Confidence level solutions for stochastic programming

Yu. Nesterov*, J.-Ph.Vial†

January, 2000

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. The associated objective value is doubly random, since it depends on two outcomes: the event in the stochastic program and the randomized algorithm. We propose a solution concept in which the probability that the randomized 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 for this process. We show that by repeating the basic process on independent sample, one can significantly sharpen the complexity bounds.

Keywords: Stochastic programming, Stochastic subgradient, Complexity estimate.

1 Introduction

The handling general probability distributions in stochastic programming is a delicate issue. Indeed, one cannot, in general, compute exact expectations, and thus one cannot make direct use of standard optimization techniques. Therefore, the first step is often to construct finite distributions, close enough to the original one. The initial problem is then replaced by a computationally tractable approximation. This approach raises issues on the quality of the computed solutions and on the computational effort to achieve a given level of optimality with respect to the original problem.

Sampling offers an alternative to deal with continuous distributions. Assuming that samples can be generated by some Monte-Carlo technique, one generates a sample of given size and compute a candidate solution with respect to that sample [4, 5, 9, 12, 14]. Statistical estimation theory is then used to study the convergence of

*CORE, Catholic University of Louvain, 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium; e-mail: nesterov@core.ucl.ac.be

†HEC, Department of Management Studies, University of Geneva, 40 Bd du Pont d'Arve, CH-1211 Geneva 4, Switzerland; e-mail: jean-philippe.vial@hec.unige.ch

the candidate solution towards the optimal set and to construct confidence intervals [6, 3, 10, 12, 13]. 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.

The aim of this paper is to bridge the present gap. To this end, we propose an approach based on Monte-Carlo sampling and stochastic gradient optimization. We introduce a new solution concept; ϵ-optimal solutions with a certain confidence level. We propose a stochastic gradient algorithm [4] to compute a solution to this problem, and we give a complexity bound on the number of iterations.

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 sometime 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 the 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 $\| x - x_0 \| \leq R$ for all $x \in Q$.
2. The function $f(x, \xi)$ is convex in x for any $\xi \in \Xi$.

(In the sequel, it might be easier to think of ξ as the outcome of a random variable $\tilde{\xi}$, but this is not necessary.)

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

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

where ϵ is positive and small enough. We argue that (2) is not satisfactory. Indeed, computing the exact value of $\phi(x)$ may not be numerically possible, even when

the distribution of ξ is known.[1] 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 randomized 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 randomized algorithm. Decision theory recommends that the solution satisfies

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

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 to 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 the problem (1), if*

$$\text{Prob}\left(\phi(\tilde{x}) - \phi^* \geq \epsilon\right) \leq 1 - \beta.$$

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.

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

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 [16] and the books of Birge and Louveaux [1] and Higle and Sen [7]. We already mentioned

[1] Complexity-wise, 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 $\mid \hat{\phi} - \phi(x) \mid \leq \hat{\epsilon}$, may involve up to $O\left(\frac{1}{\hat{\epsilon}^m}\right)$ computations of $f(x, \xi)$ at different $\xi \in \Xi$.

[6, 3, 10, 12, 13] for the analysis of statistical convergence of sampling methods for stochastic programs. Let us also mention the concept of importance sampling of Infanger [9]. The idea of working with randomized algorithms in stochastic programming is not new. The stochastic gradient algorithm goes back to Ermoliev [4]. More recently, Higle and Sen [5, 6] proposed an optimization scheme, in which scenarios are drawn at random sequentially. None of the quoted contributions includes complexity estimates.

The paper is organized as follows. In Section 2, we propose Monte-Carlo schemes to compute expectations. Assuming the existence of a randomized algorithm with known complexity, we resort to a standard probability result to estimate the probability in (1). We introduce the expert pooling process and formulate the associated probability result. In Section 3, 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 4, we specialize the randomized algorithm to strongly convex functions. In the last section, we discuss possible tradeoff between randomized optimization on large samples with few experts, and smaller samples with many experts.

2 Two-stage stochastic minimization

The solution procedure we propose involves several phases. We present them under separate subsections.

2.1 Cost evaluation

First of all, let us check what we can do with 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 $\hat{\epsilon} > 0$,

$$\mathrm{pr}\left(\mid \frac{1}{T} \sum_{t=1}^{T} f(x, \xi_t) - \phi(x) \mid \geq \hat{\epsilon} \right) \to 0.$$

From a computational point of view, the above statement is very weak: for multidimensional ξ the problem of approximating the value $\phi(x)$ at a particular x may turn out to be very hard indeed. However, when $f(x, \xi)$ is uniformly bounded in ξ, we can use a stronger convergence result to devise a fully polynomial time scheme for estimating $\phi(x)$. To this end, we resort to a standard theorem in probability theory [8].

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

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

we have

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

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

Cost production

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* $T = \frac{V^2}{2\hat{\epsilon}^2} \ln \frac{2}{1-\beta}$, *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*

$$\Pr\left(\mid \hat{\phi} - \phi(x) \mid \geq \hat{\epsilon}\right) \leq 1 - \beta.$$

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

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

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

$$\mathrm{pr}\left(\mid \hat{\phi} - \phi(x) \mid \geq \bar{\epsilon}\right) \leq 2e^{-2T\bar{\epsilon}^2/(LR)^2} = 1 - \beta.$$

$\triangle$

2.2 Stochastic optimization

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, which solves the problem (1).

Instead, we propose a stochastic optimization process to compute a strategy x, with the property that its cost is a good approximation for the optimal value of the problem with a high probability. It may look as if this objective should be very difficult to achieve, since 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 randomized version of a deterministic minimization scheme. 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 his personal experience. The second scheme consists of pooling the information from a group of experts having performed similar experiments on independent random sequences.

2.2.1 The learning process of an 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 $\mathcal{O}(f(\cdot), x)$ 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 seen in the following way:

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 randomized variant for the scheme $\mathcal{M}$ can be written as follows:

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)

Clearly, $\bar{x} = \widetilde{\mathcal{M}}(x_0)$ can be seen as a realization of some random variable. Later on we will see that for a randomized variant of deterministic minimization scheme we can prove the following inequality:

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

where $\kappa_{\mathcal{M}}(N) \to 0$ as $N \to \infty$. Other reasonable deterministic schemes may possibly satisfy (6). The next theorem applies to them.

Theorem 2 *Assume (6). Process (5) produces an (ϵ, β)-solution when N satisfies*

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

Proof:
Let $\psi(\bar{x}) = E(\phi(\bar{x})) - \phi^*$. Since $\psi(\bar{x}) \geq 0$, then for $T > 0$

$$E_{\bar{x}}[\psi(\bar{x})] \geq T\Pr(\phi(\bar{x}) \geq T).$$

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

2.2.2 Pooling information from the experts

The idea of the second stage process is to accumulate the outcomes of many decision-makers confronted to different realizations of the stochastic process $\widetilde{\mathcal{M}}(x_0)$. This idea naturally comes from the inequality (6) where we see that such average experience gives us 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 formalize now 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:*

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

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

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

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 and Corollary 1, we have from (6)

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

△

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

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

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

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

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

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

$$M = \frac{2\kappa_{\mathcal{M}}^{-1}(\frac{1}{2}\bar{\epsilon}V_\phi)}{\bar{\epsilon}^2}\ln\frac{1}{1-\beta}.$$

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

Thus, $\hat{x}$, the output of the algorithm (7), can be generated in fully polynomial time. However, this process does not yield 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

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

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 the process (7) is that the number of experts K depends only on the accuracy parameters $\bar{\epsilon}$ and β. We see also that it is much easier to increase the probability of a good answer rather than increase the quality of the answer.

3 Stochastic minimization for nonsmooth functions

In this section we consider the 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 step sizes $\{h_k\}_{k=0}^{N-1}$. Denote by $\pi_Q(x)$ the Euclidean projection of the point x onto Q. Consider the following deterministic subgradient scheme:

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

In accordance to the results of Section 2, we need to show that the expected value of the objective function at point $\bar{x}$, generated by the randomized version $\widetilde{\mathcal{SG}}$ of method (9), is good enough.

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

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

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

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

Expanding the right-hand side, we get

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

Each point x_k, $1 \leq k \leq N$, in the 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

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

In the second inequality we take the expectations on both sides of the convexity inequality $\langle g_k, x_k - x^* \rangle \geq f(x_k, \xi_k) - f(x^*, \xi_k)$. 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:

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

The result follows immediately. $\triangle$

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

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

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 as follows:

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

Under that step-size strategy, we have the following simple bound.

Corollary 1 *The expected value of the cost associated with*

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

satisfies:

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

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 aggregation process (7) as applied to the process $\widetilde{SG}(x_0)$.

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

$$\begin{aligned} &N = \left(\tfrac{2}{\bar{\epsilon}}\right)^2 \cdot \left(\tfrac{LR}{V_\phi}\right)^2, \quad K = \tfrac{2}{\bar{\epsilon}^2} \ln \tfrac{1}{1-\beta}, \\ &h_k = \tfrac{R}{L\sqrt{N}}, \ k = 0, \ldots, N-1. \end{aligned} \tag{13}$$

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

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

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

$$M = \tfrac{1}{2}\left(\frac{2}{\bar{\epsilon}}\right)^4 \cdot \left(\frac{LR}{V_\phi}\right)^2 \ln \frac{1}{1-\beta}.$$

Proof:
Since $\kappa^{-1}_{\mathcal{SG}}(\tau) = (LR/\tau)^2$, the result immediately follows from Theorem 3. $\triangle$

Note that the choice of parameters, recommended by (13), leads to

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

4 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 randomized 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 *Let the functions $f(x, \xi)$ are uniformly strongly convex:*

$$f(y, \xi) \geq f(x, \xi) + \langle g, y - x \rangle + \tfrac{1}{2}\lambda \parallel y - x \parallel^2$$

for any x, $y \in Q$, $g \in \partial_x f(x, \xi)$ and $\xi \in \Xi$. 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 $\sigma \in (0, 1)$.

$$\mathcal{SG}_1 : \left\{ \begin{array}{rcl} x_{k+1} & = & \pi_Q(x_k - h g_k),\ g_k \in \partial_x f(x_k),\ k = 0, \ldots, N-1. \\ \bar{x} & = & \frac{1-\sigma}{1-\sigma^N} \sum\limits_{k=0}^{N-1} \sigma^{N-1-k} x_k. \end{array} \right. \tag{14}$$

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

$$\begin{aligned} \bar{x}_0 &= x_0, \quad \gamma = \tfrac{1-\sigma}{1-\sigma^N}, \\ \bar{x}_k &= \gamma x_k + (1-\gamma)\bar{x}_{k-1},\ k = 1, \ldots, N-1, \\ \bar{x} &= \bar{x}_{N-1}. \end{aligned}$$

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 randomized version $\widetilde{\mathcal{SG}}_1$ of method (14), is close to ϕ^*.

Lemma 5 *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 in the process $\widetilde{SG}_1$

$$h = \frac{2}{\lambda N} \ln\left(1 + \frac{\lambda R}{2L}\sqrt{N}\right), \quad \sigma = 1 - \lambda h.$$

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

$$E(\phi(\bar{x})) - \phi^* \leq \tfrac{2L^2}{\lambda N} \ln\left(1 + \tfrac{\lambda R}{2L}\sqrt{N}\right). \tag{15}$$

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

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

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

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

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

$$\begin{aligned} E(\phi(\bar{x})) - \phi^* &\leq \tfrac{1-\sigma}{1-\sigma^N} \sum_{k=0}^{N-1} \sigma^{N-1-k} (E(\phi(x_k)) - \phi^*) \\ &\leq \tfrac{R^2}{2h} \cdot \tfrac{(1-\sigma)\sigma^N}{1-\sigma^N} + \tfrac{h}{2} L^2 = \tfrac{1}{2}\left[\lambda R^2 \cdot \tfrac{\sigma^N}{1-\sigma^N} + hL^2\right] \\ &\leq \tfrac{1}{2}\left[\tfrac{\lambda R^2}{e^{\lambda h N} - 1} + hL^2\right] = \tfrac{1}{2}\left[\tfrac{\lambda R^2}{\left(1 + \frac{\lambda R}{2L}\sqrt{N}\right)^2 - 1} + hL^2\right] \\ &\leq hL^2 = \tfrac{2L^2}{\lambda N} \ln\left(1 + \tfrac{\lambda R}{2L}\sqrt{N}\right). \end{aligned}$$

(In the next to last step we used inequality $\frac{2\xi^2}{(1+\xi)^2 - 1} \leq \ln(1+\xi)$ with $\xi = \frac{\lambda R}{2L}\sqrt{N}$.)
$\triangle$

Note that the right-hand side of the inequality (15) tends to $\frac{LR}{\sqrt{N}}$ 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:

$$N \approx \frac{1}{\bar{\epsilon}} \ln \frac{1}{\bar{\epsilon}}.$$

5 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 ϵ-close (in relative terms) to the optimal value with a probability, or confidence level, as 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".

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.

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

Table 1: Complexity estimates with subgradient optimization

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 the learning processes can be fully distributed on parallel processors, the pooling scheme appears to be attractive.

The comparison between solution with a single expert and a pool of experts reveals that the pool is particularly efficient when a high level of confidence is required. 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 and 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 $O(1/\epsilon^2)$.

A natural area of application for our method is, as in [5], the class of two-stage linear programming with recourse [2, 15]. For problem of this class one can construct elements of the subgradient to $f(x, \xi)$. Under suitable assumptions, the feasible set may be bounded by some constant R the Lipschitz constant L and the variation of ϕ can possibly be estimated from the special structure of the recourse problem. Then the length of the learning process and the number of experts could be determined a priori to fit the objective of the user.

References

[1] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.

[2] G.B. Dantzig. "Linear programming under uncertainty". *Management Science* 1 (1995), 197–206.

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

[4] Y. Ermoliev. Stochastic quasi-gradient methods, in *Numerical Techniques for Stochastic Optimization* Y. Ermoliev and R. J.-B. Wets (eds.), Springer Verlag, Berlin, 1988.

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

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

[7] J.L. Higle and S. Sen. *Stochastic Decomposition. A Statistical method for Large Scale Stochastic Linear Programming*, Kluwer Academic Publishers, Doordrecht, 1996.

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

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

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

[11] A.J. King and R.J.-B. Wets. Epi-consistency of convex stochastic programs, *Stochastic and Statistic Reports*, 34 (1989), 83–92.

[12] 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.

[13] A. Shapiro. "Asymptotic properties of statistical estimators in stochastic programming". *Annals of Statistics*, 17 (1989), 841–858.

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

[15] R. Van Slyke and R.J.-B. Wets. "L-shaped linear programs with applications to optimal control and stochastic programming". *SIAM Journal of Applied Mathematics*, 17 (1969), 638–661.

[16] R.J.-B. Wets. Stochastic Programming, in *Optimization* G.L. Nemhauser, A.H. G. Rinnoy Kan and M.J. Todd (eds.), North-Holland, Amsterdam, 1989.