On complexity of stochastic programming problems

Alexander Shapiro[*] and Arkadi Nemirovski[†]

Abstract

The main focus of this paper is in a discussion of complexity of stochastic programming problems. We argue that two-stage (linear) stochastic programming problems with recourse can be solved with a reasonable accuracy by using Monte Carlo sampling techniques, while multi-stage stochastic programs, in general, are intractable. We also discuss complexity of chance constrained problems and multi-stage stochastic programs with linear decision rules.

Key words: stochastic programming, complete recourse, chance constraints, Monte Carlo sampling, SAA method, large deviations bounds, convex programming, multistage stochastic programming.

[*] Georgia Institute of Technology, Atlanta, Georgia 30332-0205, USA, e-mail: ashapiro@isye.gatech.edu

[†] Technion – Israel Institute of Technology, Haifa 32000, Israel, e-mail: nemirovs@ie.technion.ac.il

1 Introduction

In real life we constantly have to make decisions under uncertainty and, moreover, we would like to make such decisions in a reasonably optimal way. Then for a specified objective function $F(x, \xi)$, depending on decision vector $x \in \mathbb{R}^n$ and vector $\xi \in \mathbb{R}^d$ of uncertain parameters, we are faced with the problem of optimizing (say minimizing) $F(x, \xi)$ over x varying in a permissible (feasible) set $X \subset \mathbb{R}^n$. Of course, such an optimization problem is not well defined since our objective depends on an unknown value of ξ. A way of dealing with this is to optimize the objective on *average*. That is, it is assumed that ξ is a random vector[1)], with known probability distribution P having support $\Xi \subset \mathbb{R}^d$, and the following optimization problem is formulated

$$\operatorname*{Min}_{x \in X} \big\{ f(x) := \mathbb{E}_P[F(x, \xi)] \big\}. \tag{1.1}$$

We assume throughout the paper that considered expectations are well defined, e.g., $F(x, \cdot)$ is measurable and P-integrable.

In particular, the above formulation can be applied to two-stage stochastic programming problem with recourse, pioneered by Beale [4] and Dantzig [13]. That is, an optimization problem is divided into two stages. At the first stage one has to make a decision on the basis of some available information. At the second stage, after a realization of the uncertain data becomes known, an optimal second stage decision is made. Such stochastic programming problem can be written in the form (1.1) with $F(x, \xi)$ being the optimal value of the second stage problem.

It should be noted that in the formulation (1.1) all uncertainties are concentrated in the objective function while the feasible set X is supposed to be known (deterministic). Quite often the feasible set itself is defined by constraints which depend on uncertain parameters. In some cases one can reasonably formulate such problems in the form (1.1) by introducing penalties for possible infeasibilities. Alternatively one can try to optimize the objective subject to satisfying constraints for *all* values of unknown parameters in a chosen (uncertain) region. This is the approach of robust optimization (cf., Ben-Tal and Nemirovski [6]). Satisfying the constraints for all possible realizations of random data may be too conservative and, more reasonably, one may try to satisfy the constraints with a high (close to one) probability. This leads to the chance, or probabilistic, constraints formulation which is going back to Charnes and Cooper [11].

There are several natural questions which arise with respect to formulation (1.1).

(i) How do we know the probability distribution P? In some cases one has historical data which can be used to obtain a reasonably accurate estimate of the corresponding probability distribution. However, this happens in rather specific

[1)] Sometimes, in the sequel, ξ denotes a random vector and sometimes its particular realization (numerical value). Which one of these two meanings is used will be clear from the context.

situations and often the probability distribution either cannot be accurately estimated or changes with time. Even worse, in many cases one deals with *scenarios* (i.e., possible realizations of the random data) with the associated probabilities assigned by a subjective judgment.

(ii) Why, at the first stage, do we optimize the *expected* value of the second stage optimization problem? If the optimization procedure is repeated many times, with the same probability distribution of the data, then it could be argued by employing the Law of Large Numbers that this gives an optimal decision on average. However, if in the process, because of the variability of the data one looses all its capital, it does not help that the decisions were optimal on average.

(iii) How difficult is it to solve the stochastic programming problem (1.1)? Evaluation of the expected value function $f(x)$ involves calculation of the corresponding multivariate integrals. Only in rather specific cases it can be done analytically. Therefore, typically, one employs a finite discretization of the random data which allows to write the expectation in a form of summation. Note, however, that if random vector ξ has d elements each with just 3 possible realizations independent of each other, then the total number of scenarios is 3^d, i.e., the number of scenarios grows exponentially fast with dimension d of the data vector.

(iv) Finally, what can be said about multi-stage stochastic programming, when decisions are made in several stages based on available information at the time of making the sequential decisions?

It turns out that there is a close relation between questions (i) and (ii). As far as question (i) is concerned, one can approach it from the following point of view. Suppose that a plausible family $\mathfrak{P}$ of probability distributions, of the random data vector ξ, can be identified. Consequently, the "worst-case-distribution" minimax problem

$$\operatorname*{Min}_{x\in X}\left\{ f(x) := \sup_{P\in\mathfrak{P}} \mathbb{E}_P[F(x,\xi)]\right\} \tag{1.2}$$

is formulated. The worst-case approach to decision analysis, of course, is not new. It was also discussed extensively in the stochastic programming literature (e.g., [15, 16, 19, 22, 38, 48]).

Again we are facing the question of how to choose the set $\mathfrak{P}$ of possible distributions. Traditionally this problem is approached by assuming knowledge of certain moments of the involved random parameters. This leads to the so-called Problem of Moments, where the set $\mathfrak{P}$ is formed by probability measures P satisfying moment constraints $\mathbb{E}_P[\psi_i(\xi)] = b_i$, $i = 1, ..., m$ (see, e.g., [28]). In that case the extreme (worst case) distributions are measures with a finite support of at most $m + 1$ points.

On the other hand, it often happens in applications that one is given a deterministic value μ of the uncertain data vector ξ and does not have an idea what a corresponding distribution may be. For example, ξ could represent an uncertain demand and μ is viewed as its mean vector given by a forecast. It is well recognized now that solving a corresponding optimization problem for the deterministic value $\xi = \mu$ may give a poor solution from a robustness point of view. It is natural then to introduce random perturbations to the deterministic vector μ and to solve the obtained stochastic program. For instance, one can assume that components ξ_i of the uncertain data vector are independent and have a certain type (say, log-normal if ξ_i should be nonnegative) distribution with means μ_i and standard deviations σ_i which are defined within a certain percentage of μ_i, $i = 1, ..., d$. Often this quickly stabilizes optimal solutions of the corresponding stochastic programs irrespective of the underlying distribution (cf., [36]). Furthermore, we can approach this setup from the minimax point of view by considering a worst distribution supported on, say, a box region around vector μ. If, moreover, we consider unimodal type families of distributions, then the worst case distribution is uniform (cf., [43]). For a given x, even unimodal distributions and $F(x, \cdot) := -\mathbb{1}_S(\cdot)$, where $\mathbb{1}_S(\cdot)$ is the indicator function of a symmetric convex set S, this result was first established by Barmish and Lagoa [3], where it was called the "Uniformity Principle".

Question (ii) has also a long history. One can optimize a weighted sum of the expected value and a term representing variability of the second stage objective function. For example, we can try to minimize

$$f(x) := \mathbb{E}[F(x, \xi)] + c\mathrm{Var}[F(x, \xi)], \tag{1.3}$$

where $c \geq 0$ is a chosen constant. This approach goes back to Markowitz [27]. The additional (variance) term in (1.3) can be viewed as a risk measure of the second stage (optimal) outcome. It could be noted, however, that adding the variance term may destroy convexity of the function $f(\cdot)$ even if $F(\cdot, \xi)$ is convex for all realizations of ξ (cf., [45]). An axiomatic approach to a mathematical theory of risk measures was suggested recently by Artzner et al. [1]. That is, value of a random variable Z is measured by a function $\rho(Z)$ satisfying certain axioms. An example of such function $\rho(Z)$, called coherent risk measure, is the mean-semideviation

$$\rho(Z) := \mathbb{E}[Z] + c \left\{ \mathbb{E} \left(\left[Z - \mathbb{E}[Z] \right]_+^2 \right) \right\}^{1/2},$$

where $c \in [0, 1]$.

It turns out that $\rho(Z)$ is a coherent risk measure if and only if it can be represented in the form $\rho(Z) = \sup_{P \in \mathfrak{P}} \mathbb{E}_P[Z]$, where $\mathfrak{P}$ is a set of probability measures. In different frameworks this dual representation was derived in [1, 20, 33, 34]. Therefore, the min-max problem (1.2) and the problem of minimization of a coherent risk measure, of $F(x, \xi)$, in fact are equivalent. We may refer to [2, 18, 32, 35] for extensions of this approach to a multi-stage setting.

2 Complexity of two-stage stochastic programs

In this section we discuss question (iii) mentioned in the introduction, that is, how difficult is to solve a stochastic program. Problem (1.1) is a problem of minimizing a deterministic *implicitly given* objective $f(x)$. We should expect that this problem is at least as difficult as minimizing $f(x)$, $x \in X$, in the case where $f(x)$ is given explicitly, say by a "closed form analytic expression", or, more general, by an "oracle" capable to compute the values and the derivatives of $f(x)$ at every given point. As far as problems of minimization of $f(x)$, $x \in X$, with explicitly given objective, are concerned, the "solvable case" is known, this is the Convex Programming case. That is, X is a closed convex set and $f : X \to \mathbb{R}$ is a convex function. It is known that generic Convex Programming problems satisfying mild computability and boundedness assumptions can be solved in polynomial time. In contrast to this, typical nonconvex problems turn out to be NP-hard[2)]. It follows that when speaking about conditions under which the stochastic program (1.1) is efficiently solvable, it makes sense to assume that X is a closed convex set, and $f(\cdot)$ is convex on X. We gain from a technical point (and do not lose much from practical viewpoint) by assuming X to be bounded. These assumptions, plus mild technical conditions, would be sufficient to make (1.1) easy, if $f(x)$ were given explicitly. However, in Stochastic Programming it makes no sense to assume that we can compute efficiently the expectation in (1.1), thus arriving at an explicit representation of $f(x)$. Would it be the case, there would be no necessity to treat (1.1) as a stochastic program.

We argue now that stochastic programming problems of the form (1.1) can be solved reasonably efficiently by using Monte Carlo sampling techniques provided that the probability distribution of the random data is not "too bad" and certain general conditions are met. In this respect we should explain what do we mean by "solving" stochastic programming problems. Let us consider, for example, two-stage linear stochastic programming problems with recourse. Such problems can be written in the form (1.1) with[3)]

$$X := \{x : Ax = b,\ x \geq 0\} \ \text{ and } \ F(x,\xi) := \langle c, x\rangle + Q(x,\xi),$$

where $Q(x,\xi)$ is the optimal value of the second stage problem:

$$\operatorname*{Min}_{y \geq 0} \langle q, y\rangle \ \text{ subject to } \ Tx + Wy \geq h. \tag{2.1}$$

[2)] It is beyond the scope of this paper to give a detailed explanation of what "polynomial time solvability" and "NP-hardness" mean. Informally speaking, the former property of a problem P means that P is "easy to solve" – it admits a computationally efficient solution algorithm. NP-hardness of P means that no efficient solution algorithms for P are known, and there are strong theoretical reasons to believe that they do not exist. For formal treatment of these issues in Continuous Optimization, see, e.g. [6, Chapter 5].

We should also stress that a claim "such and such problem is difficult" relates to a *generic* problem in question and does *not* imply that the problem has no solvable particular cases.

[3)] By $\langle x, y\rangle$ we denote the standard scalar product of two vectors $x, y \in \mathbb{R}^n$.

Here T and W are matrices of an appropriate order and $\xi \in \mathbb{R}^d$ is a vector whose elements are composed from elements of vectors q and h and matrices T and W which, in a considered problem, are assumed to be random. If we assume that the random data vector has a finite number of realizations (scenarios) $\xi_k = (q_k, V_k, T_k, h_k)$ with respective probabilities p_k, $k = 1, ..., K$, then the obtained two-stage problem can be written as one large linear programming problem:

$$\begin{array}{ll} \text{Min}_{x,y_1,\dots,y_K} & \langle c, x\rangle + \sum_{k=1}^{K} p_k \langle q_k, y_k\rangle \\ \text{s.t} & Ax = b,\ T_k x + W_k y_k \geq h_k,\ k = 1, ..., K, \\ & x \geq 0,\ y_k \geq 0,\ k = 1, ..., K. \end{array} \tag{2.2}$$

If the number of scenarios K is not “too large”, then the above linear programming problem (2.2) can be solved accurately in a reasonable time. However, even a crude discretization of the probability distribution of ξ typically results in an exponential growth of the number of scenarios with increase of the number d of random parameters. Suppose, for example, that components of the random vector ξ are mutually independently distributed each having a small number r of possible realizations. Then the size of the corresponding input data grows linearly in d (and r) while the number of scenarios $K = r^d$ grows exponentially. Yet in some cases problem (2.2) can be solved numerically in a reasonable time. For example, suppose that matrices T and W are constant (deterministic) and only h is random and, moreover, $Q(x, \xi)$ decomposes into the sum $Q(x, \xi) = Q_1(x_1, h_1) + ... + Q_n(x_n, h_n)$. This happens in the case of the so-called simple recourse with

$$Q_i(x_i, h_i) = q_i^+[x_i - h_i]_+ + q_i^-[h_i - x_i]_+,\ i = 1, ..., n,$$

where q_i^+ and q_i^- are some positive numbers. Then $\mathbb{E}[Q(x, \xi)] = \mathbb{E}[Q_1(x_1, h_1)] + ... + \mathbb{E}[Q_n(x_n, h_n)]$, i.e., calculation of the multidimensional expectation is reduced to calculations of one dimensional expectations. Of course, the above is a rather specific case and in a general situation there is no hope to solve problem (2.2) accurately (say with machine precision) even for moderate values of r and d (cf., [17]).

It should be said at this point that from a practical point of view, typically, it does not make sense to try to solve a stochastic programming problem with a high precision. A numerical error resulting from an inaccurate estimation of the involved probability distributions, modeling errors, etc., can be far bigger than such an optimization error. We argue now that two-stage stochastic problems can be solved efficiently with a reasonable accuracy provided that the following conditions are met:

(a) The feasible set X is fixed (deterministic).

(b) For all $x \in X$ and $\xi \in \Xi$ the objective function $F(x, \xi)$ is real valued.

(c) The considered stochastic programming problem can be solved efficiently (by a deterministic algorithm) if the number of scenarios is not “too large”.

When applied to two-stage stochastic programming, the above conditions (a) and (b) mean that the recourse is relatively complete[4)] and the second stage problem is bounded from below. The above condition (c) certainly holds in the case of two-stage *linear* stochastic programming with recourse.

In order to proceed let us consider the following Monte Carlo sampling approach. Suppose that we can generate an iid (independent identically distributed) random sample $\xi^1, ..., \xi^N$ of N realizations of the considered random vector. Then we can estimate the expected value function $f(x)$ by the sample average[5)]

$$\hat{f}_N(x) := \frac{1}{N}\sum_{j=1}^{N} F(x, \xi^j). \tag{2.3}$$

Consequently, we approximate the true problem (1.1) by the problem:

$$\operatorname*{Min}_{x\in X} \hat{f}_N(x). \tag{2.4}$$

We refer to (2.4) as the Sample Average Approximation (SAA) problem. The optimal value $\hat{v}_N$ and the set $\hat{S}_N$ of optimal solutions of the SAA problem (2.4) provide estimates of their true counterparts of problem (1.1). It should be noted that once the sample is generated, $\hat{f}_N(x)$ becomes a deterministic function and problem (2.4) becomes a stochastic programming problem with N scenarios $\xi^1, ..., \xi^N$ taken with equal probabilities $1/N$. It also should be mentioned that the SAA method is *not* an algorithm. One still has to solve the obtained problem (2.4) by employing an appropriate (deterministic) algorithm.

By the Law of Large Numbers we have that $\hat{f}_N(x)$ converges (pointwise in x) w.p.1 to $f(x)$ as N tends to infinity. Therefore it is reasonable to expect for $\hat{v}_N$ and $\hat{S}_N$ to converge to their counterparts of the true problem (1.1) with probability one (w.p.1) as N tends to infinity. And, indeed, such convergence can be proved under mild regularity conditions. However, for a fixed $x \in X$, convergence of $\hat{f}_N(x)$ to $f(x)$ is notoriously slow. By the Central Limit Theorem it is of order $O_p(N^{-1/2})$. The rate of convergence can be improved, sometimes significantly, by variance reduction methods. However, by using Monte Carlo (Quasi-Monte Carlo) techniques one cannot evaluate the expected value $f(x)$ very accurately.

The following analysis is based on exponential bounds of the Large Deviations (LD) theory (see, e.g., [14] for a general discussion of LD theory). Denote by S^ε and $\hat{S}^\varepsilon_N$ the sets of ε-optimal solutions of the true and SAA problems, respectively, i.e., $\bar{x} \in S^\varepsilon$ iff $\bar{x} \in X$ and $f(\bar{x}) \le \inf_{x\in X} f(x) + \varepsilon$. Choose accuracy constants $\varepsilon > 0$ and $0 \le \delta < \varepsilon$, and significance level $\alpha \in (0, 1)$. Suppose for the moment that the set X

[4)] It is said that the recourse is *relatively complete* if for every $x \in X$ and every possible realization of random data, the second stage problem is feasible.

[5)] In order to simplify notation we only write in the subscript the sample size N while actually $\hat{f}_N(\cdot)$ depends on the generated sample, and in that sense is random.

is finite although its cardinality $|X|$ can be very large. Then by using Cramér's LD theorem it is not difficult to show that the sample size

$$N \geq \frac{1}{\eta(\varepsilon,\delta)} \log\left(\frac{|X|}{\alpha}\right) \tag{2.5}$$

guarantees that probability of the event $\{\hat{S}_N^\delta \subset S^\varepsilon\}$ is at least $1-\alpha$ (see [21],[41, section 3.1]). That is, for any N bigger than the right hand side of (2.5) we are guaranteed that any δ-optimal solution of the corresponding SAA problem provides an ε-optimal solution of the true problem with probability at least $1-\alpha$, in other words, solving the SAA problem with accuracy δ guarantees solving the true problem with accuracy ε with probability at least $1-\alpha$.

The number $\eta(\varepsilon,\delta)$ in the estimate (2.5) is defined as follows. Consider a mapping $u : X \setminus S^\varepsilon \to X$ such that $f(u(x)) \leq f(x) - \varepsilon$ for all $x \in X \setminus S^\varepsilon$. Such mappings do exist, although not unique. For example, any mapping $u : X \setminus S^\varepsilon \to S$ satisfies this condition. Choice of such a mapping gives a certain flexibility to the corresponding estimate of the sample size. For $x \in X$, consider random variable

$$Y_x := F(u(x),\xi) - F(x,\xi),$$

its moment generating function $M_x(t) := \mathbb{E}\left[e^{tY_x}\right]$ and the LD rate function[6)]

$$I_x(z) := \sup_{t\in\mathbb{R}} \left\{tz - \log M_x(t)\right\}.$$

Note that, by construction of mapping $u(x)$, the inequality

$$\mu_x := \mathbb{E}\left[Y_x\right] = f(u(x)) - f(x) \leq -\varepsilon \tag{2.6}$$

holds for all $x \in X \setminus S^\varepsilon$. Finally, we define

$$\eta(\varepsilon,\delta) := \operatorname*{Min}_{x\in X\setminus S^\varepsilon} I_x(-\delta). \tag{2.7}$$

Because of (2.6) and since $\delta < \varepsilon$, the number $I_x(-\delta)$ is positive provided that the probability distribution of Y_x is not "too bad". Specifically, if we assume that the moment generating function $M_x(t)$, of Y_x, is finite valued for all t in a neighborhood of 0, then the random variable Y_x has finite moments and $I_x(\mu_x) = I'(\mu_x) = 0$, and $I''(\mu_x) = 1/\sigma_x^2$ where $\sigma_x^2 := \operatorname{Var}\left[Y_x\right]$. Consequently, $I_x(-\delta)$ can be approximated, by using second order Taylor expansion, as follows

$$I_x(-\delta) \approx \frac{(-\delta-\mu_x)^2}{2\sigma_x^2} \geq \frac{(\varepsilon-\delta)^2}{2\sigma_x^2}.$$

This suggests that one can expect the constant $\eta(\varepsilon,\delta)$ to be of order of $(\varepsilon-\delta)^2$. And, indeed, this can be ensured by various conditions. Consider the following condition.

[6)] That is, $I_x(\cdot)$ is the conjugate of the function $\log M_x(\cdot)$ in the sense of convex analysis.

(A1) There exists constant $\sigma > 0$ such that for any $x', x \in X$, the moment generating function $M^*(t)$ of $F(x',\xi) - F(x,\xi) - \mathbb{E}\left[F(x',\xi) - F(x,\xi)\right]$ satisfies:

$$M^*(t) \le \exp\left(\tfrac{1}{2}\sigma^2 t^2\right), \quad \forall\, t \in \mathbb{R}. \tag{2.8}$$

Note that random variable $F(x',\xi) - F(x,\xi) - \mathbb{E}\left[F(x',\xi) - F(x,\xi)\right]$ has zero mean. Moreover, if it has a normal distribution, with variance σ^2, then its moment generating function is equal to the right hand side of (2.8). Condition (2.8) means that tail probabilities $\mathrm{Prob}\big(|F(x',\xi) - F(x,\xi)| > t\big)$ are bounded from above[7)] by $O(1)\exp\left(-\frac{t^2}{2\sigma^2}\right)$. This condition certainly holds if the distribution of the considered random variable has a bounded support.

For $x' = u(x)$, random variable $F(x',\xi) - F(x,\xi)$ coincides with Y_x, and hence (2.8) implies that $M_x(t) \le \exp(\mu_x t + \sigma^2 t^2/2)$. It follows that

$$I_x(z) \ge \sup_{t\in\mathbb{R}} \left\{ zt - \mu_x t - \sigma^2 t^2/2 \right\} = \frac{(z-\mu_x)^2}{2\sigma^2}, \tag{2.9}$$

and hence for any $\varepsilon > 0$ and $\delta \in [0, \varepsilon)$:

$$\eta(\varepsilon, \delta) \ge \frac{(-\delta - \mu_x)^2}{2\sigma^2} \ge \frac{(\varepsilon - \delta)^2}{2\sigma^2}. \tag{2.10}$$

It follows that, under assumption (A1), the estimate (2.5) can be written as

$$N \ge \frac{2\sigma^2}{(\varepsilon-\delta)^2} \log\left(\frac{|X|}{\alpha}\right). \tag{2.11}$$

Remark 1 Condition (2.8) can be replaced by a more general condition

$$M^*(t) \le \exp(\psi(t)), \quad \forall\, t \in \mathbb{R}, \tag{2.12}$$

where $\psi(t)$ is a convex even function with $\psi(0) = 0$. Then $\log M_x(t) \le \mu_x t + \psi(t)$ and hence $I_x(z) \ge \psi^*(z - \mu_x)$, where ψ^* is the conjugate of the function ψ. It follows then that

$$\eta(\varepsilon, \delta) \ge \psi^*(-\delta - \mu_x) \ge \psi^*(\varepsilon - \delta). \tag{2.13}$$

For example, instead of assuming that the bound (2.8) holds for all $t \in \mathbb{R}$, we can assume that it holds for all t in a finite interval $[-a, a]$, where $a > 0$ is a given constant. That is, we can take $\psi(t) := \frac{1}{2}\sigma^2 t$ if $|t| \le a$, and $\psi(t) := +\infty$ otherwise. In that case $\psi^*(z) = z^2/(2\sigma^2)$ for $|z| \le a\sigma^2$, and $\psi^*(z) = a|z| - \frac{1}{2}a^2\sigma^2$ for $|z| > a\sigma^2$.

[7)] By $O(1)$ we denote generic absolute constants.

Now let X be a bounded, not necessary finite, subset of $\mathbb{R}^n$ of diameter

$$D := \sup_{x',x\in X} \|x'-x\|.$$

Then for $\tau > 0$ we can construct a set $X_\tau \subset X$ such that for any $x \in X$ there is $x' \in X_\tau$ satisfying $\|x - x'\| \le \tau$, and $|X_\tau| = O(1)(D/\tau)^n$. Suppose that condition (A1) holds. Then by (2.11), for $\varepsilon' > \delta$, we can estimate the corresponding sample size required to solve the reduced optimization problem, obtained by replacing X with X_τ, as

$$N \ge \frac{2\sigma^2}{(\varepsilon'-\delta)^2}\left[n\left(\log D - \log\tau\right) + \log\big(O(1)/\alpha\big)\right]. \tag{2.14}$$

Suppose, further, that there exists a (measurable) function $\kappa : \Xi \to \mathbb{R}_+$ and $\gamma > 0$ such that

$$|F(x',\xi) - F(x,\xi)| \le \kappa(\xi)\|x'-x\|^\gamma \tag{2.15}$$

holds for all $x', x \in X$ and all $\xi \in \Xi$. It follows by (2.15) that

$$|\hat f_N(x') - \hat f_N(x)| \le N^{-1}\sum_{j=1}^N |F(x',\xi^j) - F(x,\xi^j)| \le \hat\kappa_N\|x'-x\|^\gamma, \tag{2.16}$$

where $\hat\kappa_N := N^{-1}\sum_{j=1}^N \kappa(\xi^j)$.

Let us assume, further, the following:

(A2) The moment generating function $M_\kappa(t) := \mathbb{E}\left[e^{t\kappa(\xi)}\right]$ of $\kappa(\xi)$ is finite valued for all t in a neighborhood of 0.

It follows then that the expectation $L := \mathbb{E}[\kappa(\xi)]$ is finite, and moreover, by Cramér's LD Theorem that for any $L' > L$ there exists a positive constant $\beta = \beta(L')$ such that

$$P\left(\hat\kappa_N > L'\right) \le e^{-N\beta}. \tag{2.17}$$

Let $\hat x_N$ be a δ-optimal solution of the SAA problem and $\tilde x_N \in X_\tau$ be a point such that $\|\hat x_N - \tilde x_N\| \le \tau$. Let us take $N \ge \beta^{-1}\log(2/\alpha)$, so that by (2.17) we have that

$$\text{Prob}\left(\hat\kappa_N > L'\right) \le \alpha/2. \tag{2.18}$$

Then with probability at least $1-\alpha/2$, the point $\tilde x_N$ is a $(\delta + L'\tau^\gamma)$-optimal solution of the reduced SAA problem. Setting

$$\tau := [(\varepsilon-\delta)/(2L')]^{1/\gamma},$$

we obtain that with probability at least $1-\alpha/2$, the point $\tilde x_N$ is an ε'-optimal solution of the reduced SAA problem with $\varepsilon' := (\varepsilon+\delta)/2$. Moreover, by taking a sample size satisfying (2.14), we obtain that $\tilde{x}_N$ is an ε'-optimal solution of the reduced expected value problem with probability at least $1-\alpha/2$. It follows that $\hat{x}_N$ is an ε''-optimal solution of the SAA problem (1.1) with probability at least $1-\alpha$ and $\varepsilon'' = \varepsilon' + L\tau^\gamma \le \varepsilon$. We obtain the following estimate

$$N \ge \frac{4\sigma^2}{(\varepsilon-\delta)^2}\left[n\left(\log D + \gamma^{-1}\log\frac{2L'}{\varepsilon-\delta}\right) + \log\left(\frac{O(1)}{\alpha}\right)\right] \vee \left[\beta^{-1}\log(2/\alpha)\right] \quad (2.19)$$

for the sample size (cf., [41, section 3.2]).

The above result is quite general and does not involve the assumption of convexity. Estimate (2.19) of the sample size contains various constants and is too conservative for practical applications. However, in a sense, it gives an estimate of complexity of two-stage stochastic programming problems. We will discuss this in the next section. In typical applications (e.g., in the convex case) the constant $\gamma = 1$, in which case condition (2.15) means that $F(\cdot,\xi)$ is Lipschitz continuous on X with constant $\kappa(\xi)$. However, there are also some applications where γ could be less than 1 (cf., [42]).

We obtain the following basic positive result.

Theorem 1 *Suppose that assumptions* (A1) *and* (A2) *hold and X has a finite diameter D. Then for $\varepsilon > 0$, $0 \le \delta < \varepsilon$ and sample size N satisfying* (2.19)*, we are guaranteed that any δ-optimal solution of the SAA problem is an ε-optimal solution of the true problem with probability at least $1-\alpha$.*

Let us also consider the following simplified variant of Theorem 1. Suppose that:

(A3) There is a positive constant C such that $|F(x',\xi) - F(x,\xi)| \le C$ for all $x', x \in X$ and $\xi \in \Xi$.

Under assumption (A3) we have that for any $\varepsilon > 0$ and $\delta \in [0,\varepsilon]$:

$$I_x(-\delta) \ge O(1)\frac{(\varepsilon-\delta)^2}{C^2}, \;\; \text{for all } x \in X \setminus S^\varepsilon, \; \xi \in \Xi, \quad (2.20)$$

and hence $\eta(\varepsilon,\delta) \ge O(1)(\varepsilon-\delta)^2/C^2$. Consequently, the bound (2.5) for the sample size which is required to solve the true problem with accuracy $\varepsilon > 0$ and probability at least $1-\alpha$, by solving the SAA problem with accuracy $\delta := \varepsilon/2$, takes the form

$$N \ge O(1)\left(\frac{C}{\varepsilon}\right)^2 \log\left(\frac{|X|}{\alpha}\right). \quad (2.21)$$

The estimate (2.21) can be also derived by using Hoeffding's inequality[8)] instead of Cramér's LD bound.

[8)] Recall that Hoeffding's inequality states that if $Z_1, ..., Z_N$ is an iid random sample from a distribution supported on a bounded interval $[a,b]$, then for any $t > 0$,

$$\text{Prob}\left(\bar{Z} - \mu \ge t\right) \le e^{-2t^2N/(b-a)^2},$$

where $\bar{Z}$ is the sample average and $\mu = \mathbb{E}[Z_i]$.

In particular, if we assume that $\gamma = 1$ and $\kappa(\xi) = L$ for all $\xi \in \Xi$, i.e., $F(\cdot, \xi)$ is Lipschitz continuous on X with constant L independent of $\xi \in \Xi$, then we can take $C = DL$ and remove the term $\beta^{-1} \log(2/\alpha)$ in the right hand side of (2.19). By taking, further, $\delta := \varepsilon/2$ we obtain in that case the following estimate of the sample size

$$N \geq O(1) \left(\frac{DL}{\varepsilon}\right)^2 \left[n \log\left(\frac{DL}{\varepsilon}\right) + \log\left(\frac{O(1)}{\alpha}\right)\right]. \tag{2.22}$$

We can write the following simplified version of Theorem 1.

Theorem 2 *Suppose that X has a finite diameter D and condition* (2.15) *holds with $\gamma = 1$ and $\kappa(\xi) = L$ for all $\xi \in \Xi$. Then with sample size N satisfying* (2.22) *we are guaranteed that every $(\varepsilon/2)$-optimal solution of the SAA problem is an ε-optimal solution of the true problem with probability at least $1 - \alpha$.*

In the next section we compare complexity estimates implied by the bound (2.22) with complexity of "deterministic" convex programming.

3 What is easy and what is difficult in stochastic programming?

Since, generically, nonconvex problems are difficult already in the deterministic case, when discussing the question of what is easy and what is not in Stochastic Programming, it makes sense to restrict ourselves with convex problems (1.1). Thus, in the sequel it is assumed by default that X is a closed and bounded convex set, and $f : X \to \mathbb{R}$ is convex. These assumptions, plus mild technical conditions, would be sufficient to make (1.1) easy, provided that $f(x)$ were given explicitly, but the latter is *not* what we assume in SP. What we usually (and everywhere below) do assume in SP is that:

(i) The function $F(x, \xi)$ is given explicitly, so that we can compute efficiently its value (and perhaps the derivatives in x) at every given pair $(x, \xi) \in X \times \Xi$.

(ii) We have access to a mechanism which is capable of sampling from the distribution P, that is, we can generate a sample ξ^1, ξ^2,... of independent realizations of ξ.

For the sake of discussion to follow we assume in this section that we are under the premise of Theorem 2 and that problem (1.1) is *convex*. To proceed, let us compare the complexity bound given by Theorem 2 with a typical result on the "black box" complexity of the usual (deterministic) Convex Programming.

Theorem 3 *Consider a convex problem*

$$\operatorname*{Min}_{x \in X} f(x), \tag{CP}$$

where $X \subset \mathbb{R}^n$ is a closed convex set which is contained in a centered at the origin ball of diameter D and contains a ball of given diameter $d > 0$, and that $f : X \to \mathbb{R}$ is convex Lipschitz continuous, with constant L. Assume that X is given by a Separation Oracle which, given on input a point $x \in \mathbb{R}^n$, reports whether $x \in X$, and if it is not the case, returns $e \in \mathbb{R}^n$ which separates x and X: such that $\langle e, x \rangle > \max_{y \in X} \langle e, y \rangle$. Assume, further, that f is given by a First Order oracle which, given on input $x \in X$, returns on output the value $f(x)$ and a subgradient $\nabla f(x)$, $\|\nabla f(x)\|_2 \le L$, of f at x.

In this framework, for every $\varepsilon > 0$ one can find an ε-solution to (CP) *by an algorithm which requires at most*

$$M = O(1)n^2 \left[\log\left(\frac{DL}{\varepsilon}\right) + \log\left(\frac{D}{d}\right) \right] \tag{3.1}$$

calls to the Separation and First Order oracles, with a call accompanied by $O(n^2)$ arithmetic operations to process oracle's answer.

In our context, the role of Theorem 3 is twofold. First, it can be viewed as a necessary follow-up to Theorem 2 which reduces solving (1.1) to solving the corresponding SAA problem and says nothing on how difficult is the latter task. This question is answered by Theorem 3 in the convex case[9)]. However, the main role of Theorem 3 in our context is the one of a benchmark for the SP complexity results. Let us use this benchmark to evaluate the result stated in Theorem 2.

Observation 1 In contrast to Theorem 3, Theorem 2 provides us with no more than probabilistic quality guarantees. That is, the random approximate solution to (1.1) implied by the outlined SAA approach, being ε-solution to (1.1) with probability $1 - \alpha$, can be very bad with the remaining probability α. In our "black box" informational environment (the distribution of ξ is not given in advance, all we have is an access to a black box generating independent realizations of ξ), this "shortcoming" is unavoidable. Note, however, that the sample size N as given by (2.19) is "nearly independent" of α, i.e., to reduce unreliability from 10^{-2} to 10^{-12} requires at most 6-fold increase in the sample size. Note that unreliability as small as 10^{-12} is, for all practical purposes, the same as 100% reliability.

Observation 2 To proceed with our comparison, it makes sense to measure the complexity of the SAA method merely by the number of scenarios N required to

[9)] In our context, Theorem 3 allows to handle the most general "black box" situation – no assumptions on $F(\cdot, \xi)$ and X except for convexity and computability. When $F(\cdot, \xi)$ possesses appropriate analytic structure, the complexity of solving the SAA problem can be reduced by using a solver adjusted to this structure.

get an ε-solution with probability at least $1-\alpha$, and to measure the complexity of deterministic convex optimization as presented in Theorem 3 by the number M of oracle calls required to get an ε-solution. The rationale behind is that "very large" N definitely makes the SAA method impractical, while with a "moderate" N, the method becomes practical, provided that $F(\cdot,\cdot)$ and X are not too complicated, and similarly for M in the context of Theorem 3.

When comparing bounds (2.22) and (3.1), our first observation is that both of them depend polynomially on the design dimension n of the problem, which is nice. What does make difference between these bounds, is their dependence on the required accuracy ε, or, better to say, on the relative accuracy[10] $\nu := \varepsilon/(DL)$. In contrast to bound (3.1) which is polynomial in $\log(1/\nu)$, bound (2.22) is polynomial (specifically, quadratic) in $1/\nu$. In reality this means that the SAA method could solve in a reasonable time to a moderate relative accuracy, like $\nu = 10\%$ or even $\nu = 1\%$, stochastic problems involving an astronomically large, or even infinite, number of scenarios. This was verified in a number of numerical experiments (e.g., [24, 26, 36, 46]). On the other hand, in general, the SAA method does *not* allow to solve, even simply-looking, problems to high relative accuracy[11]: according to (2.22), the *estimated* sample size N required to achieve $\nu = 10^{-3}$ ($\nu = 10^{-5}$) is at least of order of millions (respectively, tens of billions). In sharp contrast to this, bound (3.1) says that in the deterministic case, relative accuracy $\nu = 10^{-5}$ is just by factor 5 "more costly" than $\nu = 0.1$.

It should be stressed that in our general setting the outlined phenomenon is not a shortcoming of the SAA method – it is unavoidable. Indeed, given positive constants L,D and ε such that $\nu = \varepsilon/(LD) \leq 0.1$, consider the pair of stochastic problems:

$$\operatorname*{Min}_{x\in[0,D]} \left\{ f_\chi(x) := \mathbb{E}_{P_\chi}[x\xi] \right\} \tag{SP$_\chi$}$$

indexed by $\chi = \pm 1$, and with distribution P_χ of ξ supported on the two-point set $\{-L; L\}$ on the axis. Specifically, P_1 assigns the mass $1/2 - 4\nu$ to the point $-L$ and the mass $1/2 + 4\nu$ to the point L, while P_{-1} assigns to the same points $-L, L$ the masses $1/2 + 4\nu$, $1/2 - 4\nu$, respectively. Of course, $f_1(x) = 4\varepsilon D^{-1}x$, $f_{-1}(x) = -4\varepsilon D^{-1}x$, the solution to (SP$_1$) is $x_1 = 0$, while the solution to (SP$_{-1}$) is $x_{-1} = D$. Note, however, that the situation is that trivial *only when we know in advance what is the distribution P_χ we deal with*. If it is not the case and all we can see is a sample of N independent realizations of ξ, the situation changes dramatically: an algorithm capable of solving with accuracy ε and reliability $1-\alpha = 0.9$ every one of the problems (SP$_{\pm 1}$) using

[10] Recall that, under assumptions of Theorem 2, DL gives an upper bound on the variation of the objective on the feasible domain. While using bound (2.19) we can take $\nu := \varepsilon/\sigma$. Passing from ε to ν, means quantifying inaccuracies as fractions of the variation, which is quite natural.

[11] It is possible to solve true problem (1.1) by the SAA method with high (machine) accuracy in some specific situations, for example, in some cases of linear two-stage stochastic programming with a finite (although very large) number of scenarios, see [37, 39].

sample of size N, would, as a byproduct, imply a procedure which, given the sample, decides, with the same reliability, which one of the two possible distributions $P_{\pm 1}$ underlies the sample. The laws of Statistics say that such a reliable identification of the underlying distribution is possible only when $N \geq O(1)\frac{D^2L^2}{\varepsilon^2}$ (compare with bound (2.22)). Note that both stochastic problems in question satisfy all the assumptions in Theorem 2, so that in the situation considered in this statement the bound (2.22) is the best possible (up to logarithmic term) as far as the dependence on D, L and ε is concerned.

To make our presentation self-contained, we explain here what are the "laws of Statistics" which underlie the above conclusions. First, an algorithm $\mathcal{A}$ capable of solving within accuracy ε and reliability 0.9 every one of the problems (SP$_{\pm 1}$), given an N-element sample drawn from the corresponding distribution, indeed implies a "0.9-reliable" procedure which decides, based on the same sample, what is the distribution; this procedure accepts hypothesis I stating that the sample is drawn from distribution P_1 if and only if the approximate solution generated by $\mathcal{A}$ is in $[0, D/2]$; if it is not the case, the procedure accepts hypothesis II "the sample is drawn from P_{-1}". Note that if the first of the hypotheses is true and the outlined procedure accepts the second one, the approximate solution produced by $\mathcal{A}$ is *not* and ε-solution to (SP$_1$), so that the probability p^{I} to accept the second hypothesis when the first is true is $\leq 1 - 0.9 = 0.1$. Similarly, probability p^{II} for the procedure to accept the first hypothesis when the second is true is ≤ 0.1. The announced lower bound on N is given by the following observation: *Consider a decision rule which, given on input a sequence ξ^N of N independent realizations of ξ known in advance to be drawn either from the distribution P_1, or from the distribution P_{-1}, decides which one of these two options takes place, and let p^{I}, p^{II} be the associated probabilities of wrong decisions. Then*

$$\max\{p^{\mathrm{I}}, p^{\mathrm{II}}\} \leq 0.1 \textit{ implies that } N \geq O(1)\nu^{-2}, \tag{3.2}$$

where $O(1)$ is a positive absolute constant.

Indeed, a candidate decision rule can be identified with a subset $\mathcal{S}$ of $\mathcal{L}^N$; this set is comprised of all realizations ξ^N resulting, via the decision rule in question, in acceptance of hypothesis I. Let P_1^N, P_{-1}^N be the distributions of ξ^N corresponding to hypotheses I, II. We clearly have

$$p^{\mathrm{I}} = \sum_{\xi^N \notin \mathcal{S}} P_1^N(\xi^N), \; p^{\mathrm{II}} = \sum_{\xi^N \in \mathcal{S}} P_{-1}^N(\xi^N).$$

Now consider the Kullback distance from P_1^N to P_{-1}^N:

$$\mathcal{K} = \sum_{\xi^N \in \mathcal{L}^N} \log\left(\frac{P_1^N(\xi^N)}{P_{-1}^N(\xi^N)}\right) P_1^N(\xi^N).$$

the function $p \log \frac{p}{q}$ of two positive variables p, q is jointly convex; denoting by $\bar{\mathcal{S}}$ the complement of S in $\mathcal{L}^N$ and by $|A|$ the cardinality of a finite set A, it follows that

$$\begin{aligned} &|\bar{\mathcal{S}}|^{-1} \sum_{\xi^N \in \bar{\mathcal{S}}} \log\left(\frac{P_1^N(\xi^N)}{P_{-1}^N(\xi^N)}\right) P_1^N(\xi^N) \\ &\qquad \geq \left(|\bar{\mathcal{S}}|^{-1} \sum_{\xi^N \in \bar{\mathcal{S}}} P_1^N(\xi^N)\right) \log\left(\frac{|\bar{\mathcal{S}}|^{-1} \sum_{\xi^N \in \bar{\mathcal{S}}} P_1^N(\xi^N)}{|\bar{\mathcal{S}}|^{-1} \sum_{\xi^N \in \bar{\mathcal{S}}} P_{-1}^N(\xi^N)}\right), \end{aligned}$$

whence

$$\sum_{\xi^N \in \bar{\mathcal{S}}} \log\left(\frac{P_1^N(\xi^N)}{P_{-1}^N(\xi^N)}\right) P_1^N(\xi^N) \geq p^{\mathrm{I}} \log\left(\frac{p^{\mathrm{I}}}{1-p^{\mathrm{II}}}\right),$$

and similarly

$$\sum_{\xi^N \in \mathcal{S}} \log\left(\frac{P_1^N(\xi^N)}{P_{-1}^N(\xi^N)}\right) P_1^N(\xi^N) \geq (1-p^{\mathrm{I}}) \log\left(\frac{1-p^{\mathrm{I}}}{p^{\mathrm{II}}}\right),$$

whence

$$\mathcal{K} \geq p^{\mathrm{I}} \log\left(\frac{p^{\mathrm{I}}}{1-p^{\mathrm{II}}}\right) + (1-p^{\mathrm{I}}) \log\left(\frac{1-p^{\mathrm{I}}}{p^{\mathrm{II}}}\right).$$

For every $p \in (0, 1/2)$, the minimum of the left hand side in the latter inequality in $p^{\mathrm{I}}, p^{\mathrm{II}} \in (0, p]$ is achieved when $p^{\mathrm{I}} = p^{\mathrm{II}} = p$ and is equal to $p \log \frac{p}{1-p} + (1-p) \log \frac{1-p}{p} \geq 4(p-1/2)^2$. Thus,

$$p := \max[p^{\mathrm{I}}, p^{\mathrm{II}}] \leq 1/2 \text{ implies that } \mathcal{K} \geq (2p-1)^2. \tag{3.3}$$

On the other hand, taking into account the product structure of $P_{\pm 1}^N$, we have

$$\begin{aligned} \mathcal{K} &= N\left[P_1(-L) \log \tfrac{P_1(-L)}{P_{-1}(-L)} + P_1(L) \log \tfrac{P_1(L)}{P_{-1}(L)}\right] \\ &= N\left[(1/2 - 4\nu) \log\left(\tfrac{1-8\nu}{1+8\nu}\right) + (1/2 + 4\nu) \log\left(\tfrac{1+8\nu}{1-8\nu}\right)\right] = 8N\nu \log\left(\tfrac{1+8\nu}{1-8\nu}\right). \end{aligned}$$

The concluding quantity is $\leq O(1)N\nu^2$, provided that $\nu \leq 0.1$. Combining this observation and (3.3), we arrive at (3.2).

Observation 3 One can argue that the phenomenon discussed in Observation 2 is not too dangerous from the practical viewpoint. In reality, especially in an "uncertain one", treated in stochastic models, relative accuracy like 1% or 5% is more than satisfactory. This indeed is true in numerous applications, which, in our opinion, is the intrinsic reason for Stochastic Programming to be of significant practical value. At the same time, there are some unpleasant exceptions; the most disturbing, from applied viewpoint, is the one related to problems without *relatively complete recourse*. This is the issue we are consider next.

The above analysis, summarized in Theorem 2, implicitly depends on the assumptions (i) and (ii) formulated in the beginning of this section (which are parallel to the assumptions (a)-(c) specified in the previous section). When applied to two-stage stochastic programming with recourse these assumptions imply that the recourse is relatively complete, i.e., for every $x \in X$ and every possible realization of ξ, the second stage problem is feasible. If, on the other hand, for some $x \in X$ and $\xi \in \Xi$ the second stage problem is infeasible, we can formally set the value $F(x, \xi)$ of the second stage problem to be $+\infty$. In order to avoid such infinite penalizations and to restore the applicability of Theorem 2 one can introduce a finite penalty for infeasibility. In some cases this can reasonably solve the problem. However, in some situations the infeasibility may result in a catastrophic event. In that case the penalty could be huge. Translated into the sample size bounds considered in the previous section, this means huge variances in the estimate (2.19) or huge Lipschitz constant in (2.22), which makes these estimates useless. In a sense, in such situation "nothing works".

It is NP-hard even to check whether a given first-stage decision $x \in X$ leads to feasible, with probability 1, second-stage problem, and even in the case when the second-stage problem is as simple as

$$\operatorname*{Min}_{y} \langle q, y \rangle \ \text{ subject to } \ Tx + Wy \geq h, \tag{3.4}$$

with only the second-stage right hand side vector $h = h(\xi)$ being random.

> To see that a generic problem of checking whether (3.4) is feasible for a given x is NP-hard, consider the case when the constraints $Tx+Wy \geq h(\xi)$ read $y \leq 0, \ \ y + x \geq h(\xi)$, where $x, y \in \mathbb{R}$,
>
> $$h(\xi) := \textstyle\sum_{i,j} Q_{ij} \xi_i \xi_j,$$
>
> $Q = [Q_{ij}]$ is a given $d \times d$ symmetric matrix, and $\xi = (\xi_1, ..., \xi_d)$ is uniformly distributed in $[-1, 1]^d$. Here x results in feasible, with probability 1, second stage problem if and only if $x \geq \rho(Q)$, where
>
> $$\rho(Q) := \max_{\xi} \left\{ \langle \xi, Q\xi \rangle : \xi \in [-1, 1]^d \right\}.$$
>
> It is well-known that given x and Q, it is NP-hard to distinguish between the cases of $x \leq \rho(Q)$ and $x > 1.01\, \rho(Q)$. This NP-hard problem is, of course, not more difficult than to decide whether $x \geq \rho(Q)$. Note that replacing in the above example the uniform distribution on $[-1, 1]^d$ with the uniform distribution on the discrete set, of cardinality 2^d, of d-dimensional vectors with entries ± 1, we end up with an equally difficult problem.

Thus, if a two-stage (linear) problem has no relatively complete recourse (which in many applications is a rule rather than an exception), it is, in general, NP-hard just to find a feasible first-stage solution x (one which results in finite $f(x)$), not speaking about minimizing over these x's. As it was mentioned above, the standard way to avoid, to some extent, this difficulty is to pass to a penalized problem. For example, we can replace the second stage problem (2.1) with the penalized version:

$$\operatorname*{Min}_{y\geq 0,\, z\geq 0} \langle q, y\rangle + rz \ \text{ subject to } \ Tx + Wy \geq h - ze, \tag{3.5}$$

where e is vector of ones and $r >> 1$ plays the role of the penalty coefficient. With this penalization, the second stage problem becomes always feasible. At the same time, one can hope that with large enough penalty coefficient r, the first-stage optimal solution will lead to "nearly always nearly feasible" second-stage problems, provided that the original problem is feasible. Unfortunately, in the situation where one cannot tolerate arising, with probability bigger than α, a second-stage infeasibility z bigger than τ (here α and τ are given thresholds), the penalty parameter r should be of order of $(\alpha\tau)^{-1}$. In the "high reliability" case $\alpha << 1$ we end up with problem (3.5) which contains large coefficients, which can lead to large value of the Lipschitz constant L_r of the optimal value function $F_r(\cdot, \xi)$ of the penalized second stage problem. As a result, quite moderate accuracy requirements (like ε being of order of 5% of the optimal value of the true problem) can result in the necessity to solve (3.5) within a pretty high relative accuracy $\nu = \varepsilon/(DL_r)$ like 10^{-6} or less, with all unpleasant consequences of this necessity.

3.1 What is difficult in the two-stage case?

We already know partial answer to this question: generically, under the premise of Theorem 2 it is difficult to solve problem (1.1) (even a convex one) to a high relative accuracy $\nu = \varepsilon/(DL)$. Note, however, that the statistical arguments demonstrating that this difficulty lies in the nature of the problem work *only for the black-box setting of* (1.1) *considered so far*, that is, only in the case when the distribution P of ξ is not known in advance, and all we have in our disposal is a black box generating realizations of ξ. With a "good description" of P available, the results could be quite different, as it is clear when looking at problems ($\text{SP}_{\pm 1}$) – with the underlying distributions given in advance, the problems become trivial. Note that in reality stochastic models are usually equipped with known in advance and easy-to-describe distributions, like Gaussian, or Bernoulli, or uniform on $[-1, 1]^d$. Thus, it might happen that our conclusion "it is difficult to solve (1.1) to high accuracy" is an artifact coming from the black-box model we used, and we could overcome this difficulty by using more advanced solution techniques based on utilizing a given in advance and "simple" description of P. *Unfortunately, this virtual possibility does not exist in reality.* Specifically, it is shown in [17] that indeed it is difficult to solve to high accuracy already two-stage linear stochastic programs with complete recourse and easy-to-describe discrete distributions.

Another difficulty, which we have already discussed, is the case of two-stage linear problems without complete recourse or, more generally, convex problems (1.1) with only partially defined integrand $F(x,\xi)$. As we have seen, this difficulty arises already when looking for feasible first-stage solutions with known in advance simple distribution P.

3.2 Complexity of multi-stage stochastic problems

In a multi-stage stochastic programming setting random data ξ is partitioned into $T \geq 2$ blocks ξ_t, $t = 1, ..., T$, i.e., ξ_t is viewed as a (discrete time) random process, and the decisions are made at time instants $0, 1, ..., T$. At time t the decision maker already knows the realizations ξ_τ, $\tau \leq t$, of the process up to time t, while realizations of the "future" blocks are still unknown. The goal is to find the first-stage decisions x (which should not depend on ξ) and decision rules $y_t = y_t(\xi_{[t]})$ which are functions of $\xi_{[t]} := (\xi_1, ..., \xi_t)$, $t = 1, ..., T$, which satisfy a given set of constraints

$$g_i(\xi, x, y_1, ..., y_T) \leq 0,\ i = 1, ..., I, \tag{3.6}$$

and minimize under these restrictions the expected value of a given cost function $f(x, y_1, ..., y_T)$. Note that even in the case when the functions g_i do not depend of ξ, the left hand sides of the constraints (3.6) are functions of ξ, since all y_t are so, and that the interpretation of (3.6) is that these functional constraints should be satisfied with probability one.

In the sequel, we focus on the case of linear multi-stage problems

$$\begin{array}{lll}
\operatorname*{Min}\limits_{x,y(\cdot)} & \mathbb{E}_P\left\{\langle c_0, x\rangle + \sum_{t=1}^T \langle c_t(\xi_{[t]}), y_t(\xi_{[t]})\rangle\right\} & \\
\text{s.t.} & A_0^0 x \geq b^0 & (C_0) \\
& A_0^1(\xi_{[1]})x + A_1^1(\xi_{[1]})y_1(\xi_{[1]}) \geq b^1(\xi_{[1]}) & (C_1) \\
& A_0^2(\xi_{[2]})x + A_1^2(\xi_{[2]})y_1(\xi_{[1]}) + A_2^2(\xi_{[2]})y_2(\xi_{[2]}) \geq b^2(\xi_{[2]}) & (C_2) \\
& \quad\quad \cdots\cdots & \\
& A_0^T(\xi_{[T]})x + A_1^T(\xi_{[T]})y_1(\xi_{[1]}) + ... + A_T^T(\xi_{[T]})y_T(\xi_{[T]}) \geq b^T(\xi_{[T]}) & (C_T)
\end{array} \tag{3.7}$$

where $y(\cdot) = (y_1(\cdot), ..., y_T(\cdot))$ and the constraints $(C_1), ..., (C_T)$ should be satisfied with probability one. Problems (3.7) are called problems with *complete recourse*, if for every instant t and whatever decisions x, y_1,...y_{t-1} made at preceding instants, the system of constraints (C_t) (treated as a system of linear inequalities in variable y_t) is feasible for almost all realizations of ξ. The major focus of theoretical research is on multi-stage problems even simpler than (3.7), specifically, on problems with *fixed recourse* where matrices $A_t^t = A_t^t(\xi_{[t]})$, $t = 1, ..., T$, are assumed to be deterministic (independent of ξ).

We argue that multi-stage problems, even linear of the form (3.7) with complete recourse, *generically* are *computationally intractable* already when medium-accuracy solutions are sought. (**Of course, this does not mean that some specific cases**

of multi-stage stochastic programming problems cannot be solved efficiently.) Note that this claim is rather a *belief* than a statement which we can rigorously prove. It is even not a formal statement which can be true or wrong since, in particular, we do not specify what does "medium accuracy" mean[12)]. What we are trying to say is that we believe that in the multi-stage case (with T treated as varying parameter, and not as a once for ever fixed entity), even "moderately positive" results like the one stated in Theorem 2 are impossible. We are about to explain what are the reasons for our belief.

Often practitioners do not pay attention to a dramatic difference between two-stage and multi-stage case. It is argued that in both cases the problem of interest can be written in the form of (1.1), with appropriately defined integrand F. Specifically, in case of the linear two-stage problem, with relatively complete recourse, we have that $F(x, \xi) = \langle c, x \rangle + Q(x, \xi)$, where $Q(x, \xi)$ is the optimal value of the second stage problem (2.1). In the case of problem (3.7) with complete recourse, $F(x, \xi)$ is given by a recurrence as follows. We start with setting

$$F_T(x, y_1, ..., y_T, \xi_{[T]}) := \langle c_0, x \rangle + \langle c_1(\xi[1]), y_1 \rangle + ... + \langle c_{T-1}(\xi_{[T-1]}), y_{T-1} \rangle + \langle c_T(\xi_{[T]}), y_T \rangle$$

and specifying the conditional, given $\xi_{[T-1]}$, expected cost of the last-stage problem:

$$\begin{aligned} F_{T-1}(x, y_1, ..., y_{T-1}, \xi_{[T-1]}) := \; & \mathbb{E}_{|\xi_{[T-1]}} \operatorname*{Min}_{y_T} \Big\{ F_T(x, y_1, ..., y_{T-1}, y_T, \xi_{[T]}) : \\ & A_0^T(\xi_{[T]})x + A_1^T(\xi_{[T]})y_1 + ... + A_T^T(\xi_{[T]})y_T \geq b^T(\xi_{[T]}) \Big\}, \end{aligned}$$

where $\mathbb{E}_{|\xi_{[T-1]}}$ is the conditional, given $\xi_{[T-1]}$, expectation. Observe that (3.7) is equivalent to the $(T-1)$-stage problem:

$$\begin{array}{cl} \operatorname*{Min}_{x, \{y_t(\cdot)\}_{t=1}^{T-1}} & \mathbb{E}_{P^{T-1}} \left\{ F_{T-1}(x, y_1, ..., y_{T-1}, \xi_{[T-1]}) \right\} \\ \text{s.t.} & x, y_1(\cdot), ..., y_{T-1}(\cdot) \text{ satisfy } (C_0), (C_1), ..., (C_{T-1}) \text{ w.p.1}, \end{array} \qquad (P_{T-1})$$

where P^{T-1} is the distribution of $\xi_{[T-1]}$. Now we can iterate this construction, ending up with the problem

$$\operatorname*{Min}_{x \in X} \left[F_0(x) \right].$$

It can be easily seen that under the assumption of complete recourse, plus mild boundedness assumptions, all functions $F_\ell(x, y_1, ..., y_\ell, \xi_{[\ell]})$ are Lipschitz continuous in the x, y-arguments.

The "common wisdom" says that since both, two-stage and multi-stage, problems are of the same generic form (1.1), with the integrand convex in x, and both are processed numerically by generating a sample of scenarios and solving the resulting

[12)] To the best of our knowledge, the complexity status of problem (3.7), even in the case of complete and fixed recourse and known in advance easy-to-describe distribution P, remains unknown (cf., [17]).

"scenario counterpart" of the problem of interest, there should be no much difference between the two and the multi-stage case, provided that in both cases one uses the same number of scenarios. This "reasoning", however, completely ignores a crucial point as follows: in order to solve generated SAA problems efficiently, the integrand F should be efficiently computable at every pair (x, ξ). This is indeed the case for a two-stage problem, since there $F(x, \xi)$ is the optimal value in an explicit Linear Programming problem and as such can be computed in polynomial time. In contrast to this, the integrand F produced by the outlined scheme, as applied to a multi-stage problem, is *not* easy to compute. For example, in 3-stage problem this integrand is the optimal value in a 2-stage stochastic problem, so that its computation at a point is a much more computationally involving task than similar task in the two-stage case. Moreover, in order to get just consistent estimates in an SAA type procedure (not talking about rate of convergence) one needs to employ a conditional sampling which typically results in an exponential growth of the number of generated scenarios with increase of the number T of stages (cf., [40]).

Analysis demonstrates that for an algorithm of the SAA type, the total number of scenarios needed to solve T-stage problem (3.7), with complete recourse, would grow, as ε diminishes, as ε^{-2T}, so that the computational effort blows up exponentially as the number of stages grows[13)] (cf., [44]). Equivalently, *for a sampling-based algorithms with a given number of scenarios, existing theoretical quality guarantees deteriorate dramatically as the number of stages grows.* Of course, nobody told us that sampling-type algorithms are the only way to handle stochastic problems, so that the outlined reasoning does not pretend to justify "severe computational intractability" of multi-stage problems. Our goal is more modest, we only argue that the fact that when solving a particular stochastic program a sample of 10^7 scenarios was used does not say much about the quality of the resulting solution: in the two-stage case, there are good reasons to believe that this quality is reasonable, while in the 5-stage the quality may be disastrously bad.

We have described one source of severe difficulty arising when solving multi-stage stochastic problems – dramatic growth, with increase of the number of stages, in the complexity of evaluating the integrand F in representation (1.1) of the problem. We are about to demonstrate that even when this difficulty does not arise, a multi-stage problem still may be very difficult. To this end, consider the following story: at time $t = 0$, one has \$ 1, and should decide how to distribute this money between stocks and a bank account. When investing amount of money x into stocks, the value u_t of the portfolio at time t will be given by chain of t relations

$$u_1 = \rho_1(\xi_{[1]})x, u_2 = \rho_2(\xi_{[2]})u_1, ..., u_t = \rho_t(\xi_{[t]})u_{t-1},$$

where the returns $\rho_t(\xi_{[t]}) \equiv \rho_t(\xi_1, ..., \xi_t) \geq 0$ are known functions of the underlying random parameters. Amount of money $1 - x$ put to bank account reach at time t the

[13)] Note that in the considered framework, $T = 1$ corresponds to two-stage programming, $T = 2$ corresponds to 3-stage programming, and so on.

value $v_t = \rho^t(1-x)$, where $\rho > 0$ is a given constant. The goal is to maximize the total expected wealth $\mathbb{E}[u_T + v_T]$ at a given time T. The problem can be written as a simple-looking T-stage stochastic problem of the form (3.7):

$$\begin{array}{lll}
\underset{x,y(\cdot)}{\text{Min}} & \mathbb{E}_P\left[u_T(\xi^T) + v_T(\xi^T)\right] & \\
\text{s.t.} & 0 \le x \le 1 & (C_0) \\
& u_1(\xi_{[1]}) = \rho_1(\xi_{[1]})x,\ v_1(\xi_{[1]}) = \rho(1-x) & (C_1) \\
& u_2(\xi_{[2]}) = \rho_2(\xi_{[2]})u_1(\xi_{[1]}),\ v_2(\xi_{[2]}) = \rho v_1(\xi_{[1]}) & (C_2) \\
& \quad\quad \cdots\cdots & \\
& u_T(\xi_{[T]}) = \rho_{T-1}(\xi_{[T-1]})u_{T-1}(\xi_{[T-1]}),\ v_T(\xi_{[T]}) = \rho v_{T-1}(\xi_{[T-1]}) & (C_T),
\end{array} \tag{3.8}$$

where $y(\cdot) = (u_t(\cdot), v_t(\cdot))_{t=1}^T$. Now let us specify the structure and the distribution of ξ as follows: a realization of ξ is a permutation $\xi = (\xi_1, ..., \xi_T)$ of T elements $1, ..., T$, and P is the uniform distribution on the set of all $T!$ possible permutations. Further, let us specify the returns as follows: the returns are given by a $T \times T$ matrix A with 0-1 elements, and

$$\rho_t(\xi_1, ..., \xi_t) := \kappa A_{t,\xi_t}, \quad \kappa := (T!)^{1/T}$$

(Note that by Stirling's formula $\kappa = (T/e)(1 + o(1))$ as $T \to \infty$.) We end up with a simple-looking instance of (3.7) with complete recourse and given in advance "easy-to-describe" discrete distribution P; when represented in the form of (1.1), our problem becomes

$$\underset{x\in[0,1]}{\text{Min}} \left\{f(x) := \mathbb{E}_P F(x,\xi)\right\}, \quad F(x,\xi) = \rho^T(1-x) + x\prod_{t=1}^T(\kappa A_{t\xi_t}), \tag{3.9}$$

so that F indeed is easy to compute. Thus, problem (3.8) looks nice – complete recourse, simple and known in advance distribution, no large data entries, easy-to-compute F in representation (1.1). At the same time the problem is disastrously difficult. Indeed, from (3.9) it is clear that $f(x) = \rho^T(1-x) + x\,\text{per}(A)$, where $\text{per}(A)$ is the permanent of A:

$$\text{per}(A) = \sum_\xi \prod_{t=1}^T A_{t\xi_t},$$

(the summation is taken over all permutations of T elements $1, ..., T$). Now, the solution to (3.9) is either $x = 1$ or $x = 0$, depending on whether or not $\text{per}(A) \ge \rho^T$. Thus, our simple-looking T-stage problem is, essentially, the problem of computing the permanent of a $T \times T$ matrix with 0-1 entries. The latter problem is known to be really difficult. First of all, it is NP-hard, [47]. Further, there are strong theoretical reasons to doubt that the permanent can be efficiently approximated within a given relative accuracy ε, provided that $\varepsilon > 0$ can be arbitrarily small, [12]. The best known to us algorithm capable to compute permanent of a $T \times T$ 0-1 matrix within relative accuracy ε has running time as large as $\varepsilon^{-2}\exp\{O(1)T^{1/2}\log^2(T)\}$ (cf., [23]), while the best known to us efficient algorithm for approximating permanent has relative error as large as c^T with certain fixed $c > 1$, see [25]. Thus, simple-looking multi-stage stochastic problems can indeed be extremely difficult...

A reader could argue that in fact we deal with a two-stage problem (3.9) rather than with a multi-stage one, so that the outlined difficulties have nothing to do with our initial multi-stage setting. Our counter-argument is that the two-stage problem (3.9) honestly says about itself that it is very difficult: with moderate ρ and T, the data in (3.9) can be astronomically large (look at the coefficient ρ^T of $(1-x)$ or at the products $\prod_{t=1}^{T}(\kappa A_{t\xi_t})$ which can be as large as $\kappa^T = T!$), and so is the Lipschitz constant of F. In contrast to this, the structure and the data in (3.8) look completely normal. Of course, it is immediate to recognize that this "nice image" is just a disguise, and in fact we are dealing with a disastrously difficult problem. Imagine, however, that we add to (3.8) a number of redundant variables and constraints; how could your favorite algorithm (or you, for that matter) recognize in the resulting messy problem that solving it numerically is, at least at the present level of our knowledge, a completely hopeless endeavor?

4 Some novel approaches

Here we outline some novel approaches to treating uncertainty which *perhaps* can cope, to some extent, with intrinsic difficulties arising in two-stage problems without complete recourse and in multi-stage problems.

4.1 Tractable approximations of chance constraints

As it was already mentioned, a natural way to handle two-stage stochastic problems without complete recourse is to impose *chance* constraints. That is, to require that a probability of insolvability of the second-stage problem is at most $\varepsilon << 1$ instead of being 0. The rationale behind this idea is twofold: first, from the practical viewpoint, "highly unlikely" events are not too dangerous: why should we bother about a marginal chance, like 10^{-6}, for the second stage to be infeasible, given that the level of various inaccuracies in our model, especially in its probabilistic data, usually is by orders of magnitude larger than 10^{-6}? Not speaking of the fact that 5 days a week we take worse chances in the morning traffic. Second, while it might be very difficult to check whether a given first-stage solution results in a feasible, with probability 1, second-stage problem, it seems to be possible to check whether this probability is at least $1-\varepsilon$ by applying Monte-Carlo simulation. Note that chance constraints arise naturally not only in the context of two-stage problems without complete recourse, but in a much more general situation of solving a constrained optimization problem with the data affected by stochastic uncertainty. Thus, it makes sense to pose a question *how could one process numerically a chance constraint*

$$\phi(x) := \mathrm{Prob}\big\{g(x,\xi) \le 0\big\} \ge 1-\varepsilon, \tag{4.10}$$

where x is the decision vector, ξ is the random disturbance with, say, known distribution, and $\varepsilon << 1$ is a given tolerance.

The concept of chance constraints originates from [11] and is one of the oldest concepts in Operations Research. Unfortunately, in its nearly 50 year old age, this concept still cannot be treated as practical. The first reason is that typically it is extremely difficult to verify *exactly* whether this constraint is satisfied at a given point. This problem is difficult already in the case of a single linear constraint $g(x,\xi) := \langle a_* + \xi, x\rangle$ with perturbations ξ uniformly distributed in a box. Another severe problem is that usually constraint (4.10), even with very simple, say bi-affine in x and in ξ, function $g(x,\xi)$ and simple-looking distribution of ξ (like uniform in a box) defines a nonconvex feasible set in the space of decision variables, which makes problematic subsequent optimization over this set of even pretty simple – just linear – objectives.

> The difficulty we have just outlined rules out the idea to approximate (4.10) by a "sample version" of this constraint, that is, by
>
> $$\widehat{\phi}_N(x) := \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}_{\{g(x,\xi^j)\le 0\}} \ge 1-\theta\varepsilon, \tag{4.11}$$
>
> where $\xi^1, ..., \xi^N$ is a sample of N independent realizations of ξ, $\mathbb{1}_{\{g(x,\xi^j)\le 0\}}$ is the indicator function[14] of the event $\{g(x,\xi^j)\le 0\}$, and $\theta < 1$ is fixed (say, $\theta = 0.99$). When $N >> \varepsilon^{-1}$, the validity of (4.11) at a point x implies, with probability close to 1, the validity of (4.10), so that (4.11) can be thought of as a "computable approximation" of (4.10). Unfortunately, the left hand side in (4.10) is, generically, a nonconvex (and even discontinuous) function of x, so that we have no way to optimize under this constraint.

To the best of our knowledge, the only generic case where both these severe difficulties disappear is the case of linear constraint $\langle a_* + \xi, x\rangle \le 0$ with normally distributed data $\xi \sim \mathcal{N}(0,\Sigma)$. In this case, (4.10) is equivalent to the convex deterministic constraint

$$\langle a_*, x\rangle + \Omega(\varepsilon)\sqrt{\langle x, \Sigma x\rangle} \le 0, \tag{4.12}$$

where the "safety parameter" $\Omega(\varepsilon) = \sqrt{2\log(1/\varepsilon)}(1+o(1))$, $\varepsilon \to 0$, is readily given by ε (which we assume to be $\le 1/2$).

> There is another generic case when the feasible set given by a chance constraint is convex. This is the case when the constraint can be represented

[14] $\mathbb{1}_A = 1$ if the event A happens, and $\mathbb{1}_A = 0$ otherwise

in the form $(x,\xi) \in Q$, where Q is a closed and convex set, and the distribution P of the random vector $\xi \in \mathbb{R}^d$ is *logarithmically quasi-concave*, meaning that

$$P(\lambda A + (1-\lambda)B) \geq \max\left[P(A), P(B)\right]$$

for all closed and convex sets $A, B \subset \mathbb{R}^d$ (cf., Prekopa [31]). Examples include uniform distributions on closed and bounded convex domains, normal distribution and every distribution on $\mathbb{R}^d$ with density $f(\xi)$ with respect to the Lebesgue measure such that the function $f^{-1/d}(\xi)$ is convex. The related result (due to Prekopa [31]) is that in the situation in question, the set $\left\{x : P(\{\xi : (x,\xi) \in Q\}) \geq \alpha\right\}$ is closed and convex for every α. This result can be applied, e.g., to two-stage stochastic programs with chance constraints of the form

$$\operatorname*{Min}_{x \in X} \langle c, x\rangle \ \text{ s.t. } \ \text{Prob}\{\exists y \in Y : Tx + Wy \geq \xi\} \geq 1 - \varepsilon,$$

where X, Y are closed convex sets and T, W are fixed matrices. Here the chance constraint indeed is of the form $\text{Prob}\{(x,\xi) \in Q\} \geq 1 - \varepsilon$, where

$$Q = \{(x,\xi) : \exists y \in Y : Tx + Wy \geq \xi\}.$$

The set Q clearly is convex; under mild additional assumptions, it is also closed. Thus, the feasible set of the chance constraint in question is convex, provided that the distribution of ξ is logarithmically quasi-concave.

Note that the outlined convexity results are applicable only to the chance constraints coming from scalar or vector inequalities where the only term affected by uncertainty is the right hand side, not the coefficients at the variables. For example, nothing similar is known for the chance constraint

$$\text{Prob}\left\{\langle a_* + \xi, x\rangle \leq 0\right\} \geq 1 - \varepsilon,$$

except for the already mentioned case of normally distributed vector ξ.

Aside of few special cases we have mentioned, chance constraint (4.10) “as it is” seems to be too difficult for efficient numerical processing, and what we can try to do is to replace it with its “tractable approximation”. For the time being, there exist two approaches to building such an approximation: “deterministic” and “scenario”.

Tractable deterministic approximations of chance constraints. With this approach, one replaces (4.10) with a properly chosen deterministic constraint

$$\psi_\varepsilon(x) \leq 0, \tag{4.13}$$

which is a “safe computationally tractable” approximation of (4.10), with the latter notion defined as follows:

1. "Safety" means that the validity of (4.13) is a *sufficient* condition for the validity of (4.10);

2. "Tractability" means that (4.13) is an explicitly given convex constraint.

Just to give an example, consider a randomly perturbed linear constraint, that is, assume that

$$g(x,\xi) := \langle a_* + M\xi, x\rangle,$$

where the deterministic vector a_* is the "nominal data", M is a given deterministic matrix and $\xi = (\xi_1, ..., \xi_d)$ is a tuple of d independent scalar random variables with zero mean and "of order of 1":

$$\mathbb{E}\left[\exp(\xi_i^2)\right] \le \exp\{1\}, \;\; i = 1, ..., d,$$

e.g., ξ_i can have a distribution supported on the interval $[-1, 1]$, or ξ_i can have normal distribution $\mathcal{N}(0, 2^{-1/2})$, $= 1, ..., d$. In this case, applying standard results on probabilities of large deviations for sums of "light tail" independent random variables with zero means, one can easily verify that when $\varepsilon \in (0,1)$ and $\Omega(\varepsilon) = O(1)\sqrt{\log(1/\varepsilon)}$ with properly chosen absolute constant $O(1)$, then the validity of the convex constraint

$$\langle a_*, x\rangle + \Omega(\varepsilon)\sqrt{\langle x, MM^T x\rangle} \le 0 \tag{4.14}$$

is a sufficient condition for the validity of (4.10). (Note that under our assumptions MM^T is an upper bound on the covariance matrix of ξ, and compare with (4.12).)

The simple result we have just described is rather attractive. First, it does not require a detailed knowledge of the distribution of ξ. Second, the approximation, although being more complicated than a linear constraint we start with, still is pretty simple; modern convex optimization techniques can process routinely to high accuracy problems with thousands of decision variables and thousands of constraints of the form (4.14). Third, the approximation is "not too conservative" – the safety parameter $\Omega(\varepsilon)$ grows pretty slowly as $\varepsilon \to 0$ and is only by a moderate constant factor larger than the safety parameter in the case of Gaussian noise, where our approximation is not conservative at all.

Recently, "not too conservative" computationally tractable safe approximations were built (see [29]) for chance versions of well-structured *nonlinear* convex constraints with nice analytic structure, specifically, for affinely perturbed least squares constraints

$$\Big\| \big[A_* + \sum_i A_i\xi_i\big]x - \big[b_* + \sum_i \xi_i b_i\big] \Big\|_2 \le \tau$$

and Linear Matrix Inequality constraints

$$\big[A_0^0 + \sum_i \xi_i A_i^0\big] + \sum_{j=1}^m x_j\big[A_0^j + \sum_i \xi_i A_i^j\big] \succeq 0$$

(A^p_q are symmetric matrices, $A \succeq 0$ means that A is symmetric positive semidefinite). In both cases, ξ_i are independent scalar disturbances with zero mean and "of order of 1". However, the outlined approach, whatever promising we believe it is, seemingly works for a very restricted family of "well-structured" functions $g(x, \xi)$, and even in these cases requires a lot of highly nontrivial "tailoring" to a particular structure in question. Consider, for example, the case of chance constraint associated with two-stage linear stochastic problem:

$$g(x, \xi) := \operatorname*{Min}_{z,y} \{z : T(\xi)x + W(\xi)y \geq h(\xi) - ze, z \geq 0\}, \tag{4.15}$$

where e is vector of ones. Note that here $g(x, \xi)$ is convex in x, and $g(x, \xi) \leq 0$ if and only if the second-stage problem

$$\operatorname*{Min}_{y} \langle q(\xi), y\rangle \ \text{ s.t. } \ T(\xi)x + W(\xi)y \geq h(\xi)$$

is feasible (cf., (3.5)). Thus, the chance constraint requires from x to result in a feasible, with probability at least $1 - \varepsilon$, second stage problem. Even in the case of simple recourse (T, W are independent of ξ) the chance constraint in question seems to be by far too difficult to admit a safe tractable deterministic approximation.

Scenario approximation. In contrast to the "highly specialized and heavily restricted" approach we have just considered, the scenario-based approach is completely universal. We just generate a sample $\xi^1, ..., \xi^N$ of N "scenarios" – independent realizations of the random disturbance ξ – and approximate (4.10) by the random system of inequalities

$$g(x, \xi^j) \leq 0, \ j = 1, ..., N. \tag{4.16}$$

Extremely nice features of this approach are its generality and computational tractability – whenever $g(x, \xi)$ is convex in x and efficiently computable (as it is the case, e.g., with the function (4.15)), (4.16) becomes a system of explicitly given convex constraints and as such can be efficiently processed numerically, *provided that the number of scenarios N is not prohibitively large. The* question, of course, is how large should be the sample in order to ensure, with reliability close to 1, that every feasible solution to (4.16) satisfies the chance constraint (4.10). This question is by far not easy, and we do not intend to discuss relevant nice and deep results known from the literature, since in fact we are more interested in a slightly different question, namely, as follows:

(Q) *Assume we are given a convex optimization problem*

$$\operatorname*{Min}_{x \in \mathbb{R}^n} f(x) \ \text{ s.t. } g(x, \xi) \leq 0 \tag{4.17}$$

(all f, g are convex in x) with ξ being a random vector with a known distribution, and, given tolerance $\varepsilon > 0$, replace this problem with its "scenario counterpart"

$$\operatorname*{Min}_{x \in \mathbb{R}^n} f(x) \text{ s.t. } g(x, \xi^j) \leq 0, j = 1, ..., N. \tag{4.18}$$

How large should be the sample size N in order for the optimal solution $\tilde{x}_N$ of (4.18) to be feasible for (4.17) *with probability at least* $1 - \varepsilon$*?*

The difference between the latter question and the former one is that now we do not require from *all* points feasible for (4.16) to satisfy (4.10), we require this property to be possessed by a specific point, $\tilde{x}_N$, we are interested in.

As it was discovered in [9, 10], question (Q) admits a nice "universal" answer. Namely, under extremely mild assumptions it turns out that whenever $\varepsilon, \delta \in (0, 1/2)$ and

$$N \geq \frac{2n}{\varepsilon} \log\left(\frac{12}{\varepsilon}\right) + \frac{2}{\varepsilon} \log\left(\frac{2}{\delta}\right) + 2n, \tag{4.19}$$

the probability of "bad sampling" which results in $\tilde{x}_N$ *not* satisfying (4.10) is less than or equal to δ. Note that this result, which heavily utilizes the convexity of (4.17), is completely "distribution-free" – it is independent of any assumptions on the distribution of ξ and requires no knowledge of this distribution.

All this being said, there is a serious problem with the scenario approach as presented so far – it becomes impractical when the required value of ε is really small, like 10^{-6} or 10^{-8}. Indeed, for those ε relation (4.19) results in unrealistically large samples. Note that pretty small values of ε are completely reasonable when speaking about a "hard" constraint $g(x, \xi) \leq 0$, that is, such that its violation has very severe or even catastrophic consequences, like heavy jam in a communication network, a blackout caused by malfunctioning of a power supply network, not speaking about exploding nuclear power plants or airliners falling from the sky. In a sense, in the context of chance constraints hard restrictions and implied pretty small values of ε seem to be a rule rather than exception. Indeed, "soft" constraints – those with ε like 1% or 0.1% – can be eliminated altogether by augmenting the objective with appropriate penalties[15)].

One could be surprised by the fact that we treat as acceptable the SAA method with the complexity proportional to ε^{-2}, ε being the required tolerance in terms

[15)] It should be added that the outlined "crude" scenario approach is not completely satisfactory even when ε is not too small. Indeed, assume that your problem has $n = 100$ variables and you are ready to take 10% chances ($\varepsilon = \delta = 0.1$). To this end, you use the scenario approach with the smallest N allowed by (4.19), that is, $N = 9835$. What should be the actual probability ε' for a fixed point $\bar{x}$ to violate the constraint $g(x, \xi) \leq 0$ in order to be feasible for (4.18) with probability 0.9? The answer is: ε' should be as small as 10^{-5}. Thus, when applied with small ε, the crude scenario approach becomes impractical, while in the case of "large" ε it seems to be too conservative.

> of the objective, and are dissatisfied with the scenario approach where the sample size is merely inverse proportional to the tolerance ε. To explain our point, think whether you will agree (a) to use a portfolio management policy with the average profit by at most 0.5% less than the "ideal" – the optimal – one, and (b) to board an airliner which may crash during the flight with probability 0.5% (or 0.05%).

When handling hard chance constraints – those with really small ε, like 10^{-6} or less – we would like to have sample sizes polynomial in both $\log(1/\varepsilon)$ and $\log(1/\delta)$ rather than to be polynomial in $(1/\varepsilon)$ and $\log(1/\delta)$. We are about to explain that under favorable circumstances, such a possibility does exist; it is given by combining scenario approach with a kind of *importance sampling*. To proceed, assume that the constraint $g(x, \xi) \leq 0$ underlying (4.10) is of a specific structure as follows: there exists a closed convex set $K \subset \mathbb{R}^m$ and an affine mapping $x \mapsto A[\xi]x + b[\xi] : \mathbb{R}^n \to \mathbb{R}^m$ depending on ξ as on a parameter such that

$$g(x, \xi) \leq 0 \Leftrightarrow A[\xi]x + b[\xi] \in K. \tag{4.20}$$

Moreover, let us assume that the affine mapping in question is affinely parameterized by ξ, that is, both $A[\xi]$ and $b[\xi]$ depend affinely on ξ. Finally, we may assume without loss of generality that ξ has zero mean.

> As an instructive example, consider the feasibility constraint associated with the second-stage problem, that is, the constraint $g(x, \xi) \leq 0$ with $g(x, \xi)$ given by (4.15). Assuming fixed recourse, that is, $W(\xi) \equiv W$ being independent of ξ, let us set
>
> $$K := \{u : \exists\, y \text{ such that } u \leq Wy\}.$$
>
> Note that K is a convex polyhedral (and thus closed) set. Now, it is clear from (4.15) that $g(x, \xi) \leq 0$ if and only if $h(\xi) - T(\xi)x \in K$. It follows that when passing from uncertain parameter ξ to the new uncertain parameter $\bar{\xi} = [h(\xi), T(\xi)] - \mathbb{E}\{[h(\xi), T(\xi)]\}$ and updating accordingly the underlying distribution, we arrive at the situation described in (4.20).

Under our assumptions, the vector $A[\xi]x + b[\xi]$ is affine in ξ, and thus can be represented as $\alpha[x]\xi + \beta[x]$, where $\alpha[x]$, $\beta[x]$ are affine in x. It follows that

$$g(x, \xi) \leq 0 \Leftrightarrow A[\xi]x + b[\xi] \in K \Leftrightarrow \xi \in K_x := \{u : \alpha[x]u + \beta[x] \in K\}. \tag{4.21}$$

Note that the set K_x is closed and convex along with K. Now, numerous important distributions Π on $\mathbb{R}^p$ with zero mean (multivariate normal, uniform on a multidimensional box, etc.) possess a kind of "concentration property" as follows: if Q is a closed convex set in $\mathbb{R}^p$ and $\Pi(Q) \geq c$, where $c < 1$ is a characteristic constant of Π, then the probability of the event $\Omega^{-1}\eta \in Q$, $\eta \sim P$, rapidly approaches 1 as

$\Omega > 1$ grows, namely, $\Pi(\{\eta : \Omega^{-1}\eta \notin Q\}) \leq C^{-1}\exp\{-C\Omega^2\}$, where C is another characteristic constant of Π. For example, in the case of multivariate normal distribution Π with zero mean, then $\Pi(Q) \geq 0.8$ implies, for a closed convex set Q, that $\Pi(\{\eta : \eta/\Omega \notin Q\}) \leq \exp\{-\Omega^2/3\}$.

Now assume that we are in the situation of (4.21) and that the distribution of ξ possesses the outlined concentration property. Let us choose somehow a safety parameter $\Omega > 1$, and consider the scenario counterpart of (4.10), *where the disturbances are drawn from the distribution of* $\Omega\xi$ *rather than from the distribution of* ξ:

$$\begin{array}{c} g(x, \Omega\xi^t) \leq 0,\ t = 1, ..., N \\ \Updownarrow \\ \Omega\xi^t \in K_x,\ t = 1, ..., N \end{array} \tag{4.22}$$

where $\xi^t \sim P$ are independent. Specifying N as

$$O(1)(1-c)^{-1}\log(1/\delta) \tag{4.23}$$

with appropriate absolute constant $O(1)$, observe that if a *fixed* x satisfies (4.22), then it is "highly likely" that $\text{Prob}\{g(x, \Omega\xi) \leq 0\} \geq c$; specifically, in the case of $\text{Prob}\{g(x, \Omega\xi) \leq 0\} < c$, the probability to get a realization of N disturbances (with N given by (4.23)) which results in (4.22) is at most δ. Thus, when a given x turns out to satisfy (4.22), then, up to probability of "bad sampling" as small as δ, we have $\text{Prob}\{g(x, \Omega\xi) \leq 0\} \equiv \text{Prob}\{\Omega\xi \in K_x\} \geq c$. In the latter case, due to the concentration property of the distribution Π of $\eta = \Omega\xi$ (induced by similar property of the distribution P of ξ), we have $\text{Prob}\{g(x, \xi) > 0\} = \text{Prob}\{\xi \notin K_x\} \leq C^{-1}\exp\{-C\Omega^2\}$. When $\Omega = \sqrt{C^{-1}\log(C^{-1}\varepsilon^{-1})}$, the latter probability is $\leq \varepsilon$, that is, x satisfies the chance constraint (4.10). For example, in the case when P is a multivariate normal distribution with zero mean and ε in (4.10) is as small as 10^{-12}, the above rule results in $\Omega = 9.1$. Thus, when $\xi \sim \mathcal{N}(0, \Sigma)$, N is given by (4.23) and $\Omega = 9.1$, a fixed point x which satisfies (4.22) is, up to probability of "bad sampling" at most δ, feasible for the chance constraint (4.10) with $\varepsilon = 10^{-12}$.

The outlined idea – to apply the scenario approach with moderately amplified disturbances rather than with "true ones" – under favorable circumstances allows to approximate chance constraints via samples of size N which is polynomial in the "sizes" of the problem (the dimensions of x, ξ and K) and *logarithms* of $1/\varepsilon$, $1/\delta$, and thus allows to handle efficiently constraints (4.10) with really small tolerances ε. For detailed presentation and analysis of this approach, see [30].

4.2 Multistage Stochastic Programming in linear decision rules

Consider a linear multi-stage stochastic program

$$\operatorname*{Min}_{x, y(\cdot)} \mathbb{E}_P\Big[\langle c_0(\xi), x\rangle + \sum_{t=1}^{T}\langle c_t, y_t(\xi_{[t]})\rangle\Big] \text{ s.t. } A_0(\xi)x + \sum_{t=1}^{T} A_t y_t(\xi_{[t]}) \geq b(\xi) \tag{4.24}$$

with fixed recourse, where the cost coefficients c_t and the matrices A_t, $t \geq 1$, are not affected by uncertainty, as reflected in the notation. Besides this, in what follows we assume that the data affected by the uncertainty (that is, $c_0(\xi)$, $A_0(\xi)$, $b(\xi)$) are *affine* functions of ξ; as we remember from the previous section, this "assumption" is in fact a convention on how we use words: nobody forbids us to treat as the actual "random parameter" the collection $(c(\xi), A_0(\xi), b(\xi))$ rather than ξ itself.

As we have explained, a multistage problem (even much better structured than (4.24)) is, generically, "severely computationally intractable". We are about to propose a radical way to reduce the complexity of the problem, specifically, to pass from *arbitrary* decision rules $y_t(\cdot)$ to *affine* ones:

$$y_t(\xi) = x_t^0 + X_t Q_t \xi, \tag{4.25}$$

where x_t^0, X_t are our new – deterministic! – variables (a vector and a matrix of appropriate sizes), and $Q_t\xi$, Q_t being a given deterministic matrix, is the "portion" of uncertainty which is revealed at time t and thus can be used to make the decision y_t [16)].

Now let us look at the problem we end up with. When substituting linear decision rules (4.25) into the constraint of (4.24), the constraint takes the form

$$\mathrm{Prob}\left\{ A_0(\xi)x + \sum_{t=1}^{T} \left[A_t x_t^0 + A_t X_t Q_t \xi \right] - b(\xi) \geq 0 \right\} = 1.$$

The left hand side of the system of inequalities in the latter $\mathrm{Prob}\{\cdot\}$ is affine in ξ, thus, the constraint in question says *exactly* that the system should be satisfied for all ξ from the support Ξ of the distribution P of ξ. Since the left hand side of the system is affine in ξ, the latter requirement is equivalent to the system to be valid for all $\xi \in \mathcal{Z}$, where $\mathcal{Z}$ is the closed convex hull of Ξ. Thus, the constraint of (4.24) is nothing but the semi-infinite system of linear inequalities

$$A_0(\xi)x + \sum_{t=1}^{T} \left[A_t x_t^0 + A_t X_t Q_t \xi \right] - b(\xi) \geq 0 \quad \forall \xi \in \mathcal{Z} \tag{4.26}$$

in variables $w = \{x, \{x_t^0, X_t\}_{t=1}^T\}$. Besides this, the coefficients of the semi-infinite inequalities in (4.26) depend affinely on ξ. Now let us use the following known fact (see [5]):

> (!) *Assume that $\mathcal{Z}$ is a polyhedral set*
>
> $$\mathcal{Z} = \{\xi : \exists\, \eta \text{ such that } M\xi + N\eta + p \geq 0\},$$
>
> *given by the data M, N, p. Then the semi-infinite system* (4.26) *is equivalent to a finite system $\mathcal{S}$ of linear inequalities:*
>
> $$w \text{ satisfies (4.26) } \Leftrightarrow \exists u : \mathcal{A}w + \mathcal{B}u + q \geq 0.$$

[16)] In the notation of (3.7), $Q_t\xi = \xi_{[t]} = (\xi_1, ..., \xi_t)$.

The sizes of $\mathcal{S}$ (that is, the row and the column sizes of $\mathcal{A},\mathcal{B}$) are polynomial in the sizes of the matrices A_0, A_1,...,A_T, M, N, and the data $\mathcal{A},\mathcal{B},q$ of $\mathcal{S}$ are readily given by the data of (4.26) *and* M, N, p (that is, given the latter data, one can build $\mathcal{S}$ in polynomial time).

In fact, [5] asserts much more than stated by (!), namely, that (4.26) is computationally tractable whenever $\mathcal{Z}$ is so. We, however, intend to stay within the grasp of Linear Programming, and to this end (!) is exactly what we need.

Example: interval uncertainty. Assume that $\mathcal{Z}$ is a box; without loss of generality, we may assume that $\mathcal{Z} = \{\xi : -1 \leq \xi_i \leq 1, i = 1, ..., d\}$. Since $A_0(\xi)$, $b(\xi)$ are affine in ξ, (4.26) can be rewritten equivalently as the semi-infinite problem

$$s_0^j[X] + \sum_{i=1}^{d} s_i^j[X]\xi_i \leq 0 \ \forall \xi \in \mathcal{Z},\ j = 1, ..., J, \tag{4.27}$$

where X stands for the collection $\{x, \{x_t^0, X_t\}_{t=1}^T\}$ of design variables in (4.26), and $s_i^j[X]$ are affine functions of X readily given by the data of (4.26). With our $\mathcal{Z}$, the semi-infinite system (4.27) is clearly equivalent to the system of constraints

$$s_0^j[X] + \sum_{i=1}^{d} |S_i^j[X]| \leq 0,\ j = 1, ..., J,$$

that is, to an explicit system of convex constraints (which can be further straightforwardly converted to a system of linear inequalities).

By the outlined analysis, *when restricted to affine decision rules*, (4.24) *becomes an explicit deterministic linear program*

$$\begin{array}{c} \mathrm{Min}_{w=\left\{x,\{x_t^0,X_t\},u\right\}} \left\{\langle c, w\rangle : \mathcal{A}w + \mathcal{B}u + q \geq 0\right\}, \\ \langle c, w\rangle \equiv \mathbb{E}\left\{\langle c_0(\xi), x\rangle + \sum_{t=1}^T \langle c_t, [x_t^0 + X_t P_t \xi]\rangle\right\}. \end{array} \tag{4.28}$$

in variables $w = \{x, \{x_t^0, X_t\}_{t=1}^T\}$.

Several remarks are in order.

Remark 2 The only reason for restricting ourselves with affine decision rules stems from the desire to end up with a computationally tractable problem. We do not pretend that affine decision rules approximate well the optimal ones – whether it is so or not, it depends on the problem, and we usually have no possibility to understand how good in this respect is a particular problem we should solve. The rationale behind restricting to affine decision rules is the belief that in actual applications it is better to pose a modest and achievable goal rather than an ambitious goal which we do not know how to achieve[17)].

Remark 3 To some extent, what is affine and what is not is a matter of how we use words. Assume, e.g., that one wants to pass from affine decision rules to quadratic ones. This is exactly the same as to keep the rules affine and to add to the entries of ξ their pairwise products, and similarly for more complicated families of decision rules. Statement (!) explains what are the "limits of sophistication in the decision rules" we can achieve: representing a sophisticated decision rule as an affine one, the uncertainty vector ξ being properly extended, we need the convex hull of the support of this extended vector to be computationally tractable. In principle, this might be not the case already for "genuinely affine" decision rules; however, in typical applications distribution P of the "actual" uncertainty ξ is simple enough, so that $\mathrm{Conv}(\mathrm{supp} P)$ is computationally tractable. However, with P as simple as a uniform distribution on a box, the "quadratic extension" $\xi \mapsto (\xi, \{\xi_i \xi_j\}_{i,j})$ of ξ results in random vector with a distribution too complicated, as far as our needs are concerned. Thus, the limitations of affine decision rules are in fact limitations of our possibility to describe efficiently convex hulls of supports of nonlinear transformations of ξ.

Remark 4 One could bet that the idea of multi-stage decision making under uncertainty via linear decision rules is as old as the corresponding optimization model. It seems, however, that this idea remained completely forgotten for a long time; at least, we do not know who should be credited with it. Linear decision rules in optimization under uncertainty were recently "resurrected" in [7] in the framework of Robust Optimization. Our exposition follows the methodology developed in [7], with the only minor exception that in Robust Optimization one is aimed at minimizing the *worst-case* value of an uncertainty-affected objective under the restriction that a candidate solution remains feasible whatever be a realization of uncertainty-affected constraints, while here we intend to optimize, under the same restriction, the *expected* value of the objective.

Remark 5 We have assumed that (4.24) has a fixed recourse; the role of this assumption was to ensure affinity of the constraints in (4.26) in ξ, which in turn made it possible to use (!) in order to end up with tractable reformulation (4.28) of the problem of interest. In the case when the recourse is not fixed, that is, the matrices A_t, $t \geq 1$, in (4.24) depend affinely on ξ, the situation becomes much more complicated

[17)] In this respect, it is very instructive to look at Control, where the idea of linear feedback dominates theoretical research, and, to some extent, applications. Aside of a handful of simple particular cases, there are no reasons to believe that "the abilities" of linear feedback are as good as those of a general nonlinear feedback. However, Control community realized long ago that a bird in the hand is worth two in the bush – it is much better to restrict ourselves with something which we indeed can analyze and process numerically. We believe this is an instructive example for the optimization community.

– the left hand sides of the inequalities in (4.26) become quadratic in ξ, which makes (!) inapplicable[18]. It turns out, however, that under not too restrictive assumptions the problem of optimizing under the constraints (4.26), although NP-hard, admits tractable approximations of reasonable quality [7].

Remark 6 Passing from arbitrary decision rules to affine ones seems to reduce dramatically the flexibility of our decision-making and thus – the expected results. Note, however, that the numerical results for inventory management models reported in [7, 8] demonstrate that affinity may well be not as a severe restriction as one could expect it to be. In any case, we believe that when processing multi-stage problems, affine decision rules make a good and easy-to-implement starting point, and that it hardly makes sense to look for more sophisticated (and by far more computationally demanding) decision policies, unless there exists a clear indication of "severe non-optimality" of the affine rules.

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, Coherent measures of risk, *Mathematical Finance* 9 (1999), 203–228.

[2] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath and H. Ku, Coherent multiperiod risk measurement, Manuscript, ETH Zürich, 2003.

[3] B.R. Barmish and C.M. Lagoa, The uniform distribution: a rigorous justification for the use in robustness analysis, *Math. Control, Signals, Systems*, 10 (1997), 203-222.

[4] E.M.L. Beale, On minimizing a convex function subject to linear inequalities, *Journal of the Royal Statistical Society, Series B*, 17 (1955), 173–184.

[5] Ben-Tal, A., and Nemirovski, A., Robust Convex Optimization, *Mathematics of Operations Research*, 23 (1998).

[6] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, SIAM, Philadelphia, 2001.

[7] Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., "Adjustable Robust Solutions of Uncertain Linear Programs", *Mathematical Programming Ser A.*, 99 (2004), 351–376.

[8] Ben-Tal, A., Golany, B., Nemirovski, A., Vial J.-Ph. (2004), "Retailer-Supplier Flexible Commitments Contracts: A Robust Optimization Approach" – submitted to *Manufacturing & Service Operations Management*.

[18] In fact, in this case the semi-infinite system (4.26) can become NP-hard already with $\mathcal{Z}$ as simple as a box [7].

[9] G. Calafiore and M.C. Campi, Uncertain convex programs: Randomized solutions and confidence levels. To appear in *Math. Prog.*

[10] G. Calafiore and M.C. Campi, Decision making in an uncertain environment: the scenariobased optimization approach. Working paper, 2004.

[11] A. Charnes and W.W. Cooper, Uncertain convex programs: randomized solutions and confidence levels, *Management Science*, 6 (1959), 73-79.

[12] P. Dagum, L. Luby, M. Mihail, and U. Vazirani, Polytopes, Permanents, and Graphs with Large Factors, *Proc. 27th IEEE Symp. on Fondations of Comput. Sci. 1988*

[13] G. B. Dantzig, Linear programming under uncertainty, *Management Science*, 1 (1955), 197–206.

[14] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, Springer-Verlag, New York, NY, 1998.

[15] J. Dupačová, Minimax stochastic programs with nonseparable penalties, In *Optimization techniques (Proc. Ninth IFIP Conf., Warsaw, 1979), Part 1*, volume 22 of *Lecture Notes in Control and Information Sci.*, pages 157–163, Berlin, 1980. Springer.

[16] J. Dupačová, The minimax approach to stochastic programming and an illustrative application, *Stochastics*, 20 (1987), 73–88.

[17] M. Dyer and L. Stougie, Computational complexity of stochastic programming problems, SPOR-Report 2003-20. Dept. of Mathematics and Computer Sci., Eindhoven Technical Univ., Eindhoven, 2003.

[18] A. Eichhorn, W. Römisch. Polyhedral risk measures in stochastic programming, *SIAM J. Optimization*, to appear.

[19] Y. Ermoliev, A. Gaivoronski, and C. Nedeva, Stochastic optimization problems with partially known distribution functions, *SIAM Journal on Control and Optimization*, 23 (1985), 697–716.

[20] H. Föllmer and A. Schied, Convex measures of risk and trading constraints, *Finance and Stochastics* 6 (2002), 429–447.

[21] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello, The sample average approximation method for stochastic discrete optimization, *SIAM Journal of Optimization*, 12 (2001), 479–502.

[22] A. A. Gaivoronski, A numerical method for solving stochastic programming problems with moment constraints on a distribution function, *Annals of Operations Research*, 31 (1991), 347–370.

[23] M. Jerrum and U. Vazirani, A Mildly Exponential Approximation Algorithm for the Permanent, *Algorithmica*, 16 (1996), 392-”401.

[24] J. Linderoth, A. Shapiro, and S. Wright, The empirical behavior of sampling methods for stochastic programming, *Annals of Operations Research*, to appear.

[25] N. Linial, A. Samorodnitsky, and A. Wigderson, A deterministic strongly poilynomial algorithm for matrix scaling and approximate permanents, *Combinatorica*, 20 (2000), 531–544.

[26] W. K. Mak, D. P. Morton, and R. K. Wood, Monte Carlo bounding techniques for determining solution quality in stochastic programs, *Operations Research Letters*, 24 (1999), 47–56.

[27] H.M. Markowitz, “Portfolio selection,” *Journal of Finance*, 7 (1952), 77–91.

[28] *Moments in mathematics*, H.J. Landau (ed.), Proc. Sympos. Appl. Math., 37, Amer. Math. Soc., Providence, RI, 1987.

[29] A. Nemirovski, “On tractable approximations of randomly perturbed convex constraints” – *Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003*, 2419-2422.

[30] A. Nemirovski and A. Shapiro, Scenario Approximations of Chance Constraints, to appear in: Calafiore, G., Dabbene, F., Eds., *Probabilistic and Randomized Methods for Design under Uncertainty*, Kluwer Academic Press.

[31] A. Prékopa, *Stochastic Programming*, Kluwer, Dordrecht, Boston, 1995.

[32] F. Riedel, Dynamic coherent risk measures, Working Paper 03004, Department of Economics, Stanford University, 2003.

[33] R.T. Rockafellar, S. Uryasev and M. Zabarankin, Deviation measures in risk analysis and optimization, Research Report 2002-7, Department of Industrial and Systems Engineering, University of Florida.

[34] A. Ruszczyński and A. Shapiro, Optimization of convex risk functions, *E-print available at:* `http://www.optimization-online.org`, 2004.

[35] A. Ruszczyński and A. Shapiro, Conditional risk mappings, *E-print available at:* `http://www.optimization-online.org`, 2004.

[36] T. Santoso, S. Ahmed, M. Goetschalckx, A. Shapiro, A stochastic programming approach for supply chain network design under uncertainty, *European Journal of Operational Research*, to appear.

[37] A. Shapiro and T. Homem-de-Mello, On rate of convergence of Monte Carlo approximations of stochastic programs, *SIAM Journal on Optimization*, 11 (2000), 70–86.

[38] A. Shapiro and A. Kleywegt, Minimax analysis of stochastic programs, *Optimization Methods and Software*, 17(2002), 523–542.

[39] A. Shapiro, T. Homem de Mello, and J.C. Kim, Conditioning of stochastic programs, *Mathematical Programming*, 94 (2002), 1–19.

[40] A. Shapiro, Inference of statistical bounds for multistage stochastic programming problems, *Mathematical Methods of Operations Research*, 58 (2003), 57–68.

[41] A. Shapiro, Monte Carlo sampling methods. In: A. Rusczyński and A. Shapiro (editors), *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, North-Holland, 2003.

[42] A. Shapiro, Stochastic programming with equilibrium constraints, *E-print available at:* `http://www.optimization-online.org`, 2004.

[43] A. Shapiro, Worst-case distribution analysis of stochastic programs, *E-print available at:* `http://www.optimization-online.org`, 2004.

[44] A. Shapiro, On complexity of multistage stochastic programs, *E-print available at:* `http://www.optimization-online.org`, 2005.

[45] S. Takriti and S. Ahmed, On robust optimization of two-stage systems, *Mathematical Programming*, 99 (2004), 109-126.

[46] B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro, The sample average approximation method applied to stochastic routing problems: a computational study, *Computational Optimization and Applications*, 24 (2003), 289–333.

[47] L.G. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, 80 (1979), 189–201.

[48] J. Žáčková, "On minimax solutions of stochastic linear programming problems," *Čas. Pěst. Mat.*, 91 (1966), 423–430.