Computational complexity of stochastic programming problems *

Martin Dyer† Leen Stougie‡

March 4, 2005

Abstract

Stochastic programming is the subfield of mathematical programming that considers optimization in the presence of uncertainty. During the last four decades a vast quantity of literature on the subject has appeared. Developments in the theory of computational complexity allow us to establish the theoretical complexity of a variety of stochastic programming problems studied in this literature. Under the assumption that the stochastic parameters are independently distributed, we show that two-stage stochastic programming problems are ♯P-hard. Under the same assumption we show that certain multi-stage stochastic programming problems are PSPACE-hard. The problems we consider are non-standard in that distributions of stochastic parameters in later stages depend on decisions made in earlier stages.

1 Introduction

Stochastic programming is the subfield of mathematical programming that considers optimization problems under uncertainty. During the last four decades a vast amount of literature on the subject has appeared. The two most comprehensive textbooks that have appeared on the subject are [1] and [12]. An easily accessible introduction is given in [8]. In this paper we will give theoretical evidence that stochastic programming problems are in general hard to solve, even harder than most well known combinatorial optimization problems. This confirms the feelings that researchers in stochastic programming have always had. We determine the complexity of what is known in stochastic programming literature as two-stage decision problems. The general formulation is given by

$$\max\{c^T x + Q(x) \mid Ax \leq b, x \in X\},$$

with

$$Q(x) = E[\max\{\mathbf{q}^T y \mid Wy \leq \mathbf{h} - \mathbf{T}x,\ y \in Y\}],$$

and $X \subset \mathbb{R}^n_{\geq 0}$ specifying non-negativity and possibly integrality restrictions on the decision variables x. Similarly $Y \subset \mathbb{R}^{n_1}_{\geq 0}$ is defined for the decision variables y. All variables, matrices and vectors have consistent dimensions. Boldface characters are used to indicate randomness. Realisations of random variables are written in normal font.

*Supported by the EPSRC grant "Phase Transitions in the Complexity of Randomised Algorithms", by the EC IST project RAND-APX, and by the TMR Network DONET of the European Community (ERB MRX-CT98-0202)

†School of Computing, University of Leeds, England, dyer@comp.leeds.ac.uk

‡Department of Mathematics and Computer Science, Technische Universiteit Eindhoven and CWI, Amsterdam, The Netherlands, leen@win.tue.nl

In the stochastic programming literature this model originates from a linear or mixed integer linear programming model with uncertain parameter values:

$$\max\{c^T x \mid Ax \leq b,\ \mathbf{T}x \leq \mathbf{h},\ x \in X\}.$$

It is assumed that the imprecise information about T and h can be represented by random objects with known probability distributions. Possible infeasibilities due to a choice of x together with a realization T of $\mathbf{T}$ and h of $\mathbf{h}$ are compensated in a *second-stage recourse action* by choosing second-stage decisions y as optimal decisions of the second-stage problems

$$\max_y\{q^T y \mid Wy \leq h - Tx,\ y \in Y\},$$

where q is a realization of the random cost vector $\mathbf{q}$, and the recourse matrix W specifies the available technology. Though in the most general stochastic programming model W is also a random matrix, we comply with the majority of stochastic programming literature in assuming it to be fixed (fixed recourse).

In the two-stage decision model, $Q(x)$ is the expected optimal recourse cost associated with a first-stage decision x, often called the expected value function of the two-stage stochastic programming problem, as we will call it here.

Two-stage stochastic programming models need not reflect a decision situation with recourse actions. They may also appear as a result of a decision situation with different levels. At an aggregate level a decision has to be taken while precise detailed information is not available. At the detailed level decisions then have to be made given the aggregate decision and given the precise information. For example, think of a situation in project planning, in which at an aggregate level resources, like machines and man power, have to be acquired or hired, whereas the precise data about what will eventually be required of the resources are not available. Just before the project is to be executed, detailed information becomes available and an optimal plan has to made, given the resources acquired. Problems of this type appear in the literature under the name of *hierarchical planning problems* (see e.g. [2], [17]).

The two-stage decision problem models a situation in which all information that was imprecise at the first stage is dissolved at the same moment, so one has complete information in the second stage. The situation in which information becomes available in more than one stage can be described by a multi-stage decision problem. We come back to this problem in Section 4. Before that section we concentrate on the two-stage decision problem.

Structural properties of the objective functions of two-stage stochastic programming problems are known. Under some mild conditions on the distribution of the random parameters, the objective function is a convex function of x, if the second-stage decision variables y are real valued and the set Y is convex. The convexity of the objective function is lost if integrality conditions are imposed on y, in which case we speak of stochastic integer or stochastic mixed integer programming. For an overview of properties of objective functions of stochastic linear programming problems and stochastic (mixed) integer programming problems, like continuity, convexity, differentiability, we refer to e.g. [1], [13], [21].

The complexity of a problem, in terms of time or space to solve it, is related to input size. For each instance, a bound on the number of elementary computer operations or on the number of computer storage units required to solve the problem instance as a function of the size of its input indicates, respectively, the time or space complexity of the problem. We will see that the way in which the random parameters in stochastic programming problems are described has an important impact on the problem complexity. To illustrate this crucial item we start by studying in Section 2 a problem which in stochastic programming literature is called the *deterministic equivalent problem*. This study shows that, under a specific random model on the parameters, two-stage stochastic programming problems are either easy or NP-*hard*, depending only on the absence or presence, respectively, of integer decision variables.

Different random models lead to considerably different complexities. For example, we notice that the evaluation of the function Q at a single point of its domain requires the computation of a multiple integral, when the random parameters are continuously distributed. Most of the stochastic programming literature on this subclass of problems is concerned with ways to get around this obstacle. Indeed, it will appear in this paper that it is this feature that dominates the complexity of the problems. For two-stage problems the evaluation of Q is ♯P-*hard*.

The class ♯P consists of counting problems, for which membership in the set of items to be counted can be decided in polynomial time. We notice that strictly following this definition of ♯P, none of the stochastic programming problems we study can belong to this complexity class. We use the term ♯P-hard for an optimization problem in the same way as NP-hard is used for optimization problems whose recognition version is NP-complete. For an exposition of the definitions and structures of the various complexity classes we refer to [6] and [11].

The complexity of *two-stage decision problems* under various random models on the parameters is presented in Section 3. In Section 4 some first steps in the study of the complexity of *multi-stage decision problems* are discussed. In particular, we show that under a specific random model on the stage by stage uncertain parameters, where the distribution in later stages may depend on decisions taken earlier, multi-stage decision problems are PSPACE-*hard*. Through personal communication with people in the stochastic programming community we have learnt that our model is a very reasonable and practically relevant one. Yet it is not the standard model in stochastic programming literature, in which the distributions are independent of decisions taken in earlier stages. The complexity of this standard model remains open. We conjecture that it is PSPACE-hard as well.

2 Complexity of the deterministic equivalent problem

Many solution methods for stochastic programming problems begin by formulating the so-called *deterministic equivalent problem* (see e.g. [1],[22]). The basic assumption for formulating this problem is that the realisations of the random parameters are specified in the form of *scenarios*. Each scenario contains a complete description of $(\mathbf{q}, \mathbf{T}, \mathbf{h})$ values in one realization. Thus, the scenarios are enumerated $(q^1, T^1, h^1), (q^2, T^2, h^2), \ldots, (q^K, T^K, h^K)$, with K denoting the total number of possible realisations of $(\mathbf{q}, \mathbf{T}, \mathbf{h})$. Each realization (q^k, T^k, h^k) has a probability p^k of occurrence. The problem can now be formulated as

$$\begin{array}{lllll} \max & c^T x + \displaystyle\sum_{k=1}^{K} p^k (q^k)^T y^k & & & \\ s.t. & Ax & & \leq b, & \\ & T^k x & + \quad W y^k & \leq h^k, & k = 1, \ldots, K. \end{array}$$

If, as input of the problem, each scenario and its corresponding probability has to be specified completely, then the input size of the problem is just the size of the binary encoding of all the parameters in this deterministic equivalent problem and hence the problem is polynomially solvable in case the decision variables have a convex feasible region and NP-complete if there are integrality constraints on the decision variables.

However, consider another extreme in which all parameters are independent identically distributed random variables, each having a value α_1 with probability p and α_2 with probability $1 - p$. In that case, using m_1 for the number of rows of the T-matrices, there are $K = 2^{n_1+m_1n+m_1}$ possible scenarios, but they can be encoded with a relatively small number of bits. The size of the deterministic equivalent problem is exponential in the size of the input, and the complexity changes correspondingly, as we will show below. Indeed, most of stochastic programming research focusses on methods to overcome this curse of problem size, which is usually caused by specification of the scenarios as combinations of realisations of independent random parameters. For example, the *sample average approximation* method does not require a full listing of all scenarios. For a survey of models and methods we refer to [14]. Thus, from now on we will consider models wherein the random parameters are independently distributed.

3 Complexity of two-stage decision problems

We will treat models with discretely distributed random parameters and with continuously distributed ones separately.

3.1 Discrete distributions

We will establish ♯P-hardness of the evaluation of the second-stage expected value function $Q(x)$ for fixed x of a two-stage stochastic programming problem with discretely distributed parameters using a reduction from the problem GRAPH RELIABILITY, ♯P-completeness of which has been proved in [20].

Definition 3.1 GRAPH RELIABILITY. *Given a directed graph with m edges and n vertices, determine the reliability of the graph, defined as the probability that two given vertices u and v are connected, if each edge fails independently with probability 1/2.*

This is equivalent to the problem of counting the number of subgraphs, from among all 2^m possible subgraphs, that contain a path from u to v.

Theorem 3.1 *Two-stage stochastic programming with discrete distributions on the parameters is* ♯P*-hard.*

PROOF. Take any instance of GRAPH RELIABILITY, i.e. a network $G = (V, A)$ with two fixed vertices u and v in V. Introduce an extra edge from v to u, and introduce for each edge $(i, j) \in A$ a variable y_{ij}. Give each edge a random weight $\mathbf{q}_{ij}$ except for the edge (v, u) that gets a deterministic weight of 1. Let the weights be independent and identically distributed (i.i.d.) with distribution $\Pr\{\mathbf{q}_{ij} = -2\} = \Pr\{\mathbf{q}_{ij} = 0\} = 1/2$. The event $\{\mathbf{q}_{ij} = -2\}$ corresponds to failure of the edge (i, j) in the GRAPH RELIABILITY instance. If, for a realization of the failures of the edges, the network has a path from u to v, then there is a path from u to v consisting of edges with weight 0 only and vice versa.

Denote $A' = A \cup (v, u)$. Now define the two-stage stochastic programming problem:

$$\max\{-cx + Q(x) \mid 0 \le x \le 1\}$$

with

$$\begin{aligned} Q(x) = E[\max\{\sum_{(i,j)\in A} \mathbf{q}_{ij} y_{ij} + y_{vu} \mid \sum_{i:(i,j)\in A'} y_{ij} - \sum_{k:(j,k)\in A'} y_{jk} &= 0 \ \forall j \in V, \\ y_{ij} &\le x \ \forall (i,j) \in A\}], \end{aligned}$$

where c is a parameter.

Suppose that for a realization of the failures of the edges there is a path from u to v in the network. As we argued the costs $q_{ij} = 0$ for edges (i, j) on the path. For such a realization, the optimal solution of the second-stage problem, is obtained by setting all y_{ij}'s corresponding to edges (i, j) on this path and y_{vu} equal to x, their maximum feasible value, and setting $y_{ij} = 0$ for all (i, j) not on the path. This yields solution value x for this realization.

Suppose that for a realization the graph does not have a path from u to v, implying in the reduced instance that on each path there is an edge with weight -2 and vice versa, then the optimal solution of the realized second-stage problem is obtained by setting all y_{ij}'s equal to 0, and also $y_{vu} = 0$, yielding solution value 0.

Therefore, the network has reliability R if and only if $Q(x) = Rx$. This implies immediately that evaluation of Q in a single point $x > 0$ is ♯P-hard. We continue to prove that the two-stage stage problem is ♯P-hard (it is not excluded that finding the optimal solution to a two-stage problem requires any evaluation of the objective function).

Notice that $Q(x) = Rx$ implies that the objective function value of the two-stage problem is $(R - c)x$. Thus, if $c \leq R$ then the optimal solution is $x = 1$ with value $(R - c)$, and if $c \geq R$ then the optimal solution is $x = 0$ with value 0. Since R can take only 2^m possible values, by performing a bisection search we can compute the exact value of R by solving only m two-stage stochastic programming problems to know the exact value of R. Thus, if one could solve the two-stage stochastic programming problem then one could solve the ♯P-hard Graph reliability problem. □

The second stage of the two-stage stochastic programming problem used in the proof is not a recourse problem. A similar reduction shows that also the more special class of two-stage stochastic recourse problems are ♯P-hard.

By total unimodularity of the restriction coefficients matrix [15] in the proof, the same reduction shows that the two-stage stochastic integer programming problem with discretely distributed parameters, i.e. the problem in which second-stage decision variables are restricted to have integer values, is ♯P-hard. We notice that imposing restrictions $y_{ij} \in \{0, 1\}$ on all second-stage variables will give $Q(x) = R\lfloor x \rfloor$. Thus, evaluating $Q(x)$ is only ♯P-hard when $x = 1$, but finding the optimal value for x is still ♯P-hard.

In the two-stage linear programming problem evaluation of Q at any point x is ♯P-*easy*, since for any realization of the second-stage random parameters a linear program remains to be solved. Given a ♯P-oracle for evaluating Q at any point x, solving two-stage stochastic linear programming problems (with discretely distributed random variables) will require a polynomial number of consultations of the oracle, since Q is a concave function in x, and maximizing a concave function over a convex set is known to be easy [7]. Thus, two-stage stochastic linear programming is in the class $P^{\sharp P}$, which is essentially equivalent to ♯P [11].

Given a ♯P-oracle for evaluating Q at any point x, a two-stage stochastic integer programming problem lies in NP. In this case the expected value function is in general not convex but discontinuous piecewise linear with a finite number of points x that are candidate for optimality (see [17]). Thus, two-stage stochastic integer programming is in the class $NP^{\sharp P} = P^{\sharp P}$ [19].

3.2 Continuous distributions

For two-stage stochastic programming problems with continuously distributed parameters, ♯P-hardness of an evaluation of the expected value function Q can be established under even the mildest conditions on the distributions. For the proof we use a reduction from the problem of computing the volume of the *knapsack polytope*, ♯P-completeness of which has been proved in [3].

Definition 3.2 Volume *of* knapsack polytope. *Given the polytope* $P = \{x \in [0, 1]^n \mid \alpha^T x \leq \beta\}$*, for given* $\alpha \in \mathbb{R}^n_+$ *and* $\beta \in \mathbb{R}_+$*, compute its volume* $\mathrm{Vol}(P)$.

Theorem 3.2 *Evaluation of $Q(x)$ of a two-stage stochastic programming problem with continuously distributed parameters is $\sharp$P-hard, even if all stochastic parameters have the uniform $[0,1]$ distribution.*

PROOF. Given an instance of a knapsack polytope, define i.i.d. random variables $\mathbf{q}_1,\ldots,\mathbf{q}_n$ with distribution uniform on $[0,1]$. Now, consider the following two-stage stochastic programming problem with continuously distributed parameters:

$$\max\{-cx+Q(x) \mid 0\le x\le 1\},$$

$$Q(x)=E[\max\{\sum_{j=1}^{n}\mathbf{q}_j y_j-\beta y \mid 0\le y\le x,\ 0\le y_j\le \alpha_j y,\ j=1,\ldots,n\}].$$

For any realization $q_1,\ldots,q_n$ of $\mathbf{q}_1,\ldots,\mathbf{q}_n$, the optimal solution of the second-stage problem in the formulation above is either $y=y_j=0,\ j=1,\ldots,n$, in case $\sum_{j=1}^{n} q_j\alpha_j\le\beta$, or $y=x,\ y_j=\alpha_j x,\ j=1,\ldots,n$, otherwise. Thus, using the notation $(s)^+=\max\{0,s\}$,

$$Q(x)=E[(\sum_{j=1}^{n}\alpha_j\mathbf{q}_j-\beta)^+x]=(1-\mathrm{Vol}(P))x.$$

Hence, the objective function of the two-stage problem is $(1-\mathrm{Vol}(P)-c)x$, which has solution $x=0$ with value 0 if $c\ge 1-\mathrm{Vol}(P)$ and $x=1$ with value $1-\mathrm{Vol}(P)-c$ if $c\le 1-\mathrm{Vol}(P)$. Again, by performing a bisection search, we can approximate $\mathrm{Vol}(P)$ to within any accuracy ϵ by solving $O(\log\frac{1}{\epsilon})$ two-stage stochastic programming problems, thus solving a $\sharp$P hard problem [4]. □

Membership of this problem in $\sharp$P would require additional conditions on the input distributions. We note that a result of Lawrence[9] shows that exact computation may not even be in PSPACE.

4 Complexity of multi-stage stochastic programming problems

We will treat the complexity of the multi-stage stochastic programming problem under discrete distributions only. For easy reference we write out the S+1-stage decision problem

$$\max\{c^Tx+Q^1(x) \mid Ax\le b, x\in X\},$$

with

$$Q^s(y^{s-1})=E[\max\{\mathbf{q}^s y^s+Q^{s+1}(y^s) \mid W^s y^s\le \mathbf{T}^s y^{s-1}-\mathbf{h}^s,\ y^s\in Y^s\}],$$

for $s=1,2,\ldots,S-1$, interpreting $y^0=x$, and

$$Q^S(y^{S-1})=E[\max\{\mathbf{q}^S y^S \mid W^S y^S\le \mathbf{T}^S y^{S-1}-\mathbf{h}^S,\ y^S\in Y^S\}].$$

If $K=S+1$, the number of stages, is part of the input, we argue that the K-stage decision problem with discrete decision-dependent random parameters is in PSPACE. Consider the last stage, stage $S+1$. Take any fixed point y^{S-1} for the decision variables of the last but one stage. Then, for any realization q_k^S, T_k^S and h_k^S, of the random parameters, where k ranges over all possible joint realisations, and for each k its probability is denoted by p_k^S, a linear or integer linear programming problem is to be solved, which can be done in polynomial space. Denote its optimal solution value by $Z_k^S(y^{S-1})$. Now, $Q^S(y^{S-1}) = \sum_k p_k^S Z_k^S(y^{S-1})$, so that Q^S can be evaluated in polynomial space. If y^{S-1} is bounded integral valued then it is clear that the solution of the $S+1$-th stage can be done in polynomial space. In the case of continuous problems the function Q^S is convex, so that the ellipsoid method guarantees that the number of points in which it is to be evaluated is polynomially bounded.

We show that a version of the problem, in which distributions in later stages depend on decisions made in earlier stages is PSPACE-hard. We call this the multi-stage stochastic programming problem with decision-dependent distributions. This is not the standard definition of the multi-stage decision problem that appears in the stochastic programming literature. The complexity of the standard problem remains open, though we expect it to be PSPACE-hard as well. We include the complexity of the non-standard formulation of the problem, since it gives some idea about the difficulty of multi-stage problems, and since stochastic programming researchers have indicated in personal communication that this model is actually of interest for practical applications.

The complexity result follows rather directly from the PSPACE-hardness of problems that are called decision-making under uncertainty by Papadimitriou [10]. These problems are roughly characterized by dynamic decision making, in which decisions are based on the current state, and the next state is a random variable or object with distribution depending on the current decision, much in the same way as classical Markov Decision problems. We selected the problem called Dynamic graph reliability [10] for our reduction here.

Definition 4.1 Dynamic graph reliability. *We are given a directed acyclic graph with n vertices and m edges, which is to be traversed from some specific vertex to some other specific vertex while edges may fail. At any moment, the probability that edge e fails before the next move is $p(e, w)$, where w is the vertex that is currently being visited. Find the strategy that maximizes the probability of successful traversal.*

Theorem 4.1 *Multi-stage stochastic programming with discretely decision-dependent distributed parameters is* PSPACE*-hard.*

Proof. Take any instance of Dynamic graph reliability, i.e. a network $G = (V, A)$ with two fixed vertices u_0 and v_0. We introduce an extra vertex u_* and an extra edge (u_*, u_0). Define the number of stages $S+1$ as the length of the longest u_*, v_0-path in G. It is well known that this path can be found in polynomial time since the graph is a directed acyclic graph. Suppose this path is $(u_*, u_0, v_1, \ldots, v_{S-1}, v_0)$. The vertices are partitioned into subsets $V_0, V_1, V_2, \ldots, V_{S+1}$ with $V_0 = \{u_*\}$, $V_1 = \{u_0\}$, $V_{S+1} = \{v_0\}$, $v_s \in V_{s+1}$ $(s = 1, \ldots, S-1)$, and all other vertices in subsets in such a way that for any edge $(i, j) \in A$ we have $i \in V_s$ and $j \in V_t$ with $s < t$. If in such a case $t > s+1$ we introduce an *auxiliary chain*, with vertices $i_1, \ldots, i_{t-s-1}$, adding i_k to V_{s+k}, $k = 1, \ldots, t-s-1$, and edges $(i, i_1), \ldots, (i_{t-s-1}, j)$, adding them to A and deleting the edge (i, j) from A. In this way we obtain a layered network $G' = (V', A')$, i.e., a network with $S+2$ layers of vertices, the sets $V_0, V_1, \ldots, V_{S+1}$, in which for each edge $(i, j) \in A'$, $i \in V_s$ and $j \in V_{s+1}$ for some $0 \le s \le S$. G' is obviously directed acyclic also. Thus, a partial order $(<)$ can be defined on the edges in A', indicating for any two edges if, and in which order, they appear on a path from u_* to v_0. On any auxiliary chain we allow only the first edge to fail in order to maintain the reliability of the original graph. Thus, for any $v \in V$, we define failure probabilities $\tilde{p}((i, i_1), v) = p((i, j), v)$, $\tilde{p}((i_k, i_{k+1}), v) = 0$ $(k = 1, \ldots, t-s-2)$ and $\tilde{p}((i_{t-s-1}, j), v) = 0$. For any edge $e \in A \cap A'$ and $v \in V$ probabilities remain unchanged: $\tilde{p}(e, v) = p(e, v)$. For any auxiliary vertex i_k and $(u, v) \in A'$, we define $\tilde{p}((u, v), i_k) = 0$.

Let $A'_s = \{(u, v) \in A' : u \in V_s, v \in V_{s+1}\}$. Note that $A'_0 = \{(u_*, u_0)\}$. For any edge $(u, v) \in A'_s$, we introduce a variable y_{uv}^s indicating the amount of flow through (u, v). The superscript s is clearly redundant here, but we use it for clarity. Now we can formulate the multi-stage programming problem as follows.

$$\max\left\{-cx+Q^1(x)\mid 0\le x\le 1,\ y^0_{u_*u_0}=x\right\},$$

and, for $s=1,2,\ldots,S-1$,

$$\begin{aligned}Q^s(y^{s-1}) = E\big[\max\big\{Q^{s+1}(y^s)\mid\ & \textstyle\sum_{v\in V_s} y^s_{uv}=\sum_{t\in V_{s-2}} y^{s-1}_{tu}\quad(\forall u\in V_{s-1}),\\ & y^s_{uv}\le \mathbf{h}_{ij,uv}(\forall (u,v)\in A'_s,\ (i,j)<(u,v))\big\}\big],\end{aligned}$$

and

$$\begin{aligned}Q^S(y^{S-1}) = E\big[\max\big\{\textstyle\sum_{u\in V_S} y^S_{uv_0}\mid y^S_{uv_0} &= \textstyle\sum_{t\in V_{S-1}} y^{S-1}_{tu}\ (\forall u\in V_S),\\ y^S_{uv_0} &\le \mathbf{h}_{ij,uv_0}\ (\forall (u,v_0)\in A'_S,\ (i,j)<(u,v_0))\big\}\big].\end{aligned}$$

For any two edges $(i,j)\in A'_s,(u,v)\in A'_t$, we define the random variable $\mathbf{h}_{ij,uv}$ as follows. If $(i,j)<(u,v)$ and $y^s_{ij}>0$, then $\Pr\{\mathbf{h}_{ij,uv}=0\}=\tilde{p}((u,v),j)$. In all other cases, $\mathbf{h}_{ij,uv}=1$.

As in the proof of Theorem 4.1, the graph has dynamic reliability R if and only if $Q^1(x)=Rx$. Hence the optimal objective function is $(R-c)x$, and at most m $(S+1)$-stage stochastic programs need to be solved to determine the exact value of R. □

5 Postlude

The complexity of two-stage stochastic programming is settled in this paper, and we provide an indication that m-stage problems are even harder. These results support the claim of many stochastic programming researchers that their problems are harder than most discrete optimization problems. The complexity of the *standard* multi-stage stochastic programming problem remains open. We conjecture that this is also PSPACE-hard.

In [5] and [16] randomized approximation schemes have been designed for two-stage stochastic programming problems. In [5] problems are considered with continuously distributed parameters and continuous decision variables, when the input distributions are restricted to be *log-concave*. Their scheme is fully polynomial under the strong condition that there is a polynomial bound on the maximum value of the second-stage value function. They give an approximate solution and a corresponding approximate solution value.

The paper [16] considers problems with both discrete and continuous distributions. The distributions are given implicitly and can be accessed only through a sampling oracle. Therefore they are not considered as part of the input. Recourse problems are considered in which first and second-stage actions are equal, but there are different costs in the two stages. The scheme proposed is fully polynomial if there is a polynomial bound on the ratio between first and second-stage cost coefficients. Moreover it is shown that such a fully polynomial scheme cannot exist if this ratio is part of the problem input. The scheme gives just an approximate solution, not an approximate value.

Both schemes rely heavily on the convexity of Q, and therefore do not apply to two-stage stochastic *integer* programming problems.

The latter paper is one in a sequence culminating from a very recent growth in interest in stochastic programming of the theoretical computer science community. An overview of approximation algorithms in stochastic programming and their performance analysis is given in [18].

Acknowledgement. We gratefully acknowledge the extensive and detailed comments on an earlier version of this paper by an anonymous referee, which have helped us to improve the exposition.

References

[1] J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, New York, 1997.

[2] M. A. H. Dempster, M. L. Fisher, L. Jansen, B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan. Analysis of heuristics for stochastic programming: results for hierarchical scheduling problems. *Mathematics of Operations Research*, 8:525–537, 1983.

[3] M. E. Dyer and A. M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM Journal on Computing*, 17:967–974, 1988.

[4] M. E. Dyer and A. M. Frieze. Computing the volume of convex bodies: A case where randomness provably helps. In *Symposia in Applied Mathematics*, volume 44, pages 123–169. American Mathematical Society, 1991.

[5] M. E. Dyer, R. Kannan, and L. Stougie. A simple randomised algorithm for convex optimization. SPOR-Report 2002-05, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2002. http://www. win. tue. nl/math/bs/spor/2002-05. pdf.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[7] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, New York, 1988.

[8] P. Kall and S. W. Wallace. *Stochastic Programming*. Wiley, Chichester, 1994.

[9] J. Lawrence. Polytope volume computation. *Math. Comput.*, 57:259–271, 1991.

[10] C. H. Papadimitriou. Games against nature. *Journal of Computer and System Sciences*, 31:288–301, 1985.

[11] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[12] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, 1995.

[13] R. T. Rockafellar and R. J-B. Wets. *Variational Analysis*. Springer-Verlag, New York, 1997.

[14] A. Ruszczynski and A. Shapiro, editors. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. North-Holland, 2003.

[15] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.

[16] D. B. Shmoys and C. Swamy. Approximation schemes for two-stage stochastic optimization problems. In *Proceedings of the Fourty Firth IEEE Symposium on Foundations of Computer Science*, pages 228–237, 2004. Extended version: *An Approximation Scheme for Stochastic Linear Programming and its Application to Stochastic Integer Programs*, http://ist. caltech. edu/∼cswamy/papers/stochopt. pdf.

[17] L. Stougie. *Design and analysis of algorithms for stochastic integer programming*, volume 37 of *CWI Tract*. Centrum voor Wiskunde en Informatica, Amsterdam, 1987.

[18] L. Stougie and M. H. van der Vlerk. Approximation in stochastic integer programming. In D.S. Johnson, J.K. Lenstra, and D.B. Shmoys, editors, *Approximation Algorithms*, Handbooks in Operations Research and Management Science. Springer-Verlag, Berlin, to appear. http://www.win.tue.nl/math/bs/spor.

[19] S. Toda. PP is as hard as the polynomial time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

[20] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410–421, 1979.

[21] M. H. van der Vlerk. *Stochastic programming with integer recourse*. PhD thesis, University of Groningen, The Netherlands, 1995.

[22] R. Van Slyke and R. J-B. Wets. L-shaped linear programs with applications to control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.