Operations Research Letters 41 (2013) 622–626

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

On the information-based complexity of stochastic programming

Gabriela Tavares *, Panos Parpas

Department of Computing, Imperial College London, South Kensington Campus, London, SW7 2AZ, United Kingdom

ARTICLE INFO

Article history:
Received 7 January 2013
Received in revised form
20 August 2013
Accepted 21 August 2013
Available online 27 August 2013

Keywords:
Optimization
Stochastic programming
Uncertainty
Complexity
Information-based complexity

ABSTRACT

Existing complexity results in stochastic linear programming using the Turing model depend only on problem dimensionality. We apply techniques from the information-based complexity literature to show that the smoothness of the recourse function is just as important. We derive approximation error bounds for the recourse function of two-stage stochastic linear programs and show that their worst case is exponential and depends on the solution tolerance, the dimensionality of the uncertain parameters and the smoothness of the recourse function.

1. Introduction

The computational complexity of a problem is defined as the amount of resources required to approximate its solution with an error below a specified tolerance. Here, we study the computational complexity of the linear two-stage stochastic programming (TSSP) problem. Stochastic programming problems are optimization problems in which one or more parameters are uncertain. In the TSSP, decisions must be made in two separate stages: in the first stage, a decision must be made before the outcomes of the uncertain parameters are known; in the second stage, these parameters are observed and a second decision must be made given these observations [2].

In the traditional setting, complexity is calculated in terms of the size of the input to the problem in question. However, in the case of stochastic programming, the input size is not the only factor that impacts complexity; therefore, the standard approach may not provide us with a reliable predictive model of the amount of time required to solve a problem. This issue has been observed in practice. For example, using numerical methods, Parpas and Webster showed that the complexity of solving a class of stochastic optimization models does not only depend on the dimension but also on whether the optimal decision is smooth [7]. The smoothness of the optimization model is usually not taken into account by the existing complexity results in this area. This paper is a first attempt toward addressing this issue.

In order to incorporate the smoothness of the underlying model, we use information-based complexity (IBC); we assume there is an oracle which can provide certain information about the problem, and this information is generally partial, noisy and priced. The complexity of the problem is then defined in terms of the total price of the information provided by the oracle in order to approximate the solution. The purpose of this paper is to investigate the complexity of the two-stage stochastic programming problem under the IBC framework. We consider the use of a measure of smoothness for determining complexity, as opposed to just using the dimension of the model.

Our main contribution is the application of IBC to the two-stage stochastic programming problem in order to demonstrate how the smoothness of this kind of problem impacts computational complexity. We show that even two-stage stochastic linear programs are intractable when the randomness comes from a continuous distribution, and derive error bounds for the approximation of the recourse function of the TSSP.

2. Background review

2.1. The two-stage stochastic programming problem

The two-stage stochastic programming problem is an optimization problem with uncertain parameters and for which decisions must be made in two separate stages. TSSP models are defined as follows:

$$\begin{aligned} \min_x \quad & \mathcal{Q}(x), \\ & Ax \le b, \\ & x \in X. \end{aligned} \tag{1}$$

* Corresponding author. Tel.: +44 7530819759.
E-mail addresses: gabriela.tavares11@imperial.ac.uk, gabrielaoptavares@gmail.com (G. Tavares).

0167-6377/$ – see front matter
http://dx.doi.org/10.1016/j.orl.2013.08.011

The objective function $\mathcal{Q}(x)$ is defined as

$$\mathcal{Q}(x) = c^T x + \mathbb{E}[Q(x,\xi)], \tag{2}$$

$$Q(x,\xi) = \min_{y \in Y} \{q(\xi)^T y | W(\xi) y \le h(\xi) - T(\xi) x\} \tag{3}$$

where $X \subset \mathbb{R}^{n_1}$, $Y \subset \mathbb{R}^{n_2}$, and $\xi \in \mathbb{R}^d$ is the random vector formed by the components of $q(\xi) \in \mathbb{R}^{n_2}$, $h(\xi) \in \mathbb{R}^k$, $W(\xi) \in \mathbb{R}^{k \times n_2}$ and $T(\xi) \in \mathbb{R}^{k \times n_1}$. $\mathbb{E}[Q(x,\xi)]$ is the expectation with respect to ξ. Parameters ξ, $q(\xi)$, $h(\xi)$, $W(\xi)$ and $T(\xi)$ are all uncertain. We will assume that ξ has its support in the d-dimensional hypercube $\Omega = [0,1]^d$ and that it is distributed according to some continuous probability density function $f(\xi)$. The equations above can be interpreted as follows: Eq. (1) denotes the first-stage decision, where we minimize over decision variable x before knowing the actual values of the uncertain parameters; to account for this uncertainty, we take the expected value of the second-stage function, $Q(x,\xi)$. In Eq. (3), $Q(x,\xi)$ constitutes another minimization problem in which we sample the uncertain parameters and minimize over decision variable y. $\mathcal{Q}(x)$ is called the expected value function or the recourse function of the TSSP. More information about the theoretical properties and the applications of the TSSP can be found in the textbook by Birge and Louveaux [2].

2.2. Information-based complexity

We now briefly introduce the concepts of IBC and present a result for the complexity of the TSSP given by this framework. From Eqs. (1)–(3), it is clear that the recourse function $\mathcal{Q}(x)$ dominates the complexity of the TSSP. Note that for a fixed x and ξ, $Q(x,\xi)$ can be evaluated in polynomial time using (for example) an interior point algorithm. Once the recourse function is determined, we are left with a simple optimization problem, which can generally be solved in polynomial time. Following Traub and Werschulz [9], we assume that there is an oracle which can approximate the recourse function using a reasonable algorithm to compute its answers. We apply the real-number model to determine what operations we can perform with the algorithm and the cost of each operation. At time $t = 1$, the algorithm chooses an arbitrary value for x_1, with $x_1 \in X$ as specified in the TSSP, and the oracle responds with an upper bound $ub(x_1)$ and a lower bound $lb(x_1)$. Based on that answer, the algorithm chooses x_2, and so on. We assume that the algorithm will always choose x_i in such a way that the interval given by the oracle for x_i is smaller than the interval given for x_{i-1}. The algorithm stops when the answer provided by the oracle converges, i.e., when we obtain a value x_n which approximately minimizes function $\mathcal{Q}(x)$. Note that popular algorithms for TSSP such as Bender's decomposition and stochastic dual dynamic programming fit exactly with this framework. Therefore IBC is a natural framework to study this class of optimization models.

The complexity of the TSSP is defined in IBC as the total cost of approximating $\mathcal{Q}(x)$. This cost is obtained by adding the total information cost, as determined by the oracle, and the total combinatorial cost, as determined by the real-number model. Since the algorithm only requires a small number of combinatory operations in order to compare the bounds given by the oracle, and assuming that the algorithm chooses each x_i in polynomial time (say in $\mathcal{P}(n_1)$ for some polynomial $\mathcal{P}$), it follows that the computational complexity of the TSSP will be $\mathcal{O}(n\mathcal{P}(n_1))$, where n is the total number of queries the algorithm makes to the oracle.

3. Related work

Our main reference in IBC is the book by Traub and Werschulz [9], in which the authors describe the IBC framework and discuss its applications. Their study of the complexity of the integration problem is of particular interest to us, since the recourse function requires the computation of an integral which dominates the complexity of the TSSP. For the integration problem, the authors derive bounds on the computational complexity and show that using the modified trapezoidal rule with n sample points provides minimal error among all other algorithms that use information of cardinality at most n.

Previous work on the complexity of stochastic programming problems focused on using the Turing model of computation. Dyer and Stougie study the complexity of the TSSP and of the multi-stage stochastic programming problem (the general case) by performing reduction from other problems [4]. They show that the evaluation of the expected value function $\mathcal{Q}(x)$ for a fixed x is #P-hard both in the case of discrete uncertain parameters and in the continuous case, in which the parameters are continuously distributed. These results do not explain how the smoothness of the functions that define the problems influences their computation time. As shown in [7], as well as in the example of Section 6 of this paper, the smoothness of the model can have a substantial impact on the solution.

Shapiro and Nemirovski study the complexity of stochastic programming problems using Monte Carlo sampling algorithms [8]. The authors show that the TSSP can be solved reasonably efficiently through Monte Carlo methods and apply large deviations theory to derive bounds on the number of samples and on the number of steps needed to solve the TSSP. Their analysis differs from ours in two crucial respects. First, the resulting complexity results do not depend on the smoothness of the problem. Second, we only look at deterministic algorithms for the evaluation of the recourse function. It will be an interesting extension to this paper to study non-deterministic algorithms as well as to extend the results to the multi-stage case.

In a recent paper, Agarwal et al. investigate the complexity of convex optimization problems under the IBC framework [1]. The authors describe a new measure for discrepancy between functions. They apply this measure to derive lower bounds for several convex optimization problems by reducing them to statistical parameter estimation. In their work, Agarwal et al. assume Lipschitz continuity in order to derive results for the complexity of stochastic optimization problems. Our work differs from theirs in that we first smooth the second-stage function of our problem, and then derive error bounds in terms of the measure of smoothness obtained.

4. Smoothing of the second-stage function

In general, the complexity of the TSSP is dictated by how hard it is to compute the recourse function $\mathcal{Q}(x)$. The expected value $\mathbb{E}[Q(x,\xi)]$ is generally not easy to compute; it requires the integration of a function f over the domain in which the uncertain parameter ξ is defined, but the exact form of function f is undetermined. For $\xi \in \mathbb{R}^d$, this integral becomes d-dimensional, which is hard to compute for large d. In order to derive the approximation error bounds for the recourse function, we first need to ensure that the function we are integrating is differentiable. The second-stage decision of the TSSP, described in Eq. (3), is in general a non-differentiable minimization problem. In order to transform the second-stage function into a differentiable function, we smooth $Q(x,\xi)$ with respect to ξ for fixed x, deriving an approximation which is guaranteed to be differentiable.

Theorem 4.1. *Suppose that the TSSP model in Eqs. (1)–(3) has complete recourse and that ξ has its support in $[0,1]^d$. Then for fixed x and ξ, the following holds:*

$$\hat{Q}(x,\xi) = \lim_{\epsilon \to 0} \left[-\epsilon \ln \left(\sum_{y \in Y^B(x,\xi)} \exp\left(\frac{-q(\xi)^T y}{\epsilon} \right) \right) \right],$$

where $Y^B(x,\xi)$ denotes the set of basic feasible solutions of the second-stage linear program for a fixed first-stage solution.

Proof. We follow the smoothing technique described by Parpas et al. in [6]. Since we assume complete recourse, the set Y^B is non-empty. Furthermore, the smoothness with respect to the x variable is not required and so the dependency of Y^B on x is not a concern in our analysis. For clarity, we drop the dependence of Y and Y^B on x. We begin by rewriting the second-stage decision:

$$Q(x,\xi) = \min_{y(\xi)\in Y} \{q(\xi)^T y(\xi)\},$$
$$Y = \{y(\xi) \in [0,1] | W(\xi)y(\xi) + T(\xi)x \le h(\xi)\}. \tag{4}$$

Let $\hat{y}(\xi)$ denote a global minimizer of (4). Then the optimal solution for the problem is given by $\hat{Q}(x,\xi) = q(\xi)^T\hat{y}(\xi)$. Moreover,

$$q(\xi)^T\hat{y}(\xi) \le q(\xi)^T y(\xi), \quad \forall y(\xi) \in Y. \tag{5}$$

Let $\epsilon > 0$ denote the smoothness parameter. Details about the influence of parameter ϵ in the smoothness of a function are discussed by Tsoukalas et al. in [10]. Multiplying both sides of Eq. (5) by $-1/\epsilon$ and taking the exponential, we get

$$\exp\left(-\frac{q(\xi)^T\hat{y}(\xi)}{\epsilon}\right) \ge \exp\left(-\frac{q(\xi)^T y(\xi)}{\epsilon}\right). \tag{6}$$

In order to avoid the emergence of an integral in the approximation, we perform the smoothing using a summation over the set of m basic feasible solutions, $Y^B = \{y_1, y_2, \ldots, y_m\}$. The basic feasible solutions are the vertices, or extremes points, of the polytope corresponding to the feasible region, Y. It follows from Eq. (6) that

$$\exp\left(\frac{-q(\xi)^T y(\hat{\xi})}{\epsilon}\right) \le \sum_{Y^B} \exp\left(\frac{-q(\xi)^T y(\xi)}{\epsilon}\right)$$
$$\le m \exp\left(\frac{-q(\xi)^T y(\hat{\xi})}{\epsilon}\right). \tag{7}$$

Taking the log and multiplying by $-\epsilon$ we get

$$q(\xi)^T\hat{y}(\xi) \ge -\epsilon \ln\left(\sum_{Y^B} \exp\left(\frac{-q(\xi)^T y(\xi)}{\epsilon}\right)\right)$$
$$\ge -\epsilon\ln(m) + q(\xi)^T\hat{y}(\xi). \tag{8}$$

Finally, taking the limit where $\epsilon \downarrow 0$, we obtain the required result. □

5. Approximation error bounds

In the last section we obtained the following approximation for $\hat{Q}(x,\xi)$:

$$\hat{Q}^\epsilon(x,\xi) = \left[-\epsilon \ln\left(\sum_{y\in Y^B(x,\xi)} \exp\left(\frac{-q(\xi)^T y}{\epsilon}\right)\right)\right]. \tag{9}$$

Note that the preceding equation does not contain a minimization with respect to the second-stage variables. We will assume that $\hat{Q}^\epsilon(x,\xi)$ is smooth with respect to the ξ variables. We make our assumption precise below.

Assumption 5.1. The function $\hat{Q}^\epsilon(x,\xi)$ is differentiable with respect to its second argument up to order $r(\epsilon)$.

To make the notation easier to follow we will write r for $r(\epsilon)$. Based on the approximation in (9) and the smoothness assumption, we derive bounds on the approximation error of the recourse function, $\mathcal{Q}(x)$, in terms of the smoothness of the second-stage function. Before we state our main result, we provide some definitions from the literature on information-based complexity. More information on this setup can be obtained in [9].

The class of permissible functions is denoted by

$$W_r^p(G) = \left\{ f : G \to \mathbb{R} \,\middle|\, \sum_{k=1}^{r} \|\partial f^k\|_p \le 1 \right\}, \tag{10}$$

where ∂f^k is the k-order derivative of f. The solution operator $S : W_r^p \to Y$, where Y is the solution space, is given by

$$S(f) = \int_G f(x)dx. \tag{11}$$

The information operator N is defined as the evaluation of function f for points in the domain G, $N(f) = \Delta = \{f(x) | x \in G\}$. We also define $\mathcal{A}(\Delta)$, the class of all algorithms that use information Δ, as $\mathcal{A}(\Delta) = \{\varphi : N(f) \to Y\}$. Using information Δ and an algorithm $\varphi \in \mathcal{A}$, we can obtain an approximation for $S(f)$ given by $A(f) = \varphi(N(f))$. The approximation error is then defined as

$$e(S,\Delta) = \inf_{\varphi\in\mathcal{A}(\Delta)} \sup_{f\in W_p^r(G)} \|S(f) - A(f)\|. \tag{12}$$

Eq. (12) takes the supremum over all possible functions in class W_p^r and the infimum over all possible algorithms that use information Δ. This means that we are using the worst-case setting in terms of the class of functions, but taking the best possible algorithm in order to obtain the approximation.

We define n basis functions, $h_1, \ldots, h_n \in C^\infty(G)$, which we will use to obtain an approximation of any function in W_p^r in terms of a linear combination of h_i. The approximation $P_n(f)$ is defined as

$$P_n : C^\infty(G) \to C^\infty(G), \tag{13}$$

where

$$P_n(f)(x) = \sum_{i=1}^{n} f(x_i)h_i(x) \tag{14}$$

and $\forall n \in \mathbb{N}$, with $x_1, \ldots, x_n \in G$.

We will make use of the following upper bound on the approximation error, derived by Ciarlet [3]:

$$\|f - P_n(f)\|_{Lq} \le c \begin{cases} n^{(-r/d+1/p-1/q)}\|f\|_{W_p^r} \\ n^{-r/d}\|f\|_{W_p^r} \end{cases} \tag{15}$$

where $c > 0$ is a constant and Lq denotes the Lq-norm.

We will also make the following assumptions regarding the data of the problem.

Assumption 5.2. The class of permissible functions W_r^p is defined over the domain $G = [0,1]^d$.

Assumption 5.3. The random variable ξ has its support in $[0,1]^d$.

We are finally ready to derive an upper bound on the complexity of solving two-stage stochastic programming problems.

Theorem 5.1. *The upper bound on the approximation error of the recourse function is given by*

$$e(S(f), A(f)) \le cn^{-r(\epsilon)/d}\|f(\xi)\|,$$

where n is the number of queries made to the oracle and c is an arbitrary constant.

Proof. We will use the approximation of f given in (13) and (14) to obtain an upper bound for the approximation error between the solution operator $S(f)$ and our approximation $A(f)$. To this end, let $f \in W_p^r(G)$ be arbitrary. Then we construct an approximation for $S(f)$ using Eq. (14)

$$Q_n(f) = \int_G P_n(f)(x)dx = \sum_{i=1}^{n} f(x_i)w_i. \tag{16}$$

Taking the information operator $N(f)$ and an algorithm φ such that $\varphi(y_1, \ldots, y_n) = \sum_i w_i y_i$, we obtain

$$\begin{aligned} e(S(f), A(f)) &= \sup_f |S(f) - \varphi(N(f))| \\ &= \sup_f |S(f) - Q_n(f)| \\ &= \left| \int \left(f(x) - \left(\sum_{i=1}^{n} f(x_i) \int_G h_i(x)dx \right) \right) dx \right| \\ &\le n^{-r/d} c \|f\|_{W_p^r} \\ &\le cn^{-r/d}, \end{aligned}$$

where we have used the fact that $\|f\|_{W_p^r} = 1$. Next, we apply the approximation obtained for $\hat{Q}(x, \xi)$ in Theorem 4.1 in order to derive the upper bound in terms of the smoothing parameter ϵ. Our aim is to obtain a value for $\mathbb{E}[Q(x, \xi)]$ by integrating over all possible values of ξ. The integrand $f(\xi)$ is given by

$$f(\xi) = p(\xi) \left[-\epsilon \ln \left(\sum_{Y^B} \exp \left(\frac{-q(\xi)^T y(\xi)}{\epsilon} \right) \right) \right], \tag{17}$$

where $p(\xi)$ is the known probability distribution for parameter ξ. We can define the solution operator $S(f)$ as

$$S(f) = \mathbb{E}[Q(x, \xi)] = \int_G f(\xi)d\xi. \tag{18}$$

As before, we obtain an approximation for $S(f)$:

$$P_n(f) = \sum_{i=1}^{n} p(\xi_i)\hat{Q}(x, \xi_i) \int_G h_i(\xi_i)d\xi, \tag{19}$$

and the approximation error will be given by

$$e(S(f), A(f)) = \left| \int_G f(\xi) - \sum_{i=1}^{n} p(\xi_i)\hat{Q}(x, \xi_i) \int_G h_i(\xi_i)d\xi \right|. \tag{20}$$

Finally, we can apply the upper bound in (15) to obtain the required result. □

We conclude this section by proving a lower bound on the complexity of the TSSP which shows that our bound is tight.

Theorem 5.2. *The lower bound on the approximation error of the recourse function is given by*

$$e(S(f), A(f)) \ge cn^{-r(\epsilon)/d},$$

where n is the number of queries made to the oracle and c is an arbitrary constant.

Proof. Let m be such that $n = m^d$, we subdivide the domain G into $\tilde{n} = (2m)^d$ closed cubes of length $1/2m$: $\{G_i\}_{i=1}^{\tilde{n}}$. Let g_i denote the point of G_i with the smallest coordinate in each direction. We take function $\psi \in C^\infty(\mathbb{R}^d)$ such that $\text{supp}(\psi) \subseteq \text{int}(G)$, and $\int_G \psi(x)dx = 1$. Let $\psi_i(x) = \psi(2m(x - g_i))$. Now using $y = 2m(x - g_i)$ and $dy = 2mdx$, we obtain

$$\int_G \psi_i(x)dx = \int_G \psi(y) \left(\frac{1}{2m} \right)^d dy = (2m)^{-d}.$$

We take information $N_f = (f(x_1), \ldots, f(x_n))$ and an arbitrary algorithm $A = \varphi \cdot N$, $A \in \mathcal{A}_n(\Delta)$. We define J as the set of all points not interior to G:

$$\begin{aligned} &J = \{j : 1 \le j \le \tilde{n}, \{x_1, \ldots, x_n\} \cap \text{int}(G_j) = \emptyset\}, \\ &|J| \ge \tilde{n} - n = (2^d - 1)n. \end{aligned}$$

We can now derive the lower bound for the approximation error in terms of the smoothing parameter ϵ. We replace x in the previous equations with ξ and redefine the integrand $f(\xi)$ in terms of functions ψ_j to obtain f_0 as follows:

$$f_0(\xi) = \frac{\sum_{j \in J} \psi_j(\xi)}{\left\| \sum_{j \in J} \psi_j(\xi) \right\|_{W_p^{r(\epsilon)}(G)}}.$$

Integrating f_0 over G, we obtain

$$\begin{aligned} \int_G f_0(\xi)d\xi &\ge \frac{(2m)^{-d}(\tilde{n} - n)}{cn^{r(\epsilon)/d - 1/p}} \\ &= cn^{-r(\epsilon)/d} n^{1/p} \frac{2^{-d}}{n}(2^d - 1)n \\ &= cn^{-r(\epsilon)/d}(n^{1/p}2^{-d}) \ge cn^{-r(\epsilon)/d}. \end{aligned}$$

Finally, we obtain the required result:

$$\begin{aligned} e(S(f), A(f)) &= \sup_f |S(f) - \varphi(N(f))| \\ &\ge \max_{\sigma = \pm 1} |S(\sigma f_0) - \varphi(0)| \ge |S(f_0)| \ge cn^{-r(\epsilon)/d}. \quad \square \end{aligned}$$

We can see in Theorems 5.1 and 5.2 that the upper and lower bounds derived for the approximation error are functions of both the dimensionality and the smoothness $r(\epsilon)$ of the second-stage function. We can also see that these bounds are tight. Therefore, we conclude that the complexity of the TSSP increases exponentially with the dimension of the problem, but decreases exponentially when $Q(x, \xi)$ becomes smoother (when $r(\epsilon)$ increases). The error bounds given by Jensen's and Edmundson–Madansky's inequalities provide us with a theoretical result which cannot be applied to the measurement of computational complexity. These bounds require an exponential cost in computation in order to become tighter. Even though the evaluation of the bounds can be done in polynomial time (by solving linear programs in our setup), an exponential number of LPs needs to be solved. This result is of course in line with our main result. In addition, our theory predicts that the smoother the recourse function, the less LPs need to be solved to obtain the Jensen and Edmundson–Madansky bounds. For example, if the recourse function is linear, then the Jensen and EM bounds are tight and only two LPs are required in order to confirm this.

6. Illustrative example

We now present an example to illustrate the theoretical results shown in the previous sections. We illustrate our point using a simple static stochastic optimization model (i.e. we do not incorporate any first-stage decisions). We consider two optimization models,

$$Q_1(\xi) = \min_{y(\xi)} \left\{ -y(\xi)1_{\{\xi \ge c_0\}} \mid 0 \le y(\xi) \le 1 \right\} \tag{21}$$

$$Q_2(\xi) = \min_{y(\xi)} \left\{ -y(\xi)1_{\{c_1 \le \xi \le c_2\}} \mid 0 \le y(\xi) \le 1 \right\}, \tag{22}$$

Table 1
Percentage increase in the number of evaluations required to solve the problem as a function of the number of dimensions of the integrand (compared to the 2-dimensional model), using both $Q_1(\xi)$ and $Q_2(\xi)$. Since $Q_2(\xi)$ is less smooth, the number of evaluations increases much more rapidly as the number of dimensions increases.

Dimensions	% increase in number of evaluations	
	Estimation of $Q_1(\xi)$	Estimation of $Q_2(\xi)$
4	0.0136	0
6	0.0652	323.60
8	0	2840.0
10	0.2900	5743.7

where 1_A denotes the indicator function on the set A. In the numerical experiments, we used constants $c_0 = 5$, $c_1 = 4.75$ and $c_2 = 7.5$. To solve the integration problem, we applied the Cuhre algorithm, which is provided as part of the Cuba library [5].

Table 1 shows the percentage increase in the number of evaluations performed by the algorithm as a function of the number of dimensions of the integrand. This increase is calculated using the 2-dimensional model. We can see from this table that the number of evaluations for $Q_1(\xi)$, the smoother function, grows slowly when compared to $Q_2(\xi)$. This can be explained by our results for the approximation error bounds in the previous section: when a function is reasonably smooth, the smoothness parameter r is large; therefore, an increase in the dimension d of the integrand does not considerably affect the complexity of the problem. On the other hand, for a non-smooth function, the value of r is small, and even a small increase in d will affect the complexity, causing a dramatic increase in the number of evaluations necessary to solve the problem.

This simple example illustrates the importance of smoothness in the complexity of stochastic optimization models. It would be interesting to see if this approach can be extended to the multi-stage case. Another direction for future research is the study of the complexity of stochastic programming with random algorithms (e.g. Monte-Carlo based methods).

Acknowledgments

The work of the second author was partially supported by a FP7 Marie Curie Career Integration Grant (PCIG11-GA-2012-321698 SOC-MP-ES).

References

[1] A. Agarwal, P. Bartlett, P. Ravikumar, M. Wainwright, Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization, IEEE Trans. Inform. Theory 58 (2012) 3235–3249.
[2] J.R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer, 1997.
[3] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, in: Stud. Math. Appl., vol. 4, North-Holland, 1978.
[4] M. Dyer, L. Stougie, Computational complexity of stochastic programming problems, Math. Program. 106 (2006) 423–432.
[5] T. Hahn, Cuba—a library for multidimensional numerical integration, Comput. Phys. Comm. 168 (2005) 78–95.
[6] P. Parpas, B. Rustem, E.N. Pistikopoulos, Linearly constrained global optimization and stochastic differential equations, J. Global Optim. 36 (2006) 191–217.
[7] P. Parpas, M. Webster, A stochastic minimum principle and an adaptive pathwise algorithm for stochastic optimal control, Automatica 49 (2013) 1663–1671.
[8] A. Shapiro, A. Nemirovski, On complexity of stochastic programming problems, Contin. Optim. (2005) 111–146.
[9] J.F. Traub, A.G. Werschulz, Complexity and Information, Cambridge University Press, 1998.
[10] A. Tsoukalas, P. Parpas, B. Rustem, A smoothing algorithm for finite min–max–min problems, Optim. Lett. 3 (2009) 49–62.