Available online at www.sciencedirect.com
ScienceDirect
Automatica 44 (2008) 1559–1568
automatica
www.elsevier.com/locate/automatica

Confidence level solutions for stochastic programming☆

Yu. Nesterov[a], J.-Ph. Vial[b,c,*]

[a] CORE, Catholic University of Louvain, 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium
[b] University of Geneva, Switzerland
[c] ORDECSYS, Place de l'Etrier, 4, CH-1224 Geneva, Switzerland

Received 28 November 2006; received in revised form 15 November 2007; accepted 24 January 2008

Abstract

We propose an alternative approach to stochastic programming based on Monte-Carlo sampling and stochastic gradient optimization. The procedure is by essence probabilistic and the computed solution is a random variable. We propose a solution concept in which the probability that the random algorithm produces a solution with an expected objective value departing from the optimal one by more than ϵ is small enough. We derive complexity bounds on the number of iterations of this process. We show that by repeating the basic process on independent samples, one can significantly reduce the number of iterations.

Keywords: Stochastic programming; Stochastic subgradient; Complexity estimate

1. Introduction

Optimization under uncertainty is an extension of deterministic optimization, when some parameters of the model are imperfectly known or determined by some stochastic process. Actually, optimization under uncertainty is the natural framework of many decision problems. The fields of global warming and energy planning, which currently draw much public attention, well illustrate this claim. Policies to fight against global warming are essentially geared towards drastic reduction of greenhouse gas emissions, and the schedule of these reductions is analyzed and determined by optimization models linking the economy and physical models of climate evolution (Bahn et al., 2006). Despite the sophistication of climate models, specialists increasingly emphasize the inherent uncertainty of climate evolution and advocate making uncertainty an intrinsic part of the models (IPCC, 2007). On a more national or regional basis, energy planning and power generation are also driven by stochastic phenomena: weather directly influences demand for electricity, the prices of oil and gas are volatile, water inflows in dams are random, etc. Clearly, there is a high demand for tools to handle uncertainty in optimization models.

With the help of decision theory and the calculus of probabilities, it is possible to give a perfectly consistent mathematical formulation of optimization problems under uncertainty. For static problems, the deterministic objective is replaced by its expectation (or an expected utility) and constraints are either deterministic or must be satisfied with some reasonable probability. In the dynamic case, the situation is more complex, because the formulation must account for a learning effect. In stochastic control, this effect is interpreted as a closed-loop control, which adjusts to the observed current state, as opposed to open-loop control, which is fixed once and for all (but is definitely suboptimal). In stochastic programming, one talks of decision variables adapted to past history or recourses. Optimizing over closed-loop controls or adapted recourses induces a complexity that is not present in the static case. Even though it is still possible to give a consistent mathematical formulation, the issue is then how to obtain usable solutions for the practitioners from the mathematical formulation. To-date, the field of stochastic optimization is far from bringing to users models and tools as efficient as in the deterministic case. There are many serious reasons for this state of affair. To start with, building a model in stochastic programming calls for

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Editor Alain Haurie.

* Corresponding author at: University of Geneva, Switzerland.
E-mail addresses: nesterov@core.ucl.ac.be (Yu. Nesterov), jphvial@bluewin.ch, jpvial@ordecsys.com (J.-Ph. Vial).

0005-1098/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2008.01.017

preliminary statistical analysis to identify the stochastic process and estimate parameters of probability distributions. Admitting that this step is successfully passed, one is then confronted with the complexity of evaluating expectations and probabilities, computations that involve multiple integrals, whose complexity dramatically increases with the number of random variables under consideration. Finally, optimization must be performed. The road to successful applications is indeed a long one.

To make those considerations more precise, in particular concerning the computational effort involved in solving the end-line optimization problem with a prescribed level of accuracy, one should resort to the theory of complexity for continuous[1] optimization, introduced by Nemirovsky and Yudin (1983) in the early 1980s. This theory has been at the core of most developments in deterministic optimization for the last 20 years, but not so much in stochastic programming. However, if one analyzes stochastic optimization from the view point of worse-case performance, it appears that there is a drastic difference between static and dynamic stochastic optimization problems. As early as 1969, Ermoliev (1969) proposed a stochastic gradient method to solve the static case and analyzed global convergence. The rate of convergence of the method for the static case was given in Nemirovsky and Yudin (1983) in terms of expected values of the objective function. In Shapiro and Nemirovski (2005) the authors study the sample average approximation (SAA) method to solve the static case and obtain a similar complexity bound. This same paper extends the result to the two-stage case. However, the sample size is the square of the static case, and for more stages the growth would be exponential with the number of stages. (See also Shapiro (2006).) This short discussion gives our motivation to study the static case. Let us mention that in the same issue, the paper (Thénié & Vial, 2008) proposes a method for the multi-stage case, but, in accordance with the above discussion, the method is presented as a heuristic approach and its validation is empirical.

In this paper, we focus on the static case and revisit the stochastic gradient scheme (SG). In essence, the SG algorithm is a stochastic process: its output, a value for the decision variable, is a random variable. The expected value of the objective function evaluated at this solution is also a random variable, and the classical result is that the expectation (over all possible repetitions of the stochastic gradient scheme) of this expectation is indeed close to the optimal expected value (the one associated with an optimal solution). In this paper we address the question of how good the solution produced by a *single* run of the stochastic gradient scheme might be. In other words, is it possible to bound the probability that the expected value associated with a particular solution of the gradient process departs from the optimal value by more than a certain threshold? We give a positive answer to this problem and compute a complexity bound to achieve a given value for this probability. By increasing the number of iterations of the stochastic gradient scheme, or alternatively said, by enlarging the sample size, the accuracy and the confidence level can be both improved. We also examine an alternative approach, in which several independent stochastic gradient schemes are performed and their solution outputs are averaged to form a final answer. Of course, less iterations are performed on each of these independent processes, and, as a result, the guaranteed accuracy and the confidence level on individual runs of the method are not as good, but the averaged solution is better. We examine the tradeoff between a single long run of SG process and many independent short runs pooled in an averaged solution and conclude that for the same total computational effort, the second approach yields better performance on the confidence level.

There exists an abundant literature on stochastic programming, in particular on methods based on sampling. Assuming that samples can be generated by some Monte-Carlo technique, one generates a sample of pre-specified size and computes a candidate solution with respect to that sample (Higle & Sen, 1991; Infanger, 1993; Mak, Morton, & Wood, 1999; Shapiro & Homem-de-Mello, 1998). Statistical estimation theory is then used to study the convergence of the candidate solution towards the optimal set and to construct confidence intervals (Dupačovà, 1988; Higle & Sen, 1992; King & Rockafellar, 1993; Mak et al., 1999). This literature provides limit theorems and ways of constructing confidence intervals. However, it remains silent on the computational effort that is required to reach a satisfactory solution. Since the approximation of continuous distributions is at the heart of practical implementations of stochastic programming, there exists a substantial literature on that topic. For extensive discussions, we refer to survey paper by Wets (1989) and the books of Birge and Louveaux (1997) and Higle and Sen (1996). We already mentioned Dupačovà (1988), Higle and Sen (1992), King and Rockafellar (1993) and Mak et al. (1999) for the analysis of statistical convergence of sampling methods for stochastic programs. Let us also mention the concept of importance sampling of Infanger (1993). The idea of working with random algorithms in stochastic programming is not new. The stochastic gradient algorithm goes back to Ermoliev (1969). More recently, Higle and Sen (1991, 1992) proposed an optimization scheme, in which scenarios are drawn at random sequentially. None of the quoted contributions includes complexity estimates.

The main results of this paper were released by the authors in Nesterov and Vial (2000) in the year 2000. At that time, the worst-case complexity analysis in Stochastic Optimization was not a common practice. The majority of the results were related to description of "average" efficiency of the computed solutions. However, the attention received by the recent publication by Shapiro and Nemirovski (2005), confirms that the community is ready to accept the new concepts. In Shapiro and Nemirovski (2005) the authors derive the worst-case lower complexity bounds for a traditional technique of Stochastic Optimization, based on preliminary generation of scenarios. From the viewpoint of computational complexity, their scheme complements our approach. Namely, for general convex nonsmooth black-box objective function, our approach

[1] In this paper we exclusively deal with continuous variables, not discrete ones.

is more efficient. At the same time, the standard technique can be better suited to the problems with explicit structures. We discuss their relations in detail in Section 6.

The paper is organized as follows. In Section 2, we give a formal definition of confidence levels for ϵ-optimal solution. In Section 3, we propose Monte-Carlo schemes to compute expectations. Assuming the existence of a random algorithm with known complexity, we resort to a standard probability result to estimate the probability in Definition 1. We introduce the expert pooling process and formulate the associated probability result. In Section 4, we discuss a stochastic gradient scheme for general convex function and analyze its complexity. The output of the method differs from the standard stochastic subgradient, since we take the average of the sequence of iterates and not the last iterate. In Section 5, we specialize the random algorithm to strongly convex functions. In the last section, we discuss possible tradeoff between random optimization on large samples with few experts, and smaller samples with many experts.

2. Confidence levels for ϵ-optimal solution

Let us first elaborate on the new solution concept. The stochastic gradient procedure is by essence probabilistic; hence, the computed solution is a random variable. If the decision-maker implements the computed solution, the objective function, which is an expectation, may sometimes fall short of the optimum. Therefore, the quality of the solution must be assessed in probabilistic terms. A reasonable requirement is that the probability of departure from the optimum be close enough to zero. With this solution concept we are able to derive complexity bounds on the number of iterations required to achieved a certain level of precision.

In order to make the above statement more precise, let Ξ be a probability space endowed with a probability measure, and let Q be subset of R^n. We are given a cost function $f : Q \times \Xi \mapsto R$. This mapping defines a family of random variables $f(x, \xi)$ on Ξ. We assume that the expectation in ξ, $E_\xi(f(x, \xi))$ is well defined for all $x \in Q$. Our problem of interest is the stochastic optimization problem:

$$\min_x \{\phi(x) \equiv E_\xi(f(x, \xi)) : \ x \in Q\}. \tag{1}$$

We always assume that problem (1) is solvable. Denote by x^* one of its solutions and $\phi^* = \phi(x^*)$. By V_ϕ we denote the variation of the function $\phi(x)$ on Q:

$$V_\phi = \max\{\phi(x) - \phi^* : \ x \in Q\}.$$

We assume that $V_\phi < \infty$.

In the rest of the paper, we shall make the following additional assumptions:

1. Q is bounded and closed convex set, and we can fix some point $x_0 \in Q$ such that $\|x - x_0\| \le R$ for all $x \in Q$.
2. The function $f(x, \xi)$ is convex in x for any $\xi \in \Xi$.
3. The variation of the function $f(x, \xi)$ in x and ξ is uniformly bounded.

Under the above assumptions, the function $\phi(x)$ is convex. Since exact minimization is usually not possible, we should replace problem (1) with

$$\text{Find } x \in Q : \quad \phi(x) - \phi^* \le \epsilon, \tag{2}$$

where ϵ is positive and small enough. We argue that (2) is not satisfactory. Indeed, computing the exact value of $\phi(x)$ may be very difficult, even when the distribution of ξ is known.[2] The standard deterministic notion of approximate solution to problem (1) becomes useless; clearly, we need an alternative definition, allowing comparisons of different solutions in a reasonable computational time.

In search of a suitable relaxation, we observe that no solution concept can exclude failures in actual implementation. This is the very nature of decision under uncertainty not to have full control over the consequences of a given decision. In this context, there is no reason to require that the computed solution be the result of some deterministic process. A random computational scheme would be equally acceptable, if the solution meets some probabilistic criterion. To fix ideas, let $\tilde{x}$ be the random solution of some random algorithm. Decision theory recommends that the solution satisfies

$$\text{Generate } \tilde{x} \in Q : \quad E_{\tilde{x}}(\phi(\tilde{x})) - \phi^* \le \epsilon. \tag{3}$$

Unfortunately, there is no much difference between (2) and (3), since both are defined with respect to expectations that cannot be computed exactly.

Our suggestion is to relax (3), and propose solutions that yield near optimal expected objective function values, with a probability close enough to one. In other terms, we substitute for the expectation over $\tilde{x}$ the weaker requirement of a suitable confidence level. The formal definition is as follows.

Definition 1. A random variable $\tilde{x}$ is called an (ϵ, β)-solution to problem (1), if

$$\text{Prob}\,(\phi(\tilde{x}) - \phi^* \ge \epsilon) \le 1 - \beta.$$

We call β the *confidence level* of the solution.

Note that solutions with confidence level $\beta = 1$ correspond to the deterministic solutions in the sense (2). A solution with β close enough to one may be acceptable to the decision-maker; but can it be computed? Indeed, the concept of (ϵ, β)-solutions also involves the expectation $\phi(\tilde{x})$ (there, $\tilde{x}$ is considered as deterministic) and we argued earlier that this quantity cannot be computed easily. Fortunately enough, this difficulty can be overcome. In the paper, we shall prove that the proposed alteration of the standard definition of optimality introduces enough flexibility to make our computational goal achievable.

[2] In the worst case, the computation of an approximation of $\phi(x)$ for a given x may already be very hard. Indeed, if $\Xi \subset R^m$, computing $\hat{\phi}$, such that $|\hat{\phi} - \phi(x)| \le \hat{\epsilon}$, may involve up to $O\left(\frac{1}{\hat{\epsilon}^m}\right)$ computations of $f(x, \xi)$ at different $\xi \in \Xi$. On the other hand, there is a large literature on quasi-Monte-Carlo methods, where it is shown that on some classes of functions the situation is much better (see Shapiro and Homem-de-Mello (2000)).

Finally, we discuss the opportunity of replacing an extensive random computational scheme with several parallel shorter implementations. We interpret this approach as putting several independent experts to work simultaneously. Each computes a solution via a random scheme and their output is averaged. Complexity-wise, this pooling process may be advantageous.

3. Two-level stochastic minimization

The solution procedure we propose involves several components. One consists in estimating the value of the objective function. The second component is the learning processes with different rules for generating the result. We present these components under separate subsections.

3.1. Cost evaluation

First of all, let us find out how we can get our hands on the values of the objective function of our problem. In what follows we assume that there is a known Monte-Carlo scheme to generate sequences of outcomes $\{\xi_t\}_{t=1}^T$ of independent random experiments. If the sample size T grows, we have, for any $x \in Q$ and $\epsilon > 0$,

$$\text{Prob}\left(\left|\frac{1}{T}\sum_{t=1}^{T} f(x,\xi_t) - \phi(x)\right| \geq \epsilon\right) \to 0.$$

From a computational point of view, the above statement is very weak: for multi-dimensional ξ the problem of approximating the value $\phi(x)$ at a particular x may turn out to be very hard indeed. However, with the assumption that the variation of $f(x,\xi)$ is bounded, we can use a stronger convergence result to devise a scheme for estimating $\phi(x)$. To this end, we resort to a standard theorem in probability theory (Hoeffding, 1963).

Theorem 1. *Let $Z_1, \ldots, Z_K$ be independent random variables with the same expectation $\mu > 0$ and* $\text{Prob}\,(0 \leq Z_j \leq V) = 1$, $j = 1, \ldots, K$. *Then, for*

$$\bar{Z}_K = \frac{Z_1 + \cdots + Z_K}{K}$$

we have

$$\text{Prob}\,(\bar{Z}_K \geq \mu + \epsilon) \leq \mathrm{e}^{-2K\epsilon^2/V^2}.$$

The above theorem can be used to estimate the complexity of the following scheme.

Cost prediction

1. Generate the sequence of random values ξ_t, $t = 1, \ldots T$.
2. Compute $\hat{\phi} = \frac{1}{T}\sum_{t=1}^{T} f(x,\xi_t)$.

(4)

Lemma 1. *Let us choose $T = \lceil \frac{V^2}{2\epsilon^2} \ln \frac{2}{1-\beta} \rceil$, where*

$$V = \max\{f(x_1,\xi_1) - f(x_2,\xi_2) : x_1, x_2 \in Q,\ \xi_1, \xi_2 \in \Xi\}.$$

Then, the outcome $\hat{\phi}$ of process (4) satisfies

$$\text{Prob}\left(|\hat{\phi} - \phi(x)| \geq \epsilon\right) \leq 1 - \beta.$$

Proof. Let $\underline{f} = \inf_{x,\xi} f(x,\xi)$. Define the random variables $Z(\xi) = f(x,\xi) - \underline{f}$ and $W(\xi) = V - Z(\xi)$. Clearly, $0 \leq Z(\xi) \leq V$ (resp. $0 \leq W(\xi) \leq V$) and $\mu = E(Z) = \phi(x) - \underline{f}$ (resp. $E(W) = \mu' = V - \mu$). Note that

$$\begin{aligned}\text{Prob}\,(|\bar{Z}_K - \mu| \geq \epsilon) &= \text{Prob}\,(\bar{Z}_K - \mu \geq \epsilon) + \text{Prob}\,(\bar{Z}_K - \mu \leq -\epsilon),\\ &= \text{Prob}\,(\bar{Z}_K - \mu \geq \hat{\epsilon}) + \text{Prob}\,(\bar{W}_K - \mu' \geq \epsilon).\end{aligned}$$

By applying Theorem 1 to Z and W, and using our choice of T, we get

$$\text{Prob}\left(|\hat{\phi} - \phi(x)| \geq \epsilon\right) \leq 2\mathrm{e}^{-2T\epsilon^2/V^2} = 1 - \beta. \quad \triangle$$

3.2. Learning processes

In the previous subsection we just showed that, at any x we can find an estimate for $\phi(x)$, which is valid with arbitrary a priori given probability. However, it is quite difficult to use this observation directly for devising a numerical scheme to solve problem (1).

Instead, we propose a stochastic optimization process to compute a strategy x, with the property that with a high probability its expected cost is a good approximation for the optimal value of the problem. It may appear that this objective will be be very difficult to achieve, because the computation of $\phi(x)$ for a fixed x is already an issue. Actually our process bypasses the computation of the value function; rather, it directly estimates a near optimal strategy. Then, the expected cost associated with this strategy is estimated in a final stage via Lemma 1.

The computation of a near-optimal strategy is done via two Monte-Carlo computational schemes. The first scheme is a strategy improvement procedure based on a random version of a deterministic minimization method. This scheme can be thought of as a random learning experiment performed by an expert. Based on the observation of successive outcomes, the expert adapts his/her strategy using the accumulated personal experience. The second scheme consists of pooling the information from a group of experts having performed similar experiments on independent random sequences.

3.2.1. The learning process of a single expert

Let us fix a priori the number of steps $N > 1$ and consider an arbitrary iterative deterministic process $\mathcal{M}$ for solving the following problem:

$$\min\{f(x) : x \in Q\}.$$

Such schemes usually generate a sequence of test points $\{x_k\}_{k=0}^N$. The next test point is formed on the basis of some information $\mathcal{I}_k$, received from the oracle at the previous test points. Denote by $\mathcal{O}(f(\cdot), x)$ the data, related to the function $f(\cdot)$, which is reported by the oracle at point x. Formally, the

scheme $\mathcal{M}$ with a given starting point x_0 can be written in the following way:

Deterministic method $\mathcal{M}$

0. **Initialization.** Set $\mathcal{I}_0 = \emptyset$.
1. **Iterations.**$(k = 0, \ldots, N-1)$
   Set $\mathcal{I}_{k+1} = \mathcal{I}_k \bigcup \mathcal{O}(f(\cdot), x_k)$.
   Compute x_{k+1} on the basis of $\mathcal{I}_{k+1}$.
2. Generate the output $\bar{x} = \mathcal{M}(x_0)$ on the basis of $\mathcal{I}_N$.

The random variant for the scheme $\mathcal{M}$ denoted by $\widetilde{\mathcal{M}}$ can be written as follows:

Random version of $\mathcal{M}$:

Individual learning process $\widetilde{\mathcal{M}}$

0. **Initialization.** Set $\mathcal{I}_0 = \emptyset$.
1. **Iterations.**$(k = 0, \ldots, N-1)$
   Generate randomly a single ξ_k in accordance with the distribution of ξ.
   Set $\mathcal{I}_{k+1} = \mathcal{I}_k \bigcup \mathcal{O}(f(\cdot, \xi_k), x_k)$.
   Compute x_{k+1} on the basis of $\mathcal{I}_{k+1}$.
2. Generate the output $\bar{x} = \widetilde{\mathcal{M}}(x_0)$ on the basis of $\mathcal{I}_N$.

(5)

The most important example of the oracle $\mathcal{O}$ is the first-order oracle, which computes at the point x_k an arbitrary subgradient in x of the function $f(x, \xi_k)$ with ξ_k being fixed.

Clearly, $\bar{x} = \widetilde{\mathcal{M}}(x_0)$ can be seen as a realization of some random variable. We will describe specific learning processes suitable for the problem at hand in Sections 3 and 4; for now, we will simply state the important common property of these processes. Namely, for the deterministic method $\mathcal{M}$ there exists a monotone function $\kappa_{\mathcal{M}}(N) \geq 0$ such that inequality

$$E(\phi(\bar{x})) - \phi^* \leq \kappa_{\mathcal{M}}(N) \tag{6}$$

holds and $\kappa_{\mathcal{M}}(N) \to 0$ as $N \to \infty$. The next theorem applies to method $\mathcal{M}$ satisfying (6).

Theorem 2. *Assume (6). Process (5) produces an $(\bar{\epsilon} V_\phi, \beta)$-solution when N satisfies*

$$\kappa_{\mathcal{M}}(N) \leq \bar{\epsilon}(1-\beta) V_\phi.$$

Proof. Let $Z = \phi(\bar{x}) - \phi^*$. Since $Z \geq 0$, using Chebyshev inequality for $T > 0$ we get

$$E(Z) \geq T \,\mathrm{Prob}\,(Z \geq T).$$

Letting $T = \bar{\epsilon} V_\phi$ and using (6) yield the result. △

3.2.2. Pooling information from the experts

The idea of the second stage process is to accumulate the outcomes of many decision-makers confronting different realizations of the stochastic process $\widetilde{\mathcal{M}}(x_0)$. This idea comes naturally from inequality (6) where we see that such average experience generates a strategy close enough to an optimal one. To this end, we need just to repeat several times the same process in order to see the different realization of $\widetilde{\mathcal{M}}(x_0)$.

Let us now formalize the above mentioned aggregation process. We fix the number of expert decision-makers $K \geq 1$.

Pooling experience process for $\widetilde{\mathcal{M}}(x_0)$

1. Compute the strategy $\bar{x}_j$ for the expert j as a realization of the learning process $\widetilde{\mathcal{M}}(x_0)$, $j = 1, \ldots, K$.
2. Compute the aggregate strategy $\hat{x} = \frac{1}{K} \sum_{j=1}^{K} \bar{x}_j$.

(7)

Lemma 2. *Assume (6). The quality of the outcome $\hat{x}$ of process (7) is described by the following inequality:*

$$\mathrm{Prob}\,\left(\phi(\hat{x}) - \phi^* \geq \kappa_{\mathcal{M}}(N) + \delta\right) \leq \mathrm{e}^{-2K\delta^2/V_\phi^2},$$

which is valid for any $\delta > 0$.

Proof. Since $\phi(\hat{x}) \leq \frac{1}{K} \sum_{j=1}^{K} \phi(\bar{x}_j)$, then

$$\begin{aligned}\mathrm{Prob}\,&\left(\phi(\hat{x}) - \phi^* \geq \kappa_{\mathcal{M}}(N) + \delta\right) \\ &\leq \mathrm{Prob}\,\left(\frac{1}{K}\sum_{j=1}^{K} \phi(\bar{x}_j) - \phi^* \geq \kappa_{\mathcal{M}}(N) + \delta\right).\end{aligned}$$

Consider the random variable $Z = \phi(\bar{x}) - \phi^*$. Then, from our assumptions we have that Z is bounded:

$$0 \leq Z \leq V_\phi.$$

Denote $\mu = E(\phi(\bar{x})) - \phi^*$. Clearly, $E(Z) = \mu$. Thus, from Theorem 1, we have from (6)

$$\begin{aligned}\mathrm{e}^{-2K\delta^2/V_\phi^2} &\geq \mathrm{Prob}\,\left(\bar{Z}_K \geq \mu + \delta\right) \\ &= \mathrm{Prob}\,\left(\bar{Z}_K \geq E(\phi(\bar{x})) - \phi^* + \delta\right) \\ &\geq \mathrm{Prob}\,\left(\frac{1}{K}\sum_{j=1}^{K} \phi(\bar{x}_j) - \phi^* \geq \kappa_{\mathcal{M}}(N) + \delta\right).\end{aligned}$$

△

In the following statement we give a complexity estimate for process (7) in the relative scale.

Theorem 3. *Assume (6) and let N and K be as follows:*

$$N = \left\lceil \kappa_{\mathcal{M}}^{-1}\left(\frac{1}{2}\bar{\epsilon} V_\phi\right)\right\rceil, \qquad K = \left\lceil \frac{2}{\bar{\epsilon}^2} \ln \frac{1}{1-\beta}\right\rceil. \tag{8}$$

Then, the point $\hat{x}$ generated by process (7) satisfies

$$\mathrm{Prob}\,\left(\phi(\hat{x}) - \phi^* \geq \bar{\epsilon}\, V_\phi\right) \leq 1 - \beta.$$

Moreover, the total number of computations of calls of oracle $\mathcal{O}(f(\cdot, \xi), x)$ does not exceed

$$M = K \cdot N \leq \left(1 + \kappa_{\mathcal{M}}^{-1}\left(\frac{1}{2}\bar{\epsilon} V_\phi\right)\right) \cdot \left(1 + \frac{2}{\bar{\epsilon}^2} \ln \frac{1}{1-\beta}\right).$$

Proof. Indeed, taking N and K as in (8) and $\delta = \frac{1}{2}\bar{\epsilon}\, V_\phi$, we immediately get the result from Lemma 2. △

Note that the above method does not provide information about the value $\phi(\hat{x})$. To estimate this value, we must use the cost prediction scheme (4). From Lemma 1 the required sample size is

$$T = \left\lceil \frac{1}{2\bar{\epsilon}^2} \left(\frac{V}{V_\phi} \right)^2 \ln \frac{2}{1-\beta} \right\rceil.$$

We can see, that the parameter T in process (4) is close to the parameter K in process (7) only if $V/(V_\phi)$ is close to one. The interesting feature of process (7) is that the number of experts K depends only on β and the relative accuracy parameter $\bar{\epsilon}$. We also see that it is much easier to increase the probability of a good answer rather than increase the quality of the answer.

4. Stochastic minimization for nonsmooth functions

In this section we consider problem (1) with nonsmooth functions $f(x, \xi)$. We assume that the Euclidean norm of stochastic subgradients $g \in \partial_x f(x, \xi)$ is uniformly bounded on $Q \times \Xi$ by some constant L.

Let us fix a priori the number of steps $N > 1$ in the learning scheme, and choose a finite sequence of positive steps $\{h_k\}_{k=0}^{N-1}$. Denote by $\pi_Q(x)$ the Euclidean projection of the point x onto Q. Consider the following deterministic subgradient scheme:

$$\mathcal{SG}: \begin{cases} x_{k+1} = \pi_Q(x_k - h_k g_k), \quad g_k \in \partial f(x_k), \\ \qquad k = 0, \ldots, N-1. \\ \bar{x} = \sum\limits_{k=0}^{N-1} h_k x_k \Big/ \sum\limits_{k=0}^{N-1} h_k. \end{cases} \tag{9}$$

In accordance with the results of Section 3, we need to show that the expected value of the objective function at point $\bar{x}$, generated by the random version $\widetilde{\mathcal{SG}}$ of method (9), is good enough, i.e., satisfies (6).

The following statement is well known (e.g. Nemirovsky and Yudin (1983)); we present its simple proof just for the reader's convenience.

Lemma 3. *The random strategy $\bar{x} = \widetilde{\mathcal{SG}}(x_0)$ satisfies the following relation:*

$$E(\phi(\bar{x})) - \phi^* \le \frac{R^2 + L^2 \sum\limits_{k=0}^{N-1} h_k^2}{2 \sum\limits_{k=0}^{N-1} h_k}. \tag{10}$$

Proof. Let x^* be an optimal solution ($\phi(x^*) = \phi^*$). Define $r_k = \|x^* - x_k\|$. Since x_{k+1} is the projection of $x_k - h_k g_k$ onto Q, it is closer to any point of Q. Thus

$$r_{k+1} \le \|x_k - h_k g_k - x^*\|.$$

Squaring and expanding the right-hand side, we get

$$r_{k+1}^2 \le r_k^2 - 2h_k \langle g_k, x_k - x^* \rangle + h_k^2 \|g_k\|^2. \tag{11}$$

Each point x_k, $1 \le k \le N$, in process (5) can be seen as a realization of a random variable. Thus, for k being fixed, the value $r_k^2 = \|x_k - x^*\|^2$ is also a realization of some random variable. Let us estimate its expectation. In view of (11), we have

$$\begin{aligned} E(r_{k+1}^2) &\le E(r_k^2) - 2h_k E(\langle g_k, x_k - x^* \rangle) + h_k^2 E(\|g_k\|^2), \\ &\le E(r_k^2) - 2h_k (E(f(x_k, \xi_k)) - \phi(x^*)) + h_k^2 L^2, \\ &= E(r_k^2) - 2h_k (E(\phi(x_k)) - \phi(x^*)) + h_k^2 L^2. \end{aligned}$$

In the second inequality we take the expectations on both sides of the convexity inequality $\langle g_k, x_k - x^* \rangle \ge f(x_k, \xi_k) - f(x^*, \xi_k)$. In the last inequality we replaced $E(f(x_k, \xi_k))$ by $E(\phi(x_k))$, since ξ_k is independent of x_k. Summing the above inequalities for $k = 0, \ldots, N-1$, and using the convexity of ϕ, we get:

$$\begin{aligned} r_0^2 + L^2 \sum_{k=0}^{N-1} h_k^2 &\ge r_N^2 + 2 \sum_{k=0}^{N-1} h_k (E(\phi(x_k)) - \phi(x^*)) \\ &\ge 2 \cdot \left(\sum_{k=0}^{N-1} h_k \right) \cdot (E(\phi(\bar{x})) - \phi^*). \end{aligned}$$

The result follows immediately. △

To ensure that the right-hand side of (10) goes to zero as $N \to \infty$, one has to make some assumptions on the step-sizes h_k. The standard assumption is

$$h_k > 0, h_k \to 0, \quad \text{and} \quad \sum_{k=0}^{N-1} h_k \to \infty \quad \text{as } N \to \infty.$$

Since the number of steps in process $\widetilde{\mathcal{SG}}$ is fixed, we can make the alternative choice of fixed step sizes of same length h. In order to simplify the presentation, let us assume that the constants R and L are known. Then the best choice of the step h is the one minimizing the right-hand side of (10), i.e.,

$$h = \frac{R}{L\sqrt{N}}.$$

Under this step-size strategy, we get from Lemma 3 the following simple bound.

Corollary 1. *Let us choose in $\mathcal{SG}$ $h_k = \frac{R}{L\sqrt{N}}$, $k = 0, \ldots, N-1$. Then, the expected value of the cost associated with*

$$\bar{x} = \widetilde{\mathcal{SG}}(x_0)$$

satisfies:

$$E(\phi(\bar{x})) - \phi^* \le \frac{LR}{\sqrt{N}} \equiv \kappa_{\mathcal{SG}}(N).$$

In the case when L and R are not known, we can take $h = \frac{\gamma}{\sqrt{N}}$ with some positive γ. In view of Lemma 3, this choice also ensures the right-hand side of inequality (10) to be of the order of $O(1/\sqrt{N})$. Note that in both cases with constant step-sizes h_k the rule for generating $\bar{x}$ becomes very simple:

$$\bar{x} = \frac{1}{N} \sum_{k=0}^{N-1} x_k. \tag{12}$$

Now we can give the complexity results for the single-expert process (5) and the aggregation process (7) based on the random method $\widetilde{SG}(x_0)$. Since $\kappa^{-1}_{\widetilde{SG}}(\tau) = (LR/\tau)^2$, the results stated in the corollary immediately follow from Theorems 2 and 3.

Theorem 4. *Let the parameters N and h_k be as follows:*

$$N = \left\lceil \frac{1}{\bar\epsilon^2} \cdot \frac{1}{(1-\beta)^2} \cdot \left(\frac{LR}{V_\phi}\right)^2 \right\rceil, \qquad h_k = \frac{R}{L\sqrt{N}}, \quad k = 0, \dots, N-1. \tag{13}$$

The point $\hat x$ generated by (5) from $\widetilde{SG}(x_0)$ satisfies the inequality

$$\text{Prob}\left(\phi(\hat x) - \phi^* \ge \bar\epsilon\, V_\phi\right) \le 1 - \beta.$$

Theorem 5. *Let the parameters N, K and h_k be as follows:*

$$N = \left\lceil \left(\frac{2}{\bar\epsilon}\right)^2 \cdot \left(\frac{LR}{V_\phi}\right)^2 \right\rceil, \qquad K = \left\lceil \frac{2}{\bar\epsilon^2} \ln \frac{1}{1-\beta} \right\rceil, \qquad h_k = \frac{R}{L\sqrt{N}}, \quad k = 0, \dots, N-1. \tag{14}$$

The point $\hat x$ generated by (7) from $\widetilde{SG}(x_0)$ satisfies the inequality

$$\text{Prob}\left(\phi(\hat x) - \phi^* \ge \bar\epsilon\, V_\phi\right) \le 1 - \beta.$$

The total number of computations of $g \in \partial_x f(x,\xi)$ does not exceed

$$M = K \cdot N \le \left(1 + \left(\frac{2}{\bar\epsilon}\right)^2 \cdot \left(\frac{LR}{V_\phi}\right)^2\right) \cdot \left(1 + \frac{2}{\bar\epsilon^2} \ln \frac{1}{1-\beta}\right).$$

Note that the choice of parameters, recommended by (14), leads to

$$h_k = \frac{\bar\epsilon\, V_\phi}{2L^2}, \quad k = 0, \dots, N-1.$$

5. Stochastic minimization for strongly convex functions

When in problem (1) the functions $f(x,\xi)$ have better properties than just being convex, we can expect that our problem becomes easier. Indeed, many properties of $f(x,\xi)$, like differentiability, type of smoothness, strong convexity, are inherited by $\phi(x)$. Unfortunately, since we are obliged to use for minimization the random versions of deterministic schemes, almost none of the above mentioned properties can really help. In this section we analyze the situation when the functions $f(x,\xi)$ are strongly convex.

Lemma 4. *Assume that the functions $f(\cdot,\xi)$ are uniformly strongly convex for every $\xi \in \Xi$, i.e.,*

$$f(y,\xi) \ge f(x,\xi) + \langle g, y - x\rangle + \frac{1}{2}\lambda\|y - x\|^2 \tag{15}$$

for all x, $y \in Q$, $g \in \partial_x f(x,\xi)$. Then the function $\phi(x)$ is also strongly convex.

The proof is straightforward.

Consider the following deterministic gradient scheme. Let us choose the step-size parameter $h > 0$ and $\sigma \in (0,1)$.

$$\mathcal{SG}_1 : \begin{cases} x_{k+1} = \pi_Q(x_k - h g_k), \quad g_k \in \partial_x f(x_k), \\ k = 0, \dots, N-1. \bar x = \dfrac{1-\sigma}{1-\sigma^N} \displaystyle\sum_{k=0}^{N-1} \sigma^{N-1-k} x_k. \end{cases} \tag{16}$$

Note, that the point $\bar x$ in this scheme can be updated recursively:

$$y_0 = x_0,$$
$$y_k = \sigma y_{k-1} + (1-\sigma)x_{k-1}, \quad k = 1, \dots, N-1,$$
$$\bar x = \frac{y_{N-1}}{1-\sigma^{N-1}}.$$

From (16) it is clear that this scheme becomes $\mathcal{SG}$ as $\sigma \to 1$.

We need to show that the expected value of the objective function at point $\bar x$, generated by the random version $\widetilde{\mathcal{SG}}_1$ of method (16), is close to ϕ^*.

Theorem 6. *Let us assume that the functions $f(x,\xi)$ are uniformly strongly convex in x with some constant $\lambda > 0$ and the Euclidean norm of subgradients of these functions is uniformly bounded on Q by some constant L.*

Let us choose h and σ in process $\widetilde{SG}_1$ as follows: $h = \frac{1}{\lambda}(1-\sigma)$, where σ is the solution of the equation

$$N\lambda^2 R^2 \sigma^N = L^2\left(1 - \sigma^N\right)^2, \quad \sigma \in (0,1).$$

Then the random variable $\bar x = \widetilde{\mathcal{SG}}_1(x_0)$ satisfies the following relation:

$$E(\phi(\bar x)) - \phi^* \le \frac{2L^2}{\lambda N} \ln\left(1 + \frac{\lambda R}{L}\sqrt{2N}\right). \tag{17}$$

Proof. Consider the random values $r_k^2 = \|x_k - x^*\|^2$. In view of (11) we have:

$$\begin{aligned} E(r_{k+1}^2) &\le E(r_k^2) - 2hE(\langle g_k, x_k - x^*\rangle) + h^2 E(\|g_k\|^2) \\ &\le E(r_k^2) - 2hE\left(f(x_k,\xi_k) - f(x^*,\xi_k) + \frac{\lambda}{2} r_k^2\right) \\ &\quad + h^2 L^2 \\ &= (1 - \lambda h)E(r_k^2) - 2h(E(\phi(x_k)) - \phi^*) + h^2 L^2. \end{aligned}$$

In view of the choice of the parameters in $\widetilde{SG}_1$, $\sigma \in (0,1)$. Summing up the above inequalities for $k = 0, \dots, N-1$, we get the following:

$$2h \sum_{k=0}^{N-1} \sigma^{N-1-k}(E(\phi(x_k)) - \phi^*) \le \sigma^N R^2 + h^2 L^2 \sum_{k=0}^{N-1} \sigma^{N-1-k}.$$

Since $\sum_{k=0}^{N-1} \sigma^{N-1-k} = \frac{1-\sigma^N}{1-\sigma}$, we obtain

$$\begin{aligned} E(\phi(\bar x)) - \phi^* &\le \frac{1-\sigma}{1-\sigma^N} \sum_{k=0}^{N-1} \sigma^{N-1-k}(E(\phi(x_k)) - \phi^*) \\ &\le \frac{R^2}{2h} \cdot \frac{(1-\sigma)\sigma^N}{1-\sigma^N} + \frac{h}{2} L^2 \end{aligned}$$

$$= \frac{1}{2}\left[\lambda R^2 \cdot \frac{\sigma^N}{1-\sigma^N} + hL^2\right]$$
$$= \frac{1}{2\lambda}\left[\lambda^2 R^2 \cdot \frac{\sigma^N}{1-\sigma^N} + L^2(1-\sigma)\right]$$
$$= \frac{L^2}{2\lambda}\left[\frac{1}{N}(1-\sigma^N) + (1-\sigma)\right]$$
$$\leq \frac{1}{\lambda}L^2 \cdot (1-\sigma).$$

It remains to estimate the value $1-\sigma$ from above.

Denote $\kappa = \frac{\lambda^2 R^2}{L^2}$. Let $x \in [0,1]$ be the solution of the quadratic equation:

$$N\kappa x = (1-x)^2.$$

(Thus, $x = \sigma^N$.) Then $(1-x)^2 \leq N\kappa x(2-x) = N\kappa(1-(1-x)^2)$. Therefore

$$1 - x \leq \left[\frac{N\kappa}{1+N\kappa}\right]^{1/2}.$$

Since we choose $\sigma^N = x$, we conclude that

$$1 - \sigma \leq 1 - \left(1 - \left[\frac{N\kappa}{1+N\kappa}\right]^{1/2}\right)^{1/N}.$$

Note that $1 + \alpha\ln(1-t) \leq \mathrm{e}^{\alpha\ln(1-t)}$. That is

$$1 - (1-t)^\alpha \leq -\alpha\ln(1-t), \quad t \in [0,1), \alpha \in R.$$

Using this inequality with $\alpha = \frac{1}{N}$ and $t = \left[\frac{N\kappa}{1+N\kappa}\right]^{1/2}$, we conclude that

$$1 - \sigma \leq -\frac{1}{N}\ln\left(1 - \left[\frac{N\kappa}{1+N\kappa}\right]^{1/2}\right)$$
$$= \frac{1}{N}\ln\left([1+N\kappa]^{1/2}([1+N\kappa]^{1/2} + [N\kappa]^{1/2})\right)$$
$$\leq \frac{2}{N}\ln\left(1+\sqrt{2N\kappa}\right). \quad \triangle$$

Note that the right-hand side of the inequality (17) tends to $\frac{4LR}{\sqrt{2N}}$ as $\lambda \to 0$.

We can now apply Theorem 3. We see that in the case of λ large enough, we can significantly reduce the length of the learning process.

6. Conclusion

Stochastic programming involves the computation of expected values. In the case of general distributions, the expectations cannot be computed exactly, and δ-approximation in an m-dimensional space may require $O(1/\delta^m)$ operations. Consequently, one cannot expect good complexity estimates for general stochastic programming.

To overcome this difficulty, we introduced the new concept of (ϵ, β)-solution with the property that their associated expected objective value is $\bar{\epsilon}$-close (in relative terms) to the optimal value with a probability, or confidence level, at least equal to $(1-\beta)$. We showed that an (ϵ, β)-solution can be constructed by means of Monte-Carlo sampling. We propose a mechanism which simulates a learning process of a so-called expert. To improve the confidence level, we suggest to repeat the learning process on independent samples. The outcomes of the individual experts are averaged to form the (ϵ, β)-solution. We called this process "pooling the experts".

Table 1
Complexity estimates with subgradient optimization

	Single expert	Pool of experts
Number of experts K	1	$\frac{2}{\epsilon^2}\ln\frac{1}{1-\beta}$
Length of process N	$\frac{1}{\bar{\epsilon}^2(1-\beta)^2}\left(\frac{LR}{V_\phi}\right)^2$	$\frac{4}{\epsilon^2}\left(\frac{LR}{V_\phi}\right)^2$
Computational effort M	$\frac{1}{\bar{\epsilon}^2(1-\beta)^2}\left(\frac{LR}{V_\phi}\right)^2$	$\frac{8}{\epsilon^4}\ln\frac{1}{1-\beta}\left(\frac{LR}{V_\phi}\right)^2$

The complexity estimates are the length N of the individual learning process, the number K of experts and the total number $M = K \cdot N$ of calls to the oracle. Using a stochastic gradient process we obtain values that are reported on Table 1. (Since we are interested only in the order of magnitude of M, K and N, we put in the table their real approximations.)

We note first that the complexity estimates are independent of the dimension of the underlying space. In view of the effort to compute m-dimensional integrals, our procedure is computationally reasonable. We also note that the number of experts is independent of the problem instance. Since the learning processes can be fully distributed on parallel processors, the pooling scheme appears to be attractive.

Table 1 indicates that there is a tradeoff between the precision and the confidence level of the solution. If we do not need too high confidence level, it advisable to rely on the experience of a single expert. However, when a high level of confidence is required the pool becomes particularly efficient. Indeed, dividing $(1-\beta)$ by a fixed number, say 10, requires adding a constant number of experts, while the single expert must extend his/her learning process by the multiplicative factor 100. This suggests that averaging the experience of a large population of young people improves reliability much more than letting a single expert refine his/her experience. Note that we do not need too smart experts: their confidence level is only one-half. We leave it to the reader to give any generality to this assertion.

The learning scheme is based on stochastic subgradient optimization. The scheme applies to general convex functions $f(\cdot, \xi)$. In the stochastic scheme one cannot exploit special properties such as differentiability. However, strong convexity helps in reducing the length of the learning process: $O(1/\epsilon)$ instead of $O(1/\epsilon^2)$.

Table 1 reveals that the good choice of the constants L, R and V_ϕ is important. A wrong choice can significantly increase the length of the learning process. Nevertheless, we can hope that there are many situations, in which the described process can be very useful. Let us further note that a high degree of accuracy is often not of a real value for decision-makers. Improving moderately upon the average may be quite satisfactory. Instead, putting emphasis on the confidence level of the chosen solution is more fitted to the needs of the decision-

makers. It seems that this aspect is generally neglected in the practice of stochastic optimization.

Indeed, in many practical applications the desired level of optimality is not very high.

Finally, let us compare our results with the complexity bounds presented in Shapiro and Nemirovski (2005). For the reader's convenience, we sketch the main developments and results of the paper. The authors follow the standard two-step approach. First, they draw a large number of samples $\{\xi_i\}_{i=1}^N$ and construct the deterministic objective function

$$\tilde{\phi}(x) = \frac{1}{N}\sum_{i=1}^{N} f(x, \xi_i).$$

The second step consists in minimizing this function by an appropriate numerical scheme:

$$\min_x \{\tilde{\phi}(x) : \ x \in Q\}. \tag{18}$$

Note that the computation of the value and the gradient of function $\tilde{\phi}$ needs N calls of oracle for function f.

The main result of Shapiro and Nemirovski (2005) is the lower bound on the sample size N, which is sufficient to guarantee that the probability of the event

$$\frac{1}{2}\epsilon\text{-solution of problem (18) gives an } \epsilon\text{-solution of problem (1)}$$

is higher than β. In our notation, this bound looks as follows (see Theorem 2, Shapiro and Nemirovski (2005)):

$$N \geq O\left(\left(\frac{LR}{\epsilon}\right)^2 \left[n \ln \frac{LR}{\epsilon} + \ln \frac{1}{1-\beta}\right]\right), \tag{19}$$

where n is the dimension of the decision vector x. Let us compare this bound with the estimates given in Table 1 (recall that we measure the accuracy of solution in the scale relative to the variation V_ϕ).

It is clear that for Shapiro and Nemirovski (2005), the main computational efforts are related to the solution process of problem (18). Depending on the structure of function $f(x, \xi)$ and dimension n, these efforts can be quite different. The most favorable situation corresponds to a smooth strongly convex f. In this case, problem (18) can be solved by $O(\ln \frac{1}{\epsilon})$ iterations of the gradient-type schemes. Hence, the total efforts will be of the order $O(N \ln \frac{1}{\epsilon}) = O(\frac{n}{\epsilon^2} \ln^2 \frac{1}{\epsilon})$ calls of f-oracle.[3] However, if the function f is convex and nonsmooth, the optimization schemes need $O(\frac{1}{\epsilon^2})$ iterations. In this case, the total computational efforts can reach the level

$$O\left(\frac{n}{\epsilon^4} \ln \frac{1}{\epsilon}\right)$$

calls of oracle, which is worse (for high dimension) than the bound guaranteed by Table 1. Note also, that our approach does not need any significant storage resources, necessary for maintaining all generated scenarios.

[3] In our analysis we discuss the dependence of the complexity in accuracy ϵ and dimension of the space n only.

Thus, we conclude that our approach complements the computational strategies described in Shapiro and Nemirovski (2005).

Acknowledgements

The authors are grateful to the referees for their valuable comments.

References

Bahn, O., Drouet, L., Edwards, N., Haurie, A., Knutti, R., Kypreos, S., et al. (2006). The coupling of optimal economic growth and climate models. In H. Wanner, M. Grosjean, R. Röthlisberger, & E. Xoplak (Eds.), *Climate variability, predictability and climate risks: A European perspective*: *Vol. 79* (pp. 103–119). Climatic Change.

Birge, J., & Louveaux, F. (1997). *Springer series in operations research. Introduction to stochastic programming*. New York: Springer-Verlag.

Dupačovà, J., & Wets, R. J.-B. (1988). Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems. *Annals of Statistics*, *16*, 1517–1549.

Ermoliev, Y. (1969). About generalized stochastic gradient methods and stochastic quasi-Fejer sequences. In *Kibernetica*: *v.2* (in Russian).

Higle, J. L., & Sen, S. (1991). Stochastic decomposition: An algorithm for two stage linear programs with recourse. *Mathematics of Operations Research*, *16*, 650–699.

Higle, J. L., & Sen, S. (1992). On the convergence of algorithms with implications for stochastic and nondifferentiable optimization. *Mathematics of Operations Research*, *17*, 112–131.

Higle, J. L., & Sen, S. (1996). *Stochastic decomposition. A Statistical method for large scale stochastic linear programming*. Dordrecht: Kluwer Academic Publishers.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*, 13–30.

Infanger, G. (1993). Monte-Carlo (importance) sampling within a Benders' decomposition for stochastic linear programs. *Annals of Operations Research*, *39*, 69–95.

IPCC. (2007). Climate change 2007: The physical science basis Draft report IPCC Secretariat, 7bis, Avenue de la Paix, Geneva, Switzerland.

King, A. J., & Rockafellar, R. T. (1993). Asymptotic theory for solutions in statistical estimation and stochastic optimization. *Mathematics of Operations Research*, *18*, 148–162.

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

Nemirovsky, A., & Yudin, D. (1983). *Problem complexity and method efficiency in optimization*. Wiley.

Nesterov, Yu., & Vial, J.-Ph. (2000). Confidence level solutions for stochastic programming, *CORE discussion paper 2000/13*.

Shapiro, A., & Homem-de-Mello, T. (1998). A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, *81*, 301–325.

Shapiro, A., & Homem-de-Mello, T. (2000). On rate of convergence of Monte Carlo approximations of stochastic programs. *SIOPT*, *11*, 70–86.

Shapiro, A. (2006). On complexity of multistage stochastic programs. *Operations Research Letters*, *34*, 1–8.

Shapiro, A., & Nemirovski, A. (2005). On complexity of stochastic programming problems. In V. Jeyakumar, & A. M. Rubinov (Eds.), *Continuous optimization: Current trends and applications* (pp. 111–144). Springer.

Thénié, J., & Vial, J.-Ph. (2008). Step decision rules for multi-stage stochastic programming: A heuristic approach, in this issue (doi:10.1016/j.automatica.2008.02.001).

Wets, R. J.-B. (1989). In G. L. Nemhauser, A. H. G. Rinnoy Kan, & M. J. Todd (Eds.), *Optimization*, *Stochastic programming*. Amsterdam: North-Holland.

Yu. Nesterov Born: 1956, Moscow.

Master degree: 1977, Moscow State University, Faculty of Computational Mathematics and Cybernetics.

Doctor degree: 1984, Institute of Control Sciences, Acad. Sci. USSR, Moscow.

Professional experience:

1977–1992: Different research positions at Central Economical and Mathematical Institute, Acad. Sci. USSR, Moscow.

1993–present: Professor at Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), Louvain-la-Neuve, Belgium.

Author of 4 monographs and more than 70 refereed papers in the leading optimization journals. Winner of the triennial Dantzig Prize 2000 awarded by SIAM and Mathematical Programming Society for a research having a major impact on the field of mathematical programming.

J.-Ph. Vial is Emeritus Professor at the University of Geneva. He currently co-chairs ORDECSYS, a consulting company in Operations Research and Systems. Jean-Philippe Vial is a specialist in Logistics, Operations Research and Algorithmic. He has been active in developing new techniques for solving large-scale optimization problems, such as those arising in multi-regional planning and in the dealing of uncertainty. He has contributed new numerical methods for convex optimization and has developed specialized software in this area. He has strong interest in applications of logistics, in particular in the area of environmental management. Jean-Philippe Vial has published a book and over 80 refereed papers in leading journals. He is a former President of the Mathematical Programming Society.