Math. Program., Ser. A (2005)

Digital Object Identifier (DOI) 10.1007/s10107-005-0673-5

R. Ravi · Amitabh Sinha

Hedging Uncertainty: Approximation Algorithms for Stochastic Optimization Problems

Received: August 28, 2003 / Accepted: September 24, 2005
Published online: December 30, 2005 –

Abstract. We study two-stage, finite-scenario stochastic versions of several combinatorial optimization problems, and provide nearly tight approximation algorithms for them. Our problems range from the graph-theoretic (shortest path, vertex cover, facility location) to set-theoretic (set cover, bin packing), and contain representatives with different approximation ratios.

The approximation ratio of the stochastic variant of a typical problem is found to be of the same order of magnitude as its deterministic counterpart. Furthermore, we show that common techniques for designing approximation algorithms such as LP rounding, the primal-dual method, and the greedy algorithm, can be adapted to obtain these results.

1. Introduction

With the increasing success of optimization algorithms in process optimization, these methods are making inroads into earlier planning stages of large scale projects. The inherent difference between optimization at the planning stage and *post-facto* optimization is that in the former, data is not fully available. Yet costly decisions with wide-ranging implications need to be taken in the face of incomplete data. Nevertheless, quite often forecasts of future uncertainty are available that can be used in the planning model. Forecasts, by nature, are imprecise and provide at best a range of possible futures. The field of stochastic optimization is an attempt to model such situations. For a detailed introduction, please consult one of the recent texts on the topic [4, 25, 18].

In a parallel development, the field of approximation algorithms evolved to counter the prohibitive resource requirements for the exact solution of NP-hard combinatorial optimization problems. Informally, these algorithms run in polynomial time and deliver a performance ratio on the worst-case quality of the output solution over all instances. As the size of the models being solved increases in scale, this solution approach gains in importance. For a detailed exposition of this topic, the reader is referred to Vazirani [44] and Ausiello et al. [2].

However, as approximation algorithms become more sophisticated in scope and technique, the refrain from real-world practitioners who have the need for such algorithms

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

A. Sinha: Ross School of Business, University of Michigan, Ann Arbor MI 48109, USA.
e-mail: amitabh@umich.edu

Mathematics Subject Classification (1991): 20E28, 20G40, 20C20

is that the input data is seldom well-defined, thus diminishing the value of the solutions and guarantees provided by the algorithm. Conversely, while the field of stochastic optimization models the uncertainty in data fairly well, the running times of the exact algorithms developed in the stochastic optimization community often prove prohibitive. This paper combines the best of both worlds, by providing approximation algorithms for a stochastic version of several classical optimization problems.

2. Background

2.1. Two-stage model

Among the most popular models in stochastic optimization is the two-stage model with recourse. At the outset, some data may be known deterministically, whereas the uncertain future is characterized only by a probability distribution. The decisions made at this point are referred to as the first-stage decisions. Subsequently, the actual future is realized, and then there may be the opportunity to augment the first-stage solution in order to optimize for the realized scenario. This second stage of decision making is called the recourse stage. The goal is to optimize the first-stage decision variables so as to minimize the expected cost over both stages, where the expectation is over the probability distribution defining the second stage.

2.2. Mathematical formulation

Formally, a two-stage stochastic optimization problem with recourse is defined as follows. Vector x^0 is the set of decision variables which have to be fixed in the first stage, when only partial information is available. Later, full information is made available and we choose some second-stage (or recourse) variables x^1 to augment the first-stage solution x^0. Let ξ represent the random vector which defines the constraint matrix T, cost vector q and requirement vector h when the full information is made available, and let A, c, and b define the same for the first stage. Given a vector (or matrix) a, let a' denote the transpose of a. Let P denote additional constraints such as non-negativity or integrality which the components of x^0 and x^1 need to satisfy. The stochastic program can be written as:

$$\begin{array}{rrcll} \min & c'x^0 & + & E_\xi\, Q(x^0, \xi) & (IP_{S1}) \\ \text{s.t.} & Ax^0 & = & b & \\ & x^0 & \in & P & \end{array}$$

$$\begin{array}{rll} \text{where } Q(x^0, \xi) & = \min & q'x^1 \\ & \text{s.t.} & T(x^0, x^1) = h \\ & & x^1 \in P \end{array}$$

Here $Q(x^0, \xi)$ represents the optimal cost of the second stage, conditioned on scenario $\xi = (q, T, h)$ having been realized and a first-stage setting of the variables x^0. The expectation is taken with respect to ξ.

2.3. Stochastic optimization with finite scenarios

A popular restriction of the two-stage model is where the second stage is characterized by a finite set of m scenarios. In scenario k, the constraint matrix, cost vector and requirement vector take on values T^k, q^k and h^k respectively, and scenario k occurs with probability p_k. In this case, we can write IP_{S1} in *extensive* form as follows, where x^k represents our choice for the variable x^1 if scenario k materializes:

$$\begin{aligned} \min \quad & c'x^0 + \sum_{k=1}^{m} p_k (q^k)' x^k \quad (IP_{S2}) \\ \text{s.t.} \quad & Ax^0 = b \\ & T^k(x^0, x^k) = h^k \qquad k = 1, 2, \ldots, m \\ & (x^0, x^k) \in P \qquad k = 1, 2, \ldots, m \end{aligned}$$

The interested reader may refer to any of the texts cited above for a more complete description of models of stochastic optimization and their uses. Schultz, Stougie and Van der Vlerk [38] provide an excellent survey of two-stage stochastic integer programming, while Kong and Schaefer [28] recently provided approximation algorithms for a class of such problems. The relevance of the finite scenario model becomes more pronounced in light of the recent work on scenario reduction by Heitsch and Römisch [21].

2.4. Approximation algorithms

The field of approximation algorithms provides one means of tackling the "curse of dimensionality" that afflicts several integer programming and combinatorial optimization problems, due to their NP-completeness. Since exact algorithms that run in time polynomial in the size of the input are not known (and unlikely to exist) for NP-complete problems, approximation algorithms trade-off the precision of the solution with speed, by guaranteeing to run in polynomial time. At the same time, they provide guarantees in the form of approximation ratios. If C is the set of all possible inputs of a certain minimization problem and $OPT(I)$ and $A(I)$ respectively denote the value of an optimal solution and the solution output by algorithm A for that problem, then the approximation ratio $\rho(A)$ of algorithm A is defined as follows:

$$\rho(A) = \max_{I \in C} \frac{A(I)}{OPT(I)}$$

The interested reader is referred to two recent books [44, 2] for further pointers on approximation algorithms.

2.5. Literature review

While both the areas of approximation algorithms and stochastic optimization have been extremely active areas of optimization in the past decade (and longer), relatively little work exists on approximation algorithms for stochastic optimization. Among the earliest papers which provided an approximation algorithm for stochastic optimization was

the work on service provisioning for a telecommunication network by Dye, Stougie and Tomasgard [9]. By its very nature, scheduling algorithms often have to account for uncertainty in the sizes and arrival times of future jobs; some approximation algorithms which account for such uncertainty in various models of scheduling include the works of Möhring, Schulz and Uetz [33], Kleinberg, Tardos and Rabani [27], and Skutella and Uetz [43].

Karger and Minkoff [26] considered a Steiner tree problem with uncertainty in the terminal set, which falls under a slightly different model of stochastic optimization. Kong and Schaefer [28] provided approximation algorithms for a class of matching problems in a stochastic setting, and their work provided some of the initial framework on which our work was built. Immorlica, et al. [22] also studied two-stage stochastic versions of some combinatorial optimization problems. Their model and results are distinct from ours, and they obtained their results independently of ours.

Subsequent to the first appearance of this work, other papers have provided further approximation algorithms for stochastic versions of combinatorial optimization problems. Gupta, Ravi and Sinha [16] considered the two-stage finite-scenario version of the rooted Steiner tree problem, and provided a constant factor approximation.

Shmoys and Swamy [39, 40] provide a sampling-based approach which uses interior point linear programming algorithms to provide approximation algorithms for two-stage as well as multi-stage stochastic versions of a certain class of combinatorial optimization problems. Gupta, et al. [14, 15] also provide sampling-based approximation algorithms for two-stage and multi-stage stochastic problems, using cost-sharing functions to bound the cost of the solution. Both of these streams of work allow for exponentially many scenarios and have running times independent of the number of scenarios, which constitutes an improvement over our work.

Other recent work providing approximation algorithms for stochastic versions of combinatorial optimization problems include that of Hayrapetyan, Swamy and Tardos [20] (information networks), Gupta and Pál [13] (Steiner trees), and Dhamdhere, Ravi and Singh [7] (minimum spanning trees).

2.6. Our results

In this paper, we provide polynomial-time approximation algorithms for several classical combinatorial optimization problems, in a two-stage stochastic optimization setting with finitely many scenarios. Our results are summarized in Figure 1. The current best known deterministic approximations are listed, with a "1" meaning that the problem can be solved optimally in polynomial time. All of the stochastic approximation ratios are derived in this paper. Some of the hardness results are carried over from the underlying deterministic problems; these are mentioned with citations. The remaining are proved in this paper.

In the table in Figure 1, we use m to refer to the number of scenarios and n to refer to the number of combinatorial elements (the number of vertices in the shortest paths problems and the number of elements in the set cover problem). An APTAS (asymptotic polynomial time approximation scheme) is an algorithm whose performance ratio approaches 1 as the number of objects increases [2].

Problem	Det. approx.	Stochastic elements	Stochastic approximation	Hardness
Shortest paths	1[8]	Sink	O(1)	NP-hard
		Sink and metric	$O(\log^2 n \log m)$	$\Omega(\log^2 n)$
Bin packing	APTAS[24]	Object sizes	APTAS	NP-complete[24]
Facility location	1.52[32]	Client demands, facility costs	8	1.46[11]
Vertex cover	2[34]	Vertex weights, Incidence	2	1.16[19]
Set cover	$O(\log n)$[23]	Set weights, Set inclusions	$O(\log nm)$	$\Omega(\log n)$[1], $\Omega(\log m)$

Fig. 1. Summary of results

In the sequel, we consider the five problems in the order listed in Figure 1. Approximation algorithms and hardness results for the underlying deterministic versions of each of these problems are surveyed, and the algorithms for the stochastic versions are presented.

3. Shortest paths

Motivation Consider a supplier who wishes to ship a single unit of a good to a single destination t from a single source s, in a graph where the shipping cost is just the cost of the edge. The solution to this problem is to compute a shortest path from s to t, and this can be easily done in polynomial time, for example by using the algorithm due to Dijkstra [8].

Now consider the following stochastic extension. The supplier does not know in advance what the destination is going to be. In particular, any of m scenarios could materialize, with the destination being t_k in scenario k. The supplier could enter into contracts in stage 1 to ship the good along edge e at cost c_e, but this might be disadvantageous if the destination turns out to be in the opposite direction. However, if the supplier decides to wait for the scenarios to materialize, then the cost of edge e in scenario k changes to $f_k c_e$, which could be disadvantageous if f_k is large. Hence the supplier might wish to reserve some edges now at cost c_e, and augment the network in scenario k if necessary.

Problem definition We are given a graph $G = (V, E)$, with edge costs c_e and a single source $s \in V$. Without loss of generality, we assume that the edge costs c_e satisfy the triangle inequality (form a metric), since any edge which violates the triangle inequality can be replaced by an edge with length equal to the shortest path between its end points. We also have a set of m scenarios, with scenario k specified by a destination vertex $t_k \in V$, a cost scale factor f_k, and a probability p_k. A feasible solution is specified by a set of edges $E' \subset E$. The first-stage cost of this solution is $\sum_{e \in E'} c_e$, and in scenario k, a second stage solution is a path P_k from s to t_k; for the second stage costs, we assume the edges in P_k bought in the first stage, namely in E', have cost zero,

while the remaining edges are increased in cost by factor f_k, giving second-stage cost $f_k \sum_{e \in P_k \setminus E'} c_e$. The objective is to compute E' which minimizes the sum of first stage edge costs and expected second stage edge costs. We abbreviate this problem as SSP (stochastic shortest paths).

While it is not obvious that E' even induces a connected component, the following lemma proves that E' is indeed connected; in fact, it is a tree.

Lemma 1. *There exists an optimal solution to SSP where the set of edges E' bought in the first stage induces a tree containing the source s.*

Proof. Suppose for a contradiction there is another connected component $C \not\ni s$. Let K' be the set of scenarios for which the optimal solution uses at least one edge in C, and let E_s be the connected component of first-stage edges which include the source s. For optimality, it must be the case that for every edge $e \in C$, we have $\sum_{P_k \ni e} p_k f_k \geq 1$, implying that $\sum_{k \in K'} f_k \geq 1$.

Now consider the paths used in the scenarios in K'. Let k^0 be the scenario in which the second-stage cost of the segment to C from the source is the cheapest. If we re-route the paths of all scenarios in K' to using the path of k^0 from the source until the point the other scenario paths intersect C, then since $\sum_{k \in K'} f_k \geq 1$, the total cost cannot increase. Therefore, we can purchase these edges (which we used for re-routing), and this does not increase the cost.

Proceeding this way for other components, we infer that E^* induces a connected graph containing s, which need be no more than a tree since the second stage solutions only look for a single path from s.

Interpretation as a network design problem Armed with the above lemma, SSP can be interpreted as the tree-star network design problem, defined as follows. In tree-star network design, demand nodes have a demand for d_j units of goods to be shipped from a source. A feasible solution is specified by a tree, with the cost of the solution being M times the cost of the tree (for pre-specified M) plus the length of the shortest path to each demand node from the tree, weighted by the demand at the node. A constant-factor approximation algorithm for this problem was first provided by Ravi and Salman [35], and it has also been studied subsequently as the connected facility location problem [26, 29], and the asymmetric VPN design problem [12].

Theorem 1. *There is a polynomial-time constant-factor approximation algorithm for SSP.*

Proof. SSP is equivalent to the tree-star network design problem, via the following transformation. The fixed cost multiplier of the tree M is set to 1. The demand of each node t_k is set to $f_k p_k$. Now, purchasing a tree T in stage 1 for SSP is equivalent to building T in the tree-star problem. The expected second stage cost is exactly $\sum_{k=1}^{m} p_k f_k \text{dist}(t_k, T)$, which is the same as incurred in the tree-star problem when the demand at node t_k is $p_k f_k$. Here dist (t_k, T) is the distance from t_k to the nearest point in T using the original metric c.

While there may exist multiple optimal solutions to SSP, some of which need not induce a first-stage tree, our algorithm finds an optimal solution where the first stage is a

tree. Since there always exists an optimal solution to SSP whose first-stage component is a tree (Lemma 1), the equivalence of SSP and tree-star network design as described above remains valid. The equivalence of SSP and tree-star network design also implies the NP-hardness of SSP. Note that SSP is different from the maybecast problem studied by Karger and Minkoff [26], because in their model, each node is a terminal independently with a certain probability, edge costs do not change across scenarios, and the solution is required to be a single tree spanning all potential terminals.

3.1. Stochastic metric

The problem becomes even more interesting (and harder) when the metric itself is allowed to change arbitrarily across scenarios. This might happen, for example, because shipping by sea becomes much cheaper than air transport in one scenario, and vice-versa in another. The problem is defined exactly as in Section 3, except that the cost of edge e in the first stage is c_e^0 and in scenario k is c_e^k. We call this the stochastic metric shortest paths (SMSP) problem.

In general, the first-stage component of an optimal solution for SMSP need not be a tree. Consider the following example, where there is only one second-stage scenario. The graph is a path with five vertices $s = v_0, \ldots, v_4 = t$, where s and t are the source and the sink respectively. Let M be a large constant. The costs of the four edges $(v_0, v_1), \ldots, (v_3, v_4)$ in the first stage are respectively $1, M, 1, M$, and in the second stage are $M, 1, M, 1$. The optimal solution is clearly to purchase edges (v_0, v_1) and (v_2, v_3) in the first stage, and the others in the second stage; this solution has cost 4. Any solution which requires the first stage to be a tree has cost at least M.

Nevertheless, in certain situations it is natural to expect or require that the first-stage component be a tree. Most extant literature dealing with stochastic versions of network design problems either explicitly require the first-stage component to be a tree, or end up with such solutions nevertheless [14, 16, 20, 22, 26]. For the rest of this section, we consider the stochastic metric shortest paths problem with the explicit additional requirement that the first-stage component of any solution must be a tree. We label this problem Tree-SMSP.

Hardness The Tree-SMSP problem is as hard as the group Steiner tree problem (GST), defined as follows. $G = (V, E)$ is an undirected graph with edge weights c_e, and there are m vertex subsets (called groups) S_k. The objective is to compute a minimum cost tree which includes at least one vertex from every group. This problem was studied by Garg, Konjevod and Ravi [10] who also gave an approximation algorithm with performance ratio roughly $O(\log^2 n \log m)$, and recently Halperin and Krauthgamer [17] showed an inapproximability threshold of $\Omega(\log^2 n)$ even when G is a tree. An $\Omega(\log^2 n)$ hardness for Tree-SMSP follows from the reduction of GST to Tree-SMSP, shown below.

Theorem 2. *An instance of GST can be modeled as a special case of Tree-SMSP.*

Proof. Suppose we are given an instance of group Steiner tree, specified by $G = (V, E)$, metric edge costs c, and groups $S_1, S_2, \ldots, S_m$. We create an instance of Tree-SMSP with one scenario for every group. The graph remains the same, and the first stage edge costs c^0 are the same as c, the edge costs in the GST instance. In scenario k, the metric is

as follows. The distance between any two vertices in S_k is zero, and all other distances are infinity. Any vertex in S_k is defined to be the destination t_k for scenario k. All scenarios are equally likely.

An optimal solution to this instance of Tree-SMSP must select a first stage tree which includes at least one vertex from each S_k, to avoid infinite cost. If the tree includes any vertex in S_k, it can be augmented at cost zero to a tree which includes t_k if scenario k materializes.

Approximation algorithm Our approximation algorithm relies on the following IP formulation of Tree-SMSP. Variable r^k_{uv} is 1 if edge (u, v) (in the direction $u \to v$) is part of the path traversed to t_k from s and edge (u, v) is chosen in the recourse solution. Variable f^k_{uv} is 1 if edge (u, v) is chosen in the path to t_k from s and edge (u, v) is part of the first-stage solution. Variable x_{uv} is 1 if edge (u, v) is chosen in the first-stage tree.

$$\begin{aligned}
\min \sum_e c_e x_e + \sum_{k=1}^{m} p_k \sum_e r^k_e c^k_e \qquad & (IP_{SMSP}) \\
\text{s.t.} \sum_v (r^k_{v,t_k} + f^k_{v,t_k}) \geq 1 \qquad & \forall k \\
\sum_v (r^k_{uv} + f^k_{uv}) = \sum_v (r^k_{vu} + f^k_{vu}) \qquad & \forall u \in V \setminus \{t_k, s\}, \forall k \\
\sum_v r^k_{uv} \geq \sum_v r^k_{vu} \qquad & \forall u \in V \setminus \{t_k\}, \forall k \\
f^k_e \leq x_e \qquad & \forall e \in E, \quad \forall k \\
f, r, x \quad \text{non-negative integers}
\end{aligned}$$

The third set of inequalities are strengthenings valid only for the tree version of SMSP, insisting that flows along recourse arcs to t_k from s via any node are non-decreasing; they are also crucial for obtaining the result below. IP_{SMSP} is polynomial in size, so its linear relaxation LP_{SMSP} can be solved optimally in polynomial time. Let (f, r, x) denote an optimal solution to the linear program LP_{SMSP}, and $OPT_{Tree-SMSP}$ be its value. The following theorem describes our rounding algorithm.

Theorem 3. *The fractional solution (f, r, x) can be rounded in polynomial time to an integer solution $(\hat{f}, \hat{r}, \hat{x})$ of cost $O(\log^2 n \log m) OPT_{Tree-SMSP}$.*

Proof. For each destination t_k, let $r^*(k) = \sum_e r^k_e c^k_e$ be the cost incurred by the recourse component of the fractional path for t_k. Let S_k be the set of all nodes within distance $2r^*(k)$ of t_k in the metric c^k. The idea is that we can incur a factor of 2 and pay for a path to t_k from any node in S_k by charging it to $r^*(k)$, and hence we need a first stage tree which reaches at least one node in S_k. We construct sets S_k for every scenario k, and create an instance of the group Steiner tree problem using the metric c.

Using Markov's inequality[1], it can be shown that if $s \notin S_k$, then we have $\sum_{e=(u,v):u \notin S_k, v \in S_k} x_e \geq \frac{1}{2}$. Hence $2x$ is a fractional solution to the linear relaxation

[1] Markov's inequality is used in the following form here: If the recourse components of the paths from s to t_k are considered, and α_i and β_i denote the path lengths and flow amounts on path i respectively, then we have $r^*(k) = \sum_i \alpha_i \beta_i$. Since in any optimal solution we must have $\sum_i \beta_i = 1$, and $\beta_i \geq 0$ for all i, we can view β_i as probabilities; specifically, we define a random variable $\mathbf{r}$ and let β_i be the probability that $\mathbf{r} = \alpha_i$. We then have $E[\mathbf{r}] = r^*(k)$, and Markov's inequality indicates that $Pr(\mathbf{r} \geq 2r^*(k)) \leq 1/2$.

of the following IP formulation of the group Steiner tree problem: min $\sum_e c_e x_e$ such that $\sum_{e=(u,v):u\notin S, v\in S} x_e \geq 1 \quad \forall S\ \exists k \ : \ S_k \subseteq S$. Using the result of Garg, Konjevod and Ravi [10], we can construct an integer solution $\hat{x}$ at a cost which is no more than $O(\log^2 n \log m \cdot OPT_{SMSP})$ which is a tree and includes at least one vertex of every S_k. Since for every scenario k we can augment this tree to include t_k at cost no more than $2r^*(k)$, our approximation ratio follows.

4. Bin packing

Problem definition and algorithm Stochastic bin packing is motivated by applications where storage capacity has to be reserved in advance of the arrival of the objects, and if the reserved capacity is insufficient, we have to purchase additional capacity at possibly higher costs. Formally, we are given a bin capacity B, known in advance. There is a set of m possible scenarios, with scenario k specified by a probability p_k of occurrence, a set S_k of objects (each with size $s_i^k \leq B$), and a bin cost f_k. A feasible solution is specified by a integral number x of bins purchased in stage 1, at unit cost per bin. If scenario k materializes, the objects in S_k need to be packed into bins of capacity B, which may necessitate the purchase of an additional integral number of bins at cost f_k per bin. The objective is to compute x so as to minimize the expected total cost. For any y, let $\lceil y \rceil$ denote the smallest integer no smaller than y, and $[y]$ denote the integer nearest to y.

For the deterministic bin-packing problem, an APTAS was given by Fernandez de la Vega and Lueker [6], which uses at most $(1+\epsilon)OPT + 1$ bins, for any $\epsilon > 0$ (although the running time depends on ϵ). In general, suppose we have an algorithm for the deterministic version of bin-packing which always uses at most $\rho.OPT + \beta$ bins, for fixed ρ and β. Asymptotically (as $OPT \to \infty$), such an algorithm becomes a ρ-approximation. Any locally optimal algorithm (first-fit, for example) achieves $\rho = 2$ with $\beta = 0$. The following theorem shows how to extend any deterministic bin-packing algorithm to handle our version of stochastic bin-packing.

Theorem 4. *Order the scenarios so that we have $\sum_i s_i^1 \geq \sum_i s_i^2 \geq \cdots \geq \sum_i s_i^m$. Let k^* be the largest integer such that $\sum_{k=1}^{k^*} f_k p_k \geq 1$. Then purchasing $x = \lceil \rho \sum_i s_i^{k^*} \rceil + \beta + 2$ bins in the first stage yields a solution of cost no more than $\rho.OPT + \beta + 2$, where OPT is the expected cost of the optimal solution to the stochastic bin-packing problem.*

Proof. Consider the fractional relaxation of the problem, when we can pack items fractionally into bins. In that case, $x^* = [\sum_i s_i^{k^*}]$ is the optimal solution, because it is the point where the expected marginal cost of buying an additional bin in recourse goes below 1. The expected total cost if we purchase x^* bins is $x^* + \sum_{k>k^*} p_k f_k (\lceil \sum_i s_i^k \rceil - x^*)$, which is a lower bound on the value of an optimal solution of stochastic bin packing.

Since $\lceil \rho \sum_i s_i^k \rceil + \beta + 2$ bins are sufficient to pack the objects in S_k, we will need to purchase at most $\lceil \rho \sum_i s_i^k \rceil - \lceil \rho x^* \rceil - 1$ additional bins if scenario $k > k^*$ materializes. If scenario $k \leq k^*$ is realized, then $\lceil \rho x^* \rceil + \beta + 2$ bins are sufficient and no additional bins are needed. Hence the expected cost of our solution is bounded from above by $\lceil \rho x^* \rceil + \sum_{k>k^*} p_k f_k (\lceil \rho \sum_i s_i^k \rceil - \lceil \rho x^* \rceil - 1) + \beta + 2$, which is no more than ρ times our lower bound plus an additional $\beta + 2$ bins.

Since the algorithm by Fernandez de la Vega and Lueker [6] achieves $\rho = 1 + \epsilon$ and $\beta = 1$, we can obtain an APTAS for our stochastic version of bin-packing which uses no more than $(1 + \epsilon)OPT + 3$ bins for any $\epsilon > 0$ (and running time dependent on ϵ).

5. Facility location

5.1. Definition

As in the classical uncapacitated facility location problem, we are given a set of facilities F and a set of clients D, with a metric c_{ij} specifying the distances between every client and every facility. However, the demand of each client is not known at the first stage. In scenario k, client j has demand d_j^k, which may be zero. Facility i has a first-stage opening cost of f_i^0, and recourse costs of f_i^k in scenario k. These may be infinity, reflecting the unavailability of the facilities in various scenarios. We abbreviate this problem as SFL.

$$\begin{aligned}
&\min \sum_{i \in F} f_i y_i^0 + \sum_{k=1}^{m} p_k \Big(\sum_{i \in F} f_i^k y_i^k + \sum_{i \in F, j \in D} d_j^k c_{ij} x_{ij}^k \Big) && (IP_{SFL}) \\
&\text{s.t.} \sum_{i \in F} x_{ij}^k \geq d_j^k && \forall j \in D, \forall k \\
&\qquad x_{ij}^k \leq y_i^0 + y_i^k && \forall i \in F, \forall j \in D, \forall k \\
&\qquad x, y \quad \text{non-negative integers}
\end{aligned}$$

The problem is best explained by the integer program formulation IP_{SFL} above. While our algorithms extend to arbitrary demands at every client, for simplicity we only study the case when all d_j^k's are either 0 or 1. Variable x_{ij}^k is 1 if and only if client j is served by facility i in scenario k. If $x_{ij}^k = 1$, then facility i must either be opened at the first stage ($y_i^0 = 1$) or in recourse in scenario k ($y_i^k = 1$) (or both).

5.2. History and non-triviality of the problem

The classical (deterministic) uncapacitated facility location problem has a rich history (see Cornuéjols, Nemhauser and Wolsey [5] for a survey). Balinski [3] introduced an integer programming formulation for this problem which has led to several approximation algorithms. The first constant factor approximation (which uses this formulation) is due to Shmoys, Tardos and Aardal [42], and the current best algorithm, due to Mahdian, Ye and Zhang [32] uses a formulation which differs only slightly. Indeed, our formulation (IP_{SFL}) extends Balinski's formulation to the stochastic setting. In the stochastic optimization community, Louveaux and Peeters [31] considered a slightly different version of stochastic facility location, and provided a dual-ascent based exact (non-polynomial-time) algorithm for it.

There are several preliminary insights which one can obtain by examining our formulation of stochastic facility location closely. First notice that if the second stage facility costs were identical to those in the first stage for all scenarios, then we can "de-couple"

the stochastic components of the problem and solve for each scenario independently. On the other hand, if there was no second stage and all facilities had to be opened in the first stage, then SFL reduces to an instance of the usual UFL, where the probability multipliers in the expected service costs can be incorporated into the demand terms (thinking of the demands as being scaled based on the probability of occurrence). This also extends to allowing arbitrary demand distributions at the vertices, if they are independent. In this case, existing approximations for UFL apply directly. The added difficulty, and indeed the interesting aspect of the model, arises from varying (and typically increased) second-stage facility costs under different scenarios.

In the other direction, SFL can be viewed as a special case of the multicommodity facility location problem [37, 41], where we treat each scenario as a distinct commodity and the cost of a facility depends on the commodities it serves. However, the best-known approximation ratio for such a version of the multicommodity facility location problem is $O(\log m)$ due to Ravi and Sinha [37], (where m is the number of scenarios), so we need different techniques for better approximations of SFL.

The main difficulty stems from the fact that we cannot treat each scenario by itself, since the different scenarios interact in utilizing first-stage facilities. A simple heuristic is to compare the solution obtained if all the demand is satisfied in the first stage with the solution when no first stage facilities are opened. While this heuristic works well in certain instances (particularly, maximization problems such as maximum weight matchings, as in [28]), it can easily be shown to perform badly in our case, due to the interaction across scenarios.

5.3. Algorithm

Our approximation algorithm proceeds along the lines of the LP-rounding algorithm due to Shmoys, Tardos and Aardal [42], with some crucial differences. We begin by solving the linear relaxation of IP_{SFL}. Let (x, y) denote an optimal LP solution. The first step in rounding this fractional solution is using the filtering technique of Lin and Vitter [30]. We fix a constant $0 < \alpha < 1$. For every client-scenario pair (j, k), we define its optimal fractional service cost to be $c^*_{jk} = \sum_i c_{ij} x^k_{ij}$. Order the facilities which serve the pair (j, k) according to non-decreasing distance from j. The α point $g_{j,k}(\alpha)$ for the client-scenario pair (j, k) is the smallest distance c^α_{jk} such that $\sum_{i:c_{ij} \le c^\alpha_{jk}} x^k_{ij} \ge \alpha$.

Theorem 5. *Given a feasible fractional solution (x, y), we can find a fractional solution $(\overline{x}, \overline{y})$ which is feasible for the LP relaxation of IP_{SFL} in polynomial time such that (i) $c^\alpha_{jk} \le \frac{1}{1-\alpha} c^*_{jk}$; (ii) $\overline{x}^k_{ij} > 0 \Rightarrow c_{ij} \le c^\alpha_{jk}$ for all $i \in F, \quad j \in D, \quad k = 1, 2, \ldots, m$; (iii) $\overline{y}^k_i \le \min\{1, \frac{y^k_i}{\alpha}\}$ for all $i \in F, \quad k = 0, 1, \ldots, m$.*

Proof. First, if $c^\alpha_{jk} > \frac{1}{1-\alpha} c^*_{jk}$, then we get the following contradiction (as an application of Markov's inequality, using the same logic as in Footnote 1): $c^*_{jk} \ge \alpha.0 + (1-\alpha) c^\alpha_{jk} > c^*_{jk}$, proving (i).

Next, define $\overline{x}$ as follows, which satisfies (ii) by definition:

$$\overline{x}^k_{ij} = \begin{cases} \min\{1, x^k_{ij}/\alpha\} & \text{if } c_{ij} \le c^\alpha_{jk} \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, define $\overline{y}_i^k = \max_{j \in D} \overline{x}_{ij}^k$, for all i and k. Using (i) and the definition of $\overline{x}$, it follows that $\overline{y}_i^k \leq \min\{1, \frac{y_i^k}{\alpha}\}$ for all $i \in F$, satisfying (iii).

The definitions also ensure that $(\overline{x}, \overline{y})$ is a feasible solution to the LP relaxation of IP_{SFL}.

The algorithm in [42] proceeds to iteratively round x_{ij}^k variables for which c_{jk}^α is smallest. However, this does not work in our case, because the rounding algorithm might close facilities which are needed for other scenarios $k' \neq k$. Hence we need a rounding algorithm which carefully treats the distinction between stage 1 facility variables y^0, and recourse facility variables y^k.

We proceed as in earlier algorithms by obtaining an optimal LP solution. In the next step, we progressively choose clients *across all scenarios* with minimum fractional service cost, and neglect to serve other clients conflicting (overlapping in facility utilization) with it by assigning them to be served by this client's serving facility. However, this will not work if the serving facility is not open in this neglected client's scenario. Hence, the main difference is that if a stage 1 facility is opened to serve a client, all clients that conflict with it can be served, while if a stage 2 facility variable is rounded up to serve this client, only those clients in the same scenario that conflict with this client are neglected and assigned to this client. This strategy suffices to pay for all opened facilities by the "disjointness" of the different scenarios' contributions in the objective function, while the rule of considering clients in increasing order of fractional service cost allows us to bound the service cost. Our rounding algorithm is described in detail below. Let $0 < \beta < 1$ be another fixed constant.

1. Initialize $\hat{F}^k = \emptyset$ to be the set of facilities opened in scenario k for $k = 0, 1, \ldots, m$. Mark all client-scenario pairs as "unserved".
2. Let (j, k) be an unserved client-scenario pair with smallest c_{jk}^α. Consider the following cases, in each case marking (j, k) as "served" and proceeding to the next client-scenario pair. Let S^0 be the set of facilities i such that $\overline{x}_{ij}^k > 0 \wedge \overline{y}_i^0 > 0$, and S^k be the set of facilities i such that $\overline{x}_{ij}^k > 0 \wedge \overline{y}_i^k > 0$.
   (a) If $\sum_{i \in S^0} \overline{y}_i^0 \geq \beta$, let i be the facility such that f_i^0 is smallest among all facilities in S^0. Move facility i to the set $\hat{F}^0$, and set $\hat{y}_i^0 = 1$. For all other facilities $i' \in S^0 \cup S^k$, set $\hat{y}_{i'}^0 = \hat{y}_{i'}^k = 0$. For client-scenario pairs (j', k') such that there exists a facility $i' \in S^0 \cup S^k$ with $c_{i'j'} \leq c_{j'k'}^\alpha$, set $\hat{x}_{ij'}^{k'} = 1$ and mark them as "served".
   (b) If $\sum_{i : i \in S^0} \overline{y}_i^0 < \beta$, then we must have $\sum_{i : c_{ij} \leq c_{jk}^\alpha} \overline{y}_i^k \geq 1 - \beta$. In this case, let i be the facility in S^k with smallest f_i^k. Move facility i to the set $\hat{F}^k$ and set $\hat{y}_i^k = 1$. For all other facilities $i' \in S^k$, set $\hat{y}_{i'}^k = 0$. For clients j' such that there exists a facility $i' \in S^k$ with $c_{i'j'} \leq c_{j'k}^\alpha$, set $\hat{x}_{ij'}^k = 1$ and mark them as "served".
3. Facilities in $\hat{F}^0$ are the facilities to be opened in stage 1, and facilities in $\hat{F}^k$ are the facilities to be opened in recourse if scenario k materializes. Clients are served according to the zero-one variables $\hat{x}_{ij}^k$.

The algorithm is similar in spirit to that of Shmoys, Tardos and Aardal [42]. Notice that feasibility of the fractional solution is maintained throughout the above rounding

process. In each step, we fully open (round to 1) one facility and assign several clients to this facility, while closing (rounding to 0) some others. The cost of the open facility is bounded by a constant times the costs of the facilities that are closed, as will be proved in Lemma 2 below. Each time a facility is closed, all clients dependent on it for service are assigned to the facility that was just opened in order to close the facility under consideration.

Lemma 2. *The rounding algorithm above produces an integer solution* $(\hat{x}, \hat{y})$ *which is feasible for I* P_{SFL} *such that*

(i) For every client-scenario pair (j, k)*, we have* $\hat{x}_{ij}^k = 1 \Rightarrow c_{ij} \leq 3c_{jk}^{\alpha}$.
(ii) $\sum_{i \in F} f_i^0 \hat{y}_i^0 \leq \frac{1}{\beta} \sum_{i \in F} f_i^0 \overline{y}_i^0$.
(iii) $\sum_{i \in F} f_i^k \hat{y}_i^k \leq \frac{1}{1-\beta} \sum_{i \in F} f_i^k \overline{y}_i^k$ *for all* $k = 1, 2, \ldots, m$.

Proof. When a client is assigned to a facility (ie, $\hat{x}_{ij}^k$ is set to 1), we either assign it to a facility within distance c_{jk}^{α}, or it is assigned when some other client j' with $c_{j'k}^{\alpha} \leq c_{jk}^{\alpha}$ was being considered. In either case, a simple application of triangle inequality yields $c_{ij} \leq 3c_{jk}^{\alpha}$.

When a facility i is chosen for opening in the first stage (ie, $\hat{y}_i^0$ is set to 1), case 2(a) must have occurred. In that case, we have a sufficiently large fraction (β) of facilities which have $\overline{y}_i^0 > 0$ which we are shutting, and we can charge the cost of opening i to the fractional solution. A similar argument holds for the case when a facility is opened in recourse in scenario k.

The solution produced is also feasible, because we start with a feasible solution $(\overline{x}, \overline{y})$, and in each step, we maintain feasibility by ensuring that a client-scenario pair is marked "served" only when its x_{ij}^k variable is set to 1 (ie, it is assigned to a facility) for some facility i.

Theorem 6. *There is a polynomial-time approximation algorithm with performance ratio* 8 *for SFL.*

Proof. Setting $\alpha = \frac{1}{4}$ and $\beta = \frac{1}{2}$, along with Theorem 5 and Lemma 2, yields the performance guarantee. The running time of the algorithm is polynomial in $|D|, |F|, m$.

Extensions The algorithm easily extends to allowing demands at client-scenario pairs which are non-negative real numbers instead of just 0 or 1. We may also allow the costs to transport one unit of demand per unit length in different scenarios to be different, motivated by, e.g., different scenarios having different prices of gas. In other words, each scenario has a multiplier γ_k such that the distance between i and j in scenario k is $\gamma_k c_{ij}$. Essentially, this can be incorporated into the demand variables d_j^k, and the rest of the algorithm proceeds as before to give identical results.

Subsequent to the first appearance of this paper, improved approximation ratios have been obtained [39, 14] for stochastic versions of the facility location problem, including versions which generalize the version considered by us.

6. Vertex cover

Problem definition We are given a first-stage (undirected) graph $G = (V, E_0)$. As usual, there are m possible scenarios, each consisting of a probability of occurrence p_k and a set of edges E_k, which may or may not be subsets of the first-stage edge set E_0. The first-stage cost of vertex v is c_v^0, and its cost in scenario k is c_v^k. The objective is to identify a set of vertices to be selected in the first stage, so that the expected cost of extending this set to a vertex cover of the edges of the realized second-stage scenario is minimized.

In the problem as defined above, edges in $E_k \setminus E_0$ must be covered in the second stage, while edges in $E_k \cap E_0$ may be covered in either stage. This is a generalization of the case when first-stage vertices cover all second-stage edges incident to them. We provide a 2-approximation for this generalized stochastic vertex cover problem.

The best known approximation algorithm for the deterministic version of vertex cover has performance ratio $2 - \frac{\log\log|V|}{2\log|V|}$, due to Monien and Speckenmeyer [34]. A lower bound of 1.16 on the hardness of approximating the problem was shown by Håstad [19]. Our approximation ratio of 2 for the generalized stochastic version of vertex cover asymptotically matches the best known approximation for the deterministic version.

Integer program formulation Our algorithm is a primal-dual algorithm which rounds the natural IP formulation of stochastic vertex cover. Variable x_v^k indicates whether or not vertex v is purchased in scenario k (where $k = 0$ as usual denotes the first stage). Recall that edges in $E_k \cap E_0$ may be covered in either the first or second stage, while edges in $E_k \setminus E_0$ must be covered in the second stage.

$$\begin{aligned} \min \sum_v c_v^0 x_v^0 + \sum_{k=1}^{m} \sum_v p_k c_v^k x_v^k & \qquad (IP_{SVC}) \\ \text{s.t. } x_u^0 + x_v^0 + x_u^k + x_v^k \geq 1 & \qquad \forall uv \in E_k \cap E_0, \forall k \\ x_u^k + x_v^k \geq 1 & \qquad \forall uv \in E_k \setminus E_0, \forall k \\ x \quad \text{non-negative integers} & \end{aligned}$$

Dual program The dual of the linear relaxation of IP_{SVC} is shown below. Variable y_e^k packs edge e in E_k if $e \in E_k$, and it packs $e \in E_0$ if $e \in E_k \cap E_0$.

$$\begin{aligned} \max \sum_{k=1}^{m} \sum_{e \in E_0 \cup E_k} y_e^k & \qquad (DP_{SVC}) \\ \text{s.t.} \sum_{e \in E_k : v \in e} y_e^k \leq p_k c_v^k & \ \forall v, \forall k \\ \sum_{k=1}^{m} \sum_{e \in E_0 \cap E_k : v \in e} y_e^k \leq c_v^0 & \quad \forall v \\ y \geq 0 & \end{aligned}$$

Algorithm The algorithm is a greedy dual-ascent type of primal-dual algorithm, with two phases. In Phase I, we raise the dual variables y_e^k uniformly for all edges in $E^k \setminus E_0$,

separately for each k. All vertices which become tight (have the first dual constraint packed to $p_k c_v^k$) have x_v^k set to 1, and deleted along with adjacent edges. We proceed this way until all edges in $E^k \setminus E^0$ are covered and deleted.

In Phase II, we do a greedy dual-ascent on all *uncovered* edges of E_k. These edges are contained in $E_k \cap E_0$. This time around, we use a slightly different rule for purchasing vertices. If a vertex is tight for x_v^0 (i.e., second dual constraint packed to c_v^0), then we select it in the stage 1 solution by setting $x_v^0 = 1$, and if it is not tight for x^0 but is tight for x^k (packed in the first dual constraint), then we select it in the recourse solution and set $x_v^k = 1$.

Theorem 7. *The integer program I P_{SVC} can be rounded by the primal-dual algorithm described above within a factor of* 2 *in polynomial time.*

Proof. We analyze the performance of the algorithm described above. Consider an edge $e = uv$ in scenario k. By definition of the algorithm, we must have selected one of its two end-points in either Phase I or Phase II (or both), so that the algorithm yields a feasible solution. We use linear programming duality to bound the cost of the solution by showing that the cost of our solution is no more than $2 \sum_k \sum_{u,v \in V} y_{uv}^k$, where y is the dual solution constructed by our algorithm.

We show the performance ratio of our algorithm as follows. Each time we set an x_v^k variable to 1, we assign some dual variables to it such that (i) the sum of dual variables assigned to each such x_v^k variable equals $p_k c^k$ (where $p_0 = 1$), and (ii) each dual variable is assigned at most twice.

Consider a vertex v which was selected (ie, x_v^k was set to 1) in scenario k in either Phase I or Phase II. We assign all dual variables y_e^k such that v is an end point of e to this vertex, and since v is selected only when the constraint $\sum_{e \in E_k : v \in e} y_e^k \leq p_k c_v^k$ goes tight, we maintain (i). An edge e in $E_k \setminus E_0$ is assigned to a vertex v only if x_v^k is set to 1 for $k \neq 0$, and since an edge has at most 2 end-points, we ensure (ii) for edges in E_k.

Next consider a vertex v for which we set x_v^0 to 1. For this to happen, the constraint $\sum_k \sum_{e \in E_0 \cap E_k : v \in e} y_e^k \leq c_v^0$ must have gone tight, and all edges in the sum are assigned to the variable x_v^0. This assignment once again ensures (i). This assignment only includes edges in $E_0 \cap E_k$, and these edges are not assigned to any variable x_v^k for $k \neq 0$, thus also ensuring (ii) for all edges in $E_0 \cap E_k$.

These two cases cover all the possibilities, thus proving the theorem.

7. Set cover

Problem definition The input in our version of the stochastic set cover problem consists of a universe U of $|U| = n$ elements, and a collection $\mathcal{S}$ of subsets of U. Each set $S \in \mathcal{S}$ has a stage 1 cost c_S^0 and a cost of c_S^k in scenario k, some of which might be infinity reflecting the unavailability of the set in certain scenarios. Each element $u \in U$ has a demand vector d_u with the k^{th} component d_u^k being 1 if it is required to cover u in scenario k, and 0 otherwise. A feasible solution is specified by a collection $\mathcal{S}' \subseteq \mathcal{S}$, with stage 1 cost $\sum_{S \in \mathcal{S}'} c_S^0$. If scenario k is realized, then $\mathcal{S}'$ must be extended by purchasing some more sets $\mathcal{S}^k$ to cover all elements with $d_u^k = 1$. The cost of this recourse solution

is $\sum_{S \in \mathcal{S}^k} c_S^k$, and we incur this with probability p_k. The objective is a solution which minimizes the sum of first stage and expected second stage costs.

Reduction to classical set cover

The deterministic version of set cover was among the earliest NP-hard problems to be approximated, with a $O(\log n)$ approximation was first provided by Johnson [23]. The problem was also shown to be NP-hard to approximate better than a factor of $\Omega(\log n)$ by Arora and Sudan [1].

Given an instance of deterministic set cover, we can define an instance of stochastic set cover by creating a distinct scenario for each element, and setting all second-stage set costs to infinity. This implies an inapproximability threshold of $\Omega(\log m)$ for stochastic set cover too.

We show below that any instance of stochastic set cover with n elements can be transformed to an instance of deterministic set cover with $n(m+1)$ elements. This means that there exists an $O(\log nm) = O(\log n + \log m)$ approximation for stochastic set cover by using this transformation and then applying any approximation algorithm for deterministic set cover. The approximation ratio therefore matches the inapproximability ratio upto constants. The reduction in Theorem 8 allows us to extend the model to the following generalization, for which the same approximation guarantee holds: In scenario k, each set S_k covers only a subset of the elements that the first-stage set S covers.

Theorem 8. *Any stochastic set cover problem is equivalent to a classical set cover problem with mn elements and $|\mathcal{S}|(m+1)$ sets.*

Proof. Associate an element u_k for every element-scenario pair (u, k) such that $d_u^k = 1$. Create $m+1$ copies of every set $S \in \mathcal{S}$. Set S^0 contains all elements u_k for all $k = 1, 2, \ldots, m$ such that $u \in S$, while set S^k only contains u_k for all $u \in S$. Finally, the cost of S^0 is c_S^0 and that of S^k is $p_k c_S^k$.

By construction, any solution to the stochastic set cover instance yields a solution to the transformed deterministic instance, and vice-versa. This proves the equivalence.

8. Conclusion

We have provided approximation algorithms for two-stage finite-scenario stochastic versions of several combinatorial optimization problems. As noted in Section 2.5, several recent papers have made substantial further progress in this area, by considering, for example, multi-stage models, models with exponentially many scenarios, the use of sampling, cost-sharing functions, etc. Given the richness of the fields of stochastic optimization as well as approximation algorithms, we believe that there is substantial potential for further work in approximation algorithms for stochastic optimization problems.

Acknowledgements. A preliminary version of this paper [36] appeared in the proceedings of the 10^{th} Integer Programming and Combinatorial Optimization Conference; we thank the participants thereof for some enlightening suggestions.

We gratefully thank Andrew Schaefer and Nan Kong of the University of Pittsburgh for introducing us to the topic of this research via [28]. We also thank two anonymous referees for their valuable comments which greatly improved this paper.

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

References

1. Arora, S., Sudan, M.: Improved low degree testing and applications. Combinatorica **23** (3), 365–426 (2003)
2. Ausiello, G., Crescenzi, P., Gambosi, G., Kann V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties. Springer-Verlag, Berlin, 1999
3. Balinski, M.L.: On finding integer solutions to linear programs. In: Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems, 1966, pp. 225–248
4. Birge, J., Louveaux, F.: Introduction to Stochastic Programming. Springer-Verlag, Berlin, 1997
5. Cornuéjols, G., Nemhauser, G., Wolsey, L.: The uncapacitated facility location problem. In: P. Mirchandani, R. Francis (eds.), Discrete Location Theory, Wiley, 1990, pp. 119–171
6. Fernandez de la Vega, W., Lueker, G.S.: Bin packing can be solved within $1 + \epsilon$ in linear time. Combinatorica **1**, 349–355 (1981)
7. Dhamdhere, K., Ravi, R., Singh, M.: On two-stage Stochastic Minimum Spanning Trees. In: Proceedings of the 11th Integer Programming and Combinatorial Optimization Conference, 2005, pp. 321–334
8. Dijkstra, E.: A note on two problems in connexion with graphs. Numerischke Mathematik **1**, 269–271 (1959)
9. Dye, S., Stougie, L., Tomasgard, A.: Approximation algorithms and relaxations for a service provision problem on a telecommunication network. Discrete Applied Mathematics **129** (1), 63–81 (2003)
10. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner treeproblem. J. Algorithms **37** (1), 66–84 (2000)
11. Guha, S., Khuller, S.: Greedy strikes back: Improved facility location algorithms. J. Algorithms **31** (1), 228–248 (1999)
12. Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R., Yener, B.: Provisioning a virtual private network: A network design problem for multicommodity flow. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 389–398 (2001)
13. Gupta, A., Pál, M.: Stochastic Steiner trees without a root. In: Proceedings of the 32nd International Colloquium on Automata Languages and Programming, 2005, pp. 1051–1063
14. Gupta, A., Pál, M., Ravi, R., Sinha, A.: Boosted Sampling: Approximation algorithms for stochastic optimization problems. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 417–426
15. Gupta, A., Pál, M., Ravi, R., Sinha, A.: What about Wednesday? Approximation algorithms for multistage-estochastic optimization. In: Proceedings of the 8th International Workshop on Approximation Algorithms for Combinatorial Optimization, 2005, pp. 86–98
16. Gupta, A., Ravi, R., Sinha, A.: An Edge in Time Saves Nine: LP Rounding Approximation Algorithms for Stochastic Network Design. In: Proceedings of the 45th Symposium on Foundations of Computer Science, 2004, pp. 218–227
17. Halperin, E., Krauthgamer, R. Polylogarithmic inapproximability. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing 2003, pp. 585–594
18. Klein Haneveld, W.K., van der Vlerk, M.H.: Stochastic Programming. Dept. of Econometrics and OR University of Groningen Netherlands, 2003
19. Håstad, J.: Some optimal inapproximability results. J. ACM **48** (4), 798–859 (2001)
20. Hayrapetyan, A., Swamy, C., Tardos, E.: Network Design for Information Networks. In: Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 933–942
21. Heitsch, H., Römisch, W. Scenario reduction algorithms in stochastic programming. Computational Optimization and Applications **24**, 187–206 (2003)
22. Immorlica, N., Karger, D., Minkoff, M. Mirrokni, V.: On the Costs and Benefits of Procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In: Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 684–693
23. Johnson, D., Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci. **9**, 256–278 (1974)

24. Coffman Jr, E., Garey, M., Johnson, D.: Approximation algorithms for bin-packing: a survey. In: D.S. Hochbaum (ed.), Approximation Algorithms for NP-hard Problems, 1997, PWS
25. Kall, P., Wallace, S.: Stochastic Programming. Wiley, 1994
26. Karger, D., Minkoff, M.: Building Steiner trees with incomplete global knowledge. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000, pp. 613–623
27. Kleinberg, J., Rabani, Y., Tardos, E.: Allocating bandwidth for bursty connections. SIAM J Comput. **30** (1), 191–217 (2000)
28. Kong, N., Schaefer, A.: A factor $1/2$ approximation algorithm for two-stage stochastic matching problems. Eur. J. Oper. Res. to appear, 2004
29. Kumar, A., Swamy, C.: Primal-dual algorithms for connected facility location problems. Algorithmica **40** (4), 245–269 (2004)
30. Lin, J.-H., Vitter, J.: ϵ-approximations with minimum packing constraint violation. In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing, 1992, pp. 771–782
31. Louveaux, F., Peeters, D.: A dual-based procedure for stochastic facility location. Oper. Res. **40**, 564–573 (1992)
32. Mahdian, M., Ye, Y., Zhang, J.: Improved approximation algorithms for metric facility location problems In: Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, 2002, pp. 229–242
33. Möhring, R., Schulz, A., Uetz, M.: Approximation in stochastic scheduling: The power of LP-based priority policies. J. ACM **46** (6), 924–942 (1999)
34. Monien, B., Speckenmeyer, E.: Ramsey numbers and an approximation algorithm for the vertex cover-problem. Acta Informatica **22**, 115–123 (1985)
35. Ravi, R., Salman, F.S.: Approximation algorithms for the traveling purchaser problem and its variants in network design. In: Proceedings of the European Symposium on Algorithms, 1999, pp. 29–40
36. Ravi, R., Sinha, A.: Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In: Proceedings of the 10th Integer Programming and Combinatorial Optimization Conference, 2004, pp. 101–115
37. Ravi, R., Sinha, A.: Multicommodity facility location. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 342–349
38. Schultz, R., Stougie, L., van der Vlerk, M.H.: Two-stage stochastic integer programming: a survey. Statistica Neerlandica **50** (3), 404–416 (1996)
39. Shmoys, D., Swamy, C.: Stochastic Optimization is (almost) as Easy as Deterministic Optimization. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004, pp. 228–237
40. Shmoys, D., Swamy, C.: Sampling-based approximation algorithms for multi-stage stochastic optimization. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, 2005, pp. 357–366
41. Shmoys, D., Swamy, C., Levi, R.: Facility location with service installation costs. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 1088–1097
42. Shmoys, D., Tardos, E., Aardal, K.: Approximation algorithms for facility location problems.In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 265–274
43. Skutella, M., Uetz, M.: Stochastic machine scheduling with precedence constraints. SIAM J. Comput. **34** (4), 788–802 (2005)
44. Vazirani, V.: Approximation Algorithms. Springer-Verlag, Berlin, 2001