Optimization Under Uncertainty

Vineet Goyal

Submitted in partial fulfillment of the requirements
of the degree of Doctor of Philosophy in Algorithms, Combinatorics and
Optimization

August 2008

Tepper School of Business
Carnegie Mellon University

Abstract

Most optimization problems in real life do not have accurate estimates of the problem parameters at the optimization phase. Stochastic optimization models have been studied widely in the literature to address this problem. The expected value optimization is reasonable in a repeated decision making framework. However, it does not sufficiently guard against the worst case future in more risk averse applications. The broad purpose of this thesis is to study optimization approaches under uncertainty that overcome this shortcoming of a traditional stochastic optimization model.

We consider new models of uncertainty namely, the "demand-robust" model and the "chance constrained" model and introduce these in the framework of general covering problems. We consider uncertainty in the right hand side of the constraints which is referred to as the demand uncertainty. In the two-stage model of "demand-robustness", we are interested in finding a solution such that the worst case cost over all realizations of uncertainty is minimized. We prove a general structural lemma about special types of first stage solutions and provide approximation algorithms for covering problems such as Steiner tree, min-cut, minimum multi-cut, vertex cover and facility location. The structural lemma essentially exploits the following idea: In a two-stage solution, if the first stage help is at least as costly as the second stage solution for some realization of the uncertain parameters (referred to as a scenario), then a solution for that scenario can be constructed completely in the first stage while only losing a factor two in the total cost. We further extend this idea to develop a 'guess-and-prune' algorithm where we 'guess' the worst case second stage cost which allows us to 'prune' away a set of scenarios for which a complete solution in the second stage has cost at most the worst case cost. For specific covering problems such as minimum cut and shortest path, we show that an approximate first stage solution can be constructed for the remaining scenarios using ideas from the structural lemma as well as the combinatorial structure of the problem.

The robust optimization approach guards against the worst case future but tends to be overly conservative if there are some outlier scenarios. To overcome this, we consider a chance constrained model where we are given a reliability level p and the idea is to select a "p fraction" of the scenarios and find a robust solution on the selected scenarios. The remaining (1-p) fraction of the scenarios are considered as outliers and can be ignored. We consider both one-stage and two-stage chance constrained covering problems with demand-uncertainty. While it is easy to obtain bi-criteria approximations for the chance-constrained problems that violate the chance-constraint by a small factor, we consider the problem of satisfying the chance-constraint strictly. We show that the covering problems in both one-stage and two-stage chance-constrained models where uncertainty is specified as an explicit list of scenarios (with more than one element in each scenario) are at least as hard to approximate as the dense k-subgraph (DkS) problem. We also consider the special case when each scenario has a single demand element and show that the chance-constrained models reduce to weighted partial covering versions either directly or via a guess and prune method.

We also consider the model of uncertainty where scenarios (possibly an exponential number) are specified implicitly by a probability distribution over the demand-elements. While it is not even clear if the two-stage problem is in NP in this implicit scenario model, we give approximations for the one-stage problem if the probability distribution satisfies certain fairly general properties.

In both the above models, we consider uncertainty in the right hand side of the constraints. We extend our work to consider uncertainty in the constraint matrix referred to as data uncertainty and study a chance constrained knapsack problem where each item has a known deterministic profit but the size is random and is distributed according to a known normal distribution independent of the other items. We obtain a polynomial time approximation scheme for this problem that selects a set of items that satisfy the chance-constraint strictly and achieve near optimal profit.

In the last chapter, we consider the planning problem for post-disaster logistics where we are required to open a set of emergency response centers such that the affected areas or the demand locations post disaster such as an earthquake can be reached from at least one of the emergency response location within a given time bound. This problem combines aspects of both demand and data uncertainty as both the demand and the underlying transport network depend on the disaster scenario and only realize after the disaster. We develop an efficient sampling based algorithm to estimate several parameters for a given set of emergency locations such as the fraction of disaster scenarios where all the demand can be covered and average fraction of demand covered across all disaster scenarios. We use the data for the case of Istanbul, Turkey to conduct the computational experiments and find that our algorithm is efficient and provides reasonably accurate estimates.

Contents

List of Figures

List of Tables

1
Introduction

In a classical optimization problem, all parameters such as costs and demands are assumed to be known deterministically. However, this assumption does not hold in most real-life applications where more often than not we only know estimates of the problem parameters. At best, we can assume that a probability distribution over parameter values is known from historical data. In such applications, classical deterministic optimization models are not useful as the optimal solutions found by such models can be very sensitive to even slight changes in the problem parameters. In this thesis, we study approaches for optimization under uncertainty both from a point of view of approximation algorithms as well as designing efficient heuristics to solve these problems exactly.

Stochastic optimization models have been widely studied in the operations research literature to address the problem of uncertain problem parameters [6, 37, 4]. In a two-stage stochastic optimization problem (one with two stages of decision making), certain decisions are required to be taken before the uncertainty materializes. These are called the *first stage decisions* and the corresponding period is called the *first stage*. After the uncertainty materializes i.e. the uncertain problem parameters become known, the first stage decisions can be augmented with more decisions to construct a feasible solution for the problem. These are called the *second stage decisions* or *recourse decisions* and the corresponding stage is referred to as the *second stage* or the *recourse stage*. A particular realization of all the uncertain parameters in the problem is referred to as a *scenario*. Typically, a decision in the first stage is less costly than the same decision in the second stage but it may be wasteful in hindsight due to the uncertainty in the problem. On the other hand, a second stage decision while being costlier is made after the uncertainty has been resolved completely. The goal is to find a set of first stage and second stage decisions such that the total expected cost is minimized. In most cases, the crucial part is selecting the set of first stage decisions. Finding the optimal second stage decisions after the uncertain parameters have been resolved is usually a straightforward problem. Multi-stage models can be considered where uncertain parameters are revealed over multiple stages, for e.g., in a multi-period inventory management problem, uncertain demand is revealed period by period.

Recently, two stage scenario based models of uncertainty have been studied widely with regard to finding approximation algorithms for combinatorial problems in [33, 49, 57, 30]. In these models, the uncertainty is specified as an explicit list of scenarios ($i = 1, \ldots, l$) with associated probabilities ($p_1, \ldots, p_l$ such that $\sum_{i=1}^{l} p_i = 1$). Scenario i realizes with probability p_i in the second stage. The second stage cost in scenario i, $c_s^i = \sigma_i \cdot c_f$, where σ_i is the inflation factor for scenario i and c_f denotes the first stage costs. Note that the cost of all elements are assumed to inflate by the same factor σ_i in scenario i in our model. The goal is to find a first stage solution (X_f) and for each scenario i, the second stage or recourse solution (X_s^i) such that $X_f \cup X_s^i$ satisfies scenario i. The objective is to minimize the total expected cost over all the scenarios of the second stage, i.e.,

$$c_f(X_f) + \sum_{i=1}^{l} p_i c_s^i(X_s^i)$$

The expected value minimization is reasonable in a repeated decision-making framework. However, it does not sufficiently guard against the worst case over all the possible scenarios in more risk averse or one-shot applications. Our main contribution in this thesis is to consider combinatorial problems in new models of uncertainty that overcome the shortcomings of the stochastic optimization model and extract structural properties that can be generalized to a larger class of problems. We also use the structural properties to design efficient heuristics to computationally solve a large class of problems.

1.1 Demand-Robust Model

Motivated by the above mentioned shortcoming of expected value minimization, we introduce a two stage model of demand-robustness. Robust optimization, where the objective is to minimize over the worst case costs, has been studied in literature [5, 37, 43, 51]. However, 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 demand or a subset of the constraints that the problem is required to satisfy. We refer to this as *demand uncertainty*. For example, consider the deterministic set covering problem where we are given a ground set of elements $U = \{u_1, \ldots, u_n\}$, a family of sets $\mathcal{S}$ and a cost function $c_f : \mathcal{S} \rightarrow \mathbb{R}^+$ and the goal is to find a minimum cost subset of sets from $\mathcal{S}$ that cover all the elements. In the two stage demand-robust model, we address the problem where the set of elements that require coverage is not known in advance. The uncertainty is specified as a list of l second-stage scenarios where each scenario is a subset of elements that require coverage if that scenario materializes. The goal is to find a first stage solution (X_f) and for each scenario i, a recourse or second stage solution (X_s^i) such that $X_f \cup X_s^i$ covers all the elements in scenario i. The objective is to minimize the worst case cost over all the scenarios, i.e.,

$$c_f(X_f) + \max_{i=1}^{l} c_s^i(X_s^i)$$

where c_s^i denotes the second-stage costs in scenario i. Recall that the cost of all elements inflate by the same factor σ_i in scenario i in our model. Our main contributions in this demand-robust model are the following.

1. We prove a structural result about the first stage solution of a general covering problem namely that there is a first stage solution that is a minimal feasible solution for some subset of scenarios and can be augmented in the second stage to complete the solution with a loss of a factor 2 in the total cost of the solution.

2. We obtain approximation algorithms for a variety of standard covering problems in this model including Steiner trees, minimum multi-cut, vertex cover and uncapacitated facility location. While many of the results are adaptations of algorithms recently developed for two-stage stochastic programming problems by using the structural result, we also show new applications of old metric rounding techniques for the multi-cut problem in the demand-robust model.

The details of the model and the results are presented in Chapter 2. Improved approximations for demand-robust shortest path and min-cut problems via a guess are prune algorithm are presented in Chapter 3. As a byproduct of these results, we also obtain the first constant factor approximation for the two-stage stochastic min-cut problem using a novel LP formulation and a charging argument using the Gomory-Hu cut tree [29].

1.2 Chance Constrained Model

The robust optimization approach guards against the worst case future. However, a tiny fraction of outlier scenarios can significantly increase the cost of our solution in a robust optimization approach. Avoiding such scenarios might result in a substantial reduction in the solution cost while still maintaining a high reliability of the solution. To overcome this problem, we introduce *chance-constraints* (see [11, 6]) in the robust and stochastic models. The idea of chance constrained optimization is as follows: we are given some reliability parameter $0 < \rho < 1$, and are required to output a feasible solution for only a subset of the scenarios whose total probability is at least ρ. We can think of the remaining scenarios as being outlier scenarios that we can choose to ignore. Henceforth, we refer to a subset of scenarios whose total probability is at least ρ as a ρ *fraction of the scenarios*.

We introduce the chance constrained optimization framework for combinatorial optimization problems, generalizing the framework of partial covering problems that have been widely studied in literature [10, 2, 24, 41, 25]. For example, in a partial covering version of the set covering problem defined above, we are given a target $k \leq n$ and the goal is to find a minimum cost subset of sets from $\mathcal{S}$ that cover at least k. This problem is a special case of a one-stage chance constrained set covering problem where demand uncertainty is specified by a list of n scenarios each being a singleton element occurring with probability $\frac{1}{n}$ and the required reliability $\rho = \frac{k}{n}$. For each $i = 1, \ldots, n$, scenario i contains only the element u_i, i.e., only u_i requires coverage if scenario i materializes.

We consider two models of demand uncertainty in the chance-constrained framework: *Explicit scenario model* where the demand uncertainty is specified by a list of explicit scenarios and *Implicit scenario model* where demand-scenarios (possibly an exponential number) are specified implicitly by a probability distribution over demand-elements that require coverage. As an example consider the chance-constrained set covering problem where each element e occurs with some specified probability p_e independently of other elements. Thus, an exponential number of demand scenarios and their associated probabilities are specified implicitly by this probability distribution. On the other hand, in the explicit scenario model an explicit list of demand-scenarios (subsets of elements requiring coverage) is given as an input. The goal is to find a minimum cost solution that covers at least a specified ρ fraction of the scenarios.

While it is easy to obtain bi-criteria approximation algorithms for the chance-constrained problems that violate the chance constraint by a small factor, we consider the problem of satisfying the chance constraint strictly.

1. We show that in the explicit scenario model (with more than one element in all the scenarios), both one-stage and two-stage problems are at least as hard to approximate as the dense k-subgraph (DkS) problem. The Dense k-Subgraph problem is conjectured to be $\Omega(n^\delta)$-hard to approximate for some $\delta > 0$ [23].

2. For the special case when each scenario has a single element, while the one-stage problem directly reduces to a weighted partial covering problem, we show that many two-stage problems (including set cover, facility location

etc) reduce to a weighted partial covering problem via a guess-and-prune method.

3. The two-stage shortest path problem does not reduce to a partial covering version but can be reduced to the weighted k-MST problem where the weight function is submodular. We give an $O(\log k)$-approximation for this problem.

4. We also consider an implicit scenario model of uncertainty where scenarios (possibly an exponential number) are specified implicitly by a probability distribution. In particular, we consider a model where each demand element occurs with a given probability independently of others referred from hereon as the *independent-scenarios* model. While it is not even clear if the two-stage problem in the independent-scenarios model is in NP, we show that the one-stage problem in this model can be reduced to a weighted partial covering problem. We also extend these results for the one-stage problem where the demand uncertainty is specified by a general probability distribution such that the *cumulative probability* of any demand-scenario can be computed efficiently and is *strictly-monotone* with respect to set inclusion.

5. **Computational Study** In [54], we give an efficient algorithm to solve chance constrained covering problems where the demand is random. We formulate the problem as an MIP using ideas from the reduction of the chance constrained set covering problem in the independent distribution model to a weighted version of the partial set covering problem. We then derive a family of cutting planes that can be proved to be facets of the convex hull of feasible solutions for the MIP and furthermore, can be generated very efficiently. The strengthened formulation is an extremely efficient procedure to solve the chance constrained covering problems. We corroborate our study with extensive computational results on a large testbed of instances. This work appears in the doctoral dissertation of Saxena [53] and will not be included as a part of this thesis.

The results of the chance constrained models are presented in Chapter 4.

1.3 Chance Constrained Knapsack Problem

The two models of uncertainty described in the above two sections consider problems where uncertainty is in the demand (or the right hand side of the constraint matrix). In Chapter 5, we consider the chance-constrained knapsack problem where each item has a know deterministic profit but a random size distributed according to a known distribution independent of other items. In this problem, the uncertainty appears in the constraint and we refer to such a model of uncertainty as *data uncertainty*. Given a reliability level ρ, the goal in the chance-constrained knapsack problem is to select a set of items that maximize the profit subject to the probabilistic constraint that the probability of total size of all the selected items not exceeding the knapsack size is at least ρ. We give a polynomial time approximation scheme (PTAS) for the problem i.e. given $\epsilon > 0$, we can obtain a set of items whose total profit is at least $(1 - \epsilon)$ times the optimal profit and the chance-constraint is satisfied strictly and the running time of the algorithm is $\tilde{O}(n^{\frac{1}{\epsilon}})$ where n is the number of items in the input.

1.4 Locating Emergency Response Centers for Post-Disaster Logistics

In Chapter 6, we present our work on locating emergency response centers for efficient post-disaster logistics. We consider the problem of effectively locating emergency response and distribution centers to provide services in a post-disaster scenario such as an earthquake. In a post-disaster scenario, not only is the demand uncertain but the underlying transportation network also is uncertain and depends on the disaster scenario. Thus, this problem combines the aspects of both data and demand uncertainty. To perform post-disaster relief operations effectively, planning in the pre-disaster phase is necessary. We study the problem of locating emergency response and distribution centers such that for a large fraction of disaster scenarios, all the demand locations can be reached from some response center within a specified time. In particular, we develop an efficient sampling based algorithm which allows us to estimate the *quality* of a given set of emergency facilities by estimating quantities such as the fraction of disaster scenarios for which the given set of facilities can reach all the demand locations within a given time bound and average fraction of demand satisfied by the given set of facilities over all scenarios. For the purpose of our study, we consider the seismic risk problem in Istanbul, Turkey and use the data from that problem for our computational experiments.

2

Demand-Robust Model for Two-stage Covering Problems

Robust optimization has roots in both Decision Theory [39, 42] as well as Mathematical Programming [15]. While min-max regret approaches were advanced in the former field as conservative decision rules, robust optimization was discussed along with stochastic programming [6] 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 [51] in facility design problem, Mulvey et al. [43] in mathematical programming, and most recently, Kouvelis and Yu [37] 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. [5, 4]. A recent annotated bibliography available online summarizes this line of work [46].

We consider 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 modeled 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.

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 [30, 19, 33, 49, 57]. These two-stage models (e.g., from [30]) 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.

2.1 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 $S_1, S_2, \ldots, S_m \subset U$ 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_i: 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 $c_f(x)$. In the k^{th} scenario, it becomes costlier by a factor σ_k i.e. $c_k(x) = \sigma_k c_f(x)$. In the second stage, one of the scenarios 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_f to be bought in the first stage, and for each k, a set of elements X_s^k to be bought in the recourse stage if scenario k is realized, such that $X_f \cup X_s^k$ contains a feasible solution for client set S_k. The cost of covering scenario k is $c_f(X_f) + c_s^k(X_s^k)$. 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_f even though some of them may not be required in the solution for any one fixed scenario.

Problem	Deterministic	Stochastic	Demand-robust
Steiner Tree	1.55 [50]	3.55 [30], 30 [31]	30*
Vertex Cover	2 (Primal-dual)	2 [49]	4*
Facility Location	1.52 [40]	5 [49], 3.04 [57]	5*
Min Cut	1	$O(\log m)^*$	$O(\log m)^*$
Min Multi-Cut	$O(\log r)$ [26]	$O(\log rm \cdot \log\log rm)^*$	$O(\log rm \cdot \log\log rm)^*$

Figure 2.1: *Result Summary . * denotes results in [18]. In the table, m, n and r denote the number of scenarios, number of nodes and maximum number of pairs per scenario respectively.*

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 $\{t_1^k, t_2^k, \ldots\}$. $sol(S_k)$ is the set of all edge sets that connect all terminals $\{t_1^k, t_2^k, \ldots\}$ to the root r.

2.1.1 Results

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 uncapacitated facility location, and provide approximation algorithms for these problems. Our results are summarized in Figure 2.1. 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.

One of our main contributions is to frame the demand-robust problems and show how this leads to interesting versions of well-studied problems in combinatorial optimization. In Section 2.2, 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 [26, 38]. In Section 2.3, 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 [38] 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 2.3.2).

In Section 2.4, 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 [31], finally giving us the constant-factor approximation result for robust Steiner trees.

In Section 2.5, 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 [57] such as vertex cover and the rounding methods of Ravi and Sinha [49] for facility location.

2.2 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 $c_f(e)$ in the first stage and $\sigma_k c_f(e)$ 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_f to be bought in the first stage and for each k, a set E_s^k to be bought in the recourse stage if scenario k is realized, such that removing $E_f \cup E_s^k$ from the graph G disconnects r from the terminal t_k. The objective is to minimize the maximum cost over all scenarios. The robust min-cut problem is proved to be NP-hard in [35].

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

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

Relaxing the integrality constraints to $x_e^0 \geq 0$ 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.

2.2.1 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 $\mathrm{LP_{opt}}$ denote the optimum value of the LP relaxation. To round the fractional LP solution, we use the region growing technique of Garg et al. [26]. We would like to stress that the notion of *volume* used here is different from the LP volume used in [26]. 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 for each scenario. We denote the copies by $G_1, \ldots, G_m$. We also introduce a copy G_0 for the first stage solution. Edge e costs $c_f(e)$ in the graph G_0 and $\sigma_k c_f(e)$ 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 $dist_k$. For any subset $S \subset V$, let $\delta(S) = \{(u,v) \in E | u \in S, v \notin S\}$. We define the volume $V_k(t_k, \rho)$ of the ball as

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

Here $dist_k(t_k, e)$ denotes the metric distance between t_k and the closer end-point of edge e. Note that the volume $V_k(t_k, \rho)$, for any ρ, is not same as the LP volume. However, it is bounded above by $\mathrm{LP_{opt}}$, which facilitates Claim 2.2.1. 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.

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

Algorithm Robust-Min-cut

1. Let $G_0, G_1, \ldots, G_m$ be copies of G. Initialize $E_f, E_s^1, \ldots, E_s^k \leftarrow \phi$.
2. Repeat the following:
   (a) Find a terminal t_k that is connected to r in the graph $G_k = (V, E \setminus (E_f \cup E_s^k))$.
   (b) Find a radius $\rho < 1/2$ for which $V_k(t_k, \rho)/C(t_k, \rho)$ is minimum.
   (c) If $V_k^0(t_k, \rho) \geq \frac{1}{2} V_k(t_k, \rho)$, then set $E_f \leftarrow E_f \cup \delta(B_k(t_k, \rho))$ and remove $B_k(t_k, \rho)$ from all graphs $G_0, G_1, \ldots, G_m$.
   Else $V_k^1(t_k, \rho) > \frac{1}{2} V_k(t_k, \rho)$. Set $E_s^k \leftarrow E_s^k \cup \delta(B_k(t_k, \rho))$ and remove $B_k(t_k, \rho)$ from the graph G_k.

   Until all the terminals are separated from r.

Figure 2.2: Algorithm for Robust Min-Cut

and

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

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 $C(t_k, \rho) := \sum_{e \in \delta(B_k(t_k, \rho))} c_f(e)$.

Claim 2.2.1 *The analysis technique of Garg et. al [26] can be used to show that there exists a radius $\rho < 1/2$ such that the following holds in the step 2b of the algorithm in Figure 2.2.1.*

$$C(t_k, \rho) \leq 2 \log m \cdot V_k(t_k, \rho). \qquad (2.1)$$

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

Lemma 2.2.2 *Cost of the edges E_f is at most $4 \log m \cdot (\mathrm{LP}_{\mathrm{opt}} + \sum_e c_f(e) \tilde{x}_e^0)$.*

Proof: In the algorithm, we include the edges $\delta(B_k(t_k, \rho))$ in E_f 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_f is bounded by

$$\begin{aligned} c_f(E_f) &\leq 4 \log m \textstyle\sum_k V_k^0(t_k, \rho) \\ &\leq 4 \log m(\mathrm{LP}_{\mathrm{opt}} + \textstyle\sum_e c_f(e)\tilde{x}_e^0). \end{aligned}$$

■

Lemma 2.2.3 *Cost of the edges E_s^k is at most $4 \log m \cdot \sum_e \sigma_k c_f(e)\tilde{x}_e^k$.*

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_s^k can be bounded as follows:

$$\begin{aligned} c(E_s^k) &\leq \sigma_k C(t_k, \rho) \leq 4 \log m \cdot \sigma_k V_k^1(t_k, \rho) \\ &\leq 4 \log m \textstyle\sum_e \sigma_k c_f(e)\tilde{x}_e^k. \end{aligned}$$

■

Theorem 2.2.4 *The Algorithm Robust-Min-Cut produces an $O(\log m)$-approximate solution to the robust min-cut problem.*

Proof: Using Lemmas 2.2.2 and 2.2.3 the total cost of any scenario k can be bounded as follows:

$$\begin{aligned} c(E_f) + c(E_s^k) &\leq 4 \log m \left(\mathrm{LP}_{\mathrm{opt}} + \sum_e c_f(e)(\tilde{x}_e^0 + \sigma_k \tilde{x}_e^k) \right) \\ &\leq 8 \log m \cdot \mathrm{LP}_{\mathrm{opt}} \end{aligned}$$

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

2.2.2 Multi-terminal Scenarios

The algorithm for robust min-cut can be adapted to give an $O(\log m)$-approximation for the case when each scenario contains a set of terminals rather than a single terminal. The k^{th} scenario consists of a set of terminals S_k that must be disconnected from the root r if scenario k materializes. In this case, we modify the input graph $G = (V, E)$ as follows. For each scenario $k = 1, \ldots, m$, we add a new vertex s_k and add edges $\{(v, s_k) | v \in S_k\}$ and set $c_f(v, s_k) = M$ for all $v \in S_k$, where M is the sum of all edge costs in E. Now, modify scenario k, $S_k = \{s_k\}$ which is a single terminal and can be solved using the algorithm for robust min-cut described above. Since all the edges added to a new vertices $s_1, \ldots, s_m$ have a very high cost, none of them is selected in a first stage or a second stage solution by our algorithm. Therefore, for each scenario k, separating s_k from r also leads to separating $\{v | v \in S_k\}$ from r.

2.2.3 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 2.2.5 *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.

There are m choices for the maximum second-stage cost and for each guess, the algorithm computes m minimum cuts in a tree which can be done in time linear in the number of vertices. Also, the algorithm computes one min-cut in a general graph of size $(|V|+1)$ to find the first-stage solution which can be done in $O(|V|^3)$. Therefore, the running time is $O(m^2|V|) + O(|V|^3) = O(|V|^3)$ since $m \leq |V|$. ■

2.3 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 $\{(s_1^k, t_1^k), (s_2^k, t_2^k), \ldots\}$. We want to find a set of edges E_f to buy in the first stage and E_s^k to buy in the recourse stage if scenario k is materialized such that $E_f \cup E_s^k$ separates each of the pairs $\{(s_1^k, t_1^k), (s_2^k, t_2^k), \ldots\}$. An edge e costs $c_f(e)$ in the first stage and $\sigma_k c_f(k)$ 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.

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

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 $G_1, G_2, \ldots, G_m$, one for each scenario. We also maintain G_0 for the first stage solution. However, we need to modify the ball growing procedure. In robust min-cut problem, when a boundary of a ball $B(t_k, \rho)$ 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 $B(s_i^k, \rho)$ and $B(t_i^k, \rho')$ 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 $V^k(s_i^k, \rho) \geq \frac{1}{2}V(s_i^k, \rho)$ (resp. $V^k(t_i^k, \rho') \geq \frac{1}{2}V(t_i^k, \rho')$), then we include $\delta(B(s_i^k, \rho))$ (resp. $\delta(B(t_k^k, \rho')))$ in the edge set E_k and remove the ball from the graph G_k. Otherwise, we find the ball among $B(s_i^k, \rho)$ and $B(t_i^k, \rho')$ which has smaller number of terminal pairs (from all scenarios) that have not been separated. Suppose $B(s_i^k, \rho)$ has smaller number of such pairs. Then we include the edges $\delta(B(s_i^k, \rho))$ in E_f and remove the ball $B(s_i^k, \rho)$ from all graphs $G_0, G_1, \ldots, G_m$. We run the algorithm recursively inside each of the components formed.

This algorithm is similar to the divide-and-conquer algorithm for Feedback Edge Set problem due to Leighton and Rao [38]. 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 2.3.1 *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 2.2.2 we can bound the cost of the algorithm paid for edges in G_0 as follows.

Lemma 2.3.2 *In each level of recursion, each unit of volume in the graph G_0 gets a charge of $O(\log rm)$.*

Theorem 2.3.3 *There is a polynomial-time $O(\log^2 rm)$-approximation algorithm for the robust multi-cut problem.*

Proof: Note that 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 $\log rm$ for at most $O(\log rm)$ levels of recursion. Therefore the total cost paid by the algorithm for edges in G_0 is at most $O(\log^2 rm \cdot OPT)$, where OPT is the optimum value of the LP relaxation. Hence, the total cost paid in any scenario is $O(\log^2 rm \cdot OPT) + O(\log rm \cdot OPT) = O(\log^2 rm \cdot OPT)$. ∎

2.3.1 Improved approximation

We now show how to improve the approximation factor to $O(\log rm \log\log rm)$ using the ideas from [21, 22, 56]. We modify our divide-and-conquer algorithm as follows. For a terminal s_i^k, we find a ball $B(s_i^k, \rho)$ such that $C(s_i^k, \rho) \leq V(s_i^k, \rho) \cdot 4 \log(V_0/V(s_i^k, \rho)) \log\log V_0$, where $V_0 = \sum_e c_f(e)(x_e^0 + x_e^k)$ is the total volume. The analysis technique from [21] 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. We bound the cost paid using the following recurrence relation:

$$\begin{aligned} cost(V_0) &\leq cost(V(s_i^k, \rho)) + cost(V_0 \setminus V(s_i^k, \rho)) \\ &\quad +4 \log(V_0/V(s_i^k, \rho)) \log\log V_0 \cdot V(s_i^k, \rho). \end{aligned}$$

Solving this recurrence, we get that the cost paid for the edges in graph G_0 is bounded by $O(V_0 \cdot \log rm \log\log rm)$. Hence the total cost paid by the algorithm is bounded by $O(\log rm \log\log rm) \cdot \mathrm{LP}_{\mathrm{opt}}$.

2.3.2 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_f on the edges and a root node r. We are also given a collection M of m scenarios with p_k being the probability of occurrence of scenario $k \in M$. For each scenario $k \in M$ there exists a node t_k and we demand that r and t_k must be separated if the k^{th} scenario appears in the recourse stage. An edge e costs $c_f(e)$ in the first stage and $\sigma_k c_f(e)$ if k^{th} scenario 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 while that is not the case in the algorithm of Garg et al. [26] for the deterministic multi-cut. In the case of robust min-cut or multi-cut, the volume is bounded from above by cost of the LP solution. This enables us to claim that there exists a radius $\rho \leq \frac{1}{2}$ such that the cost of the cut $C(t_k, \rho)$ is at most $O(\log m) \cdot V(t_k, \rho)$. This argument is not applicable to the stochastic min-cut as volume in a ball might not be bounded by the cost of the LP solution in the ball. Hence, we do some preprocessing before applying the region growing argument. We show how to do the transformation for stochastic min-cut.

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 $\sum_{i \in S} \sigma_i p_i \cdot OPT \leq \frac{|S|}{m^2} OPT \leq \frac{OPT}{m}$. Hence, we can ignore these scenarios while constructing our first stage solution.

Now, when we apply the region growing algorithm for scenario $i \in M \setminus S$ the total volume in the graph is at most $V = \sum_e c_f(e)(x_e^0 + x_e^i)$. The cost of the LP solution is at least

$$\begin{aligned} \sum_e c_f(e)(x_e^0 + \sigma_i p_i x_e^i) &\geq \sum_e c_f(e)(x_e^0 + \tfrac{1}{m^2} x_e^i) \\ &\geq \tfrac{1}{m^2} \sum_e c_f(e)(x_e^0 + x_e^i). \end{aligned}$$

Hence, $V \leq m^2 \cdot c(LP)$. Now, we can show using the techniques of Garg et al. [26] that there exists a radius $\rho \leq \frac{1}{2}$ such that $C(t_k, \rho) \leq 4 \log m \cdot V(t_k, \rho)$. 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 2.3.4 *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 2.3.5 *There exists a polynomial time algorithm which returns an $O(\log rm \log\log rm)$ approximate solution to the stochastic multi-cut problem.*

2.4 Special 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 extended to a complete solution in the second stage without much cost overhead. We use this structural result to obtain a constant factor approximation for the robust Steiner tree problem.

2.4.1 A Structural Lemma for the First Stage Solution

Lemma 2.4.1 *Given any problem Π in the robust two-stage model, there exists a first stage solution $\tilde{X}_f$ and a subset $\mathcal{S} \subseteq \{S_1, \ldots, S_m\}$ of scenarios, such that $\tilde{X}_f$ 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_f^* be the first stage solution and X_s^{i*} be the recourse stage solution in scenario i. Also, let X_f^{i*} be the part of first stage solution used in scenario i i.e. it is a minimal subset of X_f^* such that $X_f^{i*} \cup X_s^{i*}$ is a feasible solution for scenario i. We construct an alternate first stage solution $\tilde{X}_f$, such that it is a union of feasible solutions for a subset of scenarios. $\tilde{X}_f$ will contain elements from the optimal first stage solution X_f^*, and also from the optimal recourse stage solutions $X_s^{1*}, \ldots, X_s^{m*}$. Let A denote the elements of X_f^* in $\tilde{X}_f$. We construct $\tilde{X}_f$ as follows.

1. Initialize $A \leftarrow \phi$ and $B \leftarrow \phi$.
2. For each scenario $i = 1, 2, \ldots, m$, repeat the following
   (a) $X_f^{\prime i} = X_f^{i*} \setminus A$.
   (b) If $c_f(X_f^{\prime i}) \geq c_f(X_s^{i*})$, then $A \leftarrow A \cup X_f^{\prime i}$ and $B \leftarrow B \cup X_s^{i*}$.
3. $\tilde{X}_f \leftarrow A \cup B$.

Figure 2.3: Structural Lemma

Our new first stage solution $\tilde{X}_f = A \cup B$. Note that $A \subseteq X_f^*$. Therefore, $c_f(A) \leq c_f(X_f^*)$. Also, all elements in B are charged to disjoint parts of A. Thus, by construction $c_f(B) \leq c_f(A)$ which implies $c_f(\tilde{X}_f) \leq 2 \cdot c_f(X_f^*)$. Clearly, $\tilde{X}_f$ is a feasible solution for a subset of scenarios and it is minimal due to optimality of $X_f^*, X_s^{1*}, \ldots, X_s^{m*}$ and the minimality of X_f^{i*} for each i. Furthermore, we claim that $\tilde{X}_f$ 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 $\tilde{X}_f$, say i. This implies that when scenario i was considered in the above sequence, $c_f(X_f^{\prime i}) < c_f(X_s^{i*})$. Thus, we can buy $X_f^{\prime i}$ in the recourse stage and charge it to the cost of X_s^{i*}. Let the new recourse stage solution be $\tilde{X}_s^i = X_s^{i*} \cup (X_f^{\prime i} \setminus A)$. Hence, $c_i(\tilde{X}_s^i) \leq 2 \cdot c_i(X_s^{i*})$ as $c_i(x) = \sigma_i \cdot c_f(x)$.
Thus, the final cost of the new solution is

$$\begin{aligned} \max_i \{c_f(\tilde{X}_f) + c_i(\tilde{X}_s^i)\} &\leq \max_i \ 2 \cdot (c_f(X_f^*) + c_i(X_s^{i*})) \\ &\leq 2 \cdot OPT \end{aligned}$$

■

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 $X_f^*, X_s^{1*}, \ldots, X_s^{m*}$), the special solution can be constructed as in the procedure described above. Let the constructed solution be $X_f, X_s^1, \ldots, X_s^m$. From the proof of Lemma 2.4.1, we have that $c_i(X_s^i) \leq 2 \cdot c_i(X_s^{i*}), \ i = 0, 1, \ldots, m$. Thus, the stochastic objective for the new solution is,

$$\begin{aligned} c_f(X_f) + \sum_{i=1}^{m} p_i c_i(X_s^i) &= c_f(X_f) + \sum_{i=1}^{m} p_i \sigma_i c_f(X_s^i) \\ &\leq 2(c_f(X_f^*) + \sum_{i=1}^{m} p_i \sigma_i c_f(X_s^{i*})) \end{aligned}$$

Thus, the above lemma gives an alternate proof for a similar lemma in [31] that proves that there is a connected first-stage solution for the stochastic Steiner tree problem which costs at most three times the optimal, with a better bound of two rather than three.

2.4.2 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 $c_f(e)$ in the first stage and $c_k(e) = \sigma_k c_f(e)$ in the k^{th} scenario of the second stage. A solution to the problem is a set of edges E_f to be bought in the first stage and a set E_s^k in the recourse stage for each scenario k. The solution is feasible if $E_f \cup E_s^k$ contains a Steiner tree connecting $S_k \cup \{r\}$. The cost paid in the k^{th} scenario is $c_f(E_f) + \sigma_k \cdot c_f(E_s^k)$. The objective is to minimize the maximum cost over all scenarios.

The structural lemma (Lemma 2.4.1) 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. We consider a flow-based formulation on a directed graph where each undirected edge is bi-directed. For any subset $S \subset V$, let $\delta_+(S) = \{e = (u, v) : u \in S, v \notin S\}$ and $\delta_-(S) = \delta_+(V \setminus S)$. The IP formulation for the robust Steiner tree problem is shown in (2.2)-(2.8).

$$\min \ z \tag{2.2}$$

$$\forall\, k, \quad z \ \geq \ \sum_{e \in E} c_f(e) \cdot (x_e^0 + \sigma_k \cdot x_e^k) \tag{2.3}$$

$$\forall\, t \in S_k, \forall\, k, \quad \sum_{e \in \delta_+(t)} (r_e^0(t) + r_0^k(t)) \geq \ 1 \tag{2.4}$$

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

$$\sum_{e \in \delta_+(v)} r_e^0(t) + r_e^k(t) = \sum_{e \in \delta_-(v)} r_e^0(t) + r_e^k(t) \tag{2.5}$$

$$\sum_{e \in \delta_-(v)} r_e^0(t) \leq \sum_{e \in \delta_+(v)} r_e^0(t) \tag{2.6}$$

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

$$r_e^k(t) \le\ x_e^k \tag{2.7}$$

$$r_e^k(t), x_e^k \in\ \{0,1\} \tag{2.8}$$

This formulation is similar to one used by Gupta et al. in [31], 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, $x^1, x^2, \ldots, x^k$ 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. Note 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 2.4.1, we know that $z_{IP} \le 2OPT$. The fractional LP solution can be rounded using the same rounding scheme as that of Gupta et al. [31]. Thus, the following lemma can be derived from [31].

Lemma 2.4.2 *([31]) Let $\tilde{z}, \tilde{x}^0, \tilde{x}^1, \ldots, \tilde{x}^k$ be a fractional solution to the linear relaxation of the IP in (2.2)-(2.8). It can be rounded to obtain an integral solution $T^0, T^1, \ldots, T^k$, such that $T^0 \cup T^i$ connects $S_i \cup \{r\}, \forall\, i$. Furthermore, $c_f(T^0) \le 15 \cdot \sum_{e \in E} c_f(e) \cdot x_e^0$ and $\forall i, c_i(T^i) \le 15 \cdot \sum_{e \in E} c_i(e) \cdot x_e^i$.*

Theorem 2.4.3 *The Robust Steiner Tree Problem can be approximated within a factor of* 30 *in polynomial time.*

Proof: Lemma 2.4.2 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, $z_{IP} \le 15 \cdot \tilde{z} \le 30 \cdot OPT$. Hence, we obtain a 30-approximation for the Robust Steiner Tree problem. ■

2.5 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.

2.5.1 Covering Problems of Shmoys and Swamy [57]

Two-stage stochastic set covering problems were studied in a general setting by Shmoys and Swamy in [57], 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 [57] 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 2.5.1 *The demand-robust vertex cover problem can be approximated within a factor of 4.*

2.5.2 Robust Facility Location

In this problem we are given a set of facilities F and a set of clients $S_1, S_2, \ldots, S_m$ 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 $\sigma_1, \sigma_2, \ldots, \sigma_m$.

Our approximation algorithm proceeds along the lines of the LP-rounding algorithm due to Ravi and Sinha [49]. The algorithm in [49] 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 2.5.2 *The demand-robust facility location problem can be approximated within a factor of* 5.

[1] Although Ravi and Sinha [49] have claimed an 8-approximation, a more careful analysis of their algorithm gives a 5-approximation.

2.6 Conclusion and Open Problems

In this chapter, 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. The results presented in this chapter appear in Dhamdhere et al. [18].

3

Guess and Prune Algorithms for Demand-Robust Covering Problems

We introduced the two-stage demand-robust versions of common optimization problems in Chapter 2, where uncertainty in demand is modeled as an explicit list of demand scenarios. In this chapter, we present a new paradigm *guess and prune* and give improved approximation algorithms for shortest path and min-cut problems in the two-stage demand robust model. Specifically, we obtain a 2-approximation for the robust min-cut problem and a 7.1-approximation for the robust shortest path problem.

We crucially exploit and benefit from the structure of the demand-robust problem: *in the second stage, every scenario can pay up to the maximum second stage cost without worsening the solution cost*. This is not true for the stochastic versions where the objective is to minimize the expected cost over all scenarios. At a very high level, the algorithms for the problems considered are as follows: Guess the maximum second stage cost C in some optimal solution. Using this guess identify scenarios which do not need any first stage "help" i.e. scenarios for which the best solution costs at most a constant times C in the second stage. Such scenarios can be ignored while building the first stage solution. For the remaining scenarios or a subset of them, we build a low-cost first stage solution and prove the approximation bounds by a charging argument.

We give a 2-approximation for the demand-robust min-cut problem via a charging argument using Gomory Hu cut trees [29]. A $(1 + \sqrt{2})$-approximation based on a guess-and-prune strategy for the demand-robust min-cut problem appears in Golovin et al. [28] but the algorithm uses a different charging argument that exploits the laminarity of minimum cuts separating a given root node from other terminals. An $O(\log n)$-approximation is also known and is presented in Chapter 2 and appears in Dhamdhere et al. [18].

As a byproduct, we also obtain a first constant factor 4-approximation for the stochastic min-cut problem. The analysis uses a novel LP formulation and also a

Gomory Hu tree based charging argument similar to the robust version.

For the demand-robust shortest path problem, we give an algorithm with an improved approximation factor of 7.1 as compared to the 16-approximation that is presented in Chapter 2.

Both demand-robust shortest path and demand-robust min-cut problems are NP-hard. The shortest path problem can be proved to be NP-hard by a simple reduction from the Steiner tree problem: if the inflation factors are all ∞, then the demand-robust shortest-path is exactly a Steiner tree problem on the set of terminals defined by the scenarios. While the demand-robust min-cut problem is shown to be NP-hard in [35].

We also consider "hitting set" versions of demand-robust min-cut and shortest path problems where each scenario is a set of terminals instead of a single terminal and the requirement is to satisfy at least one terminal (separate from the root for the min-cut problem and connect to the root for the shortest path problem) in each scenario. We obtain approximation algorithms for these "hitting set" variants by relating them to two classical problems, namely Steiner multicut and group Steiner tree.

3.1 Two-stage Demand-Robust Min-Cut

Consider the two-stage demand-robust min-cut problem as defined in Section 2.2. Here, we present a 2-approximation for this problem.

To motivate our approach, let us consider the robust min-cut problem on trees. Suppose we know the maximum cost that some optimal solution pays in the second stage (say C). Any terminal t_i whose min-cut from r costs more than $\frac{C}{\sigma_i}$ should be cut away from r in the first stage. Thus, if we know C, we can identify exactly which terminals U should be cut in the first stage. The remaining terminals pay at most C to buy a cut in the second stage. If there are k scenarios, then there are only $k+1$ choices for C that matter, as there are only $k+1$ possible sets that U could be. Though we may not be able to guess C, we can try all possible values of U and find the best solution. This algorithm solves the problem exactly on trees.

The algorithm for general graphs has a similar flavor. In a general graph if for any terminal the minimum r-t_i cut costs more than $\frac{C}{\sigma_i}$, then we can only infer that the first stage should "help" this terminal i.e. buy some edges from a r-t_i cut. In the case of trees, every minimal r-t_i cut is a single edge, so the first stage cuts t_i from the root. However, this is not true for general graphs. We can prove that a similar algorithm that completely cuts t_i from the root gives a constant factor approximation using a charging argument. As in the algorithm for trees, we reduce the needed non-determinism by guessing a set of terminals rather than C itself. We refer to the first-stage cost of the minimum r-t_i cut as $\mathrm{mcut}(t_i)$.

Algorithm for Robust Min-Cut
$T = \{t_1, t_2, \ldots, t_k\}$ are the terminals, $r \leftarrow$ root .

1. For each terminal t_i, compute the cost (with respect to c) of a minimum r-t_i cut, denoted $\mathrm{mcut}(t_i)$.
2. Let C be the maximum second stage cost of some optimal solution. Guess $U := \{t_i : \sigma_i \cdot \mathrm{mcut}(t_i) > 2C\}$.
3. First stage solution: $E_f \leftarrow$ minimum r-U cut.
4. Second stage solution for scenario i:

$$E_s^i \leftarrow \text{ any minimum } r\text{-}t_i \text{ cut in } G \setminus E_f$$

Figure 3.1: A factor 2-approximation for Robust Min-Cut

If we relabel the scenarios in decreasing order of $\sigma_i \cdot \mathrm{mcut}(t_i)$, then for every choice of C, $U = \emptyset$ or $U = \{t_1, t_2, \ldots, t_j\}$ for some $j \in \{1, 2, \ldots, k\}$. Thus, we need to try only $k + 1$ values for C. This algorithm runs in $\tilde{O}(k^2 mn)$ time on undirected graphs using the max flow algorithm of Goldberg and Tarjan [27] to find min cuts.

Let OPT denote an optimal solution and let E_f^* denote the set of first stage edges in OPT. The second stage cost of our algorithm is at most $2C$ which is equal to twice second stage cost of OPT for the correct guess of C. We show that the first stage solution E_f, given by our algorithm has cost $c_f(E_f) \leq 2c_f(E_f^*)$ by constructing a cut that separates r from all the terminals in U and costs at most $2c_f(E_f^*)$. This proves that the output solution is a 2-approximation.

For any $S \subset V$, let $\delta_G(S) = \{e = (u, v) \in E(G) | u \in S, v \notin S\}$ and let $E(S) = \{e = (u, v) \in E(G) | u, v \in S\}$. Consider the graph $G' = (V, E \setminus E_f^*)$ and let $H = (V, F)$ be a Gomory-Hu tree for G' with respect to the edge costs c_f. For any two vertices $u, v \in V$, let $\mathrm{mcut}(u, v)$ denote the cost of the u, v-min-cut in graph G' with respect to edge costs c_f. The Gomory-Hu tree H is a tree on vertices V and a cost function $c_h : F \to \mathbb{R}_+$. Let $\mathcal{P}(u, v)$ denote the unique path from u to v in H. The tree H has the following property: for any two vertices $u, v \in V$, $\mathrm{mcut}(u, v) = \min_{e \in \mathcal{P}(u,v)} c_h(e)$. Furthermore, if $e_{uv} = \mathrm{argmin}_{e \in \mathcal{P}(u,v)} c_h(e)$, then the two connected components obtained by removing e_{uv} from H form a u, v-min-cut in G.

Consider H and root it at r. A vertex u is an *ancestor* of v if u occurs on the unique path $\mathcal{P}(v,r)$ from v to r. Let $U_m = \{t \in U | \nexists v \in U$ s.t. v is an ancestor of $t\}$. Consider $t \in U_m$ and let $e_t = \text{argmin}_{e \in \mathcal{P}(t,r)} c_h(e)$. Let S_t be the component containing t after removing e_t from H. As an illustrative example, consider the Gomory-Hu tree in Figure 3.2. In this example, $U = \{t_1, t_2, \ldots, t_7\}, U_m = \{t_1, t_5, t_6\}$ and $\mathcal{T}_m = \{t_1, t_5\}$.

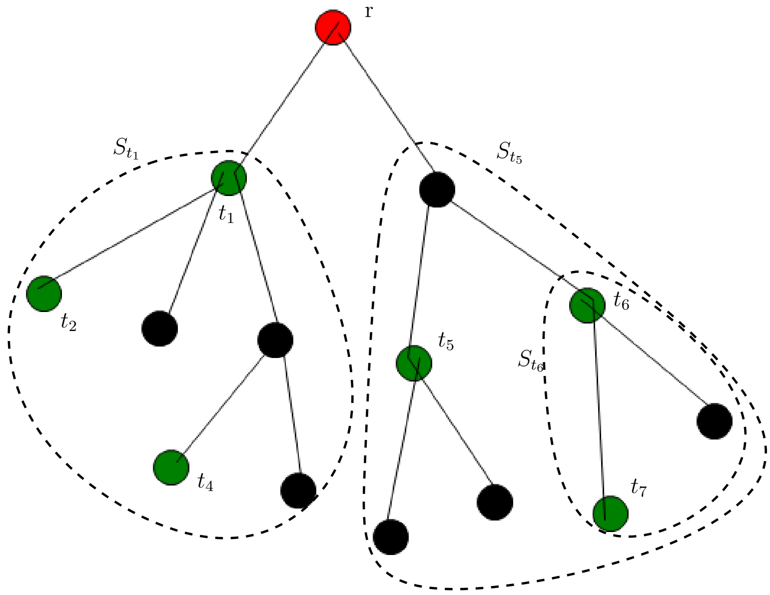

Figure 3.2: Gomory-Hu tree with root r, $U = \{t_1, t_2, \ldots, t_7\}, U_m = \{t_1, t_5, t_6\}$ and $\mathcal{T}_m = \{t_1, t_5\}$.

For all terminals $t \in U$, $c_h(e_t) \leq C$ as the second stage cost of OPT is at most C. Since $t \in U$, the r-t min-cut cost in G has cost greater than $2C$ (with respect to c_f) which implies $c_f(\delta_G(S_t)) \geq 2C$.

Lemma 3.1.1 *For any $t \in U$*

1. $c_f(\delta_G(S_t)) = c_f(E_f^* \cap \delta_G(S_t)) + c_f(\delta_G(S_t) \setminus E_f^*)$.
2. $c_f(\delta_G(S_t) \setminus E_f^*) = c_h(e_t) \leq c_f(\delta_G(S_t) \cap E_f^*)$.

Proof: $\delta_G(S_t) = (\delta_G(S_t) \cap E_f^*) \uplus (\delta_G(S_t) \setminus E_f^*)$, and thus $c_f(\delta_G(S_t)) = c_f(E_f^* \cap \delta_G(S_t)) + c_f(\delta_G(S_t) \setminus E_f^*)$. Also, $c_f(\delta_G(S_t) \setminus E_f^*) = c_h(e_t)$ as H is the Gomory-Hu tree of G' and e_t is the cheapest edge in $\mathcal{P}(r,t)$.

Since $c_f(\delta_G(S_t)\setminus E_f^*) \leq C$ and $c_f(\delta_G(S_t)) \geq 2C$, we have $c_f(\delta_G(S_t)\setminus E_f^*) \leq c_f(E_f^* \cap \delta_G(S_t))$. ■

For any terminal $t \in U_m$, recall $e_t = \text{argmin}_{e\in\mathcal{P}(t,r)} c_h(e)$ and S_t is the component containing t after removing e_t from H. We construct a first stage solution (denoted E_c) separating r from U as follows. Let $T_m = \{t \in U_m | \nexists v \in U_m \text{ s.t. } S_t \subset S_v\}$ and let

$$E_c = \bigcup_{t\in T_m} \delta_G(S_t) \setminus E(\bigcup_{t\in T_m} S_t).$$

Lemma 3.1.2 *The set of edges E_c separate r from all terminals in U.*

Proof: Consider any terminal $t \in U$. For the sake of contradiction, suppose there exists a path between t and r in $G \setminus E_c$. Since $r \notin (\cup_{t\in T_m} S_t)$, and by definition $t \in S_t \subseteq (\cup_{t\in T_m} S_t$, there exists an edge $e \in \delta_G(\cup_{t\in T_m} S_t)$. Clearly,

$$e \in (\cup_{t\in T_m} \delta_G(S_t)) \text{ and } e \notin E(\cup_{t\in T_m} S_t)$$

Therefore, $e \in E_c$ which is a contradiction. ■

Lemma 3.1.3 $c_f(E_c) \leq 2c_f(E_f^*)$

Proof: Consider any $e \in E_f^*$ and let $S_e = \{t \in T_m | e \in \delta_G(S_t)\}$. Since $S_{t_1} \cap S_{t_2} = \emptyset$ for any distinct $t_1, t_2 \in T_m$, it is easy to observe that $|S_e| \leq 2, \forall e \in E_f^*$. Let

$$E_{f,1}^* = \{e \in E_f^* \mid |S_e| = 1\}$$

$$E_{f,2}^* = \{e \in E_f^* \mid |S_e| = 2\}$$

Clearly, $E_{f,2}^* \subseteq E(\cup_{t\in T_m} S_t)$. Therefore,

$$E_c \subseteq \cup_{t\in T_m} \delta_G(S_t) \setminus E_{f,2}^* = \left(\cup_{t\in T_m} \delta_G(S_t) \setminus E_f^*\right) \cup E_{f,1}^*.$$

For any $t \in T_m$,

$$c_f(\delta_G(S_t) \setminus E_f^*) \leq c_f(\delta_G(S_t) \cap E_f^*) = c_f(\delta_G(S_t) \cap E_{f,1}^*) + c_f(\delta_G(S_t) \cap E_{f,2}^*)$$

Therefore,

$$\sum_{t\in T_m} c_f(\delta_G(S_t) \setminus E_f^*) \leq c_f(E_{f,1}^*) + 2c_f(E_{f,2}^*)$$

$$
\begin{aligned}
c_f(E_c) &\leq \sum_{t \in T_m} c_f(\delta_G(S_t) \setminus E_f^*) + c_f(E_{f,1}^*) && (3.1)\\
&\leq 2c_f(E_{f,1}^*) + 2c_f(E_{f,2}^*) && (3.2)\\
&\leq 2c_f(E_f^*) && (3.3)
\end{aligned}
$$

■

Therefore, we have the following theorem.

Theorem 3.1.4 *There is a polynomial time algorithm which gives a* 2*-approximation for the robust min-cut problem.*

3.2 Two-stage Stochastic Min-Cut Problem

In this section, we consider the stochastic version of the two-stage min-cut problem as defined in Section 2.3.2. Here, we present a 4-approximation for this problem improving from $O(\log m)$-approximation presented in Section 2.3.2.

In the robust version of the problem, we are able to exploit the fact that if the maximum second stage cost of OPT is C then all scenarios could spend up to C in the second stage without worsening the objective value. This property allows us to identify the set of terminals that should be separated from the root in the first stage. In the stochastic version however, this property does not hold. We use a novel LP formulation to identify the set of terminals that should be separated from the root in the first stage. We complete the argument to prove the required approximation factor via a Gomory-Hu tree based charging argument similar to the previous section.

Let y_i be a binary variable denoting whether terminal t_i is separated from r in the first stage or not. Let x_e be a binary variable denoting whether edge e is selected in the first stage solution or not. Let $\mathcal{P}(u, v)$ denote the set of paths from u to v in graph G. Also, let $\mathrm{mcut}(t_i)$ denote the cost of the minimum r, t_i-cut in G with respect to the cost function c_f. Consider the following integer program (IP1).

$$\min \sum_{e \in E} c_f(e) \cdot x_e + \sum_{i=1}^{k} \sigma_i p_i \cdot \mathrm{mcut}(t_i) \cdot (1 - y_i)$$

$$\sum_{e \in P} x_e \geq y_i \quad \forall P \in \mathcal{P}(r, t_i), \forall i = 1, \ldots, k \tag{3.4}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{3.5}$$

$$y_i \in \{0,1\} \qquad \forall i = 1, \ldots, k \tag{3.6}$$

Let z^* be the objective value of an optimal solution to the above program and let OPT be an optimal solution to the stochastic min-cut problem, let E_f^* be the first stage solution of OPT and E_s^i be the second stage solution for scenario i. Then we can prove the following lemma.

Lemma 3.2.1 $z^* \leq 2 \cdot (c_f(E_f^*) + \sum_{i=1}^k \sigma_i p_i \cdot c_f(E_s^i))$

Proof: We construct a feasible solution to IP1 from OPT and prove that the cost is at most $2 \cdot$ OPT using an argument similar to the proof of Lemma 3.1.3. Let $E_f^i \subset E_f^*$ be the minimal set of edges such that $E_f^i \cup E_s^i$ separates r from t_i. Also, let

$$\mathcal{T}_1 = \{t_i \in U | c_f(E_f^i) \leq c_f(E_s^i)\} \text{ and } \mathcal{T}_2 = U \setminus \mathcal{T}_1$$

Consider the following assignment for the variables in IP1.

$$y_i = \begin{cases} 1 & \text{if } t_i \in \mathcal{T}_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{i=1}^{k} \sigma_i p_i \cdot \mathrm{mcut}(t_i) \cdot (1 - y_i) = \sum_{i: t_i \in \mathcal{T}_1} \sigma_i p_i \cdot \mathrm{mcut}(t_i) \tag{3.7}$$

$$\leq \sum_{i: t_i \in \mathcal{T}_1} \sigma_i p_i \cdot (c_f(E_f^i) + c_f(E_s^i)) \tag{3.8}$$

$$\leq 2 \sum_{i: t_i \in \mathcal{T}_1} \sigma_i p_i \cdot c_f(E_s^i)) \tag{3.9}$$

Now we construct a cut E_c that separates all terminals in $\mathcal{R}$ from the root and show that it has cost at most $2c_f(E_f^*)$. Consider a Gomory-Hu tree $H = (V, F)$ on the graph $G' = (V, E \setminus E_f^*)$ with edge costs $c_h : F \to \mathbb{R}_+$. For $u, v \in V$, let $\mathcal{P}(u, v)$ denote the unique path between u and v in H. For any terminal $t \in \mathcal{R}$, let $e_t = \mathrm{argmin}_{e \in \mathcal{P}(t,r)} c_h(e)$ and S_t denote the component containing t after removing e_t from H. Let us define

$$\mathcal{T}_m = \{t \in \mathcal{T}_2 | \nexists t' \in \mathcal{T}_2 \text{ s.t. } S_t \subset S_{t'}\}$$

Now we construct the cut as follows.

$$E_c = (\bigcup_{t \in \mathcal{T}_m} \delta_G(S_t)) \setminus E(\bigcup_{t \in \mathcal{T}_m} S_t)$$

Clearly, E_c separates all terminals in $\mathcal{T}_2$ from r. We prove that $c_f(E_c) \leq 2c_f(E_f^*)$ by a similar argument as in the proof of Lemma 3.1.3. Consider any $e \in E_f^*$ and let $S_e = \{t \in T_m | e \in \delta_G(S_t)\}$. Since $S_{t_1} \cap S_{t_2} = \emptyset$ for any distinct $t_1, t_2 \in T_m$, it is easy to observe that $|S_e| \leq 2, \forall e \in E_f^*$. Let

$$E_{f,1}^* = \{e \in E_f^* \mid |S_e| = 1\}$$

$$E_{f,2}^* = \{e \in E_f^* \mid |S_e| = 2\}$$

Clearly, $E_{f,2}^* \subseteq E(\cup_{t \in T_m} S_t)$. Therefore,

$$E_c \subseteq \cup_{t \in T_m} \delta_G(S_t) \setminus E_{f,2}^* = \left(\cup_{t \in T_m} \delta_G(S_t) \setminus E_f^*\right) \cup E_{f,1}^*.$$

For any $t \in T_m$,

$$c_f(\delta_G(S_t) \setminus E_f^*) \leq c_f(\delta_G(S_t) \cap E_f^*) = c_f(\delta_G(S_t) \cap E_{f,1}^*) + c_f(\delta_G(S_t) \cap E_{f,2}^*)$$

The first inequality follows from the choice of $\mathcal{T}_m$ since for any terminal $t \in \mathcal{T}_m$, $c_f(\delta_G(S_t)) = c_f(E_f^i) + c_f(E_s^i)$ and $c_f(E_s^i) \leq c_f(E_f^i)$.
Therefore,

$$\sum_{t \in T_m} c_f(\delta_G(S_t) \setminus E_f^*) \leq c_f(E_{f,1}^*) + 2c_f(E_{f,2}^*)$$

$$\begin{aligned} c_f(E_c) &\leq \sum_{t \in T_m} c_f(\delta_G(S_t) \setminus E_f^*) + c_f(E_{f,1}^*) && (3.10) \\ &\leq 2c_f(E_{f,1}^*) + 2c_f(E_{f,2}^*) && (3.11) \\ &\leq 2c_f(E_f^*) && (3.12) \end{aligned}$$

Now, setting $x_e = 1, \forall e \in E_c$ and 0 otherwise, we obtain a feasible solution to IP1. Furthermore, $\sum_{e \in E} c_f(e) x_e = c_f(E_c) \leq 2c_f(E_f^*)$ which implies $z^* \leq 2(c_f(E_f^*) + \sum_{i=1}^k c_f(E_s^i))$. ■

Consider the LP relaxation (LP1) of IP1 where integrality conditions on x_e and y_i are relaxed for all $e \in E$ and $i = 1, \ldots, k$. Let $\tilde{z}$ be the optimal objective value of LP1 and let $(\tilde{x}, \tilde{y})$ be an optimal solution. We can prove the following lemma.

Lemma 3.2.2 *We can obtain an integral feasible solution* $(\hat{x}, \hat{y})$ *to* LP1 *in polynomial time such that*

$$\sum_{e \in E} c_f(e)\hat{x}_e + \sum_{i=1}^{k} \sigma_i p_i \cdot \text{mcut}(t_i) \cdot (1 - \hat{y}_i) \leq 2\tilde{z} \leq 2z^*$$

Proof: Let $T_1 = \{t_i \in U | \tilde{y}_i \leq \frac{1}{2}\}$ and $T_2 = U \setminus T_1$. Let E_m be a minimum cost cut that separates all terminals in T_2 from r. Consider the following solution. For all $i = 1, \ldots, k$,

$$\hat{y}_i = \begin{cases} 1 & \text{if } \tilde{y}_i \geq \frac{1}{2} \\ 0 & \text{o/w} \end{cases}$$

For all $e \in E$,

$$\hat{x}_e = \begin{cases} 1 & \text{if } e \in E_m \\ 0 & \text{o/w} \end{cases}$$

Consider the fractional solution $2\tilde{x}$. This is a fractional cut that separates all terminals in T_2 from r. Since we can round a fractional cut to an integral cut of the same (or lower) cost, the minimum cost cut E_m has cost $c_f(E_m) \leq 2\sum_{e \in E} c_f(e)\tilde{x}_e$.

Also,

$$\sum_{i=1}^{k} \sigma_i p_i \cdot \text{mcut}(t_i) \cdot (1 - \hat{y}_i) = \sum_{i:\tilde{y}_i < 1/2} \sigma_i p_i \cdot \text{mcut}(t_i) \tag{3.13}$$

$$\leq 2 \sum_{i:\tilde{y}_i < 1/2} \sigma_i p_i \cdot \text{mcut}(t_i) \cdot (1 - \tilde{y}_i) \tag{3.14}$$

$$\leq 2 \sum_{i=1}^{k} \sigma_i p_i \cdot \text{mcut}(t_i) \cdot (1 - \tilde{y}_i) \tag{3.15}$$

Thus, $(\hat{x}, \hat{y})$ is a 2-approximate integral feasible solution to LP1. ■

The cut corresponding to integral solution $\hat{x}$ is our first stage solution, say E'_f. The second stage solution for scenario i (say E'^{i}_s) is the minimum cut that separates t_i from r in the graph $G \setminus E_f$. Thus,

$$c_f(E'_f) + \sum_{i=1}^{k} \sigma_i p_i c_f(E'^{i}_s) \leq \sum_{e \in E} c_f(e)\hat{x}_e + \sum_{i:\hat{y}_i = 0} \sigma_i p_i \,\text{mcut}(t_i) \tag{3.16}$$

$$\leq 2z^* \leq 4(c_f(E^*_f) + \sum_{i=1}^{k} c_f(E^i_s)) \tag{3.17}$$

Therefore, we obtain the following theorem.

Theorem 3.2.3 *There is a polynomial time algorithm which gives a 4-approximation for the stochastic min-cut problem.*

3.3 Demand-Robust Shortest Path Problem

We consider the two-stage demand-robust shortest path problem which is a special case of the two-stage Steiner tree problem considered in Section 2.4. The problem is defined on a undirected graph $G = (V, E)$ with a root vertex r and cost c on the edges. The i^{th} scenario S_i is a singleton set $\{t_i\}$ rather than a set of terminals as in the Steiner tree problem. An edge e costs $c(e)$ in the first stage and $c_i(e) = \sigma_i \cdot c(e)$ in the i^{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_i in the recourse stage for each scenario i. The solution is feasible if $E_0 \cup E_i$ contains a path between r and t_i. The cost paid in the i^{th} scenario is $c(E_0) + \sigma_i \cdot c(E_i)$. The objective is to minimize the maximum cost over all scenarios.

The following structural result for the demand-robust shortest path problem can be obtained from the structural lemma in Sectionr 2.4.

Lemma 3.3.1 *[18] Given a demand-robust shortest path problem instance on an undirected graph, there exists a solution that costs at most twice the optimum such that the first stage solution is a tree containing the root.*

The above lemma implies that we can restrict our search in the space of solutions where first stage is a tree containing the root and lose only a factor of two. This property is exploited crucially in our algorithm.

3.3.1 Algorithm

Lemma 3.3.1 implies that there is a first stage solution which is a tree containing the root r and it can be extended to a final solution within twice the cost of an optimum solution. We call such a solution as a *connected solution*. Fix an optimal connected solution, say $E_0^*, E_1^*, \ldots, E_k^*$. Let C be the maximum second stage cost paid by this solution over all scenarios, i.e. $C = \max_{i=1}^{k}\{\sigma_i \cdot c(E_i^*)\}$. Therefore, for any scenario i, either there is path from t_i to root r in E_0^*, or there is a vertex within a distance $\frac{C}{\sigma_i}$ of t_i which is connected to r in E_0^*, where distance is with respect to the cost function c, denoted $\mathrm{dist}_c(\cdot, \cdot)$. We use this fact to obtain a constant factor approximation for our problem.

The algorithm is as follows: Let C be the maximum second stage cost paid by the connected optimal solution (fixed above) in any scenario. We need to try

Algorithm for Robust Shortest Path
Let C be the maximum second stage cost of some fixed connected optimal solution.
$T = \{t_1, t_2, \ldots, t_k\}$ are the terminals, $r \leftarrow$ root, $\alpha \leftarrow 1.775$, $V' \leftarrow \phi$.

1. $V' := \{t_i | \operatorname{dist}_c(t_i, r) > \frac{2\alpha \cdot C}{\sigma_i}\}$
2. $\mathcal{B} := \{B_i = B(t_i, \frac{\alpha \cdot C}{\sigma_i}) | \ t_i \in V'\}$, where $B(v, d)$ is a ball of radius d around v with respect to cost c. Choose a maximal set $\mathcal{B}_{\mathcal{I}}$ of non-intersecting balls from $\mathcal{B}$ in order of non-decreasing radii.
3. Guess $R^0 := \{t_i | B_i \in \mathcal{B}_{\mathcal{I}}\}$.
4. First stage solution: $E_0 \leftarrow$ The Steiner tree on terminals $R^0 \cup \{r\}$ output by the best approximation algorithm available.
5. Second stage solution for scenario i: $E_i \leftarrow$ Shortest path from t_i to the closest node in the tree E_0

Figure 3.3: Robust Shortest Path Algorithm

only $k \cdot n$ possible values of C [1], so we can assume that we have correctly guessed C. For each scenario t_i, consider a shortest path (say P_i) to r with respect to cost c. If $c(P_i) \leq \frac{2\alpha \cdot C}{\sigma_i}$, then we can handle scenario i in the second stage with cost only a factor 2α more than the optimum. Thus, t_i can be ignored in building the first stage solution. Here $\alpha > 1$ is a constant to be specified later. Let $V' = \{t_i \mid \operatorname{dist}_c(r, t_i) > \frac{2\alpha \cdot C}{\sigma_i}\}$.

For each $t_i \in V'$, let B_i be a ball of radius $\frac{\alpha \cdot C}{\sigma_i}$ around t_i. Here, we include internal points of the edges in the ball. We collectively refer to vertices in V and internal points on edges as *points*, V_P. Thus, $B_i = \{v \in V_P \mid \operatorname{dist}_c(t_i, v) \leq \frac{\alpha \cdot C}{\sigma_i}\}$.

The algorithm identifies a set of terminals $R^0 \subseteq V'$ to connect to the root in the first stage such that the remaining terminals in V' are close to some terminal in R^0 and thus, can be connected to the root in the second stage paying a low-cost.

Proposition 3.3.2 *There exist a set of terminals $R^0 \subseteq V'$ such that:*

1. *For every $t_i, t_j \in R^0$, we have $B_i \cap B_j = \phi$; and*

[1] For each scenario i, the second stage solution is a shortest path from t_i to one of the n vertices (possibly t_i), so there are at most $k \cdot n$ choices of C.

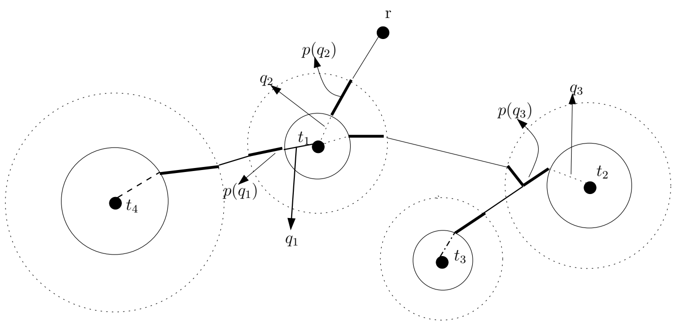

Figure 3.4: Illustration of first-stage tree computation described in Lemma 3.3.3. The balls with solid lines denote $B(t_i, \frac{C}{\sigma_i})$, while the balls with dotted lines denote $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$.

2. *For every* $t_i \in V' \setminus R^0$*, there is a representative* $\mathrm{rep}(t_i) = t_j \in R^0$ *such that* $B_i \cap B_j \neq \phi$ *and* $\frac{\alpha \cdot C}{\sigma_j} \leq \frac{\alpha \cdot C}{\sigma_i}$.

Proof: Consider terminals in V' in non-decreasing order of the radii $\frac{\alpha \cdot C}{\sigma_t}$ of the corresponding balls B_t. If terminal t_i is being examined and $B_i \cap B_j = \phi, \ \forall t_j \in R^0$, then include t_i in R^0. If not, then there exists $t_j \in R^0$ such that $B_i \cap B_j \neq \phi$; define $\mathrm{rep}(t_i) = t_j$. Note that $\frac{\alpha \cdot C}{\sigma_j} \leq \frac{\alpha \cdot C}{\sigma_i}$ as the terminals are considered in order of non-decreasing radii of the corresponding balls. ■

The First Stage Tree.

The first stage tree is a Steiner tree on the terminal set $R^0 \cup \{r\}$. However, in order to bound the cost of first stage tree we build the tree in a slightly modified way. For an illustration, refer to Figure 3.4.

Let G' be a new graph obtained when the balls $B(t_i, \frac{C}{\sigma_i})$ corresponding to every terminal $t_i \in R^0$ are contracted to singleton vertices. We then build a Steiner tree E_{01} in G' with the terminal set as the shrunk nodes corresponding to terminals in R^0 and the root vertex r. In Figure 3.4, E_{01} is the union of solid edges and the thick edges. Now, for every shrunk node corresponding to $B(t_i, \frac{C}{\sigma_i})$, we connect each tree edge incident to $B(t_i, \frac{C}{\sigma_i})$ to terminal t_i using a shortest path; these edges are shown as dotted lines in Figure 3.4 and are denoted by E_{02}. Our first stage solution is the Steiner tree $E_0 = E_{01} \cup E_{02}$.

Lemma 3.3.3 *The cost of* E_0 *is at most* $\frac{1.55\alpha}{\alpha - 1}$ *times* $c(E_0^*)$*, the first stage cost of the optimal connected solution.*

Proof: We know that the optimal first stage tree, E_0^* connects some vertex in the ball $B(t_i, \frac{C}{\sigma_i})$ to the root r for every $t_i \in R^0$, for otherwise the maximum second stage cost of OPT would be more than C. Thus, E_0^* induces a Steiner tree on the shrunk nodes in G'. We build a Steiner tree on the shrunk nodes as terminals using the algorithm due to Robins and Zelikovsky [50]. Thus,

$$c(E_{01}) \leq 1.55\ c(E_0^*) \tag{3.18}$$

Now, consider edges in E_{02}. Consider a path $q \in E_{02}$ connecting some edge incident to $B(t_i, \frac{C}{\sigma_i})$ to t_i. Since q is the shortest path between its end points, we have $c(q) \leq \frac{C}{\sigma_i}$. Now, consider a path from terminal t_i along q until it reaches $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$ and label the portion between $B(t_i, \frac{C}{\sigma_i})$ and $B(t_i, \frac{\alpha \cdot C}{\sigma_i})$ as $p(q)$. By construction, we have $c(p(q)) \geq \frac{(\alpha-1)\cdot C}{\sigma_i}$, so $c(q) \leq \frac{1}{\alpha-1} \cdot c(p(q))$.

For any two paths $q_1, q_2 \in E_{02}$, the paths $p(q_1)$ and $p(q_2)$ are edge-disjoint. Clearly, if q_1 and q_2 are incident to distinct terminals of R^0, then $p(q_1)$ and $p(q_2)$ are contained in disjoint balls and thus are edge-disjoint. If q_1 and q_2 are incident to the same terminal, then it is impossible that $p(q_1) \cap p(q_2) \neq \phi$ as E_{01} is a tree on the shrunk graph. Hence, we have

$$\sum_{e \in E_{02}} c(e) = \sum_{q \in E_{02}} c(q) \leq \sum_{q \in E_{02}} \frac{1}{\alpha - 1} \cdot c(p(q)) \leq \sum_{e \in E_{01}} \frac{1}{\alpha - 1} \cdot c(e) \tag{3.19}$$

where the last inequality is due to edge-disjointness of $p(q_1)$ and $p(q_2)$ for any two paths $q_1, q_2 \in E_{02}$. Thus, $c(E_0) = c(E_{01}) + c(E_{02}) \leq c(E_{01}) + \frac{1}{\alpha-1} \cdot c(E_{01}) \leq \frac{1.55\alpha}{\alpha-1} \cdot c(E_0^*)$, where the last inequality follows from (3.18). ■

Second Stage.

The second stage solution for each scenario is quite straightforward. For any terminal t_i, E_i is the shortest path from t_i to the closest node in E_0.

Lemma 3.3.4 *The maximum second stage cost for any scenario is at most* $2\alpha \cdot C$.

Proof: We need to consider the following cases:

1. $t_i \in R^0$: Since the first stage tree E_0 connects t_i to r, $E_i = \phi$. Thus, $c(E_i) = 0$.

2. $t_i \in V' \setminus R^0$: By Proposition 3.3.2, there exists a representative terminal $t_j \in R^0$ such that $B_i \cap B_j \neq \phi$ and $\sigma_j \geq \sigma_i$. Therefore, $\text{dist}_c(t_i, t_j) \leq$

$\frac{\alpha \cdot C}{\sigma_i} + \frac{\alpha \cdot C}{\sigma_j} \leq \frac{2\alpha \cdot C}{\sigma_i}$. We know that t_j is connected to r in E_0. Thus, the closest node to t_i in the first stage tree is at a distance at most $\frac{2\alpha \cdot C}{\sigma_i}$. Hence, $\sigma_i \cdot c(E_i) \leq 2\alpha \cdot C$.

3. $t_i \notin V'$: Then the shortest path from t_i to r with respect to cost c is at most $\frac{2\alpha \cdot C}{\sigma_i}$. Hence, the closest node to t_i in the first stage tree is at a distance at most $\frac{2\alpha \cdot C}{\sigma_i}$ and $\sigma_i \cdot c(E_i) \leq 2\alpha \cdot C$.

■

Theorem 3.3.5 *There is a polynomial time algorithm which gives a 7.1 approximation for the robust shortest path problem.*

Proof: From Lemma 3.3.3, we get that $c(E_0) \leq \frac{1.55\alpha}{\alpha-1} c(E_0^*)$. From Lemma 3.3.4, we get that the second stage cost is at most $2\alpha \cdot C$. Choose $\alpha = \frac{3.55}{2} = 1.775$. Thus, we get $c(E_0) \leq (3.55) \cdot c(E_0^*)$ and $\max_{i=1}^{k}\{\sigma_i \cdot c(E_i)\} \leq (3.55) \cdot C$. From Lemma 3.3.1 we know that $c(E_0^*) + C \leq 2 \cdot \mathsf{OPT}$, where OPT is the cost of optimal solution to the robust shortest path instance. Together the previous three inequalities imply $c(E_0) + \max_{i=1}^{k}\{\sigma_i \cdot c(E_i)\} \leq (7.1) \cdot \mathsf{OPT}$ ■

3.4 Extensions to Hitting Versions

In this problem, we introduce generalizations of demand-robust min-cut and shortest path problems that are closely related to Steiner multicut and group Steiner tree, respectively. In a Steiner multicut instance, we are given a graph $G = (V, E)$ and k sets of vertices $X_1, X_2, \ldots, X_k$ and our goal is to find the cheapest set of edges S whose removal *separates* each X_i, i.e. no X_i lies entirely within one connected component of $(V, E \setminus S)$. If $\bigcap_{i=1}^{k} X_i \neq \emptyset$, we call the instance *restricted*. In a group Steiner tree instance, we are given a graph $G = (V, E)$, a root r, and k sets of vertices $X_1, X_2, \ldots, X_k$ and our goal is to find a minimum cost set of edges S that connects at least one vertex in each $X_i, i = 1, \ldots, k$ to the root r. We show how approximation algorithms for these problems can be combined with our techniques to yield approximation algorithms for “hitting versions” of demand-robust min-cut and shortest path problems.

In the hitting version of robust min-cut (resp. shortest path), each scenario i is specified by an inflation factor σ_i and a set of nodes $T_i \subset V$ (rather than a single node). A feasible solution is a collection of edge sets $\{E_0, E_1, \ldots, E_k\}$ such that for each scenario i, $E_0 \cup E_i$ contains an root-t cut (resp. path) for some $t \in T_i$. The goal is to minimize $c(E_0) + \max_i\{\sigma_i \cdot c(E_i)\}$.

3.4.1 Robust Hitting Cuts

Robust hitting cut is $\Omega(\log k)$-hard, where k is the number of scenarios, even when the graph is a star. In fact, if we restrict ourselves to inputs in which the graph is a star, the root is the center of the star, and $\sigma = \infty$ for all scenarios, then robust hitting cut on these instances is exactly the hitting set problem. In contrast, we can obtain an $O(\log k)$ approximation for robust hitting cut on trees, and $O(\log n \cdot \log k)$ in general using results of Nagarajan and Ravi [44] in conjunction with the following theorem.

Theorem 3.4.1 *If for some class of graphs there is a ρ-approximation for Steiner multicut on restricted instances, then for that class of graphs there is a $(\rho+2)$-approximation for robust hitting cut. Conversely, if there is a ρ-approximation for robust hitting cut then there is a ρ-approximation for Steiner multicut on restricted instances.*

Algorithm: Let $\alpha = \frac{1}{2}(\rho + 1 + \sqrt{\rho^2 + 6\rho + 1})$ and let C be the cost that some optimal solution pays in the second stage. For each terminal t in some group, compute the cost of a minimum root-t cut, denoted $\mathrm{mcut}(t)$. Let $\mathcal{T}' := \{T_i : \forall t \in T_i,\, \sigma_i \cdot \mathrm{mcut}(t) > \alpha \cdot C\}$. Note that there are only $k+1$ possibilities, as in the robust min-cut algorithm. For each terminal set $T_i \in \mathcal{T}'$, separate at least one terminal in T_i from the root in the first stage using an ρ-approximation algorithm for Steiner Multicut [36, 44].

Proof of Theorem 3.4.1: We first show that a ρ-approximation for robust hitting cut implies a ρ-approximation for Steiner multicut on restricted instances. Given a restricted instance of Steiner multicut $(G, X_1, X_2, \ldots, X_k)$ build a robust hitting cut instance as follows: use the same graph and costs, set the root r to be any element of $\bigcap_i X_i$, and create scenarios $T_i = X_i \setminus r$ with $\sigma_i = \infty$ for each i. Note that solutions to this instance correspond exactly to Steiner multicuts of the same cost. Thus robust hitting cut generalizes Steiner multicut on restricted instances.

We now show the approximate converse, that a ρ-approximation for Steiner multicut on restricted instances implies a $(\rho+2)$-approximation for robust hitting cut. Let OPT be an optimal solution, and let E_0^* be the edge set it buys in stage one, and let C_1 and C_2 be the amount it pays in the first and second stage, respectively. Note we can handle every $T_i \notin \mathcal{T}'$ while paying at most $\alpha \cdot C_2$.

We prove that the first stage edges $E_0 \subset E[G]$ given by our algorithm satisfy all scenarios in $\mathcal{T}'$, and have cost $c(E_0) \leq \rho(1+\frac{2}{\alpha-1})C_1$. Thus, the total solution cost is at most $\rho(1+\frac{2}{\alpha-1})C_1 + \alpha \cdot C_2$. Compared to the optimal cost, $C_1 + C_2$, we obtain a $\max\{\alpha, \rho(1+\frac{2}{\alpha-1})\}$-approximation. Setting $\alpha = \frac{1}{2}(\rho + 1 + \sqrt{\rho^2 + 6\rho + 1})$ then yields the claimed $(\rho+2)$ approximation ratio.

A cut is called a $\mathcal{T}'$*-cut* if it separates at least one terminal in each $T \in \mathcal{T}'$ from the root. There exists a $\mathcal{T}'$-cut of cost at most $(1 + \frac{2}{\alpha-1})C_1$, by the same argument as in the proof of Theorem 3.1.4. Suppose OPT cuts away t_i^* when scenario T_i occurs. Then OPT is also an optimal solution to the robust min-cut instance on the same graph with terminals $\{t_i^* \mid i = 1, 2, \ldots, k\}$ as k scenarios. Since, for all $t \in T$ such that $T \in \mathcal{T}'$, we have $\sigma_t \cdot \mathrm{mcut}(t) > \alpha \cdot C$, we can construct a root-$\{t_i^* \mid i = 1, 2, \ldots, k\}$ cut of cost at most $(1 + \frac{2}{\alpha-1})C_1$. Thus, the cost of an optimal $\mathcal{T}'$-cut is at most $(1 + \frac{2}{\alpha-1})C_1$. Now apply the ρ-approximation for Steiner multicut on restricted instances. To build the Steiner multicut instance, we use the same graph and edge costs, and create a groups $X_i = T_i \cup \{\text{root}\}$ for each $T_i \in \mathcal{T}'$. Clearly, the instance is restricted. Note that every solution to this instance is a $\mathcal{T}'$-cut of the same cost, and vice-versa. Thus a ρ-approximation for for Steiner multicut on restricted instances yields a $\mathcal{T}'$-cut of cost at most $2(1 + \frac{2}{\alpha-1})C_1$. ■

Corollary 3.4.2 *There is a polynomial time $O(\log n \cdot \log k)$-approximation algorithm for robust hitting cut on instances with k scenarios and n nodes, and an $O(\log k)$-approximation algorithm for robust hitting cut on trees.*

3.4.2 Robust Hitting Paths

Theorem 3.4.3 *If there is a ρ-approximation for group Steiner tree then there is a 2ρ-approximation for robust hitting path. If there is a ρ-approximation for robust hitting path, then there is a ρ-approximation for group Steiner tree.*

Proof: Note that robust hitting path generalizes group Steiner tree (given a GST instance with graph G, root r and groups $X_1, X_2, \ldots, X_k$, use the same graph and root, make each group a scenario, and set $\sigma_i = \infty$ for all scenarios i). Thus a ρ-approximation for robust hitting path immediately yields a ρ-approximation for group Steiner tree.

Now suppose we have an ρ-approximation for group Steiner tree. Lemma 3.3.1 guarantees that there exists a solution $\{E_0, E_1, \ldots, E_k\}$ of cost at most $2\mathsf{OPT}$ whose first stage edges, E_0, are a tree containing root r.

The algorithm is as follows. Guess $C := \max_i\{\sigma_i c(E_i)\}$. Note that for each scenario i the tree E_0 must touch one of the balls in $\{B(t, C/\sigma_i) | t \in T_i\}$, where $B(v, x) := \{u | \operatorname{dist}_c(v, u) \leq x\}$. Thus we can construct groups $X_i := \bigcup_{t \in T_i} B(t, C/\sigma_i)$ for each scenario i and use the ρ-approximation for group Steiner tree on these groups to obtain a set of edges E_0' to buy in the first stage.

Note that $c(E_0') \leq \rho c(E_0)$ and any scenario i has a terminal $t \in T_i$ that is within distance C/σ_i of some vertex incident on an edge of tree E_0'. We conclude that the total cost is at most $\rho c(E_0) + C \leq 2\rho \cdot \mathsf{OPT}$. ■

4

Demand-Robust Covering Problems with Chance Constraints

Optimization models incorporating data and demand uncertainty have long been studied in the literature due to their vast applicability in real world scenarios. Stochastic optimization approaches optimize the expected costs over all scenarios while the robust optimization approaches optimize over the worst case scenario. However, both approaches are plagued by the presence of unlikely outlier scenarios which distort the optimization goals and the resulting solution.

A natural idea to overcome this problem is to prune away the outlier scenarios and solve the problem on remaining scenarios. This approach, referred to as *chance-constrained* optimization (see [11, 6]), has been studied in literature. A chance-constrained model incorporates probabilistic constraints in the traditional stochastic or robust optimization model. Thus, the problem of finding a minimum cost solution which is feasible for ρ fraction of the scenarios for a given reliability $\rho > 0$, can be modeled using chance constraints. This model is best introduced through an example: consider a one-stage shortest path problem on an undirected graph $G = (V, E)$ where we are required to construct a path between root r and an uncertain destination. Each vertex $v \in V$ occurs with probability $\frac{1}{n}$ as the destination and we are required to choose a minimum cost set of edges E_s such that with probability ρ (where ρ is given) there is a path between r and the destination. Note that the problem in this example reduces to finding a minimum cost rooted k-MST in G where $\frac{k}{n} \leq \rho < \frac{k+1}{n}$.

In a chance-constrained optimization approach, the parameter ρ captures the risk aversion of the optimizer. When $\rho = 1$, we return to the classical robust model, while at $\rho = 0$, the empty solution is feasible. We extend the chance constrained framework to robust covering problems with demand-uncertainty (such as considered in Dhamdhere et al. [18]) in both one-stage and two-stage models where the demand-uncertainty is either given as an explicit list of scenarios or specified implicitly. (Our methods also apply directly to the stochastic versions defined e.g., in [49] but we leave the details out).

4.0.3 Previous Work

Chance constrained programming was introduced in Charnes and Cooper [11]. Even with a long history, chance constraint models do not find wide applicability because of the inherent difficulty in solving these problems optimally; namely, the feasible region for a chance constrained problem depends on the underlying uncertainty and is generally non-convex. A detailed discussion of chance constrained programs, and more generally, stochastic programs can be found in [6]. Robust optimization and chance constrained optimization are very closely related (see [9, 13, 20]). Nemirovski and Shapiro [45] show how robust optimization framework provides an approximation of chance constrained programming while Chen et al. [13] propose robust optimization as a technique to obtain feasible solutions for chance constrained programs in [52].

For simple probability distributions, such as a uniform distribution (where each scenario occurs with the same probability), the chance-constrained problem reduces to a more familiar partial covering problem, where we are required to cover some k out of l scenarios with a minimum cost solution. Recall that the shortest path problem described above reduces to finding a minimum cost tree containing the root r that spans at least k vertices. This problem is a partial covering version of the spanning tree problem that has been studied extensively [7, 8, 14, 1] and for which a 2-approximation is known [25]. In general, in a partial set covering problem we are given a set family $\mathcal{F}$, set of elements U and a target $k \leq |U|$ and the goal is to select a minimum cost collection of sets from $\mathcal{F}$ that cover at least k elements. Partial covering versions of several combinatorial problems have been considered such as vertex cover [2, 24, 41], facility location, k-center [10]. However, to the best of our knowledge, there has not been any prior work in designing approximation algorithms for combinatorial problems in the general chance-constrained framework.

4.0.4 Our Contributions

We consider chance constraints in both one-stage as well as two-stage robust covering problems with demand-uncertainty where uncertainty is specified either as an explicit list of demand-scenarios or implicitly as a probability distribution over the demand elements that require coverage. While it is easy to obtain bi-criteria approximation algorithms for the chance-constrained problems that violate the chance constraint by a small factor, we consider the problem of satisfying the chance constraint strictly.

1. We show that in the explicit scenario model (with more than one element in all the scenarios), both one-stage and two-stage problems are at least as hard to approximate as the dense k-subgraph (DkS) problem. The Dense k-Subgraph problem is conjectured to be $\Omega(n^{\delta})$-hard to approximate for some $\delta > 0$ [23].

2. For the special case when each scenario has a single element, while the one-stage problem directly reduces to a weighted partial covering problem, we show that many two-stage problems (including set cover, facility location etc) reduce to a weighted partial covering problem via a guess-and-prune method.

3. The two-stage shortest path problem does not reduce to a partial covering version but can be reduced to the weighted k-MST problem where the weight function is submodular. We give an $O(\log k)$-approximation for this problem.

	Explicit Scenarios		
	1-**elt**		> 1 **elts**
One stage	*Reduces to partial covering*		*DkS-hard*
Two stage	Set Cover, Vertex Cover, Facility Location	*Reduce to partial covering*	*DkS-hard*
	Shortest Path	$O(\log k)$	

Table 4.1: Main results for the explicit scenario uncertainty model

4. We also consider the model of uncertainty where scenarios (possibly an exponential number) are specified implicitly by a probability distribution. In particular, we consider a model where each demand occurs with a given probability independently of others referred from hereon as the *independent-scenarios* model. While it is not even clear if the two-stage problem in the independent-scenarios model is in NP, we show that the one-stage problem in this model can be reduced to a weighted partial covering problem. We also extend these results for the one-stage problem where the demand uncertainty is specified by a general probability distribution such that the *cumulative probability* of any demand-scenario can be computed efficiently and is *strictly-monotone* with respect to set inclusion.

Outline. The rest of the chapter is organized as follows. In Section 4.1, we present the hardness of approximation of problems with more than one element per explicit scenario. In Section 4.3, we consider the explicit scenario model with only one element per scenario and present the reduction of the chance-constrained versions of many robust covering problems to weighted partial covering problems. Finally, in Section 4.4, we consider implicit models of uncertainty and show that the one-stage problems in the independent-scenario model reduce to weighted partial covering problems and also discuss extensions to the general distribution model.

4.1 Hardness of Approximation

We show that the one-stage chance constrained set cover problem in the explicit scenario model is at least as hard to approximate as Dense k-Subgraph even when every scenario has only two elements.

Problem Definition A one stage chance constrained set covering problem in the explicit scenario model (Explicit 1-CCSCP) is as follows: we are given a universe of elements U, a family of subsets $\mathcal{S}$, a cost function c on the subsets in $\mathcal{S}$, a list of l scenarios where scenario i is specified by a subset $S_i \subset U$ and its probability p_i, and a reliability factor $0 < \rho < 1$. The problem is to find a minimum cost partial set cover for elements in a subset of scenarios (say $\mathcal{I}$) such that $\sum_{i \in \mathcal{I}} p_i \geq \rho$.

We prove the following theorem.

Theorem 4.1.1 Explicit 1-CCSCP *is at least as hard to approximate as* Dense k-Subgraph *even when each scenario has only two elements.*

Proof: In a Dense k-Subgraph instance $\mathcal{I}$, we are given a graph $G = (V, E)$ and a number k, and the objective is to find a minimum size subset of vertices $V' \subseteq V$ that induces at least k edges, i.e $|E[V'] \geq k|$.

The reduction is as follows: we construct an instance $\mathcal{I}'$ of Explicit 1-CCSCP. The element set $U = \{v_i | v_i \in V\}$. For each vertex $v_i \in V$, we have a set $S_i = \{v_i\}$ in the set family $\mathcal{F}$. For each edge $e = (v_i, v_j) \in E$, we have a scenario containing two elements $\{v_i, v_j\}$. Now, in the instance $\mathcal{I}'$ of Explicit 1-CCSCP we are required to find a minimum cardinality subset $\mathcal{S}$ of sets from $\mathcal{F}$ such that the sets in $\mathcal{S}$ satisfy at least k scenarios. Note that a scenario, $\{v_i, v_j\}$ is satisfied by $\mathcal{S}$ if both v_i and v_j are contained in some sets in $\mathcal{S}$.

Suppose there is a solution $\mathcal{S}$ for instance I'. Consider $V' = \{v_i | S_i \in \mathcal{S}\}$. Consider any scenario $\{v_i, v_j\}$ that is satisfied by $\mathcal{S}$. Note that $(v_i, v_j) \in E(G)$ and $v_i, v_j \in V'$. Thus, (v_i, v_j) is an induced edge in V' which implies $|E[V']| \geq k$. Thus, $OPT(I) \leq OPT(I')$.

Conversely, consider a solution V' of I that induces at least k edges. Consider $\mathcal{S} = \{S_i | v_i \in V'\}$. It is easy to note that for each edge $(v_i, v_j) \in E[V']$, the corresponding scenario is satisfied by the solution $\mathcal{S}$. Thus, $OPT(I') \leq OPT(I)$. ■

The two-stage covering problems in the explicit scenario model can also be shown to be at least as hard to approximate as Dense k-Subgraph.

Two-stage Chance Constrained Set Covering Problem (Explicit 2-CCSCP**)** We are given a set of elements U, a family of subsets $\mathcal{S}$, cost for each set in $\mathcal{S}$, a reliability level ρ and a list of l future scenarios. Each scenario i is specified by a subset $S_i \subset U$, an inflation factor σ_i and probability p_i. In second stage in scenario i, each set $S \in \mathcal{S}$ becomes costlier by a factor σ_i. The goal is to select a ρ fraction of the scenarios $\mathcal{I}$ and a first stage solution $\mathcal{S}_f \subset \mathcal{S}$. Also, for each scenario $i \in \mathcal{I}$, find a recourse solution $\mathcal{S}_r^i \subset \mathcal{S}$ such that $\mathcal{S}_f \cup \mathcal{S}_r^i$ is a feasible set cover for S_i. The goal is to minimize

$$c(\mathcal{S}_f) + \max_{i \in \mathcal{I}} \sigma_i \cdot c(\mathcal{S}_r^i)$$

If $\sigma_i = \infty$ for all $i = 1, \ldots, l$, the two-stage problem reduces to a one-stage problem and the hardness of approximation follows from Theorem 4.1.1. Therefore, we have the following theorem.

Theorem 4.1.2 Explicit 2-CCSCP *is at least as hard to approximate as* Dense k-Subgraph *even when each scenario has only two elements.*

4.2 Bicriteria Results

We show that if the chance-constraint can be violated by a constant factor, we can obtain an $O(\alpha)$-approximation when an α-approximation is known for the robust problem without the chance-constraints. For the sake of exposition, we consider the Explicit 1-CCSCP problem but essentially the same argument extends to the two-stage problems.

Let there be l scenarios $S_1, \ldots, S_l$ with probabilities $p_1, \ldots, p_l$ respectively. The problem is to satisfy a subset of scenarios whose probabilities sum to the reliability factor ρ.

To formulate this as an integer program (IP1), let z_i be a binary variable that denotes whether scenario i is covered or not.

$$\begin{array}{rcl} \min \sum_{S \in \mathcal{S}} c_S x_S & & \\ \sum_{S: e \in S} x_S & \geq & z_i \;\; \forall e \in S_i \; \forall i = 1, \ldots, l \\ \sum_{i=1}^{l} p_i z_i & \geq & \rho \\ x_S & \in & \{0,1\} \;\; \forall S \in \mathcal{S} \\ z_i & \in & \{0,1\} \;\; \forall i = 1, \ldots, l \end{array}$$

If the deterministic set covering problem has an α-approximation, then we can give an $\frac{\alpha}{\epsilon}$-approximation for any constant $\epsilon > 0$ to the chance constrained problem that violates the chance-constraint and covers only a $\rho' = \frac{\rho-\epsilon}{1-\epsilon}$ fraction of the scenarios.

Theorem 4.2.1 *Suppose there is an α-approximation to the deterministic set covering problem. Then for the* Explicit 1-CCSCP *with reliability ρ, there is an $\frac{\alpha}{\epsilon}$-approximation for any constant $\epsilon > 0$ that covers $\rho' = \frac{\rho-\epsilon}{1-\epsilon}$ scenarios.*

Proof: Assume wlog that each scenario has probability $p = \frac{1}{l}$; otherwise we can consider multiple copies of the same scenario. Now, consider the optimal solution (say $(\tilde{x}, \tilde{z})$) of the LP relaxation of IP1. We know $\sum_{i=1}^{l} \frac{1}{l}\tilde{z}_i \geq \rho$ Let $H = \{i | \tilde{z}_i \geq \epsilon\}$ and $h = |H|$. Therefore,

$$h + (l - h) \cdot \epsilon \geq l\rho$$
$$\Rightarrow h \geq \frac{l(\rho - \epsilon)}{1 - \epsilon}$$

Consider the solution, $\hat{x} = \frac{1}{\epsilon}\tilde{x}$. Clearly $\hat{x}$ is a fractional solution that is feasible for all scenarios in H and can be rounded using the deterministic α approximation to an integer solution. Furthermore, the total probability of scenarios is H is $\frac{\rho-\epsilon}{1-\epsilon}$. Therefore, we obtain an $\frac{\alpha}{\epsilon}$-approximate solution to Explicit 1-CCSCP that violates the chance constraint and covers $\rho' = \frac{\rho-\epsilon}{1-\epsilon}$ scenarios. ■

4.3 Explicit Scenario Models

Note that even one stage versions of covering problems with more than one element per scenario are hard. For instance, we have the following corollary of Theorem4.1.1.

Corollary 4.3.1 *The one-stage (and hence, two-stage) chance-constrained versions of the following covering problems in the explicit scenario model are at least as hard to approximate as* Dense k-Subgraph *even when each scenario has only two elements.*

1. *Vertex Cover (scenario is described by a subset of edges)*
2. *Facility Location (scenario is described by a subset of demand points)*
3. *K-median (scenario is described by a subset of demand points)*
4. *K-center (scenario is described by a subset of demand points)*

5. *Steiner Tree (scenario is described by a subset of vertices)*

Hence in this section, we consider the case when each scenario has exactly one element.

One-stage versions. The one stage versions of the above problems with exactly one element per scenario are directly reducible to the respective partial covering variants. These variants have been well approximated in the literature (2-approximation for partial vertex cover [41], 3-approximation for partial facility location, partial k-center [10], constant-factor for partial k-median [12], and 2-approximation for partial shortest paths that reduce to k-MST [25]), hence we focus on the two-stage version henceforth.

We first give a logarithmic approximation for the two-stage chance-constrained robust version for the general set covering problem. Then, we show how the two stage versions of the above problems (in particular, Vertex Cover, Facility Location and Steiner tree) with one element per scenario can be approximated.

4.3.1 Two-stage Chance-Constrained Set Cover

Theorem 4.3.2 *Consider the* Explicit 2-CCSCP *where you are given a family m subsets $S_1, \ldots, S_m$ with cost function c, and l scenarios such that scenario i contains element e_i, has inflation factor σ_i and occurs with probability p_i and required reliability of the solution is ρ. This problem can be reduced to a weighted partial covering solution and thus, admits an $O(\log(\rho l))$-approximation.*

Proof: Fix an optimum solution and suppose the worst case second stage cost is B in this optimum solution. There are only l choices for B; one corresponding to the second-stage minimum cost solution for each of the l scenarios. Let c_i denote the cost of the minimum-cost set that contains e_i, $T = \{i \in [l] | \sigma_i \cdot c_i \leq B\}$. We can cover all scenarios in T in the second-stage with cost at most B. Let $\tau = \sum_{i \in T} p_i$. We need to cover a subset of scenarios from $[l] \setminus T$ whose total probability is at least $\rho - \tau$ in the first stage. Therefore, for a particular choice of B the problem reduces to a weighted partial set covering problem which admits an $O(\log k)$-approximation if you require to cover k elements. ■

The reduction in the above theorem applies to the two-stage covering problems that satisfy the following property: *If a scenario i that is covered in an optimal solution can not be independently covered in the second-stage within the worst case second-stage cost, then it must be completely covered in the first stage.*

Two-stage Chance-Constrained Vertex Cover. This problem when each scenario consists of a single edge satisfies the above property. Thus, it can be reduced to a weighted partial vertex cover problem, which implies a 2-approximation using the results of [41]. Corresponding versions of the facility location problem and the shortest path problem do not satisfy this property and thus, do not directly reduce to a partial covering problem.

4.3.2 Two-Stage Chance-Constrained Facility Location 2-CCFLP

Problem Definition Given a metric (V, d), a set of potential facilities $\mathcal{F}$ and a set of l scenarios where scenario i is specified by a demand point $v_i \in V$ and inflation factor σ_i and occurs with probability p_i and required reliability is ρ. Opening a facility $j \in \mathcal{F}$ in the first stage costs c_j while opening it in the second stage in scenario i costs $\sigma_i \cdot c_j$. The goal is to select a ρ fraction of the scenarios (say $\mathcal{I}$) and open a set of facilities F_1 to open in the first stage and for each of the selected scenario i, connect to one of the open facilities in F_1 or open a new facility and connect to it in the second stage if that scenario materializes. Let x_j be a binary variable denoting whether $j \in \mathcal{F}$ is opened in the first stage or not and let $f_i(x)$ denote the minimum second-stage cost in scenario i given the first stage solution is x. The objective is to minimize

$$\sum_{j \in \mathcal{F}} c_j x_j + \max_{i \in \mathcal{I}} f_i(x)$$

We reduce the above problem to a weighted partial covering problem and thus, give a 3-approximation for 2-CCFLP.

For the sake of simplicity, we assume that all scenarios occur with the same probability $p = \frac{1}{l}$; essentially the same algorithm and analysis extend to the general problem.

Theorem 4.3.3 *There is a 3-approximation for the 2-CCFLP with l scenarios where each scenario has only one element and the required reliability is $\rho = \frac{k}{l}$.*

Proof: Fix an optimum solution and suppose the first stage facility opening cost is C_1^* and the worst case second stage cost is C_2^* in this optimum solution. There are only $2l \cdot |\mathcal{F}|$ choices for C_2^*. Let $f_i(x)$ denote the minimum-cost solution for scenario i when the first stage solution is x. Let $T = \{i \in [l] | \sigma_i \cdot f_i(0) \leq C_2^*\}$. Note that computing $f_i(0)$ is easy: consider the minimum cost of opening (in the second stage) and connecting v_i to the open facility. We can cover all scenarios in T in the second-stage with cost at most C_2^*. Therefore, the first stage problem is to open a set of facilities such that for at least $k' = k - |T|$ scenarios from $[l] \setminus T$, there is an open facility within a distance $\alpha \cdot C_2^*$ from the demand-point for some approximation factor $\alpha > 0$. Note that there is a set of facilities of cost C_1^* such that for at least k' scenarios in $[l] \setminus T$, the demand-point is within a distance C_2^* of some open facility.

The first stage problem thus reduces to a version of the partial k-center problem considered in Charikar et al. [10] who give a 3-approximation for the problem. Therefore, we can find a set of facilities of cost at most C_1^* such that at least k' demand-scenarios are within a distance $3C_2^*$ of some open facility which gives a 3-approximation for 2-CCFLP. ■

4.3.3 Two-Stage Chance-Constrained Shortest Path (2-CCSPP)

Problem Definition Given a graph $G = (V, E)$ with edge costs c, a root vertex r, a reliability level ρ and a list of l scenarios. Each scenario i is specified by a terminal t_i, an inflation factor σ_i and a probability p_i. The goal is to select a ρ fraction of the scenarios, buy some edges E_f in the first stage and for each selected scenario i, augment the first stage solution in the recourse stage with edges E_s^i (bought at an inflated cost) such that $E_f \cup E_s^i$ contains a path from r to t_i. The objective minimizes the worst case cost over all scenarios.

For the sake of simplicity, we consider the case of uniform probabilities and a uniform inflation factor across all scenarios. Thus, the reliability level ρ translates to covering $k = l\rho$ out of l terminals. However, it is not difficult to extend this algorithm and the analysis to general problem with different probabilities and inflation factors for different scenarios.

Using the structural theorem in Dhamdhere et al. [18], we obtain the following lemma,

Lemma 4.3.4 *For the uniform robust* 2-CCSPP *that requires to cover k out of l terminals, there exists a first stage solution E_f and a set I of k scenarios such that E_f is a tree containing r and can be augmented by E_r^i to obtain a feasible solution for scenarios in I and $c(E_f) + \max_{i \in I} \sigma c(E_r^i) \leq 2\mathsf{OPT}$, where* OPT*is the optimal solution for robust* 2-CCSPP.

Fix an optimal solution to the robust 2-CCSPP such that the first stage solution is connected to r, say $O = (O_f, O_r^1, \dots, O_r^l)$ (some of the recourse edge sets may be empty). From Lemma 4.3.4, we know that $c(O_f) + \max_i \sigma c(O_r^i) \leq 2\mathsf{OPT}$. Let σC be the maximum second stage cost for any scenario in O. We can assume that σC is known (as there are only nl choices of C). Thus, the tree O_f is within a distance C from at least k of the l terminals(say $t_1^*, \dots, t_k^*$).

Algorithm Consider ball B_i of radius $2C$ around terminal t_i. We select a maximal independent set $\mathcal{I}$ on balls $B_1, \dots, B_l$ as follows:

1. Initialize $\mathcal{I} \leftarrow \phi$, $\mathcal{T} \leftarrow \{t_1, \dots, t_l\}$.

2. while ($\mathcal{T} \neq \phi$), do

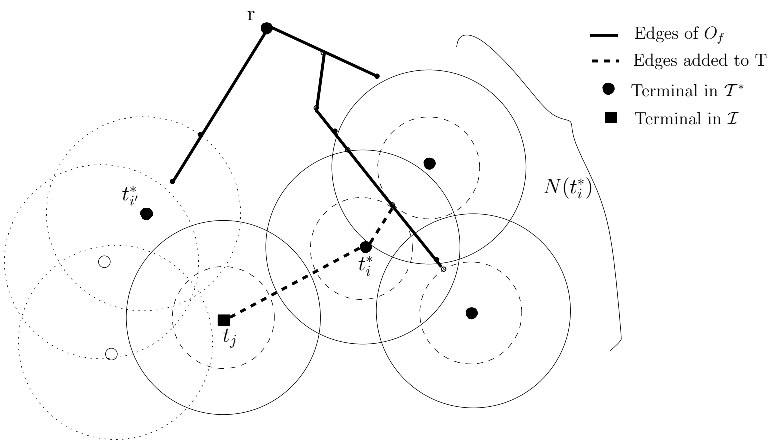

Figure 4.1: Constructing k-MST of cost $O(\mathsf{OPT})$ from O_f under our weight function

(a) Consider the terminal $t_i \in \mathcal{T}$ such that the number of terminals in $\mathcal{T}$ within a distance of at most $4C$ from t_i is maximum (resolve ties arbitrarily).

(b) Let $N(t_i)$=set of terminals in $\mathcal{T}$ that are within distance $4C$ from t_i (including itself) and let $w(t_i) = |N(t_i)|$.

(c) Add t_i to $\mathcal{I}$ and remove all terminals that are within distance $4C$ of t_i from $\mathcal{T}$.

Now, construct a minimum cost spanning tree, T_A containing r that spans terminals of weight at least k. Note that only terminals in the independent set $\mathcal{I}$ have a non-zero weight.

Lemma 4.3.5 $c(T_A) = O(\log k)c(O_f)$

Proof: Suppose $\mathcal{I}$ has terminals $t_1, \ldots, t_q$ (listed in the order they were added to $\mathcal{I}$). Note that $w(t_1) \geq w(t_2) \geq \ldots \geq w(t_q)$. Let $B(t_i^*)$ denote the ball of radius $2C$ around terminal t_i^* and $\mathcal{T}^* = \{t_1^*, \ldots, t_k^*\}$. Recall that t_i^* is a terminal within distance of at most C from O_f; therefore, $B(t_i^*)$ intersects with O_f. For the sake of argument, whenever an edge $e \in O_f$ crosses the ball $B(t_i^*)$ for any $t_i^* \in \mathcal{T}^*$, we introduce a new vertex at the point of intersection and edge e is subdivided into two. It is easy to note that $\sum_{e \in B(t_i^*) \cap O_f} c(e) \geq C$ for any $t_i^* \in \mathcal{T}^*$.

We will now construct a tree T from O_f that contains r and spans terminals in $\mathcal{I}$ of weight at least k. Initialize $T \leftarrow O_f$.

Consider the terminal $t_i^* \in \mathcal{T}^*$ such that the ball $B(t_i^*)$ intersects the ball $\hat{B}_j, j = 1, \ldots, q$ of highest weight. Let $N(t_i^*)$ denote the terminals in $\mathcal{T}^*$ that intersect with $B(t_i^*)$ (including itself) and let $n(t_i^*) = |N(t_i^*)|$. We claim that $n(t_i^*) \leq w(t_j)$. At the time t_j was added to $\mathcal{I}$, t_i^* was also a candidate. Furthermore, all the terminals in $N(t_i^*)$ were also candidates; otherwise one of them would intersect with $\hat{B}_j, j = 1, \ldots, q$ of higher weight contradicting our choice of t_i^*. Since, t_j was chosen in $\mathcal{I}$, $w(t_j) \geq n(t_i^*)$. Thus, the tree T can be extended to reach t_j by charging to the cost of edges in $O_f \cap B(t_i^*)$ since, we know $\sum_{e \in B(t_i^*) \cap O_f} c(e) \geq C$.

In doing so, we have updated the weight of T under our weight function by $w(t_j) \geq n(t_i^*)$. We update $\mathcal{T}^* \leftarrow \mathcal{T}^* \setminus (N(t_i^*) \cup N(t_j))$ and continue. Note that by updating the set terminals $\mathcal{T}^*$ by removing t_i^* and all other terminals in $N(t_i^*)$ we ensure that we do not charge to the same cost of $\mathsf{OPT}(O_f)$ in some other iteration.

Note that we might have removed $w(t_j) + n(t_i^*)$ terminals from $\mathcal{T}^*$ and added only $w(t_j) \geq 1/2(w(t_j + n(t_i^*))$ weight in T. Thus, we would obtain a tree T that has cost $O(c(O_f))$ and spans terminals of weight at least $k/2$.

We repeat this procedure on the remaining terminals of OPTthat are not covered in T. This implies that after $\log k$ rounds, we will obtain a tree T spanning terminals of cumulative weight at least k and $c(T) = O(\log k)c(O_f)$.

Thus, there exists a tree T containing r that spans a subset of terminals in $\mathcal{I} = \{t_1, \ldots, t_q\}$ whose cumulative weight is at least k and $c(T) = O(\log k)c(O_f)$. ∎

Theorem 4.3.6 *There is an $O(\log k)$-approximation to robust* 2-CCSPP.

Proof: Let T_A be the first stage tree returned by the algorithm. It is easy to note that there are at least k terminals from $t_1, \ldots, t_l$ that are within a distance of $4C$ from T_A. Lemma 4.3.5 implies that $c(T_A) = O(\log k)c(O_f)$. Thus, the cost of the solution returned by our algorithm is $(O(\log k)c(O_f) + 4\sigma C) = O(\log k)\mathsf{OPT}$. ∎

We would like to remark here that finding an approximate first stage tree that reaches within a distance C to k of the n terminals is $\Omega(\log n)$-hard by a simple reduction from a set cover problem. In the above algorithm, we find an $O(\log n)$-approximation to the first stage tree. However, we do not obey the distance bound of C strictly and find a tree that is within a distance of $4C$ from at least k terminals. Obtaining a constant approximation for robust-2-CCSPP is an interesting open problem.

4.4 Implicit Scenario Models

In this section, we consider chance-constrained covering problem with implicit scenarios. we restrict our discussion to only one stage problems since it is not even clear whether the two-stage versions are in NP (the set of scenarios satisfying the chance constraint may not be described succinctly).

For the one stage problems, we consider an *independent scenarios* model where each element occurs with a given probability independent of others and extend to a class of general distributions.

4.4.1 Independent Scenarios Model: Reduction to Partial Weighted Covering Problem

Consider the one-stage set covering problem where we are given a set family $\mathcal{F}$ and a universe of elements U. The demand-uncertainty is specified by an independent scenarios model where each element e occurs independently with probability p_e (we refer it as Independent 1-CCSCP). The probability $p(E)$ of any subset E is $\Pi_{e \in E} p_e$. Also,

$$\sum_{E' \subset E} p(E') = \Pi_{e \notin E}(1 - p_e)$$

Theorem 4.4.1 Independent 1-CCSCP *can be reduced to a weighted partial set covering problem.*

Proof: Let z_e be a $0 - 1$ variable that denotes whether e is covered or not. Also, let x_S denote whether set $S \in \mathcal{F}$ is picked in the solution or not. Then, the probabilistic constraint can be written as,

$$\Pi_{z_e = 0}(1 - p_e) \geq \rho$$

Taking logarithms on both sides, we get

$$\begin{aligned} \textstyle\sum_{z_e=0} \log(1 - p_e) &\geq \log \rho \\ \Rightarrow \textstyle\sum_{e \in U}(1 - z_e) \log(1 - p_e) &\geq \log \rho \\ \Rightarrow \textstyle\sum_{e \in U} -z_e \log(1 - p_e) &\geq \log \tfrac{\rho}{\Pi_{e \in U}(1 - p_e)} \end{aligned}$$

For each element $e \in U$, let $w_e = -\log(1 - p_e)$. Also, let $W = \log \frac{\rho}{\Pi_{e \in U}(1-p_e)}$. Note that $w_e > 0$, for all $e \in U$. Now, the chance constrained set covering problem can be reduced to a weighted partial set covering problem where weight of element e is w_e and the goal is to select a minimum-cost family of subsets from $\mathcal{F}$ that cover elements of weight at least W. ∎

For the general set-covering problem, the greedy algorithm gives an $O(\log W)$-approximation where W is the required weight target computed in the proof above.

We also present an LP-based iterative rounding algorithm for this problem that gives an f-approximation where f is the maximum number of sets that an element occurs in. We can formulate the weighted partial covering problem as the following IP.

$$\begin{array}{rcl} \min \sum_{S\in\mathcal{F}} c_S x_S & & \\ \sum_{S:e\in S} x_S & \geq & z_e \ \ \forall e \in U \\ \sum_{e\in U} w_e z_e & \geq & W \\ x_S & \in & \{0,1\} \ \ \forall S \in \mathcal{S} \\ z_e & \in & \{0,1\} \ \ \forall e \in U \end{array}$$

Let OPTdenote the cost of an optimal solution. We remove all sets that cost more than OPTfrom the instance (we can try different values of OPT). In each iteration, the iterative rounding method either selects a set S that covers some element e in the solution or selects an element e for which the corresponding covering constraint can be removed. The algorithm is as follows:

1. Initialize $\mathcal{R} \leftarrow \phi$, $m \leftarrow |\mathcal{F}|$, $n \leftarrow |U|$ and $i \leftarrow 0$.

2. In iteration i, let $(\tilde{x}^i, \tilde{z}^i)$ be a basic optimal solution of LPi (LP-relaxation of problem on ground set U and set family $\mathcal{F}$ and weight requirement W)

   (a) If there exists $e \in U$, such that $\tilde{z}^i_e = 0$, then update $U \leftarrow U \setminus \{e\}$ and go to step 3.

   (b) If there exists $e \in U$, such that $\tilde{z}^i_e = 1$, then there must be a set $S \in \mathcal{F}$ containing e such that $\tilde{x}^i_S \geq \frac{1}{f}$. Add the set S to the solution $\mathcal{R}$. Update $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S\}$, $W \leftarrow W - \sum_{e\in S\cap U} w_e$ and $U \leftarrow U \setminus S$ and go to step 3.

   (c) If none of the above two conditions hold, then at most one set $S \in \mathcal{F}$ has $0 < \tilde{x}^i_S < 1$. $\tilde{x}^i$ is integral for all other sets. Add S to the solution and for all other sets S', add S' to the solution iff $\tilde{x}^i_{S'} = 1$. Go to step 4

3. $i \leftarrow i + 1$. LPi is the LP-relaxation of the modified problem $(U, \mathcal{F}, W)$.

4. Output sets in $\mathcal{R}$.

To prove that the above algorithm is well defined, we need to show that in each iteration at least one of the conditions in steps 2a, 2b and 2c holds.

Lemma 4.4.2 *Let $(\tilde{x}^i, \tilde{z}^i)$ be a basic optimal solution of LPi in some iteration i. Then, either $i)$ there exists an element e such that $\tilde{z}_e = 0$ or $\tilde{z}_e = 1$, or $ii)$ at most one set S has $0 < \tilde{x}_S < 1$.*

Proof: The rank of the constraint matrix of LPi is at most $n+1$, where n is the number of elements in U in iteration i. Also, the number of variables in the relaxation LPi is $m+n$, where m is the number of sets in $\mathcal{S}$ in iteration i. Therefore, number of basic variables in the solution $(\tilde{x}^i, \tilde{z}^i)$ is at most $n+1$. If there is no $e \in U$ such that $\tilde{z}_e^i = 0$ or 1, then all $\tilde{z}_e^i, e \in U$ are basic. Therefore, at most one variable x_S^i corresponding to sets $S \in \mathcal{S}$ is basic which implies that at most one set has $0 < \tilde{x}_S^i < 1$. ∎

Now, we can prove that the iterative rounding algorithm is an f-approximation to the Independent 1-CCSCP, where f is the maximum number of sets that any element belongs to.

Theorem 4.4.3 *The iterative rounding algorithm is an f-approximation for* Independent 1-CCSCP.

Proof: In any iteration i of the algorithm, we add a set S to our set cover solution:

1. In step 2b, if we find an element $e \in U$ such that $\tilde{z}_e^i = 1$. Then, $\sum_{S\in\mathcal{S}, e\in S} \tilde{x}_S^i \geq 1$. Since each element occurs in at most f sets, at least one set S containing e must have $\tilde{x}_S^i \geq \frac{1}{f}$. The cost of S in LPi is $c_S \cdot \tilde{x}_S^i$ and we pay at most a factor f times that cost.

2. In step 2c, we add at most one fractional set in our solution. Since, all sets have cost at most OPT, we obtain a solution to the residual problem of iteration i that costs at most 2OPT.

Since $f \geq 2$ (wlog), the iterative rounding algorithm gives an f-approximation. ∎

The following corollary is immediate for the one-stage vertex cover problem and the spanning tree problem in the independent scenarios model by reducing the problems to the corresponding weighted partial covering versions.

Independent-Chance-Constrained Vertex Cover (Independent 1-CCVCP**).**

We are given a graph $G = (V, E)$ with costs on vertices and a reliability level ρ.. Each edge e occurs with probability p_e, independently of others. The objective is to find a minimum cost vertex cover C such that it covers ρ fraction of the scenarios (a scenario corresponds to a realization of the edges).

Independent-Chance-Constrained Shortest Path (Independent 1-CCSPP**).**

We are given a graph $G = (V, E)$ with costs on edges, a root vertex r and a reliability level ρ. Each vertex v occurs with probability p_v, independently of others. The objective is to find a minimum cost tree T containing the root r such that for ρ fraction of the scenarios there is a path in T from the root to each vertex in the scenario.

Corollary 4.4.4 *We obtain the following approximation guarantees for* Independent 1-CCVCP *and* Independent 1-CCSPP.

1. *There is a* 2*-approximation for* Independent 1-CCVCP.
2. *There is a* 5*-approximation for* Independent 1-CCSPP.

For the vertex cover problem, $f = 2$; therefore, we obtain a 2-approximation for Independent 1-CCVCP as a direct corollary to Theorem 4.4.3. From Theorem 4.4.1 we know that the Independent 1-CCSPP reduces to a weighted k-MST problem. Chudak et al. [14] give Lagrangian relaxation based 5-approximation for the unweighted k-MST problem that can be adapted to obtain a 5-approximation for the weighted version. This gives the result for Independent 1-CCSPP.

4.4.2 General Distribution Model

We consider an implicit model where scenarios come from a general distribution such that the *cumulative probability* of every demand-scenario can be computed efficiently and satisfies *strict-monotonicity* with respect to set inclusion. We show that a greedy algorithm gives a logarithmic approximation for the one-stage set cover problem in this model.

In the one-stage set cover problem in this model, we are given a set family $\mathcal{F}$, a universe of elements U and a reliability level ρ. The demand-uncertainty is specified by a probability distribution $P : 2^U \rightarrow [0, 1]$ (possibly a black-box) such that any subset $E \subset U$ occurs with probability $P(E)$. We further assume that P satisfies the following properties.

1. (**Efficiency**) Cumulative probability $F(E) = \sum_{E' \subset E} P(E')$ can be computed efficiently for any subset $E \subset U$.
2. (**Strict-Monotonicity**) For any $E_1, E_2 \subset U$, $E_1 \subsetneq E_2 \Rightarrow F(E_1) < F(E_2)$.

We obtain a logarithmic approximation for the set cover problem in this model using a greedy algorithm described below. Let OPT denote the cost of an optimal solution. We prune away all sets $S \in \mathcal{F}$ such that $c_S > \mathsf{OPT}$. Clearly, the modified instance is feasible. Also, let $S_{max} = \text{argmax}\{P(S)|S \in \mathcal{F}\}$. Since P is monotone, $p_{max} = P(S_{max}) > 0$. The algorithm is as follows:

1. Initialize $i \leftarrow 1, E_1 \leftarrow S_{max}$ and $\mathcal{C} \leftarrow \{S_{max}\}$.
2. While ($F(E_i) < \rho$)
   (a) Find a set $S \in \mathcal{F} \setminus \mathcal{C}$ that minimizes $\frac{c_S}{F(E_i \cup S) - F(E_i)}$.
   (b) Update $E_{i+1} \leftarrow E_i \cup S$ and $\mathcal{C} \leftarrow \mathcal{C} \cup S$.
   (c) Update $i \leftarrow i + 1$.

By a standard averaging argument, we obtain the following theorem.

Theorem 4.4.5 *The greedy algorithm gives an $O(\log \frac{\rho}{p_{max}})$-approximation for the one-stage set cover problem where uncertainty is given by a probability distribution (possibly a black-box) such that the cumulative probability F of any demand-scenario can be computed efficiently.*

Proof: In the greedy step i, we can find a set S_i such that $\frac{c_{S_i}}{F(E_i \cup S) - F(E_i)} \leq \frac{\mathsf{OPT}}{\rho - F(E_i)}$. Therefore,

$$\begin{aligned} c_{S_i} &\leq \mathsf{OPT} \frac{F(E_i \cup S_i) - F(E_i)}{\rho - F(E_i)} \\ &\leq \mathsf{OPT} \int_{x=F(E_i)}^{F(E_i \cup S_i)} \frac{1}{\rho - F(E_i)} dx \\ &\leq \mathsf{OPT} \int_{x=F(E_i)}^{F(E_i \cup S_i)} \frac{1}{\rho - x} dx \end{aligned}$$

Let the number of steps in the greedy algorithm be k. We know that due to the pruning step, $c_{S_k} \leq \mathsf{OPT}$ where S_k is set added in the k^{th} step. The total cost of the sets added in the first $(k-1)$ steps can be bounded as:

$$\begin{aligned} \sum_{i=1}^{k-1} c_{S_i} &\leq \sum_{i=1}^{k-1} \mathsf{OPT} \int_{x=F(E_i)}^{F(E_i \cup S_i)} \frac{1}{\rho - x} dx \\ &\leq \mathsf{OPT} \int_{x=p_{max}}^{\rho} \frac{1}{x} dx \\ &\leq \mathsf{OPT} \log \left(\frac{\rho}{p_{max}} \right) \end{aligned}$$

This proves the required approximation. ∎

5
Chance Constrained Knapsack Problem

We consider the following chance constrained knapsack problem: given n items, a knapsack size B and a reliability level $0 \leq \rho \leq 1$. Item i has a deterministic profit p_i and size S_i which is random from a known distribution and independent of the sizes of other items. The goal is to select a subset $\mathcal{S}$ of items that maximizes our profit such that $Pr(\sum_{i \in \mathcal{S}} S_i \leq B) \geq \rho$.

This problem is related to the stochastic knapsack problem considered in Dean et al. [16] where the authors consider the problem of finding an optimal policy (or ordering) to select the items that maximizes the profit while satisfying the knapsack constraint. The key difference with our model is the following. In the model in [16], the size of an item is instantiated when selected and the algorithm stops whenever we select an item that fills up the knapsack. On the other hand, we consider the problem of selecting a subset of elements that have a low probability of exceeding the knapsack size. The latter model is more appropriate in applications like project selection wherein we are required to decide today which projects should we invest in. The total investment in a project is known only during the course of the project. Thus, the problem is to decide on a subset of projects which have a low probability of exceeding our budget.

Normally distributed Sizes We consider the case when each item j has a normally distributed size with mean a_j and standard deviation σ_j independent of the other items. Let x_j denote whether item j has been selected or not. Then the stochastic knapsack problem can be formulated as follows:

$$
\begin{aligned}
\max \sum_{j=1}^{n} p_j \cdot x_j & & \\
\mathbb{P}(\sum_j S_j x_j \leq B) & \geq & \rho \\
x_j & \in & \{0,1\},\ \forall j = 1, \ldots, n
\end{aligned}
\tag{5.1}
$$

5.1 Second Order Conic and Parametric LP Formulations

When item sizes are normally distributed, we are able to rewrite the probabilistic constraint as a second-order cone constraint.

$$Pr(\sum_j S_j x_j \leq B) = Pr(\frac{\sum_j (S_j x_j - a_j x_j)}{\sum_j \sqrt{\sum_j \sigma_j^2 x_j^2}} \leq \frac{B - \sum_j a_j x_j}{\sqrt{\sum_j \sigma_j^2 x_j^2}})$$

Note that the random variable $\frac{\sum_j (S_j x_j - a_j x_j)}{\sum_j \sqrt{\sum_j \sigma_j^2 x_j^2}}$ is a standard normal variable with mean 0 and standard deviation 1. Let us denote this by Z. Also, let ϕ denote the cumulative distribution function of the standard normal variate. Therefore, the probabilistic constraint can be rewritten as

$$Pr(Z \leq \frac{B - \sum_j a_j x_j}{\sqrt{\sum_j \sigma_j^2 x_j^2}}) \geq \rho$$

$$\Rightarrow \frac{B - \sum_j a_j x_j}{\sqrt{\sum_j \sigma_j^2 x_j^2}} \geq \phi^{-1}(\rho)$$

where, $\phi^{-1}(\rho)$ is positive if $\rho > 0.5$. Thus, the chance-constraint can be simplified as,

$$\phi^{-1}(\rho)\sqrt{\sum_j \sigma_j^2 x_j^2} + \sum_j a_j x_j \leq B$$

The reformulation of the chance constrained knapsack problem with normally distributed item sizes is as follows:

$$\max \sum_{j=1}^{n} p_j \cdot x_j \tag{5.2}$$

$$\phi^{-1}(\rho)\sqrt{\sum_j \sigma_j^2 x_j^2} + \sum_j a_j x_j \leq B \tag{5.3}$$

$$x_j \in \{0,1\},\ \forall j = 1, \ldots, n \tag{5.4}$$

The relaxation (where integrality on $x_j, j = 1, \ldots, n$ is relaxed to $0 \leq x_j \leq 1$) is a second order cone program and can be solved in polynomial time. However, the integrality gap of the conic relaxation is $\Omega(\sqrt{n})$. Consider the following instance: $p_j = \sigma_j = 1, a_j = \frac{1}{\sqrt{n}} \forall j = 1, \ldots, n, B = 3, \rho = 0.95$. Any integral solution can have at most two items; therefore, the integral profit is at most 2. Whereas, consider the fractional solution $x_j = \frac{1}{\sqrt{n}}$. Then,

$$\sum_{j=1}^{n} a_j x_j + \phi^{-1}(\rho) \sqrt{\sum_{j=1}^{n} \sigma_j^2 x_j^2} = 1 + \phi^{-1}(\rho) < 3$$

Therefore, the fractional solution is feasible which shows that the integrality gap of the conic formulation is $\Omega(\sqrt{n})$.

Parametric LP Reformulation. We reformulate the second order conic constraint (5.3) as a parametric LP and obtain a fully polynomial time approximation scheme for the chance constrained knapsack problem.

Suppose we know that the sum of mean sizes of the items selected in an optimal solution is μ^*. Then, the conic constraint (5.3) can be expressed as,

$$\begin{aligned} \sum_j a_j x_j &\leq \mu^* \\ \phi^{-1}(p)^2 (\sum_j \sigma_j^2 x_j^2) &\leq (B - \mu^*)^2 \end{aligned} \tag{5.5}$$

Since $x_j^2 = x_j$ for $x_j \in \{0, 1\}$, we can simplify constraint (5.5) as

$$(\phi^{-1}(p))^2 (\sum_j \sigma_j^2 x_j) \leq (B - \mu^*)^2$$

Therefore, we can formulate the chance constrained knapsack problem as the following 2-dimensional knapsack problem where μ is the parameter corresponding to the total mean size of the selected items.

$$\max_{j=1}^{n} p_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq \mu \tag{5.6}$$

$$\phi^{-1}(p)^2 (\sum_{j=1}^{n} \sigma_j^2 x_j) \leq (B - \mu)^2 \tag{5.7}$$

$$x_j \in \{0, 1\}$$

5.2 A $(1+\epsilon)$-approximation Algorithm

We present a full polynomial time approximation scheme for the chance-constrained knapsack problem using the parametric 2-dimensional knapsack reformulation described above. We consider powers of $(1+\epsilon)$ i.e. $(1+\epsilon)^j, j = 0, \ldots, \log_{(1+\epsilon)} B$ for some constant $\epsilon > 0$ as different choices of the parameter μ. Therefore, the number of different choices of μ is $O(\frac{\log B}{\epsilon})$ which is polynomial in the input size.

We also guess the value of optimal profit OPT by considering powers of $(1+\epsilon)$. Let $P = \sum_{j=1}^{n} p_j$; we consider $O(\frac{\log P}{\epsilon})$ different choices of OPT. At most $\frac{1}{\epsilon}$ items can have profit greater than $\epsilon\mathsf{OPT}$. Therefore, for each guess of $\mathsf{OPT} = (1+\epsilon)^j$ we consider all subsets of size at most $\frac{1}{\epsilon}$ of the items that have size more than $\epsilon\mathsf{OPT}$ to include in the solution. For each guess O of OPT and each choice of subset of items of size more than $\epsilon\mathsf{OPT}$, we solve a subproblem $\Pi(S_1, S_2, O)$.

$\Pi(S_1, S_2, O)$. We are given sets of items $S_1, S_2 \subset [n]$ such that each item in S_1 has profit at most $\epsilon \cdot O$ and each item in S_2 has profit at least $\epsilon \cdot O$. Furthermore, all items in S_2 are included in our final solution. Our goal is to choose a subset of items from S_1 that together with items in S_2 maximize the total profit while satisfying the chance-constraint 5.1.

In order to solve the subproblem $\Pi(S_1, S_2, O)$, we formulate a further subproblem $\Pi(S_1, S_2, O, \mu)$ where the total mean size of all items selected from S_1 is at most μ. Therefore, we can formulate $\Pi(S_1, S_2, O, \mu)$ as the following 2-dimensional knapsack problem.

$$\max \sum_{j \in S_1} p_j x_j + \sum_{j \in S_2} p_j$$

$$\sum_{j \in S_1} a_j x_j \leq \mu \tag{5.8}$$

$$\phi^{-1}(p)^2 \Big(\sum_{j \in S_1} \sigma_j^2 x_j\Big) \leq (B - \mu - \sum_{j \in S_2} \mu_j)^2 - \phi^{-1}(p)^2 \sum_{j \in S_2} \sigma_j^2) \tag{5.9}$$

$$x_j \in \{0, 1\}$$

The algorithm $\mathcal{A}$ for the chance-constrained knapsack problem and the algorithm $\mathcal{A}(\Pi)$ for the subproblem $\Pi(S_1, S_2, O)$ are described in Figures 5.2 and 5.2 respectively.

In the following lemma, we show that we can find a good integral solution to the problem $\Pi(S_1, S_2, O, \mu)$.

Lemma 5.2.1 *Consider the problem $\Pi(S_1, S_2, O, \mu)$ such that $p_j \leq \epsilon \cdot O$ for all $j \in S_1$. If P^* is the optimal profit for $\Pi(S_1, S_2, O, \mu)$, then there is a polynomial time algorithm to find a feasible set of items whose profit is at least $(P^* - 2\epsilon \cdot O)$.*

Algorithm $\mathcal{A}$ for Chance-constrained Knapsack Problem
Given n items where item j has profit p_j and a normally distributed size with mean a_j and standard deviation σ_j, knapsack size B, reliability level ρ and a constant $\epsilon > 0$. Let $p_m = \min_{j\in[n]} p_j, P = \sum_{j\in[n]} p_j$.

Initialize $N_1 = \lfloor \log_{1+\epsilon} p_m \rfloor, N_2 = \lceil \log_{1+\epsilon} P \rceil, x_{\mathcal{A}} = 0, P_{\mathcal{A}} \leftarrow 0$.

1. For $t = N_1, \ldots, N_2$,
   (a) Let $O = (1+\epsilon)^t$ and let $S_\epsilon = \{j \in [n] | p_j \geq \epsilon \cdot O\}$.
   (b) For each set $S \subset S_\epsilon$ such that $|S| < \frac{1}{\epsilon}$,
      i. Solve $\Pi([n] \setminus S_\epsilon, S, O)$ and let x_S denote the integral solution returned by $\mathcal{A}(\Pi)$.
      ii. If $P_{\mathcal{A}} < p^T x_S$, then

$$\begin{aligned} x_{\mathcal{A}} \leftarrow & \quad x_S \\ P_{\mathcal{A}} \leftarrow & \quad p^T x_S \end{aligned}$$

2. Return the solution $x_{\mathcal{A}}$.

Figure 5.1: Algorithm for Chance Constrained Knapsack Problem

Proof: Consider the 2-dimensional knapsack formulation of $\Pi(S_1, S_2, O, \mu)$ and consider the basic optimal solution $\tilde{x}$ of the LP relaxation. Since there are only two constraints other than the bound constraints, at least $(|S_1| - 2)$ bound constraints must be tight for $\tilde{x}$. Therefore, at least $(|S_1| - 2)$ variables out of $|S_1|$ variables are integral in the basic optimal solution. Let $j_1, j_2 \in S$ such that $\tilde{x}_{j_1}, \tilde{x}_{j_2}$ are fractional. We know that $p_j \leq \epsilon \cdot O$ for all $j \in S_1$. Consider the following solution,

$$\hat{x}_j = \begin{cases} \tilde{x}_j & j \in S_1, j \neq j_1, j_2 \\ 0 & \text{otherwise} \end{cases}$$

Clearly, the solution $\hat{x}$ is feasible since we only rounded down. Also,

$$\sum_{j\in S_1} p_j \hat{x}_j \geq \sum_{j \in S_1} p_j \tilde{x}_j - 2\epsilon \cdot O.$$

Therefore, we obtain an integral solution $\hat{x}$ such that $\sum_{j\in S_1} p_j \hat{x}_j + \sum_{j\in S_2} p_j \geq P^* - 2\epsilon \cdot O$. ■

In the following lemma we show that for an appropriately chosen value of O and μ and subsets $S_1, S_2 \subset [n]$, the problem $\Pi(S_1, S_2, O, \mu)$ has optimal profit at least $\frac{\mathsf{OPT}}{1-\epsilon}$.

Algorithm $\mathcal{A}(\Pi)$ **for** $\Pi(S_1, S_2, O)$

Let $\mu_{min} = \min_{j \in S_1} \mu_j$.
Initialize $N_l = \lfloor \log_{1+\epsilon} \mu_{min} \rfloor, N_h = \lceil \log_{1+\epsilon} B \rceil, x_s = 0, P_s \leftarrow 0$.

1. For $t = N_l, \ldots, N_h$,
   (a) Let $\mu = (1+\epsilon)^t$ and let $\tilde{x}(\mu)$ be a basic optimal solution for $\Pi(S_1, S_2, O, \mu)$.
   (b) Using Lemma 5.2.1 find an integral solution $\hat{x}(\mu)$ such that
   $$\sum_{j=1}^{n} p_j \cdot \hat{x}(\mu)_j \geq \sum_{j=1}^{n} p_j \cdot \tilde{x}(\mu)_j - 2\epsilon \cdot O$$
   (c) If $P_s < \sum_{j \in S_1} p_j \hat{x}(\mu)_j + \sum_{j \in S_2} p_j$, then
   $$\begin{aligned} x_s &\leftarrow \hat{x}(\mu) \\ P_s &\leftarrow \textstyle\sum_{j \in S_1} p_j \hat{x}(\mu)_j + \sum_{j \in S_2} p_j \end{aligned}$$
2. Return the solution x_s.

Figure 5.2: Algorithm for $\Pi(S_1, S_2, O)$

Lemma 5.2.2 *Let S^* be the set of items selected by an optimal solution and let $\mathsf{OPT} = \sum_{i \in S^*} p_i$. Consider l such that $(1+\epsilon)^{l-1} \leq \mathsf{OPT} < (1+\epsilon)^l$. Let $O = (1+\epsilon)^l$ and let $S_\epsilon = \{i \in [n] | p_i \geq \epsilon \cdot O\}, S_1 = [n] \setminus S_\epsilon, S_2 = S_\epsilon \cap S^*$. Then the optimal profit for the problem $\Pi(S_1, S_2, O)$ is at least $\frac{\mathsf{OPT}}{1+\epsilon}$.*

Proof: Let $\mu^* = \sum_{j \in S^*} \mu_j, \nu_1 = \sum_{j \in S_1 \cap S^*} \mu_j, \nu_2 = \sum_{j \in S_2} \mu_j$ and let k be such that $(1+\epsilon)^{k-1} \leq \nu_1 < (1+\epsilon)^k$. Let $\beta = (1+\epsilon)^{k-1}$ and we consider the problem $\Pi(S_1, S_2, O, \beta)$. Consider the following fractional solution $\tilde{x}$ for $\Pi(S_1, S_2, O, \beta)$:

$$\tilde{x}_j = \begin{cases} \frac{1}{1+\epsilon} & j \in S_1 \cap S^* \\ 0 & \text{otherwise} \end{cases}$$

We show that $\tilde{x}$ is a feasible fractional solution for the 2-dimensional knapsack formulation of $\Pi(S_1, S_2, O, (1+\epsilon)^{k-1})$. Consider inequalities (5.8),

$$\begin{aligned}
\sum_{i \in S_1} \mu_j \tilde{x}_j &= \sum_{j \in S_1 \cap S^*} \mu_j \cdot \frac{1}{\epsilon} && (5.10)\\
&= \frac{\nu_1}{1+\epsilon} && (5.11)\\
&\leq \beta && (5.12)
\end{aligned}$$

Therefore, $\tilde{x}$ satisfies inequality (5.8). Let $\theta = \phi^{-1}(p)$. Consider inequality (5.9),

$$\begin{aligned}
\theta^2 \cdot (\sum_{j \in S_1} \sigma_j^2 \tilde{x}_j) &= \theta^2 \cdot (\sum_{j \in S_1 \cap S^*} \sigma_j^2 \frac{1}{1+\epsilon}) && (5.13)\\
&\leq \frac{\left((B-\mu^*)^2 - \theta^2 \cdot \sum_{j \in S_2} \sigma_j^2\right)}{1+\epsilon} && (5.14)\\
&= \frac{\left((B-\nu_1-\nu_2)^2 - \theta^2 \cdot (\sum_{j \in S_2} \sigma_j^2)\right)}{1+\epsilon} && (5.15)\\
&\leq \frac{\left((B-\beta-\nu_2)^2 - \theta^2 \cdot (\sum_{j \in S_2} \sigma_j^2)\right)}{1+\epsilon} && (5.16)\\
&< (B-\beta-\nu_2)^2 - \theta^2 \cdot (\sum_{j \in S_2} \sigma_j^2) && (5.17)
\end{aligned}$$

Here inequality (5.14) follows as $S^* = (S_1 \cap S^*) \cup S_2$ is an optimal solution and thus, satisfies

$$\phi^{-1}(p)^2 \cdot (\sum_{j \in S^*} \sigma_j^2) \leq (B-\mu^*)^2$$

and inequality (5.16) follows as $\beta \leq \nu_1$. This implies that $\tilde{x}$ satisfies inequality (5.9) as well and thus, is a feasible solution for $\Pi(S_1, S_2, O, \beta)$. The profit achieved by the fractional solution $\tilde{x}$ is

$$\begin{aligned}
\sum_{j \in S_1} p_j \tilde{x}_j + \sum_{j \in S_2} p_j &= \sum_{j \in S_1 \cap S^*} \frac{p_j}{1+\epsilon} + \sum_{j \in S_2} p_j && (5.18)\\
&> \frac{\sum_{j \in S_1 \cap S^*} p_j + \sum_{j \in S_2} p_j}{1+\epsilon} && (5.19)\\
&= \frac{\mathsf{OPT}}{1+\epsilon} && (5.20)
\end{aligned}$$

where the last equality follows because $S^* = (S_1 \cap S^*) \cup S_2$. Therefore, the optimal value for the problem $\Pi(S_1, S_2, O)$ is at least $\frac{\mathsf{OPT}}{1+\epsilon}$. ■

We now show that for any $\epsilon > 0$, the algorithm $\mathcal{A}$ gives a $(1-3\epsilon)$-approximation for the chance-constrained knapsack problem in running time $\tilde{O}(\frac{n^{\frac{1}{\epsilon}}}{\epsilon^2})$.

Theorem 5.2.3 *Given $\epsilon > 0$, there is a polynomial time algorithm that gives a $(1-3\epsilon)$-approximation for the chance constrained knapsack problem. Furthermore, the running time of $\mathcal{A}$ is*

$$O\left(\frac{\log(B/\mu_m)\cdot\log(P/p_m)\cdot n^{\frac{1}{\epsilon}}}{\epsilon^2}\right),$$

where $P = \sum_{j\in[n]} p_j, p_m = \min_{j\in[n]} p_j, \mu_m = \min_{j\in[n]} \mu_j$.

Proof: Let OPT denote an optimal solution and let S^* be the set of items selected in OPT. Consider l such that $(1+\epsilon)^{l-1} \leq \mathsf{OPT} < (1+\epsilon)^l$ and let $O = (1+\epsilon)^l$. Let $S_\epsilon = \{i \in [n] | p_i \geq \epsilon \cdot O\}, S_1 = [n] \setminus S_\epsilon$ and $S_2 = S \cap S^*$. Note that the algorithm $\mathcal{A}$ considers the guess O for the optimal value. Also, since $|S_2| < \frac{1}{\epsilon}$ the subproblem $\Pi(S_1, S_2, O)$ is considered as one of the subproblems in the algorithm $\mathcal{A}$. Let $\mu^1 = \sum_{j\in S_1\cap S^*} \mu_j$. Consider k such that $(1+\epsilon)^{k-1} \leq \mu^1 < (1+\epsilon)^k$ and let $\beta = (1+\epsilon)^{k-1}$. Clearly, the subproblem $\Pi(S_1, S_2, O, \beta)$ is considered in the algorithm $\mathcal{A}(\Pi)$ while solving $\Pi(S_1, S_2, O)$. From Lemma 5.2.2, we know that the optimal profit for the subproblem $\Pi(S_1, S_2, O, \beta)$ is at least $\frac{\mathsf{OPT}}{1+\epsilon}$. Furthermore, using Lemma 5.2.1 we can find a set of items $\hat{S}$ for the problem $\Pi(S_1, S_2, O, \beta)$ such that,

$$\sum_{j\in\hat{S}} p_j \geq \frac{\mathsf{OPT}}{1+\epsilon} - 2\epsilon\cdot O \tag{5.21}$$

$$\geq \left(\frac{1}{1+\epsilon} - 2\epsilon\right)\cdot\mathsf{OPT} \tag{5.22}$$

$$\geq (1-3\epsilon)\cdot\mathsf{OPT} \tag{5.23}$$

Therefore, the algorithm $\mathcal{A}$ finds an integral solution that has profit at least $(1-3\epsilon)\cdot\mathsf{OPT}$.

Running time of $\mathcal{A}$. Note that we consider $O(\frac{\log\left(\frac{P}{p_m}\right)}{\epsilon})$ different choices of the optimal profit value O, where $P = \sum_{j=1}^n p_j, p_m = \min_{j=1}^n p_j$. Also, we consider $O(n^{\frac{1}{\epsilon}})$ choices of the set of items S for the subproblem Π for each choice of O. Furthermore, in the subroutine $\mathcal{A}(\Pi)$, we solve $O(\frac{\log\left(\frac{B}{\mu_m}\right)}{\epsilon})$ different subproblems for solving one problem $\Pi(S_1, S_2, O)$ for given subsets $S_1, S_2 \subset [n]$ and a choice for optimal profit O. Therefore, the total running time of $\mathcal{A}$ is

$$O\left(\frac{\log\left(B/\mu_m\right)\cdot\log\left(P/p_m\right)\cdot n^{\frac{1}{\epsilon}}}{\epsilon^2}\right).$$

■

6 Locating Emergency Facilities for Post-Disaster Logistics

In this chapter, we consider a facility location problem that aims to locate emergency response and distribution centers (ERDC) for effective post-disaster operations such as supply of relief commodities to the affected areas in the event of a disaster such as an earthquake. Post disaster operations are faced with many challenges such as an uncertain demand and resources available due to the highly uncertain nature of the manifesting disaster and its impact. Even the infrastructure such as the transport network and the communication network available for post-disaster logistics is uncertain as some of it could have been significantly damaged or disrupted in the disaster. While the parameters are highly uncertain, the relief operations typically need to be carried out within a very short time period after the disaster. Therefore, in wake of these uncertainties it is essential to plan ahead for effective post disaster logistics. While in previous chapters in this thesis, we have considered models where either there is uncertainty in demand (see Chapters 2 and 3) or there is uncertainty in the data (see Chapter 5), the problem of locating ERDC combines aspects of both demand and data uncertainty.

6.1 Problem Description

We study the problem of locating Emergency Response and Distribution Centers (ERDC) in and around a region with seismic risk such that in the event of an earthquake, relief commodities such as water, medical aid and food can be distributed to the affected areas within a short time. Since the transport and communication networks can be disrupted in the event of an earthquake, the given set of ERDC may or may not be able to reach the affected areas within the required time depending on the impact of the disaster. Similarly, the demand for relief commodities in the affected areas is known only after the disaster. We refer to the post-disaster network and demand realizations as a *disaster scenario*. To consider opening a set of

Emergency Response and Distribution Centers (ERDC) such that all the demand is covered in all the disaster scenarios might be lead to a very expensive and conservative solution. Therefore, a better solution is to consider opening ERDC such that more than 99% or some other threshold of the disaster scenarios are covered by the ERDC. Such a solution ignores the very unlikely worst case scenarios but significantly reduces the cost of the solution to make it practical to implement. Therefore, the problem of optimally locating ERDC with a constraint that 99% of the disaster scenarios are covered is a chance-constrained optimization problem. The chance-constrained optimization problem is extremely hard to solve both computationally as well as theoretically even for a small number of scenarios as discussed in Chapter 4. When the number of scenarios are exponential in the size of the input (as is the case with the number of different transport network realizations in our problem), it is not even clear to check efficiently whether a given solution satisfies the chance-constraint or not. In this chapter, we focus on excatly this problem and give an efficient sampling based algorithm to approximately answer this question.

In this study we focus on the case of Istanbul, Turkey where seismic risk is a major concern. The Municipality of Istanbul is interested in opening ERDC around the city of Istanbul which will be used as coordination centers for distribution of relief commodities to the affected areas in the event of an earthquake. As we discussed earlier, the infrastructure available for relief operations depends on the impact of the disaster. Therefore, ideally we would like to open ERDC in locations of low seismic risk while still being close to the high seismic risk regions to distribute relief commodities to the affected areas in a short time. The Municipality has identified a set of 40 potential locations for opening ERDC that satisfy the above two conditions among other logistics constraints.

In a post-disaster scenario, the time for the relief to reach the affected areas is probably the most critical factor. The post-disaster time frame is typically divided into the first $4, 8, 12, 16, 24$ hours and so on as the services required and the chances of saving lives are different in each interval. For example, the medical first-aid and search-and-rescue teams are most critical in the first 4 or 8 hours while food supply is not that critical in the first 8 hours. Therefore, the problem of coordinating distribution of relief commodities in a post-disaster scenario is a multi-period, multi-commodity problem. For the sake of simplification, we consider a single period, single commodity distribution problem where there is a constraint of reaching all the affected areas within a given time from at least one of the open ERDC.

Since the transport network is vulnerable to the earthquake, some of the transport links may be disrupted due to the impact of the disaster. Therefore, the transport network in a post-disaster scenario is uncertain and is known only after the occurrence of the disaster. We would like to open ERDC such that in a large fraction of disaster scenarios (i.e. transport network and demand realizations), all the affected areas can be reached from some ERDC within a given time bound. Note that there are exponentially many different transport network realizations possible in a post-disaster scenario. Since the chance-constrained problem of optimally locating a set of ERDC such that a large fraction of post-disaster scenarios are covered is difficult to solve both computationally as well as theoretically in lieu of the discussion in Chapter 4, we consider the problem of estimating the fraction of disaster scenarios that are covered by a given set of ERDC. As the municipality is interested in opening only a few ERDC, it is not difficult to enumerate all possible choices of ERDC to open from the set of 40 potential locations. In particular, we consider the following two estimation problems.

Reliability(F). Given a set of open ERDC or facilities F, estimate the fraction of disaster scenarios (i.e. transport network and demand realizations) in which there is an open facility in F within a given distance bound of each demand location (or the affected area).

Max-Coverage(F). Given a set of open ERDC or facilities F, estimate the average fraction of demand satisfied by F over different disaster scenarios. A demand location in a disaster scenario is satisfied by F if there is a facility in F which is within a given distance bound of the demand location.

We also consider capacitated versions of the above problems referred to as Cap-Reliability(F) and Cap-Max-Coverage(F) where each open facility f has a given capacity which is the total demand that can be satisfied by resources at f. Since we consider only the single-commodity case, the capacity at a facility f can be thought of as the inventory level of the commodity at f. Similarly, the demand at a location in an area affected by the disaster is measured in terms of the quantity of the commodity required at that location in the post-disaster scenario.

We propose an efficient sampling based estimation algorithm for the versions of the problem above where disaster scenarios are sampled from a probabilistic link-failure model given an earthquake has occurred. Each sampled problem is modeled as a length bounded flow problem. We conduct our computational experiments using the data for the case of Istanbul. We would like to note here that the focus of this study is not to accurately model the transport link-failure probabilities in the event of an earthquake but show how sampling can be used to accurately estimate several quantities such as reliability and coverage for a set of open facilities. However, we make our best possible effort to use a reasonable probabilistic model for scenario generation in the computational experiments in this study. Furthermore, neither the computational complexity nor the accuracy of the results from our sampling algorithm are dependent on the probabilistic model.

6.2 Related Work

One of the earliest studies conducted on location of emergency facilities is due to Toregas et al. [59]. The facility location problem is modeled as a set covering problem where the affected areas are represented as demand nodes and the potential facilities are referred as supply nodes and the objective is to minimize the maximum time/distance of a demand node to its closest supply node. Haghani and Oh [32] consider a multi-commodity, multi-modal network flow model with time windows for disaster response where they assume that both the supply and demand for all the commodities is known in advance. The authors propose heuristics to solve the problem where violations in time windows are allowed and include a penalty in the objective.

Ozdamar et al. [61] analyze the problem of dispatching the commodities to distribution centers as a part of emergency logistics planning for the Marmara region. They focus on the problem of planning a detailed distribution schedule subject to vehicle capacity constraints and conduct a computational study for the case of Marmara earthquake in 1999. Yi and Ozdamar [60] consider a dynamic and fuzzy logistics coordination model for post-disaster logistics which incorporates the uncertainty in demand and supply. Dekle et al. [17] consider the problem of locating emergency response facilities in Florida that will be used by Federal Emergency Management Agency (FEMA) for post-disaster logistics. The reader is referred to [34] for a review of facility location models for emergency response. The review considers three broad models for the facility location problem: covering models, k-median and k-center.

While some of the models discussed above incorporate uncertainty in demand and supply in a post-disaster scenario, they do not consider uncertainties in the transport network which is very likely to be affected by the disaster. The most relevant work that simultaneously considers uncertainty in demand, supply and the underlying transport network is due to Barbarosoglu and Arda [3]. The authors model the uncertainty in demand, supply and transport network using a set of explicit disaster-scenarios and propose a two-stage stochastic programming solution approach. Since the number of all possible disaster scenarios is very large, Barbarosoglu and Arda [3] do not consider a complete list of scenarios and thus, it is not clear how to measure the quality of the solution over the complete set of disaster scenarios. Our work on the other hand gives an efficient sampling based algorithm to estimate the quality of the solution over the complete set of disaster scenarios.

Outline. The rest of the chapter is organized as follows. In Section 6.3 we give the mathematical formulation for the two problems Reliability(F) and Max-Coverage(F) that have been introduced earlier. We describe the sampling algorithm in Section 6.4 and finally present the computational experiment and results in Section 6.5.

6.3 Facility Location Models for ERDC

In this section, we build a mathematical model for the problem of locating ERDC subject to various logistics constraints. Let $\mathcal{F}$ denote the set of potential locations for opening ERDC and let $\pi : \mathcal{F} \to \mathbb{R}_+$ denote the ERDC opening costs. Let $\mathcal{D}$ denote the set of all possible affected areas or demand locations and let $G = (V, A)$ be the directed graph that represents the transport network and the length of each arc is given by the function $l : A \to \mathbb{R}_+$. Let $c : \mathcal{F} \to \mathbb{R}_+$ denote the capacities installed at ERDC in $\mathcal{F}$. As discussed earlier, the Municipality of Istanbul has selected a set of 40 potential locations for opening ERDC. Therefore, for the case of Istanbul $|\mathcal{F}| = 40$. Similarly, the demand locations correspond to different districts in and around Istanbul.

The actual transport network available after the disaster as well as the demand at each location depends on the impact of the disaster. Let $x_S \in \{0,1\}^A$ be a 0-1 vector that denotes the post-disaster network realization in scenario S where $x_S(e) = 1$ if $e \in A$ survives in scenario S and 0 otherwise. Also, let $d_S(j)$ denote the demand at location $j \in \mathcal{D}$ in scenario S.

We now describe a chance-constrained model for the problem of optimally locating a set of ERDC such that for a ρ fraction of disaster-scenarios, each demand location is within a distance B from some open facility. Let y_i be a binary variable that denotes whether a facility is opened at $i \in \mathcal{F}$ or not. We model the problem as a multi-commodity flow problem where each demand point j sends a flow equal to $d_S(j)$ in scenario S to some open facility within a distance B. Let $f_S^j(e)$ denote the flow from demand point j on edge e in scenario S. For each scenario S, let z_S be a binary variable denoting whether all the demand points are covered in scenario S or not. Let $\delta^+(j) = \{(i,j) | (i,j) \in A\}$ and $\delta^-(j) = \{(j,i) | (j,i) \in A\}$.

$$\min \sum_{i \in \mathcal{F}} \pi_i y_i$$

$$\sum_{e \in \delta^-(j)} f_S^j(e) - \sum_{e \in \delta^+(j)} f_S^j(e) \geq d_S(j) \cdot z_S \quad \forall j, \forall S \tag{6.1}$$

$$\sum_{e \in \delta^-(v)} f_S^j(e) - \sum_{e \in \delta^+(v)} f_S^j(e) = 0 \quad \forall v \neq j, \forall j, \forall S \tag{6.2}$$

$$f_S^j(e) \leq x_S(e) \quad \forall e \in A, \forall S \tag{6.3}$$

$$\sum_{j \in \mathcal{D}} \sum_{e \in \delta^+(i)} f_S^j(e) - \sum_{e \in \delta^-(i)} f_S^j(e) \leq y_i \cdot c_i \quad \forall i \in \mathcal{F}, \forall S \tag{6.4}$$

$$\sum_{e \in A} l_e f_S^j(e) \leq B \quad \forall j, \forall S \tag{6.5}$$

$$\sum_{S} p_S z_S \geq \rho \tag{6.6}$$

$$z_S \in \{0,1\} \quad \forall S$$

$$y_i \in \{0,1\} \quad \forall i \in \mathcal{F}$$

Constraint (6.1) requires that each demand point j sends out a flow of $d_S(j)$ in scenario S if S is covered by the open set of facilities. Constraint (6.2) enforces flow conservation while (6.3) enforces that there is a flow on arc e in scenario S only if e survives in scenario S. Constraint (6.4) requires that a potential facility is a sink for any flow only if it is open and satisfies a total demand which is not more than its capacity. Constraint (6.5) bounds the length of the flow paths and Constraint (6.6) ensures that we cover at least ρ fraction of the disaster-scenarios.

However, there is a problem with the above formulation. The number of disaster scenarios are exponential and therefore, the size of the formulation (number of variables and constraints) is too large to be able to solve practically. The IP formulation for the problem of locating ERDC such that the expected fraction of demand covered over all scenarios is maximized faces a similar problem of exponential number of variables and constraints. In fact, given a set of open facilities F it is not even clear how to check whether it is covers a ρ fraction of the disaster scenarios or how to compute the expected fraction of demand covered by F over all scenarios. We focus on these estimation problems given a set of open facilities and give an efficient sampling based additive ϵ approximation for these.

Reliability(F). Given a set of open facilities F, the problem is to determine the fraction of scenarios S (where each scenario is a post-disaster demand and network realization) such that each demand can be satisfied by some open facility within a distance B.

Max-Coverage(F). Given a set of open facilities F, the problem is to determine the expected fraction of demand over all scenarios S that can be satisfied by the set of facilities F.

To compute the estimates in the above two problems, we solve the scenario problems for a number of sampled scenarios. Let us consider the scenario versions of these problems.

Scenario version Reliability(F, S) Given a set of open facilities F and a disaster scenario S which defines the post-disaster transport network and demand realization, the problem is to determine whether all the demand can be satisfied by some open facility in F within a distance B.

The uncapacitated version of Reliability(F, S) where the facilities do not have any capacity constraints can be solved by a shortest path computation from each demand point to the closest open facility. The scenario S is feasible if all the demand points are within a distance B from some open facility in F. The capacitated version where each facility $i \in F$ has a capacity μ_i for the demand it can serve, can be formulated as a length bounded flow feasibility problem which is described below.

$$\sum_{e\in\delta^-(j)} f_S^j(e) - \sum_{e\in\delta^+(j)} f_S^j(e) \geq d_S(j) \quad \forall j \tag{6.7}$$

$$\sum_{e\in\delta^-(v)} f_S^j(e) - \sum_{e\in\delta^+(v)} f_S^j(e) = 0 \quad \forall v \neq j, \forall j \tag{6.8}$$

$$f_S^j(e) \leq x_S(e) \quad \forall e \in A \tag{6.9}$$

$$\sum_{j\in\mathcal{D}} \sum_{e\in\delta^+(i)} f_S^j(e) - \sum_{e\in\delta^-(i)} f_S^j(e) \leq \mu_i \quad \forall i \in F \tag{6.10}$$

$$\sum_{e\in A} l_e f_S^j(e) \leq B \quad \forall j \tag{6.11}$$

Scenario Version Max-Coverage(F, S). Given a set of open facilities F and a disaster scenario S, the problem is to determine maximum fraction of demand that can be satisfied by the set of facilities F. This can be formulated as max-flow problem subject to length bound and capacity constraints. The formulation is given below.

$$\max \sum_{j \in \mathcal{D}} y_S(j)$$

$$\sum_{e \in \delta^-(j)} f_S^j(e) - \sum_{e \in \delta^+(j)} f_S^j(e) \geq y_S(j) \quad \forall j \tag{6.12}$$

$$y_S(j) \leq d_S(j) \quad \forall j \tag{6.13}$$

$$\sum_{e \in \delta^-(v)} f_S^j(e) - \sum_{e \in \delta^+(v)} f_S^j(e) = 0 \quad \forall v \neq j, \forall j \tag{6.14}$$

$$f_S^j(e) \leq x_S(e) \quad \forall e \in A \tag{6.15}$$

$$\sum_{j \in \mathcal{D}} \sum_{e \in \delta^+(i)} f_S^j(e) - \sum_{e \in \delta^-(i)} f_S^j(e) \leq \mu_i \quad \forall i \in F \tag{6.16}$$

$$\sum_{e \in A} l_e f_S^j(e) \leq B \quad \forall j \tag{6.17}$$

6.4 Sampling Algorithm

In this section, we describe a generic sampling based estimation algorithm that can be used to solve the estimation problem of capacitated and uncapacitated versions of Reliability(F) and Max-Coverage(F) for a given set of facilities F. Let $\Pi(F)$ denote the estimation problem for the set of facilities F. Also, let $\Pi_S(F)$ denote the corresponding problem for a sample scenario S where the scenario S is a particular demand and transport network realization after the disaster. For instance, if $\Pi =$ Reliability then $\Pi_S(F)$ denote the problem of determining whether scenario S is covered by the set of facilities F or not. Let $X_S(F)$ denote the optimal value of $\Pi_S(F)$ and let V be an upper bound on the variance of the optimal value of $\Pi_S(F)$ across different scenarios. Let us first consider a basic sampling algorithm which is described in Figure 6.4.

We show in the following theorem that the sampling algorithm $\mathcal{A}$ outputs an estimate for $\Pi(F)$ which is an additive ϵ-approximation with high probability.

Theorem 6.4.1 *Suppose the estimate for each sample in algorithm $\mathcal{A}_0$ to compute the $\Pi(F)$ is bounded in $[a, b]$. Let ρ^* denote the true estimate for $\Pi(F)$ and let σ^2 be the variance. For any constants $\epsilon, \delta > 0$, if the number of samples*

$$N = \frac{\sigma^2}{\delta \epsilon^2},$$

then the estimate, X returned by $\mathcal{A}_0$ satisfies

$$\mathbb{P}(|X - \rho^*| > \epsilon) \leq \delta$$

Sampling Algorithm $\mathcal{A}_0$ for $\Pi(F)$
Given a set of facilities F, $\epsilon > 0$ and $\delta > 0$, the problem is to estimate $\Pi(F)$.

Initialize $N \leftarrow \frac{V}{\delta\epsilon^2}, X \leftarrow 0$

1. For $j = 1, \ldots, N$,
   (a) Generate sample scenario S_j and let X_j be the optimal value of $\Pi_{S_j}(F)$.
   (b) $X \leftarrow X + \frac{X_j}{N}$.
2. Return X as estimate for $\Pi(F)$.

Figure 6.1: Sampling Algorithm $\mathcal{A}_0$

Proof: Let X_i denote the estimate of the i^{th} sample. We know that $E[X_i] = \rho^*$ and $\mathsf{Var}(X_i) = \sigma^2$. If the sampling algorithm considers N samples and,

$$X = \frac{1}{N}(X_1 + \ldots + X_N),$$

then $E[X] = \rho^*$ and $\mathsf{Var}(X) = \frac{\sigma^2}{N}$ as the samples are independent. Using the Chebyshev's inequality, we have

$$\mathbb{P}(|X - \rho^*| \geq \epsilon) \leq \frac{\sigma^2}{N\epsilon^2}$$

Therefore, if $N = \frac{\sigma^2}{\delta\epsilon^2}$ then $\mathbb{P}(|X - \rho^*| \geq \epsilon) \leq \delta$. ■

For the problems Reliability and Max-Coverage, the variance of the estimate across all samples is bounded by 1. Therefore, to achieve an additive error of at most ϵ with probability at least $(1 - \delta)$, the required number of samples $N = \frac{1}{\delta\epsilon^2}$.

6.4.1 Improved Sampling Algorithm

We now present an improved sampling algorithm that requires a significantly smaller number of samples to achieve an additive ϵ-approximation in the estimate with probability at least $(1 - \delta)$ for some $\epsilon, \delta > 0$. The algorithm is described in Figure 6.4.1. This algorithm is adapted from the sampling algorithm of Shmoys and Swamy [58] for approximately solving large two-stage stochastic linear programs.

We prove the sampling bounds in the following theorem.

Improved Sampling Algorithm $\mathcal{A}_1$ for $\Pi(F)$
Given a set of facilities F, $\epsilon > 0$ and $\delta > 0$, the problem is to estimate $\Pi(F)$.

Initialize $N \leftarrow \frac{3V}{\epsilon^2}, t = 4 \ln\left(\frac{1}{\delta}\right)$.

1. For $i = 1, \dots, t$,
   (a) Initialize $Y_i \leftarrow 0$.
   (b) For $j = 1, \dots, N$,
      i. Generate sample scenario S_{ij} and let X_{ij} be the optimal value of $\Pi_{S_{ij}}(F)$.
      ii. $Y_i \leftarrow Y_i + \frac{X_{ij}}{N}$.
2. $Y \leftarrow$ median$(Y_1, Y_2, \dots, Y_t)$.
3. Return Y.

Figure 6.2: Improved Sampling Algorithm $\mathcal{A}_1$

Theorem 6.4.2 *Let ρ^* be the true estimate of $\Pi(F)$ and let the variance be σ^2. Also, let the estimate of each sample be bounded in $[a, b]$. If $N = \frac{3\sigma^2}{\epsilon^2}$ and $t = 4 \ln\left(\frac{1}{\delta}\right)$ for some constants $\epsilon, \delta > 0$, then the estimate Y returned by the sampling algorithm $\mathcal{A}_1$ satisfies,*

$$\mathbb{P}(|Y - \rho^*| \geq \epsilon) \leq \delta.$$

Proof: We know that $Y_i = \frac{1}{N}(X_{i1} + \ldots + X_{iN})$ for all $i = 1, \dots, t$. Since X_{ij} are i.i.d. for all $i \in [t], j \in [N]$, $E[Y_i] = \rho^*$ and $\mathsf{Var}(Y_i) = \frac{\sigma^2}{N}$. Therefore, using Chebyshev's inequality we have,

$$\mathbb{P}(|Y_i - \rho^*| \geq \epsilon) \leq \frac{\sigma^2}{N\epsilon^2}.$$

For $i = 1, \dots, t$, consider another random variable Z_i such that

$$Z_i = \begin{cases} 1 & \text{if } |Y_i - \rho^*| \geq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

and $Z = \sum_{i=1}^{t} Z_i$. Therefore,

$$E[Z] = t \cdot \mathbb{P}(Z_i = 1) \leq \frac{t\sigma^2}{N\epsilon^2} = \frac{t}{3}.$$

If $Y \notin [\rho^* - \epsilon, \rho^* + \epsilon]$, then at least $t/2$ variables Z_i should be set to one. Since the variables Z_i are independent bernoulli trials, we can use the Chernoff bounds and obtain,

$$\mathbb{P}(Z \geq \frac{t}{2}) \leq e^{-\frac{t}{4}} = \delta.$$

Therefore, $\mathbb{P}(Y \in [\rho^* - \epsilon, \rho^* + \epsilon]) \geq 1 - \delta$. ■

From Theorem 6.4.2, we obtain that the total number of samples required to obtain an additive ϵ-approximation with probability at least $(1 - \delta)$ for some $\epsilon, \delta > 0$ is,

$$N \cdot t = \frac{12\sigma^2}{\epsilon^2} \cdot \ln \left(\frac{1}{\delta} \right).$$

The dependence of number of samples on δ is improved from $\frac{1}{\delta}$ in the sampling algorithm $\mathcal{A}_0$ to $4 \ln \left(\frac{1}{\delta} \right)$ in sampling algorithm $\mathcal{A}_1$.

6.5 Computational Experiments and Results

In this section, we describe the setup for our computational study for the case of Istanbul and present our results. The North Anatolian fault line runs east to west in the Marmara sea south of Istanbul and poses a serious risk of a major earthquake in the region. In August 1999, an earthquake of magnitude 7.4 on the Richter scale (classified as M5) occurred in the region and caused significant losses to life and property. Studies [48] indicate that there is a high probability of occurrence of an earthquake of magnitude 7 or more in the next 30 years. Therefore, the Municipality of Istanbul is interested in opening ERDC in and around the city of Istanbul to improve its preparedness for post-disaster relief operations.

6.5.1 Data Collection

The Municipality of Istanbul in collaboration with several universities provides a detailed report [47] that analyses the problem of seismic risk and its impact on the population and the transport network in the event of an earthquake. The Municipality has identified a set of 40 potential locations for opening ERDC. These facilities are close to one of two major highways that run east-west through Istanbul. Thus, the facilities do not run the risk of being disconnected from the rest of the region in case several transport links are disrupted due to the earthquake. Figure 6.3 shows the major highway system near Istanbul with a number of bridges and viaducts classified as risky and less risky in [47]. The report also provides data on population in each of the 84 districts which is used in our model to estimate demand in the event of an earthquake. We assume that each district is represented by a single location in our model. Figure 6.4 shows the representative locations for all the districts. We consider a road network between these representative locations where major highways are accurately represented but the smaller links are approximate. Figure 6.5 represents the network we use in our model. The fault line which runs east to west in the south of Istanbul is also approximated by linear pieces so that the distances from the rupture on the fault line can be computed easily. Distance from the point of rupture on the fault time is one of the factors that determines the peak ground acceleration (PGA) when an earthquake occurs and thereby determines the impact at that location.

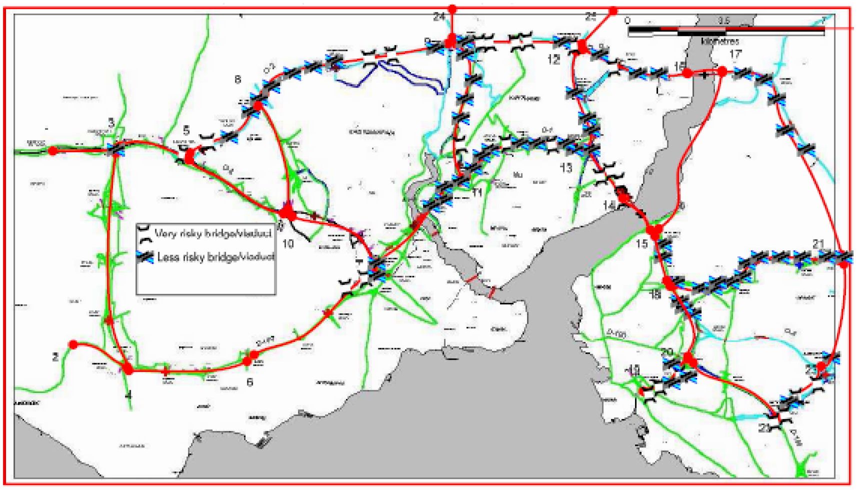

Figure 6.3: Highway Network of Istanbul

6.5.2 Probabilistic Model for Transport Network Scenarios

An earthquake occurs when there is a rupture at some point along the fault line when the techtonic plates collide and there is a sudden release of energy. The impact of earthquake at any location is determined by the amount of shaking that occurs at that location and is measured by *peak ground acceleration*(PGA). Panousis [55] models the PGA due to an earthquake of magnitude m at a distance r from the rupture as,

$$\mathsf{PGA} = \alpha \cdot \frac{e^{0.8m}}{(r+40)^{-2}}$$

where PGA is acceleration in m/s^2, r is distance in km between the location and the site of the rupture and α is a constant. We assume that the probability

Figure 6.4: District Locations

of failure of an edge is directly proportional to the highest PGA level along the link. We also classify each link as being risky and less risky based on the number of risky bridges and viaducts on the link. Let p_e denote the probability of failure of edge e by an earthquake of magnitude m when the minimum distance between edge e and the point of rupture is r_e. Then,

$$p_e = \begin{cases} \beta_h \cdot \frac{e^{0.8m}}{(r_e+40)^{-2}} & \text{if e is risky} \\ \beta_l \cdot \frac{e^{0.8m}}{(r_e+40)^{-2}} & \text{otheriwse} \end{cases} \tag{6.18}$$

We assume that a rupture occurs at a random location on the fault line between the east and west boundaries of the city and the magnitude is uniformly distributed between 6.5 and 7.5. This assumption is reasonable as it closely models the most likely scenario concluded in the report by the Municipality of Istanbul [47]. Furthermore, since we assume that the rupture can occur only within the eastern and western boundaries of the city our estimates would err only on the side of being conservative. Therefore, the transport network scenario is generated as follows.

6.5.3 Probabilistic Model for Demand Scenarios

We simplify the problem by assuming that each district is represented by single location in our network. Furthermore, we assume that the demand at a location is directly proportional to the population in the corresponding district and inversely proportional to the distance of the location from the rupture. Therefore, if the

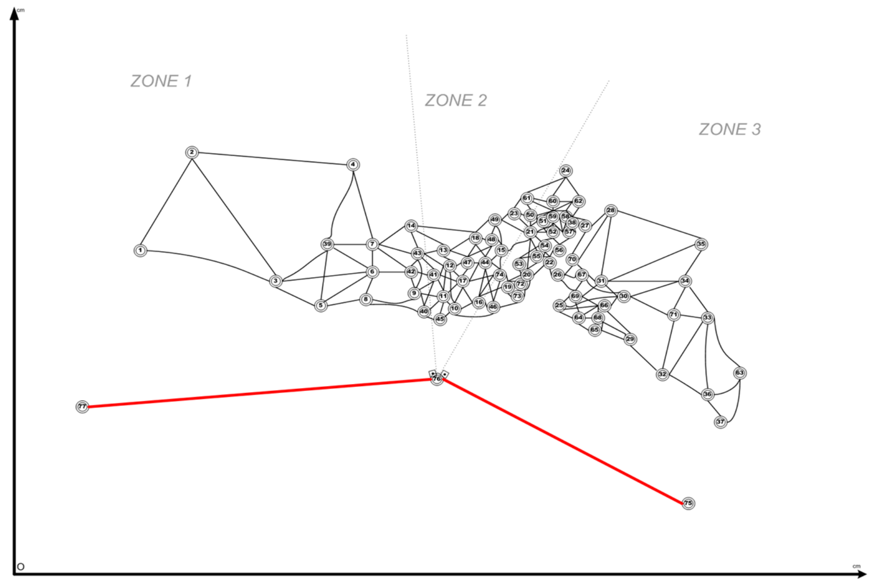

Figure 6.5: Network Model of Istanbul

population of district j is N_j and the distance from the rupture is r_j in a scenario S of earthquake magnitude m, then the demand $d_S(j)$ at location j in scenario S is given by,

$$d_S(j) = \alpha_d N_j \cdot \frac{e^{\beta_d m}}{r_S(j)}$$

where α_d and β_d are constants. Therefore, the demand at each location can be computed when the rupture location and the earthquake magnitude is given.

6.5.4 Computational Results

We use the above scenario generation models (transport network and demand) in the sampling algorithm $\mathcal{A}_1$ described in Section 6.4 to compute estimates of Reliability(F) and Max-Coverage(F) for all possible subsets $F \subset \mathcal{F}$ where $|F| = 3$ for both capacitated as well as uncapacitated versions. The results of Reliability and Max-Coverage for a few choices of F for the length bound $B = 30km$ are given in Table 6.5.4.

If the set of facilities are on the same side of the Bosphorus (the water channel that runs north-south across Istanbul), then the Reliability is very low and the estimate of Max-Coverage is significantly higher. If the emergency response facilities are on one side (say west) of the Bosphorus (for instance when $F = \{1, 2, 3\}$),

Generating a Post-disaster Transport Network Scenario, E_S

Initialize $E_S \leftarrow \phi$.

1. Generate a rupture location l uniformly at random between the east and west boundary of Istanbul and generate an earthquake magnitude m uniformly at random between 6.5 and 7.5.

2. For all edges e in the network

   (a) Compute p_e as described in Equation 6.18.

   (b) Generate a number x uniformly at random between 0 and 1.

   (c) If $x > p_e$, $E_S \leftarrow E_S \cup \{e\}$; otherwise e fails in the scenario S.

3. return E_S.

Figure 6.6: Generating a Network scenario

then they are not able to satisfy the demand on the east side if the road links between the two sides are disrupted. Therefore, all the scenarios where the road links between the two sides are disrupted are not counted in Reliability. On the other hand, these facilities are still able to satisfy almost all the demand on their side of Bosphorus in most of the scenarios which leads to a significantly higher estimate of Max-Coverage. This fact is strikingly visible in the results for the set of facilities $F = \{10, 14, 15\}$ which are all on the west of Bosphorus. In this case Reliability=0.35 and Max-Coverage=0.75. In contrast, if the facilities are split across the two sides then the estimates for both Reliability and Max-Coverage are comparable and high.

We would like to note here that the accuracy or the computational efficiency of the sampling algorithm does not depend on the scenario generation model. The focus of this study is to develop an efficient tool to estimate the quality of a set of open facilities by computing quantities such as Reliability and Max-Coverage. Any probabilistic model for scenario generation can be used in the sampling algorithm and the accuracy or the computational efficiency of the sampling algorithm are not affected other than the time required to sample scenarios from the new model.

6.6 Concluding Remarks

In this chapter, we presented an efficient sampling based algorithm to estimate parameters for which an IP-based formulations have an exponential number of

Facilities F	Reliability(F)	Max-Coverage(F)
$\{1, 2, 3\}$	0.35	0.64
$\{2, 3, 4\}$	0.39	0.65
$\{33, 36, 37\}$	0.32	0.68
$\{17, 18, 31\}$	0.85	0.91
$\{10, 21, 30\}$	0.90	0.93
$\{3, 4, 34\}$	0.65	0.73
$\{10, 14, 15\}$	0.35	0.75

Table 6.1: Reliability and Max-Coverage Results

constraints and variables. Furthermore, we are able to provably obtain an additive ϵ-approximation for the estimates with a high probability for any given $\epsilon > 0$. The goal of this study was to develop a tool that can help in locating emergency response and distribution centers for effective post-disaster logistics for the Municipality of Istanbul. The computational experiments conducted in our study bring out some interesting insights about the geographic location of these ERDC and have been discussed in the computational results section. We would like to mention that in our computational experiments, we modeled the road network of Istanbul as the only transport network available. However, since Marmara sea to the south of Istanbul provides an inexpensive medium of transport using ferries, it would be useful to consider it in the transport network; especially since this transport link would not be disrupted by the earthquake.

Bibliography

[1] S. Arora and G. Karakostas. A 2+ epsilon approximation algorithm for the k-MST problem. *Mathematical Programming*, 107(3):491–504, 2006.

[2] R. Bar-Yehuda. Using Homogeneous Weights for Approximating the Partial Cover Problem. *Journal of Algorithms*, 39(2):137–144, 2001.

[3] G. Barbarosoglu and Y. Arda. A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society*, 55(1):43–53, 2004.

[4] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71, 2003.

[5] D. Bertsimas and M. Sim. The price of robustness. *Operation Research*, 52(2):35–53, 2004.

[6] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, Berlin, 1997.

[7] A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation for the k-MST problem in the plane. *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 294–302, 1995.

[8] A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation algorithm for the k MST problem (extended abstract). *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 442–448, 1996.

[9] G. Calafiore and MC Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[10] M. Charikar, S. Khuller, D.M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651, 2001.

[11] A. Charnes and WW Cooper. Deterministic Equivalents for Optimizing and Satisficing under Chance Constraints. *Operations Research*, 11(1):18–39, 1963.

[12] K. Chen. A constant factor approximation algorithm for k-median clustering with outliers. *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 826–835, 2008.

[13] X. Chen, M. Sim, and P. Sun. A Robust Optimization Perspective of Stochastic Programming. *Optimization Online*, 2005.

[14] Fabián A. Chudak, Tim Roughgarden, and David P. Williamson. Approximate k-msts and k-steiner trees via the primal-dual method and lagrangean relaxation. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 60–70, London, UK, 2001. Springer-Verlag.

[15] G. B. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1955.

[16] Brian C. Dean, Michel X. Goemans, and Jan Vondrak. Approximating the stochastic knapsack problem: The benefit of adaptivity. *focs*, 00:208–217, 2004.

[17] J. Dekle, M.S. Lavieri, E. Martin, H. Emir-Farinas, and R.L. Francis. Fuzzy Modeling for Coordinating Logistics in Emergencies. *Interfaces*, 35:133–139, 2005.

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

[19] K. Dhamdhere, R. Ravi, and M. Singh. On two-stage stochastic minimum spanning trees. In *IPCO*, pages 321–334, 2005.

[20] E. Erdoğan and G. Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming*, 107(1):37–61, 2006.

[21] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. In *FOCS*, pages 62–71, 1995.

[22] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[23] U. Feige, D. Peleg, and G. Kortsarz. The Dense k-Subgraph Problem. *Algorithmica*, 29(3):410–421, 2001.

[24] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.

[25] N. Garg. Saving an epsilon: a 2-approximation for the k-MST problem in graphs. *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 396–402, 2005.

[26] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.

[27] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.

[28] D. Golovin, V. Goyal, and R. Ravi. Pay today for a rainy day: Improved approximations for demand-robust min-cut and shortest path problems. In *STACS*, pages 206–217, 2006.

[29] Gomory, R. E. and Hu, T. C. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, dec 1961.

[30] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *STOC*, pages 417–426, 2004.

[31] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *FOCS*, pages 218–227, 2004.

[32] A. Haghani and S.C. Oh. Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research Part A*, 30(3):231–250, 1996.

[33] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *SODA*, pages 684–693, 2004.

[34] H. Jia, F. Ordóñez, and M. Dessouky. A modeling framework for facility location of medical services for large-scale emergencies. *IIE Transactions*, 39(1):41–55, 2007.

[35] R. Khandekar, G. Kortsarz, V Mirrokni, and M.R. Salavatipour. Two Stage Network Design with Exponential Scenarios. *Proceedings of the sixteenth annual European Symposium on Algorithms*, 2008.

[36] Philip N. Klein, Serge A. Plotkin, Satish Rao, and Eva Tardos. Approximation algorithms for steiner and directed multicuts. *J. Algorithms*, 22(2):241–269, 1997.

[37] P. Kouvelis and G. Yu. *Robust Discrete Optimisation and Its Applications*. Kluwer Academic Publishers, Netherlands, 1997.

[38] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *FOCS*, pages 422–431, 1988.

[39] R. D. Luce and H. Raiffa. *Games and Decision*. Wiley, New York, 1957.

[40] M. Mahdian, Y. Ye, and J. Zhang. A 1.52 approximation algorithm for the uncapacitated facility location problem. In *APPROX*, pages 229–242, 2002.

[41] J. Mestre. A Primal-Dual Approximation Algorithm for Partial Vertex Cover: making educated guesses. *Lecture Notes in Computer Science*, 3624:182, 2005.

[42] J. W. Milnor. Games against nature. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*. Wiley, New York, 1954.

[43] J. M. Mulvey, R. J. Vanderbei, and S. J. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43:264–281, 1995.

[44] V. Nagarajan and R. Ravi. Approximation algorithms for requirement cut on graphs. In *APPROX + RANDOM*, pages 209–220, 2005.

[45] A. Nemirovski and A. Shapiro. Convex Approximations of Chance Constrained Programs. *SIAM Journal on Optimization*, 17:969, 2006.

[46] Y. Nikulin. Robustness in combinatorial optimization and scheduling theory: An annotated bibliography, 2004. `http://www.optimization-online.org/DB_FILE/2004/11/995.pdf`.

[47] Municipality of Istanbul. The study on a disaster prevention or mitigation basic plan in Istanbul including microzonation in the republic of Turkey. *Technical Report*, 2002.

[48] T. Parsons, S. Toda, R.S. Stein, A. Barka, and J.H. Dieterich. Heightened Odds of Large Earthquakes Near Istanbul: An Interaction-Based Probability Calculation. *Science*, 288(5466):661, 2000.

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

[50] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *SODA*, pages 770–779, 2000.

[51] M. J. Rosenblatt and H. L. Lee. A robustness approach to facilities design. *International Journal of Production Research*, 25:479–486, 1987.

[52] A. Ruszczynski and A. Shapiro. Stochastic programming models. *Handbooks in Operations Research and Management Science*, 10:1–64.

[53] A. Saxena. *Integer Programming, a Technology*. PhD thesis, Carnegie Mellon University, Forthcoming.

[54] A. Saxena, V. Goyal, and M. Lejeune. MIP Reformulations for the Proabilistic Covering Problem. *Optimization Online*, 2007.

[55] M. S. Selcuk, A. S. Yücemen. Reliability of Life-line Networks with Multiple Sources under Seismic Hazard. *Natural Hazards*, 21(1):1–18, 2000.

[56] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15:281–288, 1995.

[57] D. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *FOCS*, pages 228–237, 2004.

[58] D.B. Shmoys and C. Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *Journal of the ACM (JACM)*, 53(6):978–1012, 2006.

[59] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

[60] W. Yi and L. zdamar. Fuzzy Modeling for Coordinating Logistics in Emergencies. *International Scientific Journal of Methods and Models of Complexity: Special Issue on Societal Problems in Turkey*, 7(1), 2004.

[61] L. zdamar, E. Ekinci, and B. Kkyazici. Emergency Logistics Planning in Natural Disasters. *Annals of Operations Research: Models and Algorithms for Planning and Scheduling Problems*, 129:217–245, 2004.