An Edge in Time Saves Nine: LP Rounding Approximation Algorithms for Stochastic Network Design

Anupam Gupta
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213.
anupamg@cs.cmu.edu

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

Amitabh Sinha
Dept. of Operations & Mgmt. Sc.
U. of Michigan Business School
Ann Arbor MI 48103.
amitabh@umich.edu

Abstract

Real-world networks often need to be designed under uncertainty, with only partial information and predictions of demand available at the outset of the design process. The field of stochastic optimization deals with such problems where the forecasts are specified in terms of probability distributions of future data. In this paper, we broaden the set of models as well as the techniques being considered for approximating stochastic optimization problems. For example, we look at stochastic models where the cost of the elements is correlated to the set of realized demands, and risk-averse models where upper bounds are placed on the amount spent in each of the stages. These generalized models require new techniques, and our solutions are based on a novel combination of the primal-dual method truncated based on optimal LP relaxation values, followed by a tree-rounding stage. We use these to give constant-factor approximation algorithms for the stochastic Steiner tree and single sink network design problems in these generalized models.

1. Introduction

Network planning problems involve dealing with uncertain future demand patterns for network bandwidth as well as price variations in buying or leasing sufficient capacity to meet demands that arise in the future. Traditional optimization models that attempt to attack the resultant problems after the demands have materialized have formulated this in various ways as a *fixed-charge* problem [27, 29], a *network loading* problem [26], a *capacity expansion* problem [5] or as *capacitated design* [8] problems. These problems have also been studied in the framework of online algorithms, where we can make the pessimistic assumption that the uncertainty is adversarial in nature [3, 7, 19].

In light of recent data-mining advances as well as developments in forecasting techniques, there is better understanding and quantification of the nature of future uncertainty. This can be exploited in a simplified *scenario* model of future states: A finite number of scenarios are postulated, each with its own characteristics. In our example, one set of scenarios may correspond to a recovering economy with stronger demands and stable – or even rising – bandwidth prices; another set of scenarios would be associated with a low growth in demands, and falling network bandwidth prices. The field of *Stochastic Programming* [6, 24] advocates using this model of the future to cast the network planning problem in a two-stage expected cost minimization framework; let us illustrate this with an example.

1.1. Stochastic Steiner Trees

Consider the stochastic version of the basic network planning problem of connecting terminals to a root in a Steiner tree: we are given a graph $G = (V, E)$ with a specified *root* vertex $r \in V$. Each edge $e \in E$ has an associated non-negative cost c_e; we can assume (without loss of generality) that these costs satisfy the triangle inequality. We are also given a probability distribution $p : 2^V \times \mathbb{R} \to [0, 1]$; the value $p(S_k, \sigma^k)$ is the probability that the set of terminals to be connected to the root is S_k and edge costs are changed by a multiplicative factor $\sigma^k (\geq 0)$. (The assumption that the inflation parameter can depend on the scenario is a departure from previous papers; see § 2 for details.) The algorithm must output a set of edges E^0 to be purchased in the *first stage*; subsequently, given the set S_k of terminals (which are assumed to be drawn from the distribution p), it must output a *second stage* or *recourse* edges E^k, such that $E^0 \cup E^k$ is a Steiner tree connecting $\{r\} \cup S_k$.

The goal is to minimize the objective function, which is the expected cost of the solution (with the expectation taken over the randomness in p). Formally, if $c(E') = \sum_{e \in E'} c_e$

is the cost of the edges in set E', then we want to minimize

$$\mathbf{E}[\text{ solution cost }] = c(E^0) + \mathbf{E}[\,\sigma^k\, c(E^k)\,]. \qquad (1)$$

Since we are considering the finite scenario model, we assume that there are only finitely many scenarios (S_k, σ^k) for which $p(S_k, \sigma^k) > 0$; moreover, we require that these scenarios be specified in advance. Thus, the input problem size is proportional to the number of scenarios. We prove the following result:

Theorem 1 *There is a polynomial-time constant-factor approximation algorithms for the Stochastic Steiner Tree problem in the finite scenario model.*

1.2. Stochastic Single Sink network design

We then go on to consider the single sink network design problem, which models the situation in which each edge has both a fixed cost σ, and an incremental cost δ, with the cost of sending $x > 0$ units of flow on an edge being $\sigma + \delta\, x$. The relationship between this model and traditional "buy-at-bulk" network design formulations is discussed in [11]. The stochastic version of this problem models the situation where both these cost multipliers change over time, possibly uncorrelated with each other.

Formally, the input consists of the graph $G = (V, E)$ and root r, with each edge having a cost c_e. The first-stage fixed cost for edge e is now given by $\sigma^0\, c_e$ and incremental cost by $\delta^0\, c_e$; if scenario k (with terminal set S_k) materializes, then the fixed and incremental costs change to $\sigma^k\, c_e$ and $\delta^k\, c_e$. The algorithm must specify the edges E^0 to be bought in the first stage, and for each scenario k, the set of edges E^k to be bought in the second stage (if scenario k materializes) so that $E^0 \cup E^k$ spans the set $S^k \cup \{r\}$. In our solution $E^0 \cup E^k$ for scenario k, vertex $t \in S^k$ routes one unit of flow from t to the root r along a path $p_t \in E^0 \cup E^k$ which minimizes $\sum_{e \in p_t \cap E^0} \delta^0 c_e + \sum_{e \in p_t \cap E^k} \delta^k c_e$.

The objective function for stochastic single sink network design again minimizes the expected cost for each terminal to send one unit of flow to the root, which is given by the following function:

$$\sigma^0 c(E^0) + \mathbf{E}\Big[\,\sigma^k\, c(E^k)\; + \sum_{t \in V} \Big(\delta^0 \sum_{e \in p_t \cap E^0} c_e + \delta^k \sum_{e \in p_t \cap E^k} c_e\Big)\Big].$$

This problem is clearly a generalization of the Stochastic Steiner Tree problem, which can be obtained by setting $\delta^0 = \delta^k = 0$ for all k. We extend our main result to Stochastic single sink network design.

Theorem 2 *There is a polynomial-time constant-factor approximation algorithm for the Stochastic Single Sink Network Design problem.*

Note that there are two uncorrelated rates of inflation in this problem (for the σ's and δ's); this should be contrasted with the fact that previous work in approximations for stochastic problems assumed that the inflation in prices was captured by a single parameter.

The extension of the theorem to the case where arbitrary amounts of flow can originate at each terminal is immediate. Let us instead consider a more interesting extension of the problem: to control the expenditure in each of the scenarios.

1.3. Bounding the Risk

One of the shortcomings with the simple stochastic optimization model described above is that algorithms minimizing the expected cost may suggest investing large sums of money in, say, the first stage: the investors may be loath to do this, since this exposes them to a large (downside) risk.

In light of this, we define risk-bounded versions of these problems, in which we can specify a budget B_0 for the first stage, as well as budgets B_k for each scenario; these budgets constrain the expenditure in both the first and the second stages, and the latter can depend on which scenario materializes. We provide a bi-criteria approximation algorithm for this risk-bounded version of the Stochastic Network Design problem.

Theorem 3 *Consider the Stochastic network design problem, along with a bound B_0 on the cost of the first-stage solution and bounds B_k for the cost of the second-stage solution in scenario k for every scenario. There exists a polynomial time algorithm which either proves that the problem instance is infeasible (the bounds are too low) or provides a solution where each bound is violated by at most a constant factor.*

1.4. Our Techniques

To handle the aforementioned extensions to the models, our algorithms extend the traditional approach of rounding the fractional LP relaxations to integer programming formulations of the stochastic problems in two interesting ways.

1. While our algorithms begin by solving the *LP relaxations* of the problems, they subsequently run a *primal-dual* subroutine guided (or more literally, "capped") by features of the fractional solution. To the best of our knowledge, our work is the first where both steps are critical for the analysis.
2. Our algorithms round the solution using *local bounds*; that is, the cost of serving each scenario is bounded by its own fractional cost. This fact helps us in obtaining the bounds on the downside risk, and leaves open possibilities for further extensions and enhancements.

1.5. Modeling Choices and Generality

There are several possibilities for modeling the uncertainty (stochastic process) associated with the network design problems we consider. The simplest way of modeling this uncertainty is that while the topology of the underlying metric space (or graph) is known, the exact set of terminals requiring service is not known.

A preliminary *first-stage* network is constructed before the actual terminals are revealed; after they are known, a *second stage* of decision-making allows us to purchase additional edges to complete a solution to serve the revealed demand. Purchasing edges in the second stage is much costlier, to account for inflation, lower lead-time, higher costs for right-of-way, etc. We want to minimize the *total cost*: the sum of the first-stage and the *expected* second-stage costs. This is the model of *two-stage stochastic optimization with recourse*; all the work in approximation algorithms for stochastic optimization so far fall into this broad framework, as does our work. A first attempt in modeling the uncertainty in the demands could be to assume that each node has an *independent* probability of requiring service, and that edge costs *inflate by a universal constant factor* between the first and second stages [20]. One way to generalize the model is to allow arbitrary correlations between the clients, and merely ask for *sampling access* to the probability distribution over the clients [15].

These models are still overly stylized, optimistically assuming that inflation in costs is governed by a process independent of the actual demands that arise in the system. To achieve further generality, we use the *scenario* model considered for other problems in [32]. A scenario is specified by its probability of occurrence, a list of terminals which require service, and the corresponding cost inflation factors – there could be more than one. Hence, the scenario model provides a general way of modeling the stochastic process, and underlies the bulk of the discrete models in classical stochastic optimization; in fact, there is a rich body of work in scenario reduction which produces instances with relatively few scenarios but whose solutions are still close-to-optimal for the original problems [18].

2. Related Work

Stochastic Optimization: The field of stochastic optimization [6, 21] is devoted to the general study of optimization under uncertainty, and dates back to the work of Dantzig [9] and Beale [4] in 1955; we refer the interested reader to recent monographs [25, 30, 35]. Classical stochastic optimization usually deals with exact algorithms (which do not result in polynomial running-times since they are typically NP-hard).

Approximation algorithms for stochastic optimization: While this paper is the first work on approximation algorithms for stochastic single-sink network design and for risk-bounded stochastic optimization, variants of stochastic Steiner tree have been considered in several of the works on approximation algorithms for stochastic optimization [15, 20, 22, 32]; we now compare our work with them. Our work generalizes most of this extant work.

Given individual probabilities p_v for each node v to materialize as a terminal, the *maybecast* problem of Karger and Minkoff [22] requires a *single* tree as output. The cost incurred in a scenario (i.e., a realization of terminals, according to the probabilities p_v) is the cost of the subtree induced by the terminals. In contrast, we incur first-stage costs regardless of the scenario, allow different trees to be specified for different scenarios, and allow the cost function to depend on the specific second-stage scenario. Immorlica et al. [20] considered the problem with a *single inflation cost* for the second-stage ($\sigma^k = \sigma$ for all k), and with independent probabilities for nodes to be terminals; they provided a $O(1)$-approximation for Stochastic Steiner tree when the metric is an *ultrametric* and an $O(\log n)$ approximation for general metrics.

An $O(1)$-approximation for Stochastic Steiner tree in general metrics and arbitrary distributions that can be sampled efficiently was given by Gupta et al. [15]; While they also required $\sigma^k = \sigma$ for all k, their model can be extended to allow different values of σ^k. Furthermore, though our approximation ratio for Stochastic Steiner tree is weaker than theirs, it forms the basis of the algorithms for stochastic single sink network design. Also, giving risk-bounded approximations currently does not seem possible with the techniques of [15].

Finally, the *stochastic shortest path* problem was considered by [20, 32], which is a degenerate case of Stochastic Steiner tree instance where each scenario has exactly one terminal. Ravi and Sinha [32] also considered a more general model where the metric changes arbitrarily in each scenario, and provided poly-logarithmic approximations and inapproximability results.

Very recently, Shmoys and Swamy [36] presented randomized approximation algorithms for two-stage stochastic versions of many set cover problems that have deterministic rounding algorithms. Hayrapetyan et al. [17], in addition to other results, show how a multi-stage extension of the stochastic Steiner tree problem can be approximated within a factor of the order of the number of stages.

Deterministic models for Network Design: The Steiner tree problem has been very well studied in the approximation algorithms community. While the current best approximation ratio is 1.55 [33], our algorithms draw on techniques from the primal-dual 2-approximation due to Agrawal, Klein and Ravi [1], and its generalization due to Goemans and

Williamson [12].

There is much recent work on approximations for "buy-at-bulk" network design [28, 34], where the cost of buying bandwidth on edges is a concave function of the amount of bandwidth purchased, reflecting natural economies of scale. The current best result for this problem is an $O(\log n)$ approximation for the multicommodity version [2, 10], and $O(1)$ for the case when all the traffic has a single sink [13, 14, 37]. The single-sink network design problem is a special case of this general class, and has efficient algorithms due to Hassin, Ravi and Salman [16], and to the *Light Approximate Shortest-path Tree (LAST)* construction of Khuller, Raghavachari and Young [23]. The algorithm for the *traveling purchaser* problem [31] provides other leads to the development of our algorithms.

3. Stochastic Steiner tree

In this section, we give a constant factor approximation algorithm for the Stochastic Steiner tree problem in the finite scenario model of two-stage stochastic optimization with recourse. To this end, we formulate the problem as an integer program, solve the linear programming relaxation, and then round the solution using a new variant of the primal-dual method guided by the solution to the linear programming relaxation.

Recall that the input is an undirected graph $G = (V, E)$ with edge-weights c_e, and a distinguished root vertex r. Since we are working in the finite scenario model, a set of m scenarios $\{S_1, \ldots, S_m\}$ is also given. The k^{th} scenario is a set of terminals $S_k \subseteq V$, with an associated probability of occurrence $p_k = p(S_k)$. Furthermore, a scale factor σ^k is given, where the cost of buying e in the recourse network costs $\sigma^k c_e$. A feasible solution is specified by a set of edges E^0 selected in the first stage, and for each scenario k, a set of edges E^k selected in scenario k, such that $E^0 \cup E^k$ is a Steiner tree for $S_k \cup \{r\}$. The objective is to minimize the expected cost of the solution as given by (1), and this can be formulated as an integer linear program as in (IP_{SST}) below.

$$\begin{array}{lrcl} \min & \displaystyle\sum_{e \in E} c_e x_e^0 + \sum_{k=1}^{m} p_k\, \sigma^k \sum_{e \in E} c_e x_e^k & & \qquad (IP_{SST}) \\ & \text{s.t.} \displaystyle\sum_{e \in \delta(S)} (x_e^0 + x_e^k) & \geq & 1 \\ & & & \forall S : r \notin S, \quad S \cap S_k \neq \emptyset, \forall k \\ & x_e^k & \in & \mathbb{Z}_{\geq 0} \\ & & & \forall k, \quad e \in E \end{array}$$

Variables x^0 and x^k are indicator variables for the sets E^0 and E^k, defined as $E^i = \{e : x_e^i = 1\}$. The set $\delta(S)$ denotes the cut formed by edges with exactly one end-point in the set S; that is, $\delta(S) = \{e \in E : |e \cap S| = 1\}$. Edges purchased in the first stage incur a cost $\sum_{e \in E} c_e\, x_e^0$. Scenario k occurs with probability p_k, in which case we incur an additional cost of $\sigma^k \sum_{e \in E} c_e\, x_e^k$. The objective is to minimize the expected total cost. This formulation is an extension of the undirected-cut formulation of the deterministic version of Steiner tree studied in [1, 12].

The critical part for the algorithm is to compute E^0; given E^0, computing E^k is equivalent to contracting the edges in E^0 and finding a minimum Steiner tree on $S_k \cup \{r\}$, which can be well-approximated.

3.1. Tree solutions

As a first step towards obtaining a useful LP relaxation for the problem, let us prove a key structural result. We prove (in Lemma 4) that there exists a near-optimal solution where the paths from any terminal to the root are *monotone*; i.e., they consist of an initial portion of recourse edges, followed by a final portion of first-stage edges. The lemma shows that there exists a first-stage solution that is a tree containing the root r that is not too much worse than the optimum. The intuition behind the lemma is simply that if the expected cost (sum of probabilities times inflation factors) of a network in the second stage is more than the first-stage cost, then it is better to purchase it in the first stage. This key idea, while very simple, forms the crux of the arguments in [20] and [32].

Lemma 4 *Let OPT be the cost of an optimal integer solution to IP_{SST}, specified by E^{0*} and E^{k*}. Then there exists a first-stage tree solution T^0 to IP_{SST} which contains E^{0*} and costs no more than $3OPT$.*

Proof: For every k, define $\rho_k = \sigma^k\, p_k$, and let T^{k*} be the Steiner tree for the terminals S_k in scenario k. We can assume that $E^{k*} \subseteq T^{k*} \subseteq E^{0*} \cup E^{k*}$. Consider a Steiner tree instance defined on the original graph, with one terminal for each connected component in E^{0*}. For every edge e in E^{0*}, we must have $\sum_{k: T^{k*} \ni e} \rho_k \geq 1$, otherwise it is cheaper to defer the edge to recourse. Suppose we define $y_e = \sum_k \rho_k x_e^k$ for every non-E^{0*} edge e. Then $\sum_{e \in E} c_e y_e$ is exactly the cost paid in recourse by our optimal solution. Let T be a connected component of E^{0*} not containing r; Then we have for all cuts $\delta(C)$ such that $T \subseteq C$, $\sum_{k: T^{k*} \ni \text{ some edge from } T} x_k(\delta(C)) \geq 1$. From the above, it also follows that for all such cuts, $y(\delta(C)) \geq 1$ as well. Thus, y is a feasible fractional solution for the undirected-cut formulation of the deterministic Steiner tree instance in this contracted graph. Hence (using the rounding results of [1, 12]) the y's can be rounded within a factor of 2 to yield a Steiner tree in stage 1, which extends E^{0*}, at an additional cost of no more than $2OPT$. Since this extended solution contains E^{0*}, the second stage costs do not increase. □

Observe that for the Stochastic Steiner Tree problem, the optimal choice for E^0 is a forest; however, it is not necessary that E^0 be a single tree, nor that it be connected to the root r. Indeed, consider the example of the "wheel" graph, with $\ell > 1$ "spokes"; spoke i consists of a path P_i with ℓ edges. The rim R of the wheel connects the ends of these spokes by single edges, and hence forms a cycle of length ℓ. The root r hangs off the center of the wheel by a single edge, and all edges in the graph have $c_e = 1$. There are ℓ scenarios, with i-th scenario having $S_i = V(P_i) \cup V(R)$, $p_i = 1/\ell$, and $\sigma_i = \sigma$. The optimal solution for E^0 consists of the edges on the rim and the edge incident to r, and hence E^k consists of the spoke P_k corresponding to scenario k; this gives an expected cost of $\ell(1+\sigma)+1$. A simple case analysis shows that if we force E^0 to be a single tree, then we have to pay $\approx \ell(\min\{2+\sigma, 2\sigma\})$. Setting $\sigma = 2$ shows that demanding E^0 to be connected may cause the solution to be asymptotically a factor $4/3$ worse than optimum.

3.2. LP rounding algorithm

In light of Lemma 4 and the ease of dealing with trees, we will henceforth solve the problem where E^0 is a tree, which we call T^0. In this case, the path from every terminal in scenario k consists of a portion of only recourse edges, followed by a portion consisting of only first-stage edges. This in fact enables us to write a stronger IP formulation for the problem. We then round the linear relaxation of this IP formulation within a constant factor.

First we note some simplifying assumptions.

- The costs c obey the triangle inequality without loss of generality.
- Each terminal occurs in at most one scenario S_k; this is without loss of generality since we are considering finitely many scenarios and listing each scenario explicitly.
- If $p_k\sigma^k \geq 1$ for any k, then it is optimal to connect all the terminals of scenario k in stage 1. Hence we assume that this is done, and $p_k\sigma^k \in (0,1)$ for all k henceforth.

The revised IP is shown in (IP_{SST-T}). Variables $x^0, x^1, \ldots, x^k$ are indicators for the installation of edges in stage 1 and recourse. For a terminal t in scenario k, 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*; that is, each undirected edge is replaced by two anti-parallel directed edges in the graph. Given this *bi-directed* graph, the cut-sets are defined as $\delta^+(S) = \{e = (u,v) : u \in S, v \notin S\}$ and $\delta^-(S) = \delta^+(V \setminus S)$. For a singleton vertex v, we abuse notation slightly to denote $\delta^+(\{v\})$ by $\delta^+(v)$ (and likewise, $\delta^-(v)$).

$$\min \sum_{e\in E} c_e x_e^0 + \sum_{k=1}^{m} p_k \sigma^k \sum_{e\in E} c_e x_e^k \qquad IP_{SST-T}$$

$$\begin{array}{rcl}
\text{s.t. } (r^0(t)+r^k(t))(\delta^+(t)) & \geq & 1 \\
& & \forall t \in S_k, \forall k \\
(r^0(t)+r^k(t))(\delta^-(v)-\delta^+(v)) & = & 0 \\
& & \forall v \notin \{t,r\} \\
r^0(t)(\delta^-(v)) & \leq & r^0(t)(\delta^+(v)) \\
& & \forall v \notin \{t,r\} \\
r_e^k(t) & \leq & x_e^k \\
& & \forall e, \forall k, \forall t \\
r, x & \in & \mathbb{Z}_{\geq 0}
\end{array}$$

Rounding Overview We begin by solving the LP relaxation of the polynomial size LP obtained as the linear relaxation of IP_{SST-T}; let (x,r) denote an optimal LP solution. The basic approach is that if we have a graph with a set of terminals and fractional edge variables x such that any cut separating some terminals from the root has x-value at least 1, we have a fractional Steiner tree which we can round within a factor of 2, using say [1]. Our aim, therefore, is to extract a similar situation out of our fractional solution (x,r) where the cut values for the first-stage variables (x^0) is at least some constant, and round it to a first-stage Steiner tree. However, if the recourse costs dominate, we must use the recourse LP support to guide our choice of recourse trees. Our new idea here is to use the primal-dual algorithms for Steiner trees to grow such recourse trees but truncate this process when the growing moats (cuts) obey the first condition of having a constant support value for the first-stage variables crossing them. To implement this idea, we modify the graph a little in order to take care of various issues. The rounding algorithm has several stages, which we describe below without optimizing constants for the final performance ratio.

(1) Path decomposition Every terminal t is able to ship one unit of flow from itself to the root, specified by its r variables. This can be decomposed into a set of paths, with path p having a flow value $f(p)$, and a prefix from t to some point $p^* \in p$ with recourse flow ($r_e^k(t) > 0$) and the remaining suffix containing stage 1 flow ($r_e^0(t) > 0$). The point p^* is called the *critical point* of the path p for terminal t. Let P_t be the set of positive-flow paths for terminal t, where we have $\sum_{p\in P_t} f(p) = 1$ for all t.

(2) Ball definition The idea is to identify a radius for each terminal beyond which most of the flow is on stage 1 cables, and inside which most of the flow is of recourse type. Given a radius γ, define the ball around t to be $B_\gamma(t) = \{v : c_{tv} \leq \gamma\}$, its complement $\overline{B}_\gamma(t) = V \setminus B_\gamma(t)$, and the *boundary* of the ball to be $\hat{B}_\gamma(t) = \{v : c_{tv} = \gamma\}$ (We can subdivide edges to consider interior points of edges to be at the boundary as appropriate). For a terminal t, define the **critical radius** $\gamma(t)$ to be such that $\sum_{p:p^* \in B_{\gamma(t)}(t)} f(p) \geq \frac{1}{2}$ and $\sum_{p:p^* \in \hat{B}_{\gamma(t)}(t) \cup \overline{B}_{\gamma(t)}(t)} f(p) \geq \frac{1}{2}$. Since for every terminal t, every path $p \in P_t$ contains p^* as a vertex, $\gamma(t)$ exists for every terminal and can be found by a shortest path computation. We also define $\gamma(r) = 0$.

By definition, we have $x^0(\delta(S)) \geq \frac{1}{2}$ for every $S \supseteq B_{\gamma(t)}(t)$, where we use $x^i(A)$ to denote $\sum_{e \in A} x^i_e$. Thus if we select a set of balls which are disjoint, then we can round the x^0-values outside these balls to an integer Steiner tree at a cost at most twice of the linear relaxation. However, the cost of this tree can be bounded above by the cost of the fractional solution only if we contract each ball and build a Steiner tree on the contracted graph. This is not sufficient, since we need to pay for edges from the boundaries of the balls to the centers. Hence we use an additional step introduced in [31] and used subsequently in [11, 37].

We select a maximal set of terminals R^0 such that (i) $r \in R^0$, (ii) for every $t, t' \in R^0$, we have $B_{2\gamma(t)}(t) \cap B_{2\gamma(t')}(t') = \emptyset$, and (iii) for every $v \notin R^0$, there exists $t_v \in R^0$ such that $B_{2\gamma(t_v)}(t_v) \cap B_{2\gamma(v)}(v) \neq \emptyset$, and $\gamma(t_v) \leq \gamma(v)$. Such a collection can be found by examining all terminals in increasing order of their critical radii. It is crucial that all terminals *from all scenarios* are examined, recalling that each terminal belongs to exactly one scenario.

(3) First stage tree The first stage Steiner tree T^0 is simply any approximate Steiner tree on R^0. Let ρ_{ST} denote the approximation ratio of the LP-rounding algorithm we use to compute a Steiner tree on R^0 (The best-known value is currently 2 [1]).

Lemma 5 *The cost of T^0 is at most $4\rho_{ST} \cdot \sum_{e \in E} c_e x^0_e$.*

Proof: Consider an approximate Steiner tree obtained by contracting $B_{\gamma(t)}(t)$ for every $t \in R^0$; its cost can be bounded by $2\rho_{ST} \sum_{e \in E} c_e x^0_e$ since we have $x(\delta(B_{\gamma(t)}(t))) \geq \frac{1}{2}$ for every $t \in R^0$. Finally, we charge the cost of extending the edges which reach $\hat{B}_{\gamma(t)}(t)$ to actually reach t to the portion of the path in $B_{2\gamma(t)}(t) - B_{\gamma(t)}(t)$, giving a further overhead of a factor of 2. □

This completes our first-stage solution. We now have to construct second-stage solutions for each scenario k to connect the vertices in $S^k \setminus T^0$ to the first-stage tree T^0. While we have a fractional solution to work with, it is not amenable to rounding. Hence we now run a primal-dual subroutine to construct the second-stage trees, but use the fractional solution to guide and prematurely halt the primal-dual subroutine.

(4) Component growth If stage 1 costs were infinite, then we would compute Steiner trees for each scenario separately. Hence the intuition is that we will try to grow Steiner trees for each scenario, but at some point the finiteness of stage 1 costs will “cap” our growth and force us to switch to stage 1. We will use the path decomposition and the presence of representatives defined above to guide the process of capping. The growing process is as defined in the primal-dual algorithms of Agrawal, Klein and Ravi [1]. The reader is urged to examine [1] or [12] for the details of the primal-dual method in this context; we sketch our procedure below with the details deferred to a complete version of this manuscript.

We grow components (moats) for each scenario separately. Fix a scenario k, and grow moats around all terminals in $S_k \setminus R^0$ using the primal-dual algorithm [1]. These moats collide and merge, in which case the merged active moats continue to grow. We grow a moat M as long as for some terminal $t \in M$, we have $\sum_{p \in M} f(p) \leq \frac{1}{2}$. It is important that we are summing over paths all of whose points are *fully contained* inside M, and not just their critical points (all vertices in the path from the terminal t to the moat boundary are inside the moat). In other words, as long as we can guarantee that at least half unit of “recourse” flow is crossing an active moat, we keep growing the moat.

Our moat-growth procedure is slightly different from a simple breadth-first-search strategy, for the following reason. In a typical moat, it may be the case that the same path crosses the moat boundary more than once, so that $x^k(\delta(M))$ is an overestimate of $\sum_{p \notin M} f(p)$. To guard against this, our component growth procedure incorporates accounting for paths fully contained inside the moats.

A moat could therefore stop growing (get capped) in two cases: either while growing some more critical points come inside the moat and the invariant ceases to hold, or at a collision the invariant ceases to hold. For our purposes, there is no distinction between the two.

Lemma 6 *For any capped moat M constructed for scenario k, the total dual value collected during its growth denoted $y(M) \geq \frac{1}{2} \cdot \sum_{u,v \in M} x^k_{uv} c_{uv}$.*

Proof: The proof is immediate because at any instant when M is growing, there is at least half unit of recourse flow on edges along which the moat is growing. □

The performance ratio of the primal-dual rounding algorithm yields the following bound.

Lemma 7 *The cost of the forest T_M constructed for the capped moats in scenario k is at most $2\rho_{ST} \cdot \sum_{u,v\in M} x^k_{uv} c_{uv}$.*

(5) Recourse Steiner trees for scenarios Consider a moat M of scenario k. Using Lemma 7, we can pay for a Steiner tree of type k connecting all terminals in $S_k \cap M$. We now have to connect this tree to T^0 when scenario k materializes. We cannot just charge the cost of this connection to any terminal inside M, since Steiner tree fractional solutions share such costs. Instead, we charge this hook to the cost of the tree itself.

Lemma 8 *Let M be a moat with tree T_M. Then the cost of hooking T_M to T^0 is at most $8 \sum_{u,v\in M} x^k_{uv} c_{uv}$.*

Proof: Consider a terminal $t \in M$ such that $\sum_{p\in M} f(p) \geq \frac{1}{2}$; this exists by our moat capping condition. The sum of building costs along these recourse paths from t is at least $\frac{1}{2} \cdot \gamma(t)$ by the definition of $\gamma(t)$ and the assumption of metric costs. Since there exists $t' \in T^0$ at distance no more than $4\gamma(t)$ from t, the cost of connecting M to T^0 can be charged to t. Moreover, since we are only using paths of t which are fully inside M, the charging is disjoint across disjoint moats in M and we are done. □

Theorem 9 *The LP relaxation of IP_{SST-T} can be rounded within a factor of $\max(4\rho_{ST}, 8+2\rho_{ST})$ in polynomial time.*

Proof: This follows from Lemmas 5, 7 and 8. □

Putting together the above theorem with Lemma 4 gives our main Theorem 1 on Stochastic Steiner trees.

4. Stochastic network design

4.1. Problem definition

We now give our algorithm for stochastic network design, extending the algorithm for Stochastic Steiner tree to include ideas such as a subroutine for transforming the Steiner tree to approximately preserve shortest-path distances due to Khuller, Raghavachari and Young [23].

Recall the description of the problem in Section 1.2. Using the same argument as for the Steiner tree case, we can argue that the first stage solution can be completed to be a tree solution with only a factor of 2 worsening in the performance ratio. The resulting integer program formulation is shown in (IP_{SND}), where $r^0_e(t)$ and $r^k_e(t)$ denote the flow along edge e of terminal t on first-stage and recourse cables respectively, with $t \in S_k$.

$$
\begin{aligned}
\min\ & \sigma^0 c(T^0) + \sum_{k=1}^{m} p_k \Big[\sigma^k c(T^k) \\
& + \sum_{t\in S_k} \sum_{e} c_e (r^0_e(t)\delta^0 + r^k_e(t)\delta^k)\Big] \qquad IP_{SND}
\end{aligned}
$$

$$
\begin{array}{rcl}
\text{s.t. } (r^0(t) + r^k(t))(\delta^+(t)) & \geq & 1 \\
 & & \forall t \in S_k, \forall k \\
(r^0(t) + r^k(t))(\delta^-(v) - \delta^+(v)) & = & 0 \\
 & & \forall v \notin \{t, r\} \\
r^0(t)(\delta^-(v)) & \leq & r^0(t)(\delta^+(v)) \\
 & & \forall v \notin \{t, r\} \\
r^k_e(t) & \leq & x^k_e \\
 & & \forall e, \forall k, \forall t \\
r, x & \in & \mathbb{Z}_{\geq 0}
\end{array}
$$

This is a strict generalization of Stochastic Steiner tree, which can be obtained by setting $\delta^k = 0$ for all k. However, we will often be using our algorithm for Stochastic Steiner tree to obtain a partial solution for which the σ component of the cost can be easily accounted. As a normalization, we assume $\delta^0 = 1$.

4.2. Special case: $\delta^k = 1$ for all k

We first analyze a constant factor approximation for the special case when $\delta^k = 1$ for all k. This yields the main ideas which are developed further to provide an approximation algorithm for the general case.

Algorithm The algorithm is fairly straightforward. We begin by ignoring the incremental costs δ, reducing the problem to Stochastic Steiner tree. We use the algorithm in the previous section to obtain a constant factor approximation for the σ component of the cost function. Let $T, T^1, \ldots, T^m$ be the forests obtained. Note that $T \cup T^i$ is a tree for scenario-i terminals.

We convert T into a Light Approximate Shortest-path Tree (LAST). Let T^0 be the new tree, where the distance between any vertex $v \in T^0$ from r is no more than α times c_{vr}, for a fixed constant α. We then contract T^0, and consider each T^k in turn. We "LASTize" T^k to $\hat{T}^k$ with the contracted T^0 as root, which may switch terminals around from the various trees of T^k that they originally belonged to. The algorithm terminates when we have LASTized all scenario trees. The following theorem, adapted from Khuller, Raghavachari and Young [23], bounds the σ component of the cost function at this stage. The proof of the theorem is algorithmic and yields a polynomial time algorithm to compute such LASTs.

Theorem 10 *(Khuller, Raghavachari and Young [23]) There exists a constant $\alpha = 1 + \sqrt{2}$ such that given a graph G and a Steiner tree T on terminals $R \in G$ rooted at r, there exists a tree T' such that $c(T') \leq \alpha c(T)$ and for every $t \in R$, the distance from r in T' to t is no more than α times its distance in G.*

Bounding the routing cost Consider a terminal t in scenario k, and let v be the first point in T^0 in the path constructed by our solution from t to r. We bound the routing cost of t by c_{tr}, which is a lower bound. The probability of occurrence p_k is incorporated into both sides of this bound, and hence may be ignored.

We therefore have $c_{tv} + c_{vr} \leq 2c_{tv} + c_{tr} \leq (2\alpha + 1)c_{tr}$, where the first inequality follows from the triangle inequality and the second from the LAST guarantee for $T^0 \cup T^k$. This yields the following theorem.

Theorem 11 *When $\delta^k = 1\ \forall k$, a constant factor approximation algorithm can be obtained for the stochastic network design problem by converting into LASTs the trees obtained by any approximation algorithm for the corresponding stochastic Steiner tree problem as described above.*

The fact that $\delta^k = 1$ for all k is used critically in this algorithm in order to bound the routing costs by the triangle inequality. Surprisingly enough, this simple algorithm extends to the case for general δ^k with very minor modifications. The analysis, however, is a little more involved and is shown next.

4.3. General case

We now consider the case of general incremental costs, where each δ^k is different, some greater than 1 and some smaller. The algorithm, surprisingly enough, is exactly the same as for the special case of constant incremental costs discussed above. Therefore, we can use the fact that T^0 itself is already a LAST with respect to r, and the building cost (the σ component of the objective function) can therefore be paid for by the earlier analysis.

Bounding the routing cost We use two separate arguments to bound the routing costs of scenarios, depending on whether their multipliers δ^k have gone up or down relative to the first stage. For a terminal t, let its path to the root be given by $p_k \cup p_0$, where p_k is the path from t to T^0, and p_0 is the path in T^0. Therefore, the cost of this path in our solution is $c(t) = \delta^k c(p_k) + c(p_0)$. Let the LP routing cost be given by $c^*(t)$.

Lemma 12 *For a terminal $t \in S_k$ such that $\delta^k \geq 1$, we have $c(p) = O(c^*(t))$.*

Proof: Let p, p_0, p_k be defined as before, and let $t' \in T^0$ be the point where p_0 and p_k meet. We have two lower bounds: $c^*(t) \geq c_{tr}$ since $\delta^k \geq 1$, and $c^*(t) \geq \frac{\delta^k}{2}\gamma(t)$ by definition of $\gamma(t)$. Let $t'' \in T^0$ be such that $c_{tt''} \leq 4\gamma(t)$; this exists by our selection of terminals R^0 on which T^0 was built. Since T^k was LASTized, we have $c_{tt'} \leq \alpha c_{tt''} \leq 4\alpha\gamma(t)$, which bounds $\delta^k c(p_k)$ by $8\alpha c^*(t)$.

Next, we bound $c(p_0)$ using the triangle inequality as $c(p_0) \leq \alpha(c(p_k) + c_{tr})$, since T^0 is a LAST. Using the fact that $\delta^k \geq 1$ and $\delta^k c(p_k) \leq 8\alpha c^*(t)$ above, we can also bound $c(p_0)$ by $(8\alpha^2 + \alpha)c^*(t)$ and we are done. □

Lemma 13 *For a terminal $t \in S_k$ such that $\delta^k \leq 1$, we have $c(p) = O(c^*(t))$.*

Proof: We first assume that $r \notin B_{2\gamma(t)}(t)$. The bound for $c(p_k)$ goes through as in Lemma 12, because $c(p_k) \leq 4\alpha\gamma(t)$ and $\frac{1}{2}\delta^k\gamma(t)$ continues to be a lower bound in this case.

However, we use a slightly different lower bound to bound $c(p_0)$, since we cannot charge it to $c(p_k)$ because $\delta^k \leq 1$. By definition of $\gamma(t)$, we must have that $\frac{1}{2}(c_{tr} - \gamma(t))$ is a lower bound since we have $\sum_{p:p^* \notin B_{\gamma(t)}(t)} f(p) \geq \frac{1}{2}$. Since $r \notin B_{2\gamma(t)}(t)$, we must have that $\frac{\gamma(t)}{2}$ is a lower bound on $c^*(t)$, since a half fraction of t's flow paths are of type zero in the "shell" $B_{2\gamma(t)}(t) - B_{\gamma(t)}(t)$. We can then use the triangle inequality as before, and using this shell lower bound of $\frac{\gamma(t)}{2}$ and $\frac{1}{2}(c_{tr} - \gamma(t))$ as a lower bound, we can bound $c(p_0)$ by $O(c^*(t))$. Formally, $c_{t'r} \leq c_{t't} + c_{tr}$, where p_0 goes from t' to r. For the first term, we have $c_{t't} \leq 4\alpha\gamma(t) \leq 8\alpha c^*(t)$ by the definition of t'' and the "LASTing" guarantee. For the second term, $c_{tr} = 2 \cdot \frac{1}{2}(c_{tr} - \gamma(t)) + 2 \cdot \frac{\gamma(t)}{2} \leq 4c^*(t)$.

Finally, if $r \in B_{2\gamma(t)}(t)$, we handle such terminals t separately. Let $S'_k = \{t \in S_k : r \in B_{2\gamma(t)}(t)\}$. We construct a recourse Steiner tree for terminals in S'_k rooted at r (without capping), and convert it into a LAST. Such a tree can easily be paid for in both its building and routing cost because $r \in B_{2\gamma(t)}(t)$ for all $t \in S'_k$. □

We still haven't bounded the routing costs of terminals in T^0. However, this can easily be done by the following artifice - assume that there are two copies of the terminals in T^0, with the first copy being an artificial copy selected into T^0, and the second copy moving on to stage 2. Using this artifice, every terminal is assumed to be pushed to the second stage, and we have just bounded the routing costs of terminals in the second stage.

Putting all these together, we have the following theorem.

Theorem 14 *There is a poly-time constant factor approximation algorithm for stochastic network design.*

5. Risk-bounded network design

While typical stochastic optimization algorithms minimize the overall expected cost, a natural question to ask is that given a particular scenario, how much is the algorithm requiring us to pay? A reasonable solution might require that the cost incurred in the second stage is comparable to the requirement of the second stage. A general way of modeling this is to assign *budgets* B_k for each scenario k, saying, "If scenario k materializes, the solution must not cost more than B_k in the second stage." Such a budget is a means for guarding the *down-side risk*: the worst cost that could be incurred in any scenario.

A similar budget could also be specified for the first-stage, though of course that could lead to an infeasible problem. (Even though portions of the routing cost are incurred using first-stage cables, we model this as wholly being incurred in the second stage, and take this into account only in the second stage budgets.) A powerful feature of the finite scenario model and our solution technique is that this version of down-side risk can be explicitly modeled, and our algorithm provides a solution which guards against it. Our main result is the following.

Theorem 15 *Consider the Stochastic Steiner tree problem, along with a bound B_0 on the cost of the first-stage solution and bounds B_k for the cost of the second-stage solution in scenario k for every scenario. There exists a polynomial time algorithm which either proves that the problem instance is infeasible (the bounds are too low) or provides a solution where each bounded is violated by at most a constant factor.*

Proof: (Sketch) The down-side risk budget can be modeled as constraints of the form $\sum_{e \in E} c_e x_e^k \leq B_k$ for each k for which a budget exists (including $k = 0$ for the first stage). Let IP_{SST-R} denote the extension of IP_{SST-T} obtained by adding these constraints. The linear relaxation of this IP can be tested for feasibility in polynomial time, and if it is found to be infeasible, then the problem itself is infeasible because the budgets are too restrictive. Note that the problem can always be made feasible by making B_0 appropriately high, for any values of B_k for the second stage.

If a feasible LP solution is found, we proceed by using exactly the same algorithm as for Stochastic Steiner tree. The key observation is that the algorithm bounds the cost of each scenario *locally*, using only its corresponding components of the LP solution: Lemma 5 bounds the cost of T^0 by $O(\sum_{e \in E} c_e x_e^0)$, and Lemmas 7 and 8 bound the cost of each component of T^k by $O(\sigma^k \sum_{e \in E} c_e x_e^k)$. □

The critical fact required to obtain risk-bounded guarantees continues to hold for stochastic network design: the building cost analysis can be taken care of as in Theorem 15 above. The routing costs can also be taken care of, since Lemmas 12 and 13 bound the routing cost of each terminal by a constant times their corresponding fractional routing costs. This results in Theorem 3.

6. Future work

An alternative version of stochastic network design is where the routing cost multiplier (δ^k) in scenario k is applied to flow on edges purchased in the first stage as well as the second stage. For example, δ^k may represent the gasoline cost in scenario k, and σ^k represents the cost of constructing a road in scenario k. This version of the problem can be easily solved by our algorithm, with the only change required being the replacement of the δ^0 term in the objective function with δ^k.

The optimization of the performance ratios is left for a complete version of this paper. An interesting open question is whether the first-stage solution can variables, which could be useful if the cost of maintaining descriptions and solutions of all scenarios is prohibitively expensive.

7. Acknowledgments

This work was done while the third author (Amitabh Sinha) was a graduate student at the Tepper School of Business at Carnegie Mellon University. R. Ravi and Amitabh Sinha were supported in part by NSF grant CCR-0105548 and ITR grant CCR-0122581 (The ALADDIN project).

References

[1] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995. (Preliminary version in *23rd STOC*, 1991).

[2] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 542–547, 1997.

[3] Baruch Awerbuch, Yossi Azar, and Serge Plotkin. Throughput-competitive online routing. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[4] E. M. L. Beale. On minimizing a convex function subject to linear inequalities. *J. Roy. Statist. Soc. Ser. B.*, 17:173–184; discussion, 194–203, 1955. (Symposium on linear programming.).

[5] D. Bienstock and O. Günluk. Capacitated network design – polyhedral structure and computation. *INFORMS Journal on Computing*, 8(3):243–259, 1996.

[6] John R. Birge and François Louveaux. *Introduction to stochastic programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.

[7] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, New York, 1998.

[8] Robert D. Carr, Lisa Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Symposium on Discrete Algorithms*, pages 106–115, 2000.

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

[10] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 448–455. ACM Press, 2003.

[11] Naveen Garg, Rohit Khandekar, Goran Konjevod, R. Ravi, F. Sibel Salman, and Amitabh Sinha. On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design formulation. In *Proceedings of the 8th Integer Programming and Combinatorial Optimization Conference*, volume 2081 of *Lecture Notes in Computer Science*, pages 170–184, 2001.

[12] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995. (Preliminary version in *5th SODA*, 1994).

[13] Sudipto Guha, Adam Meyerson, and Kamesh Mungala. A constant factor approximation for the single sink edge installation problems. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 383–388, 2001.

[14] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 365–372, 2003.

[15] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: Approximation algorithms for stochastic optimization problems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 417–425, 2004.

[16] Refael Hassin, R. Ravi, and F. S. Salman. Approximation algorithms for a capacitated network design problem. In *APPROX*, pages 167–176, 2000.

[17] Ara Hayrapetyan, Chaitanya Swamy, and Éva Tardos. Network design for information networks. Manuscript, 2004.

[18] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24:187–206, 2003.

[19] Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

[20] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 684–693, 2004.

[21] Peter Kall and Stein W. Wallace. *Stochastic programming*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons Ltd., Chichester, 1994.

[22] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, pages 613–623, 2000.

[23] Samir Khuller, Balaji Raghavachari, and Neal E. Young. Balancing minimum spanning and shortest path trees. *Algorithmica*, 14(4):305–322, 1995.

[24] Willem K. Klein Haneveld and Maarten H. van der Vlerk. Stochastic integer programming: general models and algorithms. *Ann. Oper. Res.*, 85:39–57, 1999. Stochastic programming. State of the art, 1998 (Vancouver, BC).

[25] Willem K. Klein Haneveld and Maarten H. van der Vlerk. *Stochastic Programming*. Department of Econometrics and OR, University of Groningen, Netherlands, 2003.

[26] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the capacitated network loading problem. *Operations Research*, 43:142–157, 1995.

[27] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18:1–55, 1984.

[28] Yishay Mansour and David Peleg. An approximation algorithm for minimum-cost network design. In *Robust communication networks: interconnection and survivability (New Brunswick, NJ, 1998)*, volume 53 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 97–106. Amer. Math. Soc., Providence, RI, 2000.

[29] M. W. Padberg, T. J. Van Roy, and L. A. Wolsey. Valid linear inequalities for fixed charge problems. *Operations Research*, 33:842–861, 1985.

[30] Michael Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 1995.

[31] R. Ravi and F. S. Salman. Approximation algorithms for the traveling purchaser problem and its variants in network design. In *Algorithms—ESA '99 (Prague)*, volume 1643 of *Lecture Notes in Comput. Sci.*, pages 29–40. Springer, Berlin, 1999.

[32] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *Proceedings of the 10th Integer Programming and Combinatorial Optimization Conference*, pages 101–115, 2004.

[33] Gabriel Robins and Alexander Zelikovsky. Improved Steiner tree approximation in graphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

[34] F. Sibel Salman, Joseph Cheriyan, R. Ravi, and Sairam Subramanian. Approximating the single-sink link-installation problem in network design. *SIAM Journal on Optimization*, 11(3):595–610, 2000.

[35] R. Schultz, L. Stougie, and M. H. van der Vlerk. Two-stage stochastic integer programming: a survey. *Statist. Neerlandica*, 50(3):404–416, 1996.

[36] David Shmoys and Chaitanya Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.

[37] Kunal Talwar. Single-sink buy-at-bulk LP has constant integrality gap. In *Proceedings of the 9th Integer Programming and Combinatorial Optimization Conference*, volume 2337 of *Lecture Notes in Computer Science*, pages 475–486, 2002.