0) + \sigma_k \cdot c_0(E_k)$. The objective is to minimize the maximum cost over all scenarios.

The structural lemma (Lemma 4.1) shows that there is a first stage solution which is feasible for some subset of the scenarios. For the robust Steiner tree problem, it means there is a tree solution for the first stage that can be extended to a final solution within twice the cost of the optimum solution. Therefore, we formulate the problem with the additional constraint that the first stage solution should be a tree. This means that the path from any terminal to the root consists of a portion of only recourse edges, followed by a portion consisting of only first-stage edges. The flow-based IP formulation for the robust Steiner tree problem is shown in (4.2)-(4.8).

$$\min z \quad (4.2)$$

$$\forall k, z \geq \sum_{e \in E} c_0(e) \cdot (x_e^0 + \sigma_k \cdot x_e^k) \quad (4.3)$$

$$\forall t \in S_k, \forall k, \sum_{e \in \delta_+(t)} (r_e^0(t) + r_e^k(t)) \geq 1 \quad (4.4)$$

$$\forall v \notin \{t, r\}, \forall t \in S_k, \forall k,$$

$$\sum_{e \in \delta_+(v)} r_e^0(t) + r_e^k(t) = \sum_{e \in \delta_-(v)} r_e^0(t) + r_e^k(t) \quad (4.5)$$

$$\sum_{e \in \delta_-(v)} r_e^0(t) \leq \sum_{e \in \delta_+(v)} r_e^0(t) \quad (4.6)$$

$$\forall e, \forall t \in S_k, \forall k,$$

$$r_e^k(t) \leq x_e^k \quad (4.7)$$

$$r_e^k(t), x_e^k \in \{0, 1\} \quad (4.8)$$

This formulation is similar to one used by Gupta et al. in [10], where they give a constant factor approximation for the stochastic Steiner tree problem. The x^0 variables are indicators for the edges in the first stage, and x^1, x^2, \dots, x^k are the indicators for recourse stage edges. For a terminal t in scenario k , the variable $r_e^k(t)$ indicates whether edge e is used in the recourse portion of t 's path to the root, and $r_e^0(t)$ indicates whether it is used in the first-stage portion of the path. These flow variables are directed; for $e = (u, v)$, the variable $r_{uv}^k(t)$ denotes the flow of commodity t along a recourse edge in the direction u to v . Given these directed flow variables, the cut-sets are defined as $\delta_+(S) = \{e = (u, v) : u \in S, v \notin S\}$ and $\delta_-(S) = \delta_+(V \setminus S)$. Note, however, that the edge installation variables x_e^k refer to undirected edges.

Consider the LP relaxation of the above IP formulation obtained by dropping the integrality constraints. Let z_{IP} be the cost of the optimum IP solution, \tilde{z} be the optimum LP solution and OPT be the optimum solution of the original instance. From Lemma 4.1, we know that $z_{IP} \leq 2OPT$. The fractional LP solution can be rounded using the same rounding scheme as that of Gupta et al. [10]. Thus, the following lemma can be derived from [10].

Lemma 4.1 ([10]) *Let $\tilde{z}, \tilde{x}^0, \tilde{x}^1, \dots, \tilde{x}^k$ be a fractional solution to the linear relaxation of the IP in (4.2)-(4.8). It can be rounded to obtain an integral solution T^0, T^1, \dots, T^k , such that $T^0 \cup T^i$ connects $S_i \cup \{r\}$, $\forall i$. Furthermore, $c_0(T^0) \leq 15 \cdot \sum_{e \in E} c_0(e) \cdot x_e^0$ and $\forall i, c_i(T^i) \leq 15 \cdot \sum_{e \in E} c_i(e) \cdot x_e^i$.*

Theorem 4.2 *The Robust Steiner Tree Problem can be approximated within a factor of 30 in polynomial time.*

Proof: Lemma 4.1 shows that the optimum fractional solution of the LP relaxation can be rounded to an integral solution such that cost of each scenario is increased by at most a factor of 15. Thus, $z_{IP} \leq 15 \cdot \tilde{z} \leq 30 \cdot OPT$. Hence, we obtain a 30-approximation for the Robust Steiner Tree problem. ■

5 Other Robust Optimization Problems

In this section, we consider some other combinatorial problems in the two-stage robust model and give approximation algorithms for them.

5.1 Covering Problems of Shmoys and Swamy [23]

Two-stage stochastic set covering problems were studied in a general setting by Shmoys and Swamy in [23], where they showed how a ρ -approximation algorithm for the single stage problem gives a 2ρ -approximation for the corresponding two stage stochastic version. The key idea is to observe that every element will be at least half-covered by the first- or second-stage sets that contain it. By scaling up both first- and second-stage by a factor of two, and using the rounding algorithm on both scaled solutions, one obtains a solution with the promised guarantee. A major contribution of [23] is a polynomial-time approximation scheme to solve the two-stage stochastic programs even though the underlying problem may be $\#P$ -complete.

A simple application of the above method to polynomial-sized robust problems gives a simple 2ρ -approximation algorithm for covering problems allowing a ρ -approximate single stage rounding method.

Consider the demand-robust version of minimum vertex cover: nodes have different costs in the first stage and in each of the scenarios in the second stage, while each scenario consists of a subgraph of the complete graph on the nodes. The goal is to choose some vertices in the first stage and for every scenario, augment the chosen set at the second-stage costs to form a vertex cover of the edges in this scenario. A simple corollary of the above observation along with the classical 2-approximation rounding result for regular vertex cover gives the following simple result.

Theorem 5.1 *The demand-robust vertex cover problem can be approximated within a factor of 4.*

5.2 Robust Facility Location

In this problem we are given a set of facilities F and a set of clients S_1, S_2, \dots, S_m for each scenario. A metric

c_{ij} specifies the distances between every client and every facility. Facility i has a first-stage opening cost of f_i^0 , and a recourse cost of f_i^k in scenario k . Note that in this case we can handle general second stage costs unlike the model stated earlier where the second stage costs change by certain inflation factors $\sigma_1, \sigma_2, \dots, \sigma_m$.

Our approximation algorithm proceeds along the lines of the LP-rounding algorithm due to Ravi and Sinha [19]. The algorithm in [19] rounds a fractional solution such that the cost of each scenario in the integral solution is bounded by 5 times its cost in the fractional solution. Thus, the same techniques give a 5-approximation for robust facility location.¹

Theorem 5.2 *The demand-robust facility location problem can be approximated within a factor of 5.*

6 Conclusion and Open Problems

In this paper, we introduce a new model called demand-robustness and give approximation algorithms for some combinatorial problems in this model. There seems to be an interesting parallel between stochastic and robust settings. For example, the rounding techniques for the stochastic Steiner tree problem can be adapted to the robust version of the same problem. Similarly, the rounding technique used for robust min-cut and multi-cut can be adapted to stochastic min-cut and multi-cut with a slight modification. It would be interesting to prove a general result showing that a ρ -approximation for a stochastic optimization problem leads to a $O(\rho)$ -approximation for the robust version of the problem and vice-versa.

References

- [1] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operation Research*, 52(2):35–53.
- [2] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71.
- [3] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, Berlin, 1997.
- [4] George B. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1955.
- [5] Kedar Dhamdhere, R. Ravi, and Mohit Singh. On two-stage stochastic minimum spanning trees. In *IPCO*, 2005.
- [6] Guy Even, Joseph Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. In *IEEE Symposium on Foundations of Computer Science*, pages 62–71, 1995.
- [7] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
- [8] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.
- [9] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the 36th annual ACM symposium on Theory of computing (STOC)*, pages 417–426, 2004.
- [10] Anupam Gupta, R. Ravi, and Amitabh Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.
- [11] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems.
- [12] P. Kouvelis and G. Yu. *Robust Discrete Optimisation and Its Applications*. Kluwer Academic Publishers, Netherlands, 1997.
- [13] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 422–431, 1988.
- [14] R. D. Luce and H. Raiffa. *Games and Decision*. Wiley, New York, 1957.
- [15] M. Mahdian, Y. Ye, and J. Zhang. A 1.52 approximation algorithm for the uncapacitated facility location problem. In *Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 229–242, 2002.
- [16] J. W. Milnor. Games against nature. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*. Wiley.

¹Although Ravi and Sinha[19] have claimed an 8-approximation, a more careful analysis of their algorithm gives a 5-approximation.

- [17] J. M. Mulvey, M. J. Verderbel, and S. A. Zenios. Robust optimisation of large scale systems. Technical Report SOR-91-13, Statistics and Operations Research, Princeton University, Princeton, NJ, 1994.
- [18] Yury Nikulin. Robustness in combinatorial optimization and scheduling theory: An annotated bibliography. webpage, 2004. http://www.optimization-online.org/DB_FILE/2004/11/995.pdf.
- [19] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *IPCO*, pages 101–115, 2004.
- [20] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proc. of ACM/SIAM Symposium on Discrete Algorithms (SODA)*, pages 770–779, 2000.
- [21] M. J. Rosenblatt and H. L. Lee. A robustness approach to facilities design. *International Journal of Production Research*, 25:479–486, 1987.
- [22] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15:281–288, 1995.
- [23] David Shmoys and Chaitanya Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.