Hedging Uncertainty: Approximation Algorithms for Stochastic Optimization Problems*

R. Ravi and Amitabh Sinha

Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, USA
ravi@cmu.edu, asinha@andrew.cmu.edu

Abstract. We study the design of approximation algorithms for stochastic combinatorial optimization problems. We formulate the problems in the framework of two-stage stochastic optimization, and provide nearly tight approximations. Our problems range from the simple (shortest path, vertex cover, bin packing) to complex (facility location, set cover), and contain representatives with different approximation ratios.
The approximation ratio of the stochastic variant of a typical problem is of the same order of magnitude as its deterministic counterpart. Furthermore, common techniques for designing approximation algorithms such as LP rounding, the primal-dual method, and the greedy algorithm, can be carefully 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, the reader is referred to one of the recent texts on the topic [4,19].

In a parallel development, the field of approximation algorithms [33,2] evolved to counter the prohibitive resource requirements for exact solution of NP-hard combinatorial optimization problems. Informally, these algorithms run in polynomial time and deliver a performance ratio on the 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.

* This work was supported in part by NSF grant CCR-0105548 and ITR grant CCR-0122581 (The ALADDIN project).

D. Bienstock and G. Nemhauser (Eds.): IPCO 2004, LNCS 3064, pp. 101–115, 2004.

However, as approximation algorithms become more sophisticated in scope and technique, the refrain from real-world practitioners who have the need for such algorithms 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 the stochastic version of several classical optimization problems.

2 Background

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.

Mathematical Formulation We consider the two-stage stochastic optimization problem with recourse, with an additional restriction: finitely many scenarios. This means that the future will be one of a finite set of possibilities (scenarios), and the parameters and probability of occurence of each scenario is known up-front. The mathematical formulation of this model is given below.

Vector x^0 is the set of first-stage decision variables, with constraints $Ax^0 = b$, and a cost vector is c. There are m scenarios, with the k^{th} scenario having probability of occurence p_k, cost vector q^k, and decision variables x^k. If the k^{th} scenario is realized, then the combined solution (x^0, x^k) must satisfy the constraints given by the matrix T^k and requirement vector h^k. Let P denote additional constraints such as non-negativity or integrality.

$$\begin{array}{lll} \min & c^T x^0 + \sum_{k=1}^{m} p_k (q^k)^T x^k \ (IP_S) & \\ \text{s.t.} & Ax^0 = b & \\ & T^k(x^0, x^k) = h^k & k = 1, 2, \ldots, m \\ & (x^0, x^k) \in P & k = 1, 2, \ldots, m \end{array}$$

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 [30] provide an excellent survey of two-stage stochastic integer programming, while Kong and Schaefer [20] recently provided approximation algorithms for a class of such problems.

Approximation Algorithms The *raison d'etre* for approximation algorithms is the prohibitively high running time of exact algorithms for integer program-

Problem	**Det. approx.**	**Stochastic elements**	**Our results (Apx. ratio)**	**Hardness**
Bin packing	APTAS [5]	Object sizes	APTAS	NP-comp.[5]
Shortest paths	1[7]	Sink only	5	MAX-SNP
		Sink and metric	$O(\log^2 n \log m)$	$\Omega(\log^2 n)$
Vertex cover	2[27]	Vertex weights, incidence	2	1.16[14]
Facility location	1.52[25]	Demands, facility costs	8	1.46[10]
Set cover	$O(\log n)$[17]	Set weights, inclusions	$O(\log nm)$	$\Omega(\log n)$[1]
				$\Omega(\log m)$

Fig. 1. Summary of results. We use m to denote the number of scenarios and n to refer to the number of combinatorial elements (number of vertices in graph problems and number of elements in the set cover problem).

ming and combinatorial optimization, due to their NP-completeness. Approximation algorithms run in time polynomial in the size of the input, and also provide guarantees in the form of approximation ratios. If C is a class of minimization problems and $OPT(I)$ and $A(I)$ respectively denote the value of an optimal solution and the solution output by algorithm A, then the approximation ratio $\rho(A)$ of algorithm A is defined as $\rho(A) = \max_{I \in C} \frac{A(I)}{OPT(I)}$.

While the area of approximation algorithms has been a very active field, most approximation algorithms assume complete knowledge of the input at the outset, barring a few exceptions such as scheduling problems [26,32]. Recently, and independently of us, Immorlica et al. [15] considered approximation algorithms for stochastic optimization problems with a restriction on the cost function: in the second stage, all costs go up uniformly by a factor of λ. A generalization of their model was considered by Gupta et al. [12], who provided approximation algorithms with only *sampling access* to the second-stage realization process, thereby obtaining a framework which can handle an arbitrary number of scenarios as long as there is an efficient process to sample the second stage.

Our Results We demonstrate the relevance and applicability of developing approximation algorithms for stochastic optimization. We carefully adapt existing techniques for deterministic versions of several problems to provide approximation guarantees for the stochastic versions within constant factors.

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 the stochastic approximation ratios are derived in this paper. Some of the hardness results are carried over from the underlying deterministic problems; the remaining are proved in this paper. An APTAS is an asymptotic polynomial time approximation scheme, which is an algorithm whose performance ratio approaches 1 as the number of objects increases. A problem is said to be MAX-SNP-hard [28], abbreviated MAX-SNP in the table, if there

is a constant $c > 1$ such that it is impossible to approximate the problem with performance ratio smaller than c unless $P = NP$.

Paper Outline In the sequel, we consider the five problems listed in Figure 1. We consider Stochastic Vertex Cover in the next section, and prove our results. Subsequent sections are devoted to the stochastic versions of four other problems that we study: Facility Location, Shortest Paths, Bin Packing and Set Cover. We conclude with directions for future research in Section 8.

3 Vertex Cover

We are given a first-stage (undirected) graph $G = (V, E_0)$, with m possible scenarios, each consisting of a probability of occurrence p_k and a set of edges E_k (not necessarily subsets of 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.

We require that the edges in $E_k \setminus E_0$ have to be covered in the second stage, even if one of their end-points was chosen in the first stage. This generalizes the case when first-stage vertices cover all second-stage edges incident to them. 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 [27]. A lower bound of 1.16 on the hardness of approximating the problem was shown by Håstad [14]. Our algorithm for the generalized stochastic version of vertex cover has approximation ratio 2, asymptotically matching the best known approximation for the deterministic version.

Integer Program Formulation In formulation IP_{SVC} below, variable x_v^k indicates whether or not vertex v is purchased in scenario k (where $k = 0$ denotes the first stage). 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. Observe that a 4-approximation can be easily obtained by rounding IP_{SVC} [15]. We obtain a tighter approximation ratio using a primal-dual algorithm.

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

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 $e \in E_k \cap E_0$.

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

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 from 1 to m. 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 are 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, which are contained in $E_k \cap E_0$. We raise y_e^k for all uncovered edges for $k = 0$ to m. 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 1. *The integer program IP_{SVC} can be rounded by the primal-dual algorithm described above within a factor of* 2 *in polynomial time.*

Proof. Consider an edge $e = uv$ in scenario k. We must have selected one of its two end-points in either Phase I or Phase II (or both), so 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. 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. Therefore, the constraint $\sum_l \sum_{e \in E_0 \cap E_k : v \in e} y_e^k \leq c_v^0$ 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.

4 Facility Location

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.

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

The problem is best explained by the IP formulation IP_{SFL} above. While our algorithm extends to arbitrary demands, 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).

History and Non-triviality of the Problem The classical (deterministic) uncapacitated facility location problem has a rich history (see Cornuéjols, Nemhauser and Wolsey [6] 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 using this formulation is due to Shmoys, Tardos and Aardal [31], and the current best algorithm, due to Mahdian, Ye and Zhang [25] 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 [23] considered a slightly different version of stochastic facility location, and provided a dual-ascent based exact (non-polynomial time) algorithm for it.

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); 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. We cannot treat each scenario by itself, since the different scenarios interact in utilizing first-stage facilities. Our algorithm has to carefully balance the effects of the two stages in deciding what facilities to open.

Algorithm Our approximation algorithm proceeds along the lines of the LP-rounding algorithm of Shmoys, Tardos and Aardal [31], 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 [22]. We fix a constant $0 < \alpha < 1$. For every client-scenario pair (j, k), we define its optimal fractional service cost to be $c_{jk}^* = \sum_i c_{ij} x_{ij}^k$. 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 $\sum_{i:c_{ij}\leq c^{\alpha}_{jk}} x^k_{ij} \geq \alpha$. The following theorem was also proved in [31].

Theorem 2. *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) $c^{\alpha}_{jk} \leq \frac{1}{1-\alpha}c^*_{jk}$; (ii) $\overline{x}^k_{ij} > 0 \Rightarrow c_{ij} \leq c^{\alpha}_{jk}$ for all $i \in \mathcal{F}, \quad j \in \mathcal{D}, \quad k = 1,2,\ldots,m$; (iii) $\overline{y}^k_i \leq \min\{1, \frac{y^k_i}{\alpha}\}$ for all $i \in \mathcal{F}, \quad k = 0,1,\ldots,m$.*

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): $c^*_{jk} \geq \alpha.0 + (1-\alpha)c^{\alpha}_{jk} > c^*_{jk}$, proving (i).

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

$$\overline{x}^k_{ij} = \begin{cases} \min\{1, \frac{1}{\alpha}\}x^k_{ij} \text{ if } c_{ij} \leq c^{\alpha}_{jk} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}$$

Furthermore, define $\overline{y}^k_i = \min_{j\in\mathcal{D}} \overline{x}^k_{ij}$. Using (i) and the definition of $\overline{x}$, it follows that $\overline{y}^k_i \leq \min\{1, \frac{y^k_i}{\alpha}\}$ for all $i \in \mathcal{F}$, satisfying (iii). The definitions also ensure that $(\overline{x},\overline{y})$ is a feasible solution to the LP relaxation of IP_{SFL}.

The algorithm of Shmoys, Tardos and Aardal [31] iteratively rounds x^k_{ij} variables for which c^{α}_{jk} is smallest. 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. 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^{α}_{jk}. 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 $\overline{x}^k_{ij} > 0 \wedge \overline{y}^0_i > 0$, and S^k be the set of facilities i such that $\overline{x}^k_{ij} > 0 \wedge \overline{y}^k_i > 0$.
   (a) If $\sum_{i\in S^0} \overline{y}^0_i \geq \beta$, let i be the facility such that f^0_i is smallest among all facilities in S^0. Move facility i to the set $\hat{F}^0$, and set $\hat{y}^0_i = 1$. For all

other facilities $i' \in S^0 \cup S^k$, set $\hat{y}^0_{i'} = \hat{y}^k_{i'} = 0$. For client-scenario pairs (j', k') such that there exists a facility $i' \in S^0 \cup S^k$ with $c_{i'j'} \leq c^\alpha_{j'k'}$, set $\hat{x}^{k'}_{ij'} = 1$ and mark them as "served".

(b) If $\sum_{i:i\in S^0} \overline{y}^0_i < \beta$, then we must have $\sum_{i:c_{ij}\leq c^\alpha_{jk}} \overline{y}^k_i \geq 1-\beta$. In this case, let i be the facility in S^k with smallest f^k_i. Move facility i to the set $\hat{F}^k$ and set $\hat{y}^k_i = 1$. For all other facilities $i' \in S^k$, set $\hat{y}^k_{i'} = 0$. For clients j' such that there exists a facility $i' \in S^k$ with $c_{i'j'} \leq c^\alpha_{j'k}$, set $\hat{x}^k_{ij'} = 1$ 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 $\hat{x}^k_{ij}$.

Lemma 1. *The rounding algorithm above produces an integer solution $(\hat{x}, \hat{y})$ which is feasible for IP_{SFL} such that (i) For every client-scenario pair (j, k), we have $\hat{x}^k_{ij} = 1 \Rightarrow c_{ij} \leq 3c^\alpha_{jk}$. (ii) $\sum_{i\in F} f^0_i \hat{y}^0_i \leq \frac{1}{\beta} \sum_{i\in F} f^0_i \overline{y}^0_i$. (iii) $\sum_{i\in F} f^k_i \hat{y}^k_i \leq \frac{1}{1-\beta} \sum_{i\in F} f^k_i \overline{y}^k_i$ for all $k = 1, 2, \ldots, m$.*

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

When a facility i is chosen for opening in the first stage (ie, $\hat{y}^0_i$ is set to 1), case 2(a) must have occurred. In that case, we have a sufficiently large fraction (β) of facilities which have $\overline{y}^0_i > 0$ 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^k_{ij} variable is set to 1 (ie, it is assigned to a facility) for some facility i.

Theorem 3. *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 2 and Lemma 1, yields the performance guarantee.

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. 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^k_j. Recently, Mahdian [24] developed an approximation algorithm for SFL with approximation ratio 3, by extending the ideas of Jain and Vazirani [16].

5 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 [7].

5.1 Stochastic Sink

Now consider the case when the supplier does not know the destination in advance. In particular, any of m scenarios could materialize, with the destination being t^k in scenario k. The supplier wishes to reserve some edges now at cost c_e, and augment the network in the second stage (when edges may be more expensive) after the revelation of the actual destination.

Problem Definition We are given a graph $G = (V, E)$, with metric edge costs c_e and a single source $s \in V$. 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 2. *The set of edges E' bought in the first stage in an optimal solution to SSP 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 from C to the source is the cheapest. If we re-route the paths of all scenarios in K' to using the path to using the path of k^0 from 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 to 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 to 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 from each demand node to the tree, weighted by the demand at the node. A constant-factor approximation algorithm for this problem was first provided by Ravi and Salman [29], and it has also been studied subsequently as the connected facility location problem [18,21], and the asymmetric VPN design problem [11].

Theorem 4. *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 $p_k f_k$.

The equivalence of SSP and tree-star network design also implies the NP-hardness of SSP. The best-known approximation ratio for tree-star network design is 5, due to Kumar and Swamy [21]. This implies an approximation algorithm with the same performance ratio for stochastic sink shortest paths.

5.2 Stochastic Metric and Sink

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 5, 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.

Hardness Even with the restriction that the first stage set of edges form a tree, SMSP 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 [9] who also gave an approximation algorithm with performance ratio roughly $O(\log^2 n \log m)$, and recently Halperin and Krauthgamer [13] showed an inapproximability threshold of $\Omega(\log^2 n)$ even when G is a tree. For the rest of this section, we consider the restriction of SMSP where the first stage solution has to be a tree, which we dub *Tree-SMSP*. An $\Omega(\log^2 n)$ hardness for Tree-SMSP follows from the reduction of GST to Tree-SMSP, shown below.

Theorem 5. *A GST instance 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 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 from t_k to 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 from t_k to 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.

$$\min \textstyle\sum_e c_e x_e + \sum_{k=1}^{m} p_k \sum_e r^k_e c^k_e \quad (IP_{SMSP})$$

$$\begin{aligned}
\text{s.t. } \textstyle\sum_v (r^k_{t_k,v} + f^k_{t_k,v}) &\geq 1 && \forall k \\
\textstyle\sum_v (r^k_{uv} + f^k_{uv}) &= \textstyle\sum_v (r^k_{vu} + f^k_{vu}) && \forall u \in V \setminus \{t_k, s\}, \forall k \\
\textstyle\sum_v r^k_{uv} &\leq \textstyle\sum_v r^k_{vu} && \forall u \in V \setminus \{t_k\}, \forall k \\
f^k_e &\leq x_e && \forall e \in E, \quad \forall k \\
f, r, x & \quad \text{non-neg. integers}
\end{aligned}$$

The third set of inequalities are strengthenings valid only for the tree version of SMSP, insisting that flows along recourse arcs from t_k to s via any node are non-increasing; 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_{SMSP} be its value. The following theorem describes our rounding algorithm.

Theorem 6. *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 $r^*(k) = \sum_e r_e^k c_e^k$ 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 from t_k to 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, if $s \notin S_k$, we have $\sum_{e=(u,v):u \in S_k, v \notin S_k} x_e \geq \frac{1}{2}$. Hence $2x$ is a solution to the LP relaxation of the following IP formulation of the group Steiner tree problem: $\min \sum_e c_e x_e$ such that $\sum_{e=(u,v):u \in S, v \notin S} x_e \geq 1 \ \forall S \ \exists k : \ S_k \subseteq S$. Using the result of Garg, Konjevod and Ravi [9], we can construct an integer tree solution $\hat{x}$ at a cost $O(\log^2 n \log m \cdot OPT_{SMSP})$ which 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 at most $2r^*(k)$, our approximation ratio follows.

6 Stochastic Bin Packing

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 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 number of bins at cost f_k per bin. The objective is to compute x so as to minimize the expected total cost. Let $[x]$ denote the integer nearest to x.

Let ρ denote the approximation ratio of the best approximation algorithm for the bin-packing problem. Any locally optimal algorithm (first-fit, for example) achieves $\rho = 2$. An asymptotic PTAS was given by Fernandez de la Vega and Lueker [8], which uses at most $(1+2\epsilon)OPT+1$ bins. The following theorem shows how to extend any bin-packing algorithm to handle stochastic bin-packing.

Theorem 7. *Order the scenarios so that we have $\sum_i s_i^1 \geq \sum_i s_i^2 \geq \ldots \geq \sum_i s_i^m$. Let k^* be the largest integer such that $\sum_{k=1}^{k^*} f_k p_k \geq 1$. Then $x = [\rho \sum_i s_i^{k^*}]$ is an asymptotic ρ-approximate solution.*

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

7 Stochastic Set Cover

The input in 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. 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 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 with some more sets $\mathcal{S}^k$ to cover all elements with $d_u^k = 1$. The cost of this recourse solution is $\sum_{S \in \mathcal{S}^k} c_S^k$, incurred with probability p_k.

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 [17]. 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. 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.

8 Directions for Further Research

Much remains to be done on several classical problems for which algorithms in the two-stage stochastic model are not known. Another direction of research is to develop approximations for more complex models of stochastic optimization: The extension of the two-stage model to multiple stages allows more detailed modeling; the variant where uncertainty is modeled by a continuous distribution is also often considered. It is our hope that these models will provide a rich setting for the application of optimization in practice.

9 Acknowledgments

We would like to thank Nan Kong and Andrew Schaefer of the University of Pittsburgh for several enlightening discussions leading to this work.

References

1. Arora, S., Sudan, M. Improved low degree testing and its applications. In Proceedings of the 29^{th} Annual ACM Symposium on Theory of Computing (1997) 485-495.
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, Berlin, Germany (1999).
3. Balinski, M.L.: On finding integer solutions to linear programs. In Proc. IBM Scientific Computing Symposium on Combinatorial Problems (1966) 225-248.
4. Birge, J., Louveaux, F.: Introduction to Stochastic Programming, Springer, Berlin (1997).
5. Coffman Jr., E., Garey, M., Johnson, D.: Approximation algorithms for bin-packing: a survey. In D.S. Hochbaum, Approximation Algorithms for NP-hard Problems, PWS, Boston (1997).
6. Cornuéjols, G., Nemhauser, G., Wolsey, L.: The uncapacitated facility location problem. In P. Mirchandani and R. Francis, eds, Discrete Location Theory, Wiley, New York (1990) 119-171.
7. Dijkstra, E.: A note on two problems in connexion with graphs. Numerische Mathematik **1** (1959) 269-271.
8. Fernandez de la Vega, W., Lueker, G.S.: Bin packing can be solved within $1 + \epsilon$ in linear time. Combinatorica **1** (1981) 349-355.
9. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner tree problem. Journal of Algorithms **37(1)** (2000) 66-84.
10. Guha, S., Khuller, S.: Greedy strikes back: Improved facility location algorithms. In Proceedings of the 9^{th} ACM-SIAM Symposium on Discrete Algorithms (1998) 649-657.
11. 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 33^{rd} Annual ACM Symposium on Theory of Computing (2001) 389-398.
12. Gupta, A., Pál, M., Ravi, R., Sinha, A.: Boosted sampling: Approximation algorithms for stochastic optimization. Proceedings of the 36^{th}, Annual ACM Symposium on Theory of Computing (2004) (to appear).

13. Halperin, E., Krauthgamer, R.: Polylogarithmic inapproximability. In Proceedings of the 35^{rd} Annual ACM Symposium on Theory of Computing (2003) 585-594.
14. Håstad, J.: Some optimal inapproximability results. In Proceedings of the 29^{th} Annual ACM Symposium on Theory of Computing (1997) 1-10.
15. 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 15^{th} Annual ACM-SIAM Symposium on Discrete Algorithms (2004) 684-693.
16. Jain, K., Vazirani, V.: Primal-dual approximation algorithms for metric facility location and k-median problems. In Proceedings of the 40^{th} Annual IEEE Symposium on Foundations of Computer Science (1999) 2-13.
17. Johnson, D.: Approximation algorithms for combinatorial problems. Journal of Computer and System Sciences **9** (1974) 256-278.
18. Karger, D., Minkoff, M.: Building Steiner trees with incomplete global knowledge. In Proceedings of the 41^{st} Annual IEEE Symposium on Foundations of Computer Science (2000) 613-623.
19. Klein Haneveld, W.K., van der Vlerk, M.H.: Stochastic Programming, Dept. of Econometrics and OR, University of Groningen, Netherlands (2003).
20. Kong, N., Schaefer, A.: A factor $\frac{1}{2}$ approximation algorithm for a class of two-stage stochastic mixed-integer programs. Manuscript, submitted to INFORMS Journal of Computing (2003).
21. Kumar, A., Swamy, C.: Primal-dual algorithms for connected facility location problems. In Approximation Algorithms for Combinatorial Optimization (2002) 256-270.
22. Lin, J-H., Vitter, J.: ϵ-approximations with minimum packing constraint violation. In Proceedings of the 24^{th} Annual ACM Symposium on Theory of Computing (1992) 771-782.
23. Louveaux, F., Peeters, D.: A dual-based procedure for stochastic facility location. Operations Research **40** (1992) 564-573.
24. Mahdian, M.: Personal communication (2003).
25. Mahdian, M., Ye, Y., Zhang, J.: A 1.52 approximation algorithm for the uncapacitated facility location problem. In Approximation Algorithms for Combinatorial Optimization (2002) 229-242.
26. Möhring, R., Schulz, A., Uetz, M.: Approximation in stochastic scheduling: The power of LP-based priority policies. Journal of the ACM **46(6)** (1999) 924-942.
27. Monien, B., Speckenmeyer, E.: Ramsey numbers and an approximation algorithm for the vertex cover problem. Acta Informatica **22** (1985) 115-123.
28. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. Journal of Computer Systems and Sciences **43** (1991) 425-440.
29. Ravi, R., F.S. Salman, F.S.: Approximation algorithms for the traveling purchaser problem and its variants in network design. In European Symposium on Algorithms (1999) 29-40.
30. Schultz, R., Stougie, L., van der Vlerk, M.H.: Two-stage stochastic integer programming: A survey. Statist. Neerlandica **50(3)** (1996) 404-416.
31. Shmoys, D., Tardos, E., Aardal, K.: Approximation algorithms for facility location problems. In Proceedings of the 29^{th} ACM Symposium on Theory of Computing (1997) 265-274.
32. Skutella, M., Uetz, M.: Scheduling precedence-constrained jobs with stochastic processing times on parallel machines. In Proceedings of the 12^{th} Annual ACM-SIAM Symposium on Discrete Algorithms (2001) 589-590.
33. Vazirani, V.: Approximation Algorithms, Springer, Berlin, Germany (2001).