European Journal of Operational Research xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Stochastics and Statistics

Asymptotic formulas for the derivatives of probability functions and their Monte Carlo estimations

Josselin Garnier [a,*], Abdennebi Omrane [b], Youssef Rouchdy [c]

[a] Laboratoire de Probabilités et Modèles Aléatoires and Laboratoire Jacques-Louis Lions, Université Paris VII, 2 Place Jussieu, 75251 Paris Cedex 5, France
[b] Université des Antilles et de la Guyane, UFR Sciences Exactes et Naturelles, DMI, 97159 Pointe à Pitre, Guadeloupe, France
[c] CEREMADE, Université de Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France

ARTICLE INFO

Article history:
Received 3 November 2007
Accepted 26 September 2008
Available online xxxx

PACS:
90C15
65C05

Keywords:
Applied probability
Monte Carlo methods
Stochastic programming
Optimization with constraints
Random constraints

ABSTRACT

One of the key problems in chance constrained programming for nonlinear optimization problems is the evaluation of derivatives of joint probability functions of the form $P(x) = \mathbb{P}(g_p(x, \Lambda) \leqslant c_p, p = 1, \ldots, N_c)$. Here $x \in \mathbb{R}^{N_x}$ is the vector of physical parameters, $\Lambda \in \mathbb{R}^{N_\Lambda}$ is a random vector describing the uncertainty of the model, $g : \mathbb{R}^{N_x} \times \mathbb{R}^{N_\Lambda} \to \mathbb{R}^{N_c}$ is the constraints mapping, and $c \in \mathbb{R}^{N_c}$ is the vector of constraint levels. In this paper specific Monte Carlo tools for the estimations of the gradient and Hessian of $P(x)$ are proposed when the input random vector Λ has a multivariate normal distribution and small variances. Using the small variance hypothesis, approximate expressions for the first- and second-order derivatives are obtained, whose Monte Carlo estimations have low computational costs. The number of calls of the constraints mapping g for the proposed estimators of the gradient and Hessian of $P(x)$ is only $1 + 2N_x + 2N_\Lambda$.

These tools are implemented in penalized optimization routines adapted to stochastic optimization, and are shown to reduce the computational cost of chance constrained programming substantially.

1. Introduction

It is possible to distinguish two types of optimization problems in the presence of randomness. In the first type, the cost function is random, or it is the expectation of a random cost function, and the set of feasible solutions is deterministic. In the second type, the cost function is deterministic and the set of constraints is random. One of the key issues is then to study the gradient and/or the Hessian of the constraint function in order to apply KKT (Karush Kuhn Tucker) or related theorems in optimization theory (Ruszczyński, 2006). Chance-constrained problems are described in the books by Prékopa (1995), Ruszczyński and Shapiro (2003), Calafiore and Dabbene (2006), and Kall and Wallace (1994) as well as in survey articles by Birge (1997) and Sen and Higle (1999) for instance.

In the present work, a deterministic cost function $J(x)$ has to be minimized under a set of N_c random constraints

$$g_p(x, \Lambda) \leqslant c_p, \quad p = 1, \ldots, N_c, \tag{1}$$

where $g_p : \mathbb{R}^{N_x} \times \mathbb{R}^{N_\Lambda} \to \mathbb{R}$ is a real-valued constraints mapping and $c_p \in \mathbb{R}$ is the constraint level, for each $p = 1, \ldots, N_c, x \in \mathbb{R}^{N_x}$ is the vector of physical parameters, and $\Lambda \in \mathbb{R}^{N_\Lambda}$ is a continuous random vector describing the uncertainty of the model. In this paper, we assume that Λ has a multivariate normal distribution.

In many situations, the set of points x such that the N_c constraints (1) are satisfied for (almost) all realizations of the random vector Λ is empty. Indeed, even if constraint violation at x happens only for a very unlikely subset of realizations of Λ, the point x has to be excluded. It then makes sense to look for an admissible set of points x that satisfy the constraints with high probability $1 - \alpha$, $0 < \alpha < 1$ (or $\ll 1$), in the sense that only a small proportion α of realizations of the random vector Λ leads to constraint violation at x. The set of constraints (1) is then substituted by the probabilistic constraint

$$P(x) \geqslant 1 - \alpha, \tag{2}$$

with

$$P(x) = \mathbb{P}(g_p(x, \Lambda) \leqslant c_p, \quad p = 1, \ldots, N_c). \tag{3}$$

The probabilistic constraint prescribes a lower bound for the probability $P(x)$ of simultaneous satisfaction of the N_c random constraints. Situations where probabilistic constraints of the form (2) and (3) are encountered are listed in the book by Prékopa (1995) and in the articles by Henrion and Römisch (2000), Henrion and Strugarek (2008), and Prékopa (1971).

* Corresponding author. Tel.: +33 1 4478693.
E-mail addresses: garnier@math.jussieu.fr (J. Garnier), aomrane@univ-ag.fr (A. Omrane), rouchdy@ceremade.dauphine.fr (Y. Rouchdy).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.09.026

Optimization problems under probabilistic constraints of the form (2) and (3) are particularly difficult to solve for continuous random vectors, such as multivariate normal, because the probability (3) has the form of a high-dimensional integral that cannot be evaluated directly, but has to be approximated by numerical quadrature or by Monte Carlo (MC) simulations (Dubi, 2000). Because MC simulation appears to offer the best possibilities for higher dimensions, it seems to be the natural choice for use in stochastic programs. Birge and Louveaux (1997) describe some basic approaches built on sampling methods. They also consider some extentions of the MC method to include analytical evaluations exploiting problem structure in probabilistic constraints estimation.

The evaluation of the gradient (and possibly the Hessian) of (3) is required in nonlinear optimization routines. In the separable case $g(x,\Lambda) = \Lambda - f(x)$ this issue boils down to the calculus of normal distribution functions, since Hessians and gradients can then be exactly reduced to function values of such distribution functions (Prékopa, 1995). One can then use the existing techniques for the estimation of multivariate distribution functions (see the numerical and simulation techniques proposed by Schervish (1984), Genz (1992), Deák (1986), Szántai (2001, 2002), or the sharp bounds obtained by Prékopa (1990) using linear programming or by Bukszár and Prékopa (2001) using graph-theoretical constructions). Our approach addresses the general model $g(x,\Lambda)$, including, but not restricted to, the multiplicative model $g(x,\Lambda) = G(x)\Lambda - f(x)$. The first idea in order to estimate the gradient and Hessian of the probability $P(x)$ for the general model is to use finite differences of estimates of the probability $P(x)$. Of course, these estimates are likely to be much more precise with the use of advanced (graph-theoretical, bounding, variance-reduction, etc.) techniques described in the literature than with the use of crude MC (for function values). However, we want to emphasize the disadvantage of excessive function calls for finite differences, no matter which specific method is used for calculating function values.

We consider a situation in which the computational cost of the estimation of the gradient and Hessian of the probability (3) is proportional to the number of calls of the function g. We propose a specific MC method to evaluate the gradient and Hessian of the probability $P(x)$ and we implement this method in penalization optimization routines adapted to stochastic optimization. The proposed MC estimators are based on asymptotic expansions of probabilistic representations of the partial derivatives of the probability (3) in the case in which the variances of the input random parameters Λ_i are small. MC estimators of the asymptotic expressions can be implemented and we show that this method reduces the computational cost of the evaluations of the derivatives of the probability $P(x)$, hence reduces the total computational cost of chance constrained programming. More precisely, the proposed MC estimators of the gradient and of the Hessian of the probability $P(x)$ at some point x only requires $1 + 2N_x + 2N_\Lambda$ calls of the function g.

The main contribution of this paper is the description of the asymptotic expressions of the gradient and Hessian of the probability $P(x)$ for small variances. We then use the crude MC method in order to build estimators of these asymptotic expressions. Indeed, these expressions can be expressed as expectations of functions of normal multivariate random vectors. It could be of interest here to use a more advanced simulation technique, and in fact we would recommend to do so in practice. However, this is not relevant for the computational cost as discussed of this paper, which is defined as the number of calls of the function g, while we assume that the generation of any normal multivariate random vector is essentially cost-free.

The paper is organized as follows: In Section 2, we introduce the problem and define the probabilistic constraint. In Sections 3 and 4, we propose complete and approximate expressions of the derivatives of the probability $P(x)$ and deduce low-cost MC estimators for these quantities. We finally apply these methods to a simple demonstration example (a separable model), an academic example (a polynomial model), and an industrial case.

2. The probabilistic constraint

In this section, we consider the probabilistic constraint (2) and (3). From now on we assume that Λ is a normal multivariate random vector with mean 0 and covariance matrix K. In many applications the Λ_i's are independent and have normal distribution with mean 0 and variance σ_i^2, so that K is the diagonal $N_\Lambda \times N_\Lambda$ matrix $(\sigma_i^2)_{i=1,\dots,N_\Lambda}$, but we shall carry out our analysis with a general covariance matrix.

The main focus of the forthcoming analysis is the multiplicative model, in which the constraint mapping has the form $g(x,\Lambda) = G(x)\Lambda - f(x)$, where $G(x)$ is a matrix field and $f(x)$ is a vector field. However, the constraint mapping has a general form in the industrial application we have in mind. In this section, we show how the general model can be reduced to the multiplicative model in the case in which the variances of the random parameters are small.

The constraints are of the form (1) where $g : \mathbb{R}^{N_x} \times \mathbb{R}^{N_\Lambda} \to \mathbb{R}^{N_c}$ is a function of class $\mathscr{C}^2$. In general we can distinguish two types of constraints:

- n constraints depend on the random vector Λ:
$$g_p(x,\Lambda) \leqslant c_p, \quad p = 1,\dots,n. \tag{4}$$
- $N_c - n$ constraints are deterministic:
$$g_p(x) \leqslant c_p, \quad p = n+1,\dots,N_c. \tag{5}$$

The $N_c - n$ deterministic constraints will be dealt with separately, using standard constrained optimization tools. Of course, it may happen that $N_c = n$, i.e. all constraints are random.

As discussed in the introduction, in chance constrained programming, the set of random constraints (4) is substituted by the probabilistic constraint (2) with

$$P(x) = \mathbb{P}(g_p(x,\Lambda) \leqslant c_p,\ p = 1,\dots,n). \tag{6}$$

In the industrial example we have in mind (described in Section 3.7), the function g is a very complex numerical routine. As a consequence, the computational cost of an evaluation method of the probability $P(x)$, its gradient and Hessian, will be considered to be proportional to the number of calls of g in this paper. In other words, we assume that the random number generation of the random vector Λ is essentially cost-free. Note that this may be not always the case, and the computational cost of the generation of the random vectors could play a role. Recent numerical procedures to evaluate normal multivariate distribution functions can be found in the papers by Szántai (2001) and Gassmann et al. (2002).

Let us fix x and consider the n constraints (4) that depend on Λ. We first perform a linearization of these constraints with respect to Λ and consider the approximation

$$g_p(x,\Lambda) \simeq g_p(x,0) + \sum_{i=1}^{N_\Lambda} \frac{\partial g_p}{\partial \Lambda_i}(x,0)\Lambda_i,$$

for Λ in the hypercube $\prod_{i=1}^{N_\Lambda}[-3\sigma_i, 3\sigma_i]$. This approximation is valid when σ is small (since g is of class $\mathscr{C}^2$). Note that the choice of the constant 3 is actually determined by the level $1-\alpha$ of the admissible set (it could be replaced by the $1-\alpha$-quantile of the standard normal distribution). The set of constraints (4) can then be approximated by the set of inequalities

$$G(x)\Lambda \leqslant C(x), \tag{7}$$

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

where $G(x)$ is the deterministic $n \times N_\Lambda$ matrix and $C(x)$ is the deterministic n-dimensional vector defined by

$$G_{pi}(x) = \frac{\partial g_p}{\partial \Lambda_i}(x,0), \quad p = 1,\dots,n,\ i = 1,\dots,N_\Lambda,$$
$$C_p(x) = c_p - g_p(x,0), \quad p = 1,\dots,n.$$

The computation of C requires one evaluation of g and the one of G requires $2N_\Lambda$ evaluations of g (by second-order finite differences).

The probability $P(x)$ defined by (6) can then be approximated by

$$P(x) = \mathbb{P}(G(x)\Lambda \leqslant C(x)). \tag{8}$$

In the following we propose probabilistic representations and estimations of the gradient and Hessian of $P(x)$, when $P(x)$ has the form (8). However, we shall apply the proposed estimators to problems in which $P(x)$ has the general form (6).

The random vector $G(x)\Lambda$ has zero-mean multivariate normal distribution. Its $n \times n$ covariance matrix $\Gamma(x)$ is real, symmetric, and given by

$$\Gamma(x) = G(x)KG(x)^t, \tag{9}$$

where K is the covariance matrix of Λ. We will assume in the following that $\Gamma(x)$ is an invertible matrix. This is the case in the applications we have in mind, and this is the main motivation for introducing the distinction between the n constraints that depend on Λ and the ones that do not depend on it. This hypothesis could be removed, but this would complicate the presentation.

Therefore, the probability $P(x)$ can be expressed as the n-dimensional integral

$$P(x) = \int_{-\infty}^{C_1(x)} \cdots \int_{-\infty}^{C_n(x)} p_{\Gamma(x)}(z_1,\dots,z_n)\mathrm{d}z_1 \cdots \mathrm{d}z_n, \tag{10}$$

where p_Γ is the density of the n-dimensional normal distribution with mean 0 and covariance matrix Γ:

$$p_\Gamma(z) = \frac{1}{\sqrt{(2\pi)^n \det \Gamma}} \exp\left(-\frac{1}{2} z^t \Gamma^{-1} z\right).$$

If $\Gamma(x)$ were diagonal, then we would have simply

$$P(x) = \prod_{p=1}^{n} \Phi\left(\frac{C_p(x)}{\sqrt{\Gamma_{pp}(x)}}\right),$$

where Φ is the standard normal distribution function. In this case, the probability $P(x)$ can be evaluated with arbitrary precision, as well as its gradient and Hessian. Unfortunately, this is not the case in our applications, as well as in many other applications, so that the evaluations of the multi-dimensional integral (10) and its derivatives are necessary.

Let us discuss briefly the estimation of the probability $P(x)$ before considering the estimation of its derivatives. The problem is reduced to the evaluation of the cumulative distribution function of a multivariate normal distribution. This can be done by a crude MC method. This method is easy to implement and it does not require any new call of the function g. It is reduced to the generation of an independent and identically distributed (iid) sequence of normal random vectors with mean 0 and covariance matrix $\Gamma(x)$. Accordingly, $P(x)$ can be evaluated by MC method, and the estimator has the form:

$$\widehat{P}^{(M)}(x) = \frac{1}{M} \sum_{l=1}^{M} \left[\prod_{k=1}^{n} \mathbf{1}_{(-\infty, C_k(x)]}(Z_k^{(l)})\right], \tag{11}$$

where $Z^{(l)}, l = 1,\dots,M$ is an iid sequence of random vectors with density $p_{\Gamma(x)}$. The indicator function $\mathbf{1}_A$ is defined such that $\mathbf{1}_A(z) = 1$ if $z \in A$ and 0 otherwise. The computational cost of the MC method, i.e. the number of calls of the function g, is $1 + 2N_\Lambda$, which is necessary to evaluate $C(x), G(x)$, and therefore $\Gamma(x)$. This cost is independent of M, since we assume in this paper that the random number generation of Λ is cost-free. Note that:

- a more advanced method (graph-theoretical, bounding, variance-reduction) could be used to reduce the number M of samples, but this does not affect the computational cost defined in terms of calls of g. However, we would recommend to do so in practice.
- the cost of the full MC estimator of the probability $P(x)$ defined by (3) (the one obtained without linearizing $g_p(x,\Lambda)$ in Λ) is M:

$$\widehat{P}^{(M)}_{\text{fullMC}}(x) = \frac{1}{M} \sum_{l=1}^{M} \left[\prod_{k=1}^{n} \mathbf{1}_{(-\infty, c_k]}(g_k(x, \Lambda_k^{(l)}))\right], \tag{12}$$

where $\Lambda^{(l)}, l = 1,\dots,M$ is an iid sequence of multivariate normal random vectors with mean 0 and covariance K. Indeed, the full MC estimator calls the function g for each evaluation. Even if one uses a more advanced simulation technique than the crude MC method, the cost of the full estimator is far too expensive.

3. The gradient of the probability $P(x)$

3.1. The probabilistic representation of the gradient

The gradient of the probability $P(x)$ is needed in optimization routines. We can distinguish (at least) two methods:

Method 1. It is possible to approximate the gradient of $P(x)$ by using a second-order standard finite-difference:

$$\frac{\partial P}{\partial x_k}(x) \simeq \frac{P(x + \delta e_k) - P(x - \delta e_k)}{2\delta}, \tag{13}$$

where $x \pm \delta e_k = (x_1,\dots,x_{k-1}, x_k \pm \delta, x_{k+1},\dots,x_n)$ and δ is small. Then one can use MC estimations of $P(x \pm \delta e_k)$ (or any other advanced method) to construct an estimator of $\nabla P(x)$. Note that the calibration of δ is not easy in this case.

Method 2. It is possible to carry out MC simulations with an explicit expression of the gradient. Indeed, explicit expressions of the gradient of probability functions can be found in many situations (Kibzun and Uryasev, 1998). Here, the gradient of $P(x)$ has the form

$$\frac{\partial P}{\partial x_k}(x) = \sum_{i,j=1}^{n} A_{ij}(x) \frac{\partial \Gamma_{ij}(x)}{\partial x_k} + \sum_{i=1}^{n} B_i(x) \frac{\partial C_i(x)}{\partial x_k}, \tag{14}$$

for $k = 1,\dots,N_x$, where the matrix A and the vector B are given by

$$\begin{aligned} A_{ij}(x) &= \int_{-\infty}^{C_1(x)} \cdots \int_{-\infty}^{C_n(x)} \frac{\partial p_\Gamma}{\partial \Gamma_{ij}}(z_1,\dots,z_n) dz_1 \cdots dz_n \Big|_{\Gamma=\Gamma(x)} \\ &= \int_{-\infty}^{C_1(x)} \cdots \int_{-\infty}^{C_n(x)} \frac{\partial \ln p_\Gamma}{\partial \Gamma_{ij}}(z) p_\Gamma(z) d^n z \Big|_{\Gamma=\Gamma(x)}, \end{aligned} \tag{15}$$

$$\begin{aligned} B_i(x) &= \int_{-\infty}^{C_1(x)} \cdots \int_{-\infty}^{C_{i-1}(x)} \int_{-\infty}^{C_{i+1}(x)} \cdots \int_{-\infty}^{C_n(x)} \\ &\quad \times p_{\Gamma(x)}(z_1,\dots,z_i = C_i(x),\dots,z_n) dz_1 \cdots dz_{i-1} dz_{i+1} \cdots dz_n \\ &= \frac{1}{\sqrt{2\pi\Gamma_{ii}}} \exp\left(-\frac{C_i(x)^2}{2\Gamma_{ii}}\right) \int_{-\infty}^{C_1(x)} \cdots \int_{-\infty}^{C_{i-1}(x)} \int_{-\infty}^{C_{i+1}(x)} \cdots \int_{-\infty}^{C_n(x)} \\ &\quad \times p_{\Gamma(x)}(z' | z_i = C_i(x)) d^{n-1} z', \end{aligned} \tag{16}$$

and $p_\Gamma(z'|z_i)$ is the conditional density of the $(n-1)$-dimensional random vector $Z' = (Z_1,\dots,Z_{i-1},Z_{i+1},\dots,Z_n)$ given $Z_i = z_i$. Therefore, $p_{\Gamma(x)}(z'|z_i = C_i(x))$ is the density of the $(n-1)$-dimensional normal distribution with mean

$$\tilde{\mu}^{(i)}(x) = \left(\frac{C_i(x)\Gamma_{ji}(x)}{\Gamma_{ii}(x)}\right)_{j=1,\dots,i-1,i+1,\dots,n}, \tag{17}$$

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

and $(n-1)\times(n-1)$ covariance matrix

$$\tilde{\Gamma}^{(i)}(x) = \left(\Gamma_{kl}(x) - \frac{\Gamma_{ki}(x)\Gamma_{il}(x)}{\Gamma_{ii}(x)}\right)_{k,l=1,\dots,i-1,i+1,\dots,n}. \tag{18}$$

The logarithmic derivative of p_Γ has the form

$$\begin{aligned}\frac{\partial \ln p_\Gamma(z)}{\partial \Gamma_{ij}} &= -\frac{1}{2\det\Gamma}\frac{\partial \det\Gamma}{\partial \Gamma_{ij}} - \frac{1}{2} z^t \frac{\partial \Gamma^{-1}}{\partial \Gamma_{ij}} z \\ &= -\frac{1}{2}(\Gamma^{-1})_{ji} + \frac{1}{2}(\Gamma^{-1}z)_i(\Gamma^{-1}z)_j. \end{aligned} \tag{19}$$

Finally, $(\partial\Gamma_{ij}(x))/(\partial x_k)$ and $(\partial C_i(x))/(\partial x_k)$ are given explicitly by

$$\frac{\partial \Gamma_{ij}(x)}{\partial x_k} = \sum_{l,l'=1}^{N_\Lambda} \frac{\partial}{\partial x_k}\left[\frac{\partial g_i(x,0)}{\partial \Lambda_{l'}} K_{ll'} \frac{\partial g_j(x,0)}{\partial \Lambda_l}\right],$$

$$\frac{\partial C_i(x)}{\partial x_k} = -\frac{\partial g_i(x,0)}{\partial x_k}.$$

Remark: the expression (14) is obtained by differentiating (10) with respect to x. This formal derivation can be justified (with the hypothesis that the matrix $\Gamma(x)$ is invertible and continuously differentiable and $C(x)$ is continuously differentiable) by using Remark 2.1 and then Theorem 3.1 of (Kibzun and Uryasev, 1998).

3.2. Monte Carlo estimation

The goal of this subsection is to present the simple and crude MC estimator for the gradient of the probability $P(x)$. This estimator is based on the representation (14) and is given by (22) below. Indeed, the matrix $A(x)$ can be evaluated by MC simulations, at the same time as $P(x)$ (i.e. with the same samples $Z^{(l)}, 1 \leqslant l \leqslant M$):

$$\begin{aligned}\widehat{A}_{ij}^{(M)}(x) = \frac{1}{2M}\sum_{l=1}^{M}\left[(\Gamma(x)^{-1}Z^{(l)})_i(\Gamma(x)^{-1}Z^{(l)})_j \prod_{k=1}^{n} \mathbf{1}_{(-\infty,C_k(x)]}(Z_k^{(l)})\right] \\ -\frac{1}{2}(\Gamma_{ji}^{-1}(x))\widehat{P}^{(M)}(x),\end{aligned} \tag{20}$$

where $Z^{(l)}, l = 1,\dots,M$ is an iid sequence of n-dimensional normal random vectors with mean 0 and covariance matrix $\Gamma(x)$.

$B(x)$ can be evaluated by Monte Carlo as well, but this requires a specific computation for each $i = 1,\dots,n$. The MC estimator for $B_i(x)$ is

$$\widehat{B}_i^{(M)}(x) = \frac{1}{\sqrt{2\pi\Gamma_{ii}(x)}}\exp\left(-\frac{C_{i(x)}^2}{2\Gamma_{ii}(x)}\right)\frac{1}{M}\sum_{l=1}^{M}\left[\prod_{k=1\neq i}^{n}\mathbf{1}_{(-\infty,C_k(x)]}(Z_k'^{(l)})\right], \tag{21}$$

where $(Z_1'^{(l)},\dots,Z_{i-1}'^{(l)},Z_{i+1}'^{(l)},\dots,Z_n'^{(l)}), l = 1,\dots,M$ is an iid sequence of $(n-1)$-dimensional normal random vectors with mean $\tilde{\mu}^{(i)}(x)$ and covariance matrix $\tilde{\Gamma}^{(i)}(x)$ given by (17) and (18).

Using the MC estimators (20) and (21) to evaluate the expression (14) of the gradient, we obtain the MC estimator for the gradient of the probability $P(x)$:

$$\left(\widehat{\frac{\partial P}{\partial x_k}}(x)\right)^{(M)} = \sum_{i,j=1}^{n}\widehat{A}_{ij}^{(M)}(x)\frac{\partial\Gamma_{ij}(x)}{\partial x_k} - \sum_{i=1}^{n}\widehat{B}_i^{(M)}(x)\frac{\partial g_i(x,0)}{\partial x_k}, \tag{22}$$

where $\widehat{A}_{ij}^{(M)}$ is the MC estimator (20) of A_{ij} and $\widehat{B}_i^{(M)}$ is the MC estimator (21) of B_i.

3.3. First discussion on the computational cost

Remember that we assume that the generation of any normal multivariate random vector is essentially cost-free. That is why we can use the crude MC method to estimate (13) and (14) (which is simply the expectation of a function of a normal multivariate random vector). It could be of interest here to use a more advanced technique, but this would not affect the computational cost defined as the number of calls of the function g.

Let us first discuss the computational cost of the estimation of the gradient of the probability $P(x)$ when using the finite-difference approximation (13).

If a full MC estimation (12) of the terms $P(x \pm \delta e_k)$ is used (or a more advanced method using the full expression of $g(x,\Lambda)$), then the method has the computational cost $2MN_x$. Indeed, we need $2N_x$ estimators for the terms $P(x \pm \delta e_k), k = 1,\dots N_x$, and each estimator has the cost M.

If we use the estimator (11) to estimate $P(x \pm \delta e_k)$, then the computational cost of the estimation of (13) is reduced to $(1+2N_\Lambda)2N_x$.

Let us now discuss the computational cost of the estimation of the gradient of the probability $P(x)$ when using the representation (14). The cost of the estimator (22) is $(1+2N_x)(1+2N_\Lambda)$: Indeed, the computational cost for the evaluations of $\Gamma(x)$ and $C(x)$ is $2N_\Lambda + 1$, the computational cost for the evaluations of the terms $(\partial C_i(x))/(\partial x_k)$ is $2N_x$, and the computational cost for the evaluations of the terms $(\partial^2 g_p(x,0))/(\partial x_k \partial \Lambda_i)$ is $4N_\Lambda N_x$ (by finite differences).

Therefore, both the evaluations of the finite-difference approximation and of the probabilistic representation of the gradient of the probability $P(x)$ has computational costs of the order of $4N_xN_\Lambda$. In many industrial examples, a cost proportional to N_xN_Λ is too heavy. In the next subsection, we show that the estimation of the gradient of $P(x)$ can be dramatically simplified if the variances σ_i^2 are small enough.

3.4. Simplified computation of the gradient

If the σ_i^2's are small, of the typical order σ^2, then the expression (14) of $(\partial P)/(\partial x_k)$ can be expanded in powers of σ^2. We can estimate the order of magnitudes of the four types of terms that appear in the sum (14). We first note that the order of magnitude of $\Gamma(x)$ is σ^2, because it is linear in K. We have

- $A_{ij}(x)$ is given by (15). It is of order σ^{-2}, because $(\partial \ln p_\Gamma)/(\partial\Gamma_{ij})$ is of order σ^{-2}.
- $(\partial\Gamma_{ij}(x))/(\partial x_k)$ is of order σ^2, because it is linear in K.
- $B_i(x)$ is given by (16). It is of order σ^{-1}, because it is proportional to $1/\sqrt{\Gamma_{ii}}$.
- $(\partial C_i(x))/(\partial x_k)$ is of order 1, because it does not depend on K.

As a consequence, the dominant term in the expression (14) of $(\partial P)/(\partial x_k)$ is

$$\frac{\partial P}{\partial x_k}(x) \simeq \sum_{i=1}^{n} B_i(x)\frac{\partial C_i(x)}{\partial x_k}. \tag{23}$$

Precise arguments are given in the Appendix A to confirm that this approximation is valid when σ is small. The simplified expression (23) can be evaluated by the MC estimator

$$\left(\widehat{\frac{\partial P}{\partial x_k}}(x)\right)^{(M)} \simeq -\sum_{i=1}^{n}\widehat{B}_i^{(M)}(x)\frac{\partial g_i(x,0)}{\partial x_k}, \tag{24}$$

where $\widehat{B}_i^{(M)}$ is the MC estimator (21) of B_i, or by any other advanced simulation method to estimate B_i. The advantage of this simplified estimator is that it requires a small number of calls of the function g, namely $1+2(N_\Lambda+N_x)$, instead of $(1+2N_x)(1+2N_\Lambda)$ for (22), because only the first-order derivatives of g with respect (x,Λ) at the point $(x,0)$ are needed. In our industrial application the computational cost is dramatically reduced. In Section 3.5, we first illustrate the accuracy of the method on a simple demonstration example, where the probability $P(x)$ and its derivatives are known analytically. In Section 3.6, we consider an academic example in the form

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

of a polynomial model (in x and in Λ). In Section 3.7, we apply the method to an industrial case.

3.5. Application to a simple demonstration model

We consider a separable example where the exact values of $P(x)$ and its gradient are known, so that the evaluation of the performance of the proposed simplified method is straightforward. We consider the case where $N_x = 2, N_\Lambda = 5, n = N_c = 5, g_p(x,\Lambda) = f_p(x) - \Lambda_p, c_p = 1$, with f_p given by

$$\begin{aligned} f_1(x) &= x_1 + x_2, \\ f_2(x) &= x_1 - x_2, \\ f_3(x) &= -x_1 + x_2, \\ f_4(x) &= -(x_1+1)^2 - (x_2+1)^2 + 2, \\ f_5(x) &= -x_1 - x_2 - 1. \end{aligned} \tag{25}$$

In the non-random case $\Lambda \equiv 0$, the admissible space is

$$\mathscr{A} = \{x \in \mathbb{R}^{N_x} : f_p(x) \leqslant c_p, p = 1, \ldots, N_c\},$$

which is the domain plotted in the left panel of Fig. 1. It is in fact the limit of the probabilistic admissible set

$$\mathscr{A}_{1-\alpha} = \{x \in \mathbb{R}^{N_x} : P(x) \geqslant 1-\alpha\}, \tag{26}$$

when the variances of the Λ_p's go to 0, whatever $\alpha \in (0,1)$.

We consider the case where the Λ_p's are iid zero-mean normal random variables with standard deviation σ_p, with $\sigma_p = \sigma = 0.3$ (Figs. 1 and 2). The figures show that the simplified MC estimator for $\nabla P(x)$ is very accurate. The exact expression of $P(x)$ is here

$$P(x) = \prod_{p=1}^{5} \Phi\left(\frac{c_p - f_p(x)}{\sigma_p}\right), \tag{27}$$

where Φ is the standard normal distribution function.

3.6. Application to a polynomial model

We consider a polynomial example in which the exact values of the derivatives of $P(x)$ are not known analytically. We consider the case where $N_x = 2, N_\Lambda = 5, n = N_c = 5, c_p = 1$, and $g(x,\Lambda)$ is given by

$$\begin{aligned} g_1(x,\Lambda) &= (1+x_1^2)\Lambda_1 - x_1\Lambda_2 + \Lambda_1^2, \\ g_2(x,\Lambda) &= x_2\Lambda_1 + (1+x_2^2)\Lambda_2 + x_1\Lambda_3 + x_1\Lambda_2^2, \\ g_3(x,\Lambda) &= -x_2\Lambda_2 + (1+2x_1^2)\Lambda_3 - (x_1+x_2)\Lambda_4 + x_2\Lambda_3^2, \\ g_4(x,\Lambda) &= (x_1+x_2)\Lambda_3 + (1+2x_2^2)\Lambda_4 + (x_2-x_1)\Lambda_5 + x_1^2\Lambda_4^2, \\ g_5(x,\Lambda) &= (x_1-x_2)\Lambda_4 + (1+x_1^2+x_2^2)\Lambda_5 + x_2^2\Lambda_5^2. \end{aligned}$$

We consider the case where the Λ_p are iid zero-mean normal random variables with standard deviation $\sigma_p \equiv \sigma$, with $\sigma = 0.1$ (resp. 0.2, 0.3) in Fig. 3 (resp. 4, 5). In Figs. 3–5, we compare the results of finite differences of full MC estimates (12) of the probability $P(x)$ (with $M = 10^5$ and 10^5 calls of the function g) with the simplified MC estimates (that require only 15 calls of the function g). The figures show that the simplified MC estimator (24) of $\nabla P(x)$ is very accurate for $\sigma = 0.1$, but the quality of the estimation is slightly reduced for $\sigma = 0.2$ and becomes rather poor for $\sigma = 0.3$. As predicted by our theory, the simplified MC estimator of the gradient is accurate when σ is small.

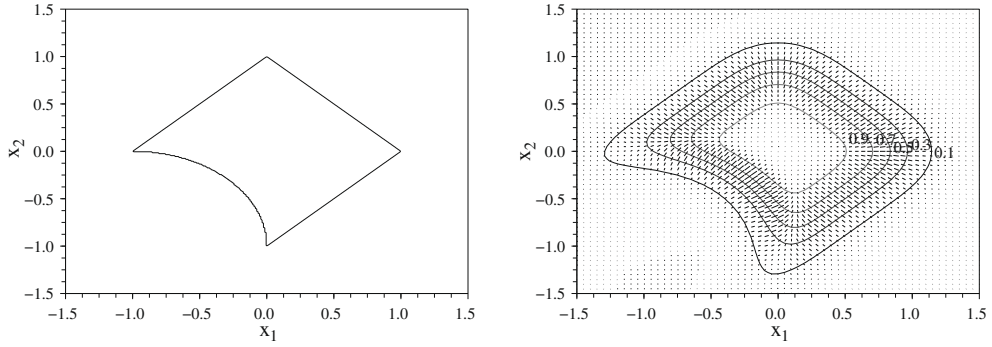

Fig. 1. Left picture: Deterministic admissible domain $\mathscr{A}$ for the separable model of Section 3.5. Right picture: The solid lines plot the level sets $P(x) = 0.1, 0.3, 0.5, 0.7$ and 0.9 of the exact expression of the probability (27). The arrows represent the gradient field $\nabla P(x)$. Here $\sigma = 0.3$.

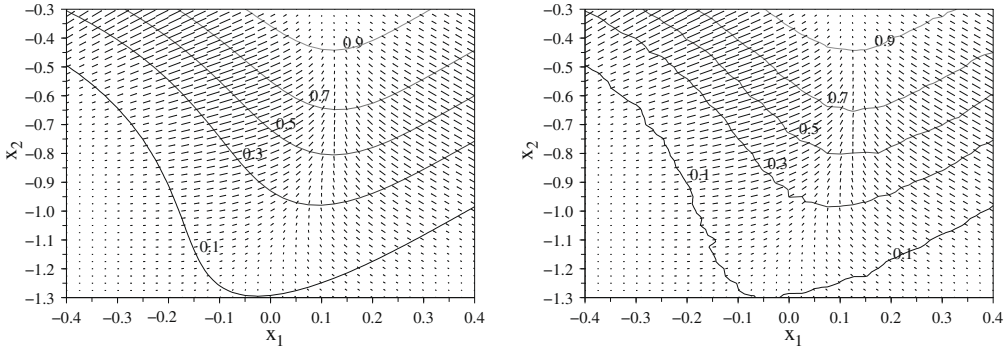

Fig. 2. The solid lines plot the level sets $P(x) = 0.1, 0.3, 0.5, 0.7$ and 0.9 of the separable model of Section 3.5 with $\sigma = 0.3$. The arrows represent the gradient field $\nabla P(x)$. Left figure: the exact expression (27) of $P(x)$ is used. Right figure: the MC estimator (11) of $P(x)$ and the simplified MC estimator (24) of $\nabla P(x)$ are used, with $M = 5000$. The error of the simplified MC estimation of $\nabla P(x)$ is approximately 2% (the maximal value of the norm of ∇P is 2.68 and the maximal absolute error is 0.051).

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

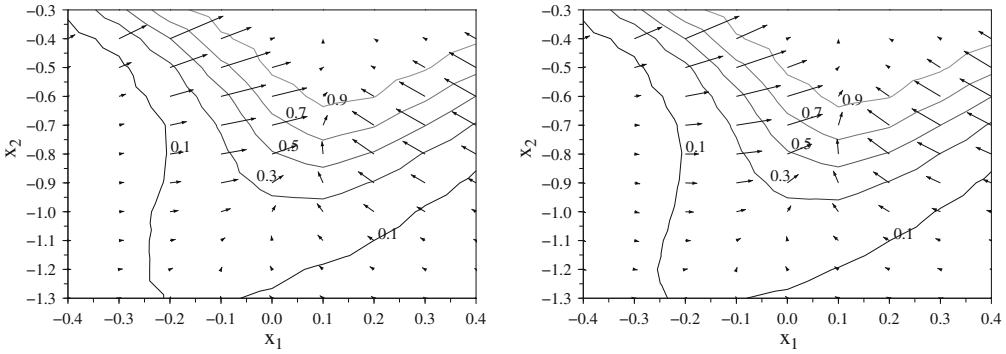

Fig. 3. The solid lines plot the level sets $P(x) = 0.1, 0.3, 0.5, 0.7$ and 0.9 of the polynomial model of Subsection 3.6. The arrows represent the gradient field $\nabla P(x)$. Left figure: the full MC estimator (12) of $P(x)$ and finite differences of the full MC estimator are used with $M = 10^5$ (and 10^5 calls of the function g). Right figure: the MC estimator (11) of $P(x)$ and the simplified MC estimator (24) of $\nabla P(x)$ are used, with $M = 5000$ (and 15 calls of the function g). The error of the simplified MC estimation of $P(x)$ is of the order of one percent (the maximal absolute error on the 9×11 grid is 0.011). The error of the simplified MC estimation of $\nabla P(x)$ is of the order of a few percent (the maximal value of the norm of ∇P is 5.8 and the maximal absolute error is 0.39).

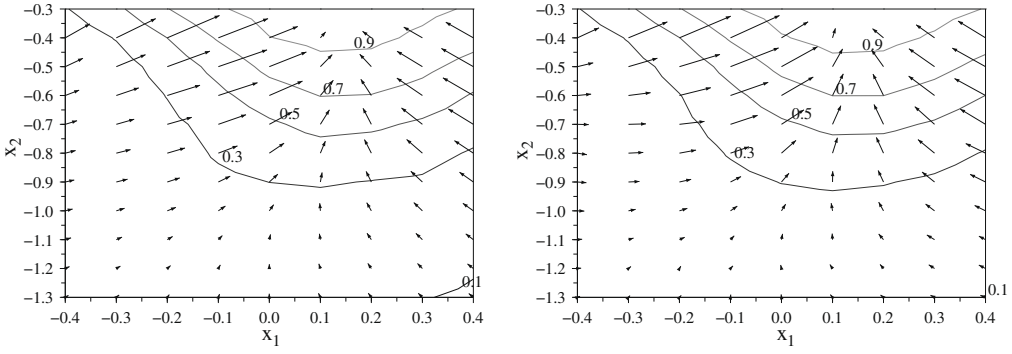

Fig. 4. Same as in Fig. 3, but $\sigma = 0.2$. The error of the simplified MC estimation of $P(x)$ is of the order of 2% (the maximal absolute error on the 9×11 grid is 0.022). The error of the simplified MC estimation of $\nabla P(x)$ is of the order of 10% (the maximal value of the norm of ∇P is 3.03 and the maximal absolute error is 0.51).

3.7. Application to an industrial case

We have applied our method to a industrial case which consists in optimizing some characteristic parameters of an aircraft at an early stage of the design process, in which many external parameters are still unknown. In this case, the function g is a big computer routine, the dimension of the input parameters is $N_x = 6$, the random parameters form a set of $N_A = 16$ independent normal random variables with means 0 and standard deviations 0.03 or 0.1, and $N_c = 6$ constraints are imposed.

More precisely, the six input parameters (on which optimization will be carried out) are associated with aircraft parameters: the sea level static reference thrust, the bypass ratio, the wing area, etc.

The 16 random variables correspond to still-unknown or variable parameters (operational empty weight, nominal landing weight, etc.). The constraints deal with the take-off performance domain (the aircraft must be able to take off at its maximum weight within a limited distance), the landing performance domain (the aircraft must be able to prepare its landing at a speed that allows the pilot to control properly its trajectory and which is supposed to allow the pilot to stop the aircraft after a distance which is no longer than its take off field length), the en-route performance domain (the aircraft must be able to start its cruise at a minimum altitude), etc. The computations of the constraints g are the computationally expensive steps of the process. We can admit that the function g is smooth (at least of class $\mathscr{C}^2$), but it is difficult to assume anything more about this black box.

It turns out that the simplified estimator of the gradient of the probability $P(x)$ has given accurate results, allowing for a standard quadratic penalization routine to be implemented. We here illustrate the accuracy of the method by considering a particular test point $x^* \in \mathbb{R}^{N_x}$. The probability $P(x^*)$ is 0.372 (evaluated with the full MC estimator (12)). We compare the estimations of the gradient $\nabla P(x^*)$ obtained on the one hand with the simplified MC method (24) and on the other hand with the full MC method (12) and the finite-difference approximation (13). We report for each method the results of three independent simulations to illustrate the dispersion of the results. The simplified MC estimator (24) for the gradient of P gives:

$$(\widehat{\nabla P(x^*)})^{(M)} = \begin{pmatrix} 6.5 \times 10^{-6} \\ 1.41 \times 10^{-2} \\ 1.27 \times 10^{-2} \\ 1.07 \times 10^{-2} \\ -9.85 \times 10^{-2} \\ 4.87 \end{pmatrix}, \quad \begin{pmatrix} 6.7 \times 10^{-6} \\ 1.43 \times 10^{-2} \\ 1.29 \times 10^{-2} \\ 1.09 \times 10^{-2} \\ -1.00 \times 10^{-1} \\ 4.90 \end{pmatrix},$$

$$\begin{pmatrix} 6.7 \times 10^{-6} \\ 1.45 \times 10^{-2} \\ 1.29 \times 10^{-2} \\ 1.10 \times 10^{-2} \\ -9.76 \times 10^{-2} \\ 5.10 \end{pmatrix},$$

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

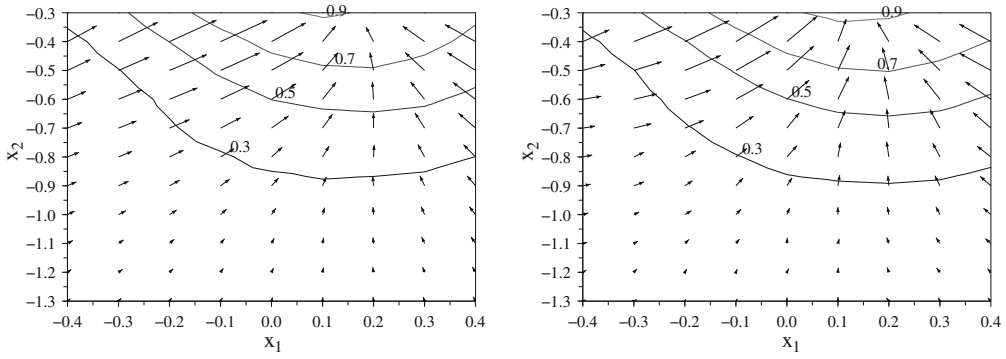

Fig. 5. Same as in Fig. 3, but $\sigma = 0.3$. The error of the simplified MC estimation of $P(x)$ is of the order of 6% (the maximal absolute error on the 9×11 grid is 0.063). The error of the simplified MC estimation of $\nabla P(x)$ is of the order of 20% (the maximal value of the norm of ∇P is 2.24 and the maximal absolute error is 0.48).

with $M = 5000$, while the full MC method (with $M = 5000$) gives:

$$(\widehat{\nabla P(x^*)})^{(M)}_{\text{fullMC}} = \begin{pmatrix} 6.6 \times 10^{-6} \\ 1.28 \times 10^{-2} \\ 1.19 \times 10^{-2} \\ 1.06 \times 10^{-2} \\ -1.19 \times 10^{-1} \\ 5.01 \end{pmatrix}, \quad \begin{pmatrix} 6.6 \times 10^{-6} \\ 1.27 \times 10^{-2} \\ 1.19 \times 10^{-2} \\ 1.04 \times 10^{-2} \\ -1.06 \times 10^{-1} \\ 4.99 \end{pmatrix},$$

$$\begin{pmatrix} 6.6 \times 10^{-6} \\ 1.22 \times 10^{-2} \\ 1.19 \times 10^{-2} \\ 1.05 \times 10^{-2} \\ -1.18 \times 10^{-1} \\ 4.97 \end{pmatrix}.$$

The difference between the simplified and full MC methods is that the first method requires one to call g exactly 45 times and takes a few minutes, while the second method requires one to call g 10,000 times and takes a few days. Besides, as we will see, it is not possible to extend the full MC method to the estimation of the Hessian of $P(x)$, because this would require a very high degree of precision in the evaluations of $P(x^* \pm \delta e_k)$, which means that a very large number of calls of the function g should be used. On the other hand, the simplified method can be extended almost cost-free to the estimation of the Hessian, as we will see in the next section.

As seen in Fig. 6, the simplified estimator (24) allows us to get in a straightforward manner the gradient field of the probability $P(x)$.

4. The Hessian of the probability $P(x)$

The Hessian of the probability $P(x)$ is needed (or can be useful) in some advanced optimization routines. From now on we assume that g is of class $\mathscr{C}^3$. Differentiating (14) with respect to x we obtain the complete expression of the Hessian. This expression contains a great many terms, including terms that involve third-order derivatives of the form $(\partial^3 g_i)(\partial \Lambda_l \partial x_k \partial x_m)$, whose evaluations by finite-differences or any other method would be very costly (about $N_\Lambda N_x^2$ calls of g). However, these terms have different orders of magnitude with respect to σ (assuming that σ is small). If we keep only the higher-order terms, we get the following simplified expression

$$\frac{\partial^2 P(x)}{\partial x_k \partial x_m} \simeq \sum_{1 \leqslant i,j \leqslant n} D_{ij}(x) \frac{\partial C_i}{\partial x_k} \frac{\partial C_j}{\partial x_m}, \tag{28}$$

for $k, m = 1, \ldots, N_x$, where D_{ij} is of order σ^{-2} and $(\partial C_i)/(\partial x_k)$ is of order 1, so that the overall expression is of order σ^{-2}. Terms of order σ^{-1} and smaller have been neglected, as we show in the next subsection.

4.1. Derivation of the simplified expression of the Hessian

To get the expression (28), we have noted that the x-derivatives of the terms in the first sum in (14):

$$\sum_{i,j=1}^{n} A_{ij}(x) \frac{\partial \Gamma_{ij}(x)}{\partial x_k}$$

are of order σ^{-1}, and the x-derivatives of the terms of the second sum

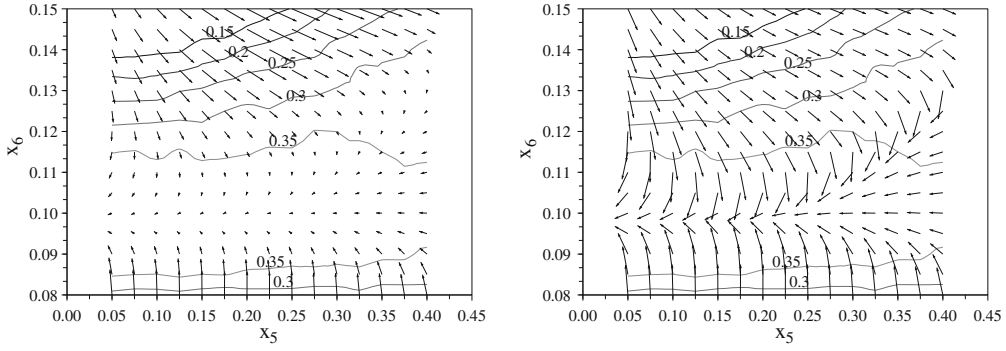

Fig. 6. The solid lines plot the level sets $P(x) = 0.15, 0.2, 0.25, 0.3$ and 0.35 of the industrial model of Section 3.7, when x_5 and x_6 are varying around the point x^*. The arrows represent the gradient field $(\partial_{x_5} P, \partial_{x_6} P)(x)$ estimated with the simplified estimator (24) with $M = 5000$. On the left figure the gradient field is plotted as it is, while we normalize its modulus on the right figure to make the gradient direction more visible.

$$\sum_{i=1}^{n} B_i(x)\frac{\partial C_i(x)}{\partial x_k}$$

have contributions of order σ^{-1}, namely

$$\sum_{i=1}^{n} B_i(x)\frac{\partial^2 C_i(x)}{\partial x_k \partial x_m},$$

and contributions of order σ^{-2}, namely

$$\sum_{i=1}^{n} \frac{\partial B_i(x)}{\partial x_m}\frac{\partial C_i(x)}{\partial x_k},$$

that give (28). Indeed:

$$\begin{aligned}\frac{\partial B_i(x)}{\partial x_m} &= \int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_n(x)}\frac{\partial p_{\Gamma(x)}(\check{z}_i, z_i = C_i(x))}{\partial x_m}d\check{z}_i \\ &+ \sum_{j\neq i}\int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_{j-1}(x)}\int_{-\infty}^{C_{j+1}(x)}\cdots\int_{-\infty}^{C_n(x)} \\ &\times p_{\Gamma(x)}(\check{z}_{i,j}, z_i = C_i(x), z_j = C_j(x))d\check{z}_{i,j}\times\frac{\partial C_j(x)}{\partial x_m},\end{aligned}$$

where $\check{z}_i = (z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_n)$ and $\check{z}_{i,j} = (z_1,\ldots,z_{i-1},z_{i+1},\ldots,z_{j-1},z_{j+1},\ldots,z_n)$. The partial derivative of the probability density can be written as

$$\begin{aligned}\frac{\partial p_{\Gamma(x)}(\check{z}_i, z_i = C_i(x))}{\partial x_m} &= \sum_{k,l=1}^{n}\left.\frac{\partial p_{\Gamma}(\check{z}_i, z_i = C_i(x))}{\partial \Gamma_{kl}}\right|_{\Gamma=\Gamma(x)}\frac{\partial \Gamma_{kl}(x)}{\partial x_m} \\ &+ \left.\frac{\partial p_{\Gamma(x)}(\check{z}_i, z_i)}{\partial z_i}\right|_{z_i=C_i(x)}\frac{\partial C_i(x)}{\partial x_m}.\end{aligned}$$

The first sum is of order p_Γ because $(\partial p_\Gamma)/(\partial\Gamma_{kl})$ is of order $\sigma^{-2}p_\Gamma$ (see (19)) while $(\partial\Gamma_{kl})/(\partial x_m)$ is of order σ^2.

The second sum is of order $\sigma^{-1}p_\Gamma$ because $(\partial C_i)/(\partial x_m)$ is of order 1 while $(\partial p_\Gamma)/(\partial z_i)$ is of order $\sigma^{-1}p_\Gamma$:

$$\frac{\partial \ln p_\Gamma(z)}{\partial z_i} = -\frac{1}{2}\frac{\partial}{\partial z_i}(z^t\Gamma^{-1}z) = -\frac{(z^t\Gamma^{-1})_i + (\Gamma^{-1}z)_i}{2} = -(\Gamma^{-1}z)_i.$$

We only keep the second sum in the simplified expression:

$$\begin{aligned}\frac{\partial B_i(x)}{\partial x_m} &\simeq \int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_n(x)}\times\frac{\partial \ln p_{\Gamma(x)}}{\partial z_i}(\check{z}_i, z_i = C_i(x)) \\ &\times p_{\Gamma(x)}(\check{z}_i, z_i = C_i(x))d\check{z}_i\times\frac{\partial C_i(x)}{\partial x_m} \\ &+ \sum_{j\neq i}\int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_{j-1}(x)}\int_{-\infty}^{C_{j+1}(x)}\cdots\int_{-\infty}^{C_n(x)} \\ &\times p_{\Gamma(x)}(\check{z}_{i,j}, z_i = C_i(x), z_j = C_j(x))d\check{z}_{i,j}\times\frac{\partial C_j(x)}{\partial x_m}.\end{aligned}$$

The explicit expressions of $D_{ij}(x)$ in (28) are the following ones. When $i<j$:

$$\begin{aligned}D_{ij}(x) &= \frac{1}{2\pi\sqrt{\Gamma_{ii}\Gamma_{jj}-\Gamma_{ij}^2}} \\ &\times\exp\left(-\frac{1}{2}\begin{pmatrix}C_i(x)\\C_j(x)\end{pmatrix}^t\begin{pmatrix}\Gamma_{ii}&\Gamma_{ij}\\\Gamma_{ij}&\Gamma_{jj}\end{pmatrix}^{-1}\begin{pmatrix}C_i(x)\\C_j(x)\end{pmatrix}\right) \\ &\times\int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_{j-1}(x)}\int_{-\infty}^{C_{j+1}(x)}\cdots\int_{-\infty}^{C_n(x)} \\ &\times p_{\Gamma(x)}(z''|z_i = C_i(x), z_j = C_j(x))d^{n-2}z''.\end{aligned} \tag{29}$$

Here $p_\Gamma(z''|z_i,z_j)$ is the conditional density of the vector

$$Z'' = (Z_1,\ldots,Z_{i-1},Z_{i+1},\ldots,Z_{j-1},Z_{j+1},\ldots,Z_n),$$

given $(Z_i,Z_j) = (z_i,z_j)$. Therefore, $p_{\Gamma(x)}(z''|z_i = C_i(x), z_j = C_j(x))$ is the density of the $(n-2)$-dimensional normal distribution with mean $\tilde{\mu}^{(ij)}(x)$

$$\tilde{\mu}_k^{(ij)}(x) = \begin{pmatrix}\Gamma_{ki}(x)\\\Gamma_{kj}(x)\end{pmatrix}^t\begin{pmatrix}\Gamma_{ii}(x)&\Gamma_{ij}(x)\\\Gamma_{ij}(x)&\Gamma_{jj}(x)\end{pmatrix}^{-1}\begin{pmatrix}C_i(x)\\C_j(x)\end{pmatrix}, \tag{30}$$

$k = 1,\ldots,i-1,i+1,\ldots,j-1,j+1,\ldots,n$, and covariance matrix $\tilde{\Gamma}^{(ij)}(x)$

$$\tilde{\Gamma}_{kl}^{(ij)}(x) = \Gamma_{kl}(x) - \begin{pmatrix}\Gamma_{ki}(x)\\\Gamma_{kj}(x)\end{pmatrix}^t\begin{pmatrix}\Gamma_{ii}(x)&\Gamma_{ij}(x)\\\Gamma_{ij}(x)&\Gamma_{jj}(x)\end{pmatrix}^{-1}\begin{pmatrix}\Gamma_{li}(x)\\\Gamma_{lj}(x)\end{pmatrix}, \tag{31}$$

$k,l = 1,\ldots,i-1,i+1,\ldots,j-1,j+1,\ldots,n$.

When $i=j$:

$$\begin{aligned}D_{ii}(x) &= -\frac{1}{\sqrt{2\pi\Gamma_{ii}}}\exp\left(-\frac{C_i(x)^2}{2\Gamma_{ii}}\right)\int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_n(x)} \\ &\times p_{\Gamma(x)}(z'|z_i = C_i(x))((\Gamma^{-1}z)_i)_{z_i=C_i(x)}d^{n-1}z'.\end{aligned}$$

The expression of $D_{ii}(x)$ can also be written as

$$\begin{aligned}D_{ii}(x) &= -(\Gamma^{-1})_{ii}C_i(x)B_i(x) - \frac{1}{\sqrt{2\pi\Gamma_{ii}}} \\ &\times\exp\left(-\frac{C_i(x)^2}{2\Gamma_{ii}}\right)\int_{-\infty}^{C_1(x)}\cdots\int_{-\infty}^{C_{i-1}(x)}\int_{-\infty}^{C_{i+1}(x)}\cdots\int_{-\infty}^{C_n(x)} \\ &\times p_{\Gamma(x)}(z'|z_i = C_i(x))\left(\sum_{k\neq i}(\Gamma^{-1})_{ik}z'_k\right)d^{n-1}z'.\end{aligned} \tag{32}$$

Here $p_\Gamma(z'|z_i = C_i(x))$ is the density of the $(n-1)$-dimensional normal random vector with mean $\tilde{\mu}^{(i)}(x)$ and covariance matrix $\tilde{\Gamma}^{(i)}(x)$ given by (17) and (18).

4.2. Monte Carlo estimation

Remember that the generation of any normal multivariate random vector is essentially cost-free in our setting. Therefore, we use the crude MC method to estimate (28), which is simply the expectation of a function of a normal multivariate random vector. Once again, it could be of interest here to use a more advanced simulation technique than the crude MC method, but this does not affect the computational cost which is defined as the number of calls of the function g.

The MC estimator for $D_{ij}(x), i<j$, is

$$\begin{aligned}\widehat{D}_{ij}^{(M)}(x) &= \frac{1}{2\pi\sqrt{\Gamma_{ii}\Gamma_{jj}(x)-\Gamma_{ij}^2(x)}} \\ &\times\exp\left(-\frac{1}{2}\begin{pmatrix}C_i(x)\\C_j(x)\end{pmatrix}^t\begin{pmatrix}\Gamma_{ii}(x)&\Gamma_{ij}(x)\\\Gamma_{ij}(x)&\Gamma_{jj}(x)\end{pmatrix}^{-1}\begin{pmatrix}C_i(x)\\C_j(x)\end{pmatrix}\right) \\ &\times\frac{1}{M}\sum_{l=1}^{M}\left[\prod_{k=1\neq i,j}^{n}\mathbf{1}_{(-\infty,C_k(x)]}(Z_k''^{(l)})\right],\end{aligned} \tag{33}$$

where $(Z_1''^{(l)},\ldots,Z_{i-1}''^{(l)},Z_{i+1}''^{(l)},\ldots,Z_{j-1}''^{(l)},Z_{j+1}''^{(l)},\ldots,Z_n''^{(l)}), l = 1,\ldots,M$ is an iid sequence of $(n-2)$-dimensional normal random vectors with mean $\tilde{\mu}^{(ij)}(x)$ and covariance matrix $\tilde{\Gamma}^{(ij)}(x)$ given by (30) and (31).

The MC estimator for $D_{ii}(x)$ is

$$\begin{aligned}\widehat{D}_{ii}^{(M)}(x) &= -(\Gamma(x)^{-1})_{ii}C_i(x)\widehat{B}_i^{(M)}(x) - \frac{1}{\sqrt{2\pi\Gamma_{ii}(x)}}\exp\left(-\frac{C_i(x)^2}{2\Gamma_{ii}(x)}\right) \\ &\times\frac{1}{M}\sum_{l=1}^{M}\left[\left(\sum_{k\neq i}(\Gamma(x)^{-1})_{ik}Z_k'^{(l)}\right)\prod_{k=1\neq i}^{n}\mathbf{1}_{(-\infty,C_k(x)]}(Z_k'^{(l)})\right],\end{aligned}$$

where $(Z_1'^{(l)},\ldots,Z_{i-1}'^{(l)},Z_{i+1}'^{(l)},\ldots,Z_n'^{(l)}), l = 1,\ldots,M$ is an iid sequence of $(n-1)$-dimensional normal random vectors with mean $\tilde{\mu}^{(i)}(x)$

given by (17) and covariance matrix $\widetilde{\Gamma}^{(i)}(x)$ given by (18). The same sequence as the one used to construct the estimator $\widehat{B}_i^{(M)}$ by (21) can be used here for the estimator $\widehat{D}_{ii}^{(M)}$. As a result, the MC estimator for the Hessian of $P(x)$ is

$$\left(\widehat{\frac{\partial^2 P(x)}{\partial x_k \partial x_m}}\right)^{(M)} = \sum_{1 \leqslant i \leqslant j \leqslant n} \widehat{D}_{ij}^{(M)}(x) \frac{\partial C_i}{\partial x_k} \frac{\partial C_j}{\partial x_m}. \tag{34}$$

To summarize, the estimation of the (simplified) Hessian (28) by the MC estimator (34) does not require any additional call of the function g (compared to the estimation of the gradient), since only C, the gradient of C, and the matrix Γ are needed.

4.3. Numerical illustrations

We illustrate the results with the separable model introduced in Section 3.5. Figs. 7 and 8 show that the simplified MC estimator for the Hessian of $P(x)$ is very accurate. This means that, with $1 + 2(N_x + N_\Lambda)$ calls of the function g, we can get accurate estimates of $P(x)$, its gradient and its Hessian.

Finally, we have implemented the MC estimation of the Hessian of the probability $P(x)$ to the industrial case described in Section 3.7. However, we cannot compare with the full MC method, because the computational cost of the full MC method to compute the Hessian is prohibitive. In Fig. 9 we plot the diagonal Hessian field $(\partial^2_{x_5} P, \partial^2_{x_6} P)(x)$ to show that the method gives what seems to be acceptable results.

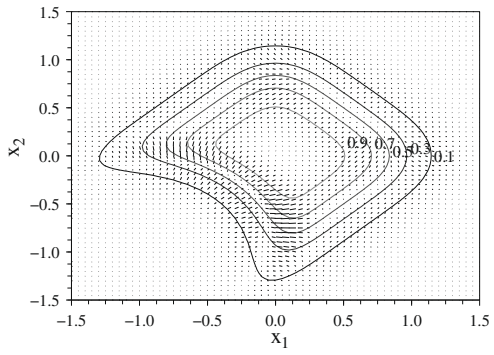

Fig. 7. The solid lines plot the level sets $P(x) = 0.1$, 0.3, 0.5, 0.7 and 0.9 of the separable model of Section 3.5 with $\sigma = 0.3$. The arrows represent the diagonal Hessian field $(\partial^2_{x_1} P, \partial^2_{x_2} P)(x)$. Here the exact expression (27) of $P(x)$ is used.

5. Application to stochastic optimization

The goal is to solve

$$\min_{x \in \mathbb{R}^{N_x}} J(x) \text{ s.t.} \left\{ \begin{array}{l} P(x) \geqslant 1 - \alpha \text{ and} \\ g_p(x) \leqslant c_p, p = n+1, \ldots, N_c \end{array} \right\}. \tag{35}$$

The previous sections show how to compute efficiently $P(x)$, its gradient and its Hessian. Therefore, the problem has been reduced to a standard constrained optimization problem. Different techniques have been proposed for solving constrained optimization problems: reduced–gradient methods, sequential linear and quadratic programming methods, and methods based on augmented Lagrangians and exact penalty functions. Fletcher (1987) discusses constrained optimization theory and sequential quadratic programming. Gill et al. (1981) discuss reduced–gradient methods. Bertsekas (1982) presents augmented Lagrangian algorithms.

Penalization techniques are well-adapted to stochastic optimization (Wang and Spall, 2003). The basic idea of the penalization approach is to transform the constrained optimization problem (35)into an unconstrained optimization problem of the form: minimize the function $\widetilde{J}$ defined by

$$\widetilde{J}(x) := J(x) + \rho Q(x),$$

where $Q : \mathbb{R}^{N_x} \to \mathbb{R}$ is the penalty function and ρ is a positive real number, referred to as the penalty parameter. The qth order penalization ($q \geqslant 2$) uses the penalty function (Ruszczyński, 2006)

$$Q(x) = \frac{1}{q} \sum_{k=1}^{K} \max\{0, q_k(x)\}^q,$$

which is continuously differentiable with

$$\nabla Q(x) = \sum_{k=1}^{K} \max\{0, q_k(x)\}^{q-1} \nabla q_k(x).$$

In our case, we have $K = N_c - n + 1, q_1(x) = 1 - \alpha - P(x)$, and $q_k(x) = g_{k+n-1}(x) - c_{k+n-1}, k = 2, \ldots, K$. The unconstrained problem obtained by penalization can be solved using a gradient method optimization. At each iteration k, $k \geqslant 1$, a descent direction d_k of the cost function $\widetilde{J}$ at the current state x_k is determined. This direction has the form $d_k = -W_k g_k$, where W_k is the current approximation of the inverse Hessian of the cost function at x_k and g_k is the gradient of the cost function at x_k. The iteration formula has the form $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a step-size determined along the direction d_k by Wolfe conditions (Nocedal and Wright, 1999).

Example. We minimize the function $J(x) = (x_1 - 1)^2 + 4x_2^2 + 4$ subject to the constraints described in Section 3.5, where the Λ_p are independent normal random variables with means 0 and stan-

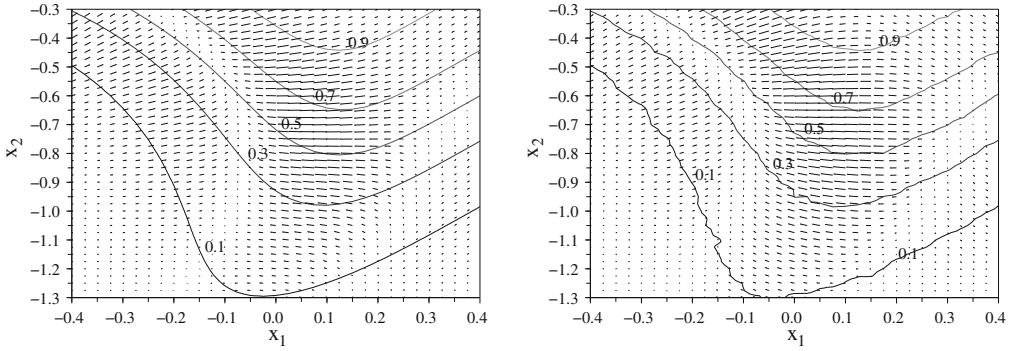

Fig. 8. The solid lines plot the level sets $P(x) = 0.1$, 0.3, 0.5, 0.7 and 0.9 of the separable model of Section 3.5 with $\sigma = 0.3$. The arrows represent the diagonal Hessian field $(\partial^2_{x_1} P, \partial^2_{x_2} P)(x)$. Left figure: the exact expression (27) of $P(x)$ is used. Right figure: the MC estimator (11) of $P(x)$ and the simplified MC estimator (34) of $(\partial^2_{x_1} P, \partial^2_{x_2} P)(x)$ are used, with $M = 5000$ and 15 calls of the function g.

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026

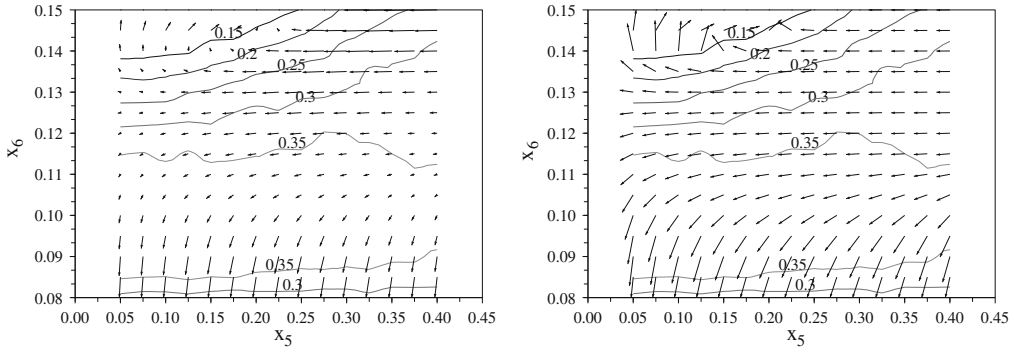

Fig. 9. The solid lines plot the level sets $P(x) = 0.15, 0.2, 0.25, 0.3$ and 0.35 of the model of Section 3.7, when x_5 and x_6 are varying around the point x^*. The arrows represent the diagonal Hessian field $(\hat{c}^2_{x_5}P, \hat{c}^2_{x_6}P)(x)$ estimated with the simplified estimator (34) with $M = 5000$. On the left figure the diagonal Hessian field is plotted as it is, while we normalize its modulus on the right figure to make the direction more visible.

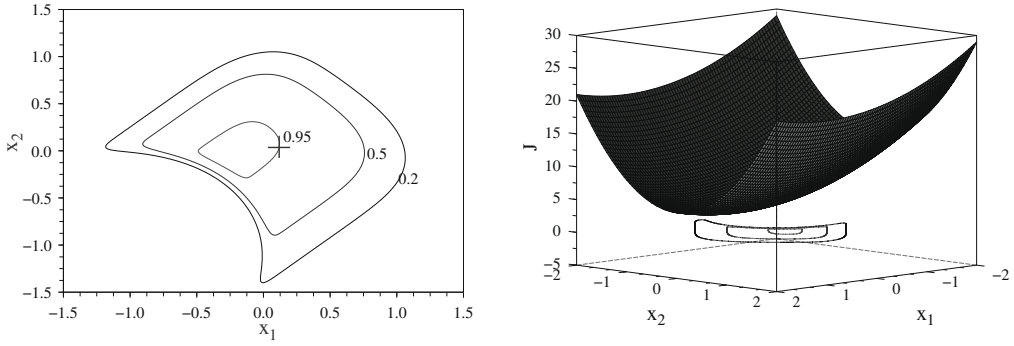

Fig. 10. Minimization of J for the probability level $1 - \alpha = 0.95$. The left picture shows that the minimum is obtained at the level set 0.95. The right picture plots the cost function and level set curves.

dard deviations $0.1p, p = 1, \ldots, 5$. Fig. 10 shows the result of the optimization routine with quadratic penalization.

6. Conclusion

The original contribution of this paper consists of a rapid and efficient method to estimate the gradient and Hessian of the probability

$$P(x) = \mathbb{P}(g_p(x, \Lambda) \leqslant 0, p = 1, \ldots, N_c).$$

This method requires a very small number of calls of the constraint functions g_p, which makes it possible to implement an optimization routine for the chance constrained problem. The method is based on (1) particular probabilistic representations of the gradient and Hessian of $P(x)$ which can be obtained when the random vector Λ has multivariate normal distribution and (2) asymptotic expansions of these representations when the variances of the input random parameters Λ_i are small. Straightforward MC estimators can then be used for the estimations of these asymptotic expressions. More advanced methods could be used for the estimation step, which are available in the literature since the asymptotic expressions can be expressed as expectations of functions of multivariate normal vectors.

It should be possible to extend the results to cases in which the conditional distributions of the vector Λ given one or two entries are explicitly known. The idea would be to use the general formulas of Kibzun and Uryasev (1998) for the gradients and to expand these formulas for small variances.

The estimation method presented in this paper could also be applied to the optimization of a random parameter, which is another type of stochastic problem. Here the constraints are assumed to be known, as well as the cost function, but the input vector $x \in \mathbb{R}^{N_x}$ is known only approximately. This problem models the production of an object by a non-perfect machine and takes into account the unavoidable imperfections of the production process. In this case the probabilistic constraint has the form $P(x) \geqslant 1 - \alpha$ with

$$P(x) = \mathbb{P}(f_p(X^{(x)}) \leqslant 0, p = 1, \ldots, N_c), \tag{36}$$

where $f_p, p = 1, \ldots, N_c$ are constraint functions, $X^{(x)}$ is a random vector with distribution function $\Psi(\cdot - x)$:

$$\mathbb{P}(X_p^{(x)} \leqslant c_p, p = 1, \ldots, N_x) = \Psi(c_1 - x_1, \ldots, c_{N_x} - x_{N_x}),$$

and Ψ is the distribution function of a zero-mean random vector that models the fluctuations of the production process. This probabilistic constraint has the same form as (6), since it is possible to write $X^{(x)} = x + \Lambda$ with Λ a random vector with distribution function Ψ, and then the probability (36) has the form

$$P(x) = \mathbb{P}(g_p(x, \Lambda) \leqslant 0, p = 1, \ldots, N_c), \text{ with } g_p(x, \Lambda) = f_p(x + \Lambda).$$

If Ψ is the distribution function of the multivariate normal distribution with mean 0 and covariance matrix K and if the variations of f_p can be linearized over hypercubes of the form $\prod_{i=1}^{N_x}[x_i - 3\sigma_i, x_i + 3\sigma_i]$, then we can apply the methodology described in the paper.

Acknowledgements

This work was initiated during the Summer Mathematical Research Center on Scientific Computing (whose french acronym is CEMRACS), which took place in Marseille in july and august 2006. This project was initiated and supported by the European Aeronau-

tic Defence and Space (EADS) company. In particular, we thank R. Lebrun and F. Mangeant for stimulating and useful discussions.

Appendix A. Complements on the simplified expression of the gradient

In Section 3.4 we gave a presentation of the simplified expression (23) of the gradient based on order-of-magnitude considerations. In this Appendix A, we provide rigorous estimates that confirm this approximation.

We assume that the covariance matrix of Λ is of the form $K = \sigma^2 K_0$ with $\sigma \ll 1$. Consequently, the matrix $\Gamma(x)$ defined by (9) is also of the form $\Gamma(x) = \sigma^2 \Gamma_0(x)$.

Estimate for the first sum in (14): We have

$$A_{ij}(x) = -\frac{1}{2}(\Gamma^{-1})_{ji} P(x) + \frac{1}{2}\int (\Gamma^{-1}z)_i (\Gamma^{-1}z)_j \prod_{k=1}^{n} \mathbf{1}_{(-\infty, C_k]}(z_k) p_\Gamma(z) d^n z.$$

Taking the absolute values

$$|A_{ij}(x)| \leqslant \frac{1}{2}|(\Gamma^{-1})_{ji}| + \frac{1}{4}\int [(\Gamma^{-1}z)_i]^2 p_\Gamma(z) d^n z + \frac{1}{4}\int [(\Gamma^{-1}z)_j]^2 p_\Gamma(z) d^n z,$$

and computing the last two terms of the right side gives

$$|A_{ij}(x)| \leqslant \frac{1}{2}|(\Gamma^{-1})_{ji}| + \frac{1}{4}(\Gamma^{-1})_{ii} + \frac{1}{4}(\Gamma^{-1})_{jj}.$$

Therefore, $A_{ij}(x)$ is smaller than σ^{-2}:

$$|A_{ij}(x)| \leqslant \left[\frac{1}{2}|(\Gamma_0^{-1})_{ji}| + \frac{1}{4}(\Gamma_0^{-1})_{ii} + \frac{1}{4}(\Gamma_0^{-1})_{jj}\right]\frac{1}{\sigma^2},$$

and

$$\left|\sum_{ij} A_{ij}(x)\frac{\partial \Gamma_{ij}(x)}{\partial x_k}\right| \leqslant \sum_{ij}\left[\frac{1}{2}|(\Gamma_0^{-1})_{ji}| + \frac{1}{4}(\Gamma_0^{-1})_{ii} + \frac{1}{4}(\Gamma_0^{-1})_{jj}\right]\left|\frac{\partial \Gamma_{0ij}(x)}{\partial x_k}\right|,$$

with the right side independent of σ.

Estimate for the second sum in (14): Let us assume that the deterministic admissible set

$$\mathscr{A} = \{x \in \mathbb{R}^{N_x} : C_p(x) \geqslant 0, p = 1, \ldots, n\}$$

is not empty and that its complementary set is non-empty as well. This is the typical situation in chance constrained programming. This hypothesis means that, when the uncertainty is reduced to zero, the admissible set is non-empty, which implies that the constrained optimization problem is well posed, and the admissible set does not extend to $\mathbb{R}^{N_x}$, which implies that the constrained optimization problem is really constrained (note: as we show below, if $\mathscr{A} = \emptyset$, then $P(x) \to 0$ as $\sigma \to 0$ uniformly on the compact sets of $\mathbb{R}^{N_x}$, which is not a situation interesting in chance constrained programming).

Let us consider a point x^* that belongs to the boundary of $\mathscr{A}$. For such a point, there exists i such that $C_i(x^*) = 0$ and $C_j(x^*) \geqslant 0$ for all other j. Therefore, we have

$$B_i(x^*) = \frac{1}{\sqrt{2\pi\Gamma_{ii}}}\int_{-\infty}^{C_1(x^*)} \cdots \int_{-\infty}^{C_{i-1}(x^*)} \int_{-\infty}^{C_{i+1}(x^*)} \cdots \int_{-\infty}^{C_n(x^*)} p_{\tilde{\Gamma}}(z') d^{n-1} z' \geqslant \frac{1}{\sqrt{2\pi\Gamma_{ii}}}\int_{-\infty}^{0} \cdots \int_{-\infty}^{0} p_{\tilde{\Gamma}}(z') d^{n-1} z',$$

with $\tilde{\Gamma}$ given by (18). $\tilde{\Gamma}$ is of the form $\tilde{\Gamma} = \sigma^2 \tilde{\Gamma}_0$ with $\tilde{\Gamma}_0$ independent of σ^2, so that

$$B_i(x^*) \geqslant \left[\frac{1}{\sqrt{2\pi\Gamma_{0ii}}}\int_{-\infty}^{0} \cdots \int_{-\infty}^{0} p_{\tilde{\Gamma}_0}(z') d^{n-1} z'\right]\frac{1}{\sigma}.$$

This shows that $B_i(x)$ takes values which are typically of order σ^{-1}. Since $\partial C_i(x)/\partial x_k$ does not depend on σ, this also shows that the sum

$$\sum_{i=1}^{n} B_i(x)\frac{\partial C_i(x)}{\partial x_k}$$

takes values of order σ^{-1}.

Remark: Let us briefly consider the situation in which $\mathscr{A} = \emptyset$. Let $R > 0$ and denote by B_R the ball of $\mathbb{R}^{N_x}$ with center at 0 and radius R. We introduce

$$c_R = \sup_{x \in B_R} \inf_{p=1,\ldots,n} C_p(x), \quad \gamma_R = \sup_{x \in B_R} \sup_{p=1,\ldots,n} \Gamma_{0pp}(x).$$

The number c_R is negative, since it is the supremum of a negative-valued function over a compact set. The number γ_R is positive and finite. We then have

$$\sup_{x \in B_R} P(x) \leqslant \sup_{x \in B_R} \inf_{p=1,\ldots,n} \Phi\left(\frac{C_p(x)}{\sqrt{\Gamma_{pp}(x)}}\right) \leqslant \Phi\left(\frac{c_R}{\sigma\sqrt{\gamma_R}}\right),$$

where Φ is the distribution function of the standard normal distribution. As $\sigma \to 0$, the right side of this inequality goes to zero. This calculation proves the assertion that, if $\mathscr{A} = \emptyset$, then $P(x) \to 0$ as $\sigma \to 0$ uniformly on the compact sets of $\mathbb{R}^{N_x}$.

References

Bertsekas, D.P., 1982. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York.
Birge, J.R., 1997. Stochastic programming computation and applications. INFORMS Journal on Computing 9, 111–133.
Birge, J.R., Louveaux, F., 1997. Introduction to Stochastic Programming. Springer, New York.
Bukszár, J., Prékopa, A., 2001. Probability bounds with cherry trees. Mathematics of Operations Research 26, 174–192.
Calafiore, G., Dabbene, F., 2006. Probabilistic and Randomized Methods for Design Under Uncertainty. Springer, London.
Deák, I., 1986. Computing probabilities of rectangles in case of multinormal distribution. Journal of Statistical Computation and Simulation 26, 101–114.
Dubi, A., 2000. Monte Carlo Applications in Systems Engineering. Wiley, Chichester.
Fletcher, R., 1987. Practical Methods of Optimization. Wiley, New York.
Gassmann, H.I., Deák, I., Szántai, T., 2002. Computing multivariate normal probabilities: A new look. Journal of Computational and Graphical Statistics 11, 920–949.
Genz, A., 1992. Numerical computation of multivariate normal probabilities. Journal of Computational and Graphical Statistics 1, 141–149.
Gill, P.E., Murray, W., Wright, M.H., 1981. Practical Optimization. Academic Press, New York.
Henrion, R., Römisch, W., 2000. Stability of solutions of chance constrained stochastic programs. In: Guddat, J., Hirabayashi, R., Jongen, HTh., Twilt, F. (Eds.), Parametric Optimization and Related Topics V, Peter Lang, Frankfurt, pp. 95–114.
Henrion, R., Strugarek, C., 2008. Convexity of chance constraints with independent random variables. Computational Optimization and Applications 41, 263–276.
Kall, P., Wallace, S.W., 1994. Stochastic Programming. Wiley, Chichester.
Kibzun, A., Uryasev, S., 1998. Differentiability of probability function. Stochastic Analysis and Applications 16, 1101–1128.
Nocedal, J., Wright, S.J., 1999. Numerical Optimization. Springer–Verlag, New York.
Prékopa, A., 1971. Logarithmic concave measures with application to stochastic programming. Acta Scientiarum Mathematicarum 32, 301–316.
Prékopa, A., 1990. Sharp bounds on probabilities using linear programming. Operations Research 38, 227–239.
Prékopa, A., 1995. Stochastic Programming. Kluwer, Dordrecht.
Ruszczyński, A., 2006. Nonlinear Optimization. Princeton University Press, Princeton.
Ruszczyński, A., Shapiro, A., 2003. Stochastic programming. Handbooks in Operations Research and Management Science, vol. 10. Elsevier, Amsterdam.
Schervish, M.J., 1984. Multivariate normal probabilities with error bound. Applied Statistician 39, 255–257.
Sen, S., Higle, J.L., 1999. An introductory tutorial on stochastic linear programming models. Interfaces 29, 33–61.
Szántai, T., 2001. Improved bounds and simulation procedures on the value of the multivariate normal probability distribution function. Annals of Operations Research 100, 85–101.
Wang, I.J., Spall, J.C., 2003. Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions. In: Proceedings of IEEE Conference on Decision and Control, Maui, HI, pp. 3808–3813.

Please cite this article in press as: Garnier, J. et al., Asymptotic formulas for the derivatives of probability functions ..., European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.09.026