2004

Reliability-based optimal design using sample average approximations

Royset, J.O.

Elsevier Ltd.

þÿProbabilistic Engineering Mechanics 19 (2004) 331 343

http://hdl.handle.net/10945/48694

ELSEVIER

Probabilistic Engineering Mechanics 19 (2004) 331–343

PROBABILISTIC
ENGINEERING
MECHANICS

www.elsevier.com/locate/probengmech

Reliability-based optimal design using sample average approximations

J.O. Royset[a,*], E. Polak[b]

[a]Department of Operations Research, Naval Postgraduate School, 239 Glasgow Hall, Monterey, CA 93943, USA
[b]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 231 Cory Hall, Berkeley, CA 94720, USA

Received 30 June 2003; revised 19 March 2004; accepted 31 March 2004

Abstract

An algorithm for reliability-based optimal design is developed using sampling techniques for estimating the failure probability. The algorithm applies a new method for sensitivity calculations of the failure probability. Initially, the estimates of the failure probability are coarse. As the algorithm progresses towards an optimal design, the number of sample points is increased in an adaptive way leading to better estimates of the failure probability. The algorithm is proven to converge to an optimal design. The applicability of the algorithm is shown in an example from the area of highway bridge design.

Keywords: Reliability-based optimal design; Sample average approximations; Monte-Carlo simulation; Bridge design

1. Introduction

In this paper, we discuss problems arising in reliability-based optimal design of structures. Such problems are characterized by the presence of functions representing the failure probabilities of one or more structural systems and their components. Reliability-based optimal design is computationally, but also theoretically, challenging. A particular source of difficulty is the construction of approximating expressions for the failure probability that can be used in conjunction with some optimization algorithm.

In the areas of applied mathematics and operations research, there is a large literature dealing with various optimization problems arising in decision making under uncertainty. Such problems are referred to as stochastic optimization problems. Two techniques for solving reliability-based optimal design problems, as well as more general stochastic optimization problems, are stochastic quasi-gradient methods [3,7,10,27,43] and sample average approximations [13,14,16,17,25,38,39].

Stochastic quasi-gradient methods employ various approximation techniques to compute search directions in an iterative scheme for finding an optimal design. These methods are not directly applicable to problems involving failure probability constraints, but they can handle problems with failure probabilities in the objective function. For deterministic constraints, the implementation of the stochastic quasi gradient methods may require numerically costly approximations to such operations as gradient projection.

A sample average approximation problem is constructed by replacing the failure probabilities in the original reliability-based optimal design problem by corresponding Monte Carlo sampling estimates. The results associated with such approximations give asymptotic properties of minimizers of sample average approximation problems as the number of samples goes to infinity, and error estimates for finite sample sizes. These results provide guidance for the selection of one or more approximation problems to be solved using some optimization algorithm. Using sample average approximations, Royset and Polak [36] develop the theoretical basis for a new implementable algorithm for reliability-based optimal design. The algorithm will be discussed in detail below.

Stochastic quasi-gradient methods and sample average approximations have been used rarely for solving reliability-based optimal design problems arising in engineering, and specialized approaches have been developed. These approaches include the use of response surface techniques [9], surrogate functions [42], stochastic linear programming [26], first-order approximations to the failure

[*] Corresponding author. Tel.: +1-510-282-7986; fax: +1-831-656-2595.
E-mail address: joroyset@nps.navy.mil (J.O. Royset).

0266-8920/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.probengmech.2004.03.001

probability [6,19,35], and the use of probabilistic models resulting in manageable expressions for the failure probability [29,30]. In the specialized approaches, the resulting optimization problems are typically solved using standard linear or nonlinear optimization algorithms. An exception to this trend is Ref. [35], which solves semi-infinite optimization problems. These specialized approaches may work satisfactorily under certain conditions, but are not proved to converge to a solution of the original design problem. For example, the approach in Ref. [35] is known to converge to a solution of a first-order approximation of the reliability-based optimal design problem, but the scheme for improving the first-order solution is based on heuristics.

Engineering efforts to use gradient-free optimization techniques can be found in Refs. [2,4,15,31]. These techniques tend to converge rather slowly, particularly in the case of many design variables. Studies focusing on applications of reliability-based optimal design techniques include Refs. [20–22,41]. See Ref. [37] for a more comprehensive review of the literature.

In view of the rapid development of computers, the high computational cost traditionally associated with sample average approximations may diminish. Hence, it appears that sample average approximations, with its potential for high-accuracy solutions of reliability-based optimal design problems, may prove to be an attractive alternative to existing specialized approaches in engineering applications. In this paper, we discuss the application of the new algorithm constructed in Ref. [36], which uses a sampling technique to estimate the failure probability and incorporates such estimates with a standard nonlinear optimization algorithm. The algorithm is illustrated with an example from highway bridge design. We also present a new sensitivity formula for the failure probability, which is of importance in structural reliability analysis.

2. Definition of failure probability

In accordance with common practice [5], we express the failure probability of a structure by means of a time-invariant probabilistic model defined in terms of an m-dimensional vector $\mathbf{V}$ of random variables. Let $\mathbf{x}$ be an n-dimensional vector of deterministic design variables, e.g. member sizes, amount of steel reinforcement, or parameters in the distribution of $\mathbf{V}$. Failure of the structure is defined in terms of K limit-state functions $G_k(\mathbf{x},\mathbf{v})$, $k \in \mathbf{K} = \{1,2,\ldots,K\}$, where $\mathbf{v}$ is a realization of the random vector $\mathbf{V}$. The limit-state functions $G_k(\mathbf{x},\mathbf{v})$ describe the performance of the structure with respect to specific requirements.

It is theoretically and computationally convenient to introduce a bijective transformation of realizations $\mathbf{v}$ of the random vector $\mathbf{V}$ into realizations $\mathbf{u}$ of a standard normal random vector $\mathbf{U}$. Such transformations can be defined under weak assumptions when the probability distribution of $\mathbf{V}$ is continuous. For a given design vector $\mathbf{x}$, let $T_{\mathbf{x}}(\mathbf{v})$ be this transformation. Replacing $\mathbf{v}$ by $T_{\mathbf{x}}^{-1}(\mathbf{u})$, gives the equivalent limit-state functions $g_k(\mathbf{x},\mathbf{u})$, where $g_k(\mathbf{x},\mathbf{u}) = G_k(\mathbf{x},T_{\mathbf{x}}^{-1}(\mathbf{u}))$.

A limit-state function $g_k(\mathbf{x},\mathbf{u})$, together with the rule that $g_k(\mathbf{x},\mathbf{u}) \le 0$ is defined as failure and $g_k(\mathbf{x},\mathbf{u}) > 0$ is defined as safe, is referred to as a component. A component may or may not be associated with a physical member or a particular failure mode of the structure.

For any positive integer q, we denote the q-dimensional Euclidean space with $\mathbb{R}^q$. Let components of vectors be given by subscripts, i.e. any vector $\mathbf{a} = (a_1,a_2,\ldots,a_q)$, and let $\varphi_m(\cdot)$ denote the m-dimensional standard normal probability density function.

We define the kth component failure probability by the m-dimensional integral

$$p_k(\mathbf{x}) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} I_{\mathscr{F}_k(\mathbf{x})}(\mathbf{u})\varphi_m(\mathbf{u})\mathrm{d}u_1\cdots\mathrm{d}u_m, \quad k \in \mathbf{K}, \tag{1}$$

where $I_{\mathscr{F}_k(\mathbf{x})}(\mathbf{u})$ is the indicator function, which is equal to one on the failure domain

$$\mathscr{F}_k(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^m | g_k(\mathbf{x},\mathbf{u}) \le 0\} \tag{2}$$

and zero elsewhere, i.e. $I_{\mathscr{F}_k(\mathbf{x})}(\mathbf{u}) = 1$ whenever $\mathbf{u} \in \mathscr{F}_k(\mathbf{x})$ and $I_{\mathscr{F}_k(\mathbf{x})}(\mathbf{u}) = 0$ otherwise.

Generally, a collection of components and a rule for determining which combinations of component failures constitute a system failure are referred to as a structural system. This study focuses on optimization problems involving component failure probabilities.

3. Problem statement

The reliability-based optimal design problem with component failure probabilities is denoted $\mathbf{P}$ and it takes the form

$$\mathbf{P} = \min_{\mathbf{x}\in\mathbb{R}^n}\left\{c_0(\mathbf{x}) + \sum_{k=1}^{K} c_k(\mathbf{x})p_k(\mathbf{x}) | p_k(\mathbf{x}) \le \hat{p}_k, k \in \mathbf{K}, \mathbf{x} \in \mathbf{X}\right\}, \tag{3a}$$

where

$$\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n | f_j(\mathbf{x}) \le 0, j \in \mathbf{J}\}, \tag{3b}$$

with $f_j(\mathbf{x})$, $j \in \mathbf{J} = \{1,2,\ldots,J\}$, being real-valued deterministic, continuously differentiable, constraint functions, and $c_k(\mathbf{x})$, $k \in \{0,1,\ldots,K\}$, are real-valued continuously differentiable cost functions describing the initial cost $c_0($ $\mathbf{x})$ and the cost $c_k(\mathbf{x})$ associated with the failure of the kth component. The values $\hat{p}_k$, $k \in \mathbf{K}$, are pre-defined bounds on the failure probabilities. The objective function in $\mathbf{P}$ can be interpreted as the initial cost plus the expected cost of failure, when expected costs of failure of the components are additive. Hence, $\mathbf{P}$ defines the problem of minimizing the initial cost plus the expected cost of failure subject to reliability and deterministic constraints.

The difficulty associated with solving **P** is two-fold. First, the failure probabilities cannot be computed exactly and, hence, must be approximated. Second, expressions, if they exist, for the gradients of the failure probabilities and their approximations are difficult to obtain. Hence, a direct application of a standard nonlinear optimization algorithm is not possible. In the following, we use the theoretical results in Ref. [36] to overcome these difficulties.

4. Gradient of the failure probability

We show that the failure probability, as defined in Eq. (1), can be rewritten in a form that leads to a formula for the gradient of the failure probability. In this section, we consider only one limit-state function. Hence, we drop the subscript k.

Suppose that the limit-state function $g(\mathbf{x},\mathbf{u})$ is sufficiently 'nice' such that we can solve, either analytical or numerically, the equation

$$g(\mathbf{x},\mathbf{u})=0 \tag{4a}$$

for one of the components of $\mathbf{u}$. Without loss of generality, we assume that we can solve for u_1. Let the remaining components in $\mathbf{u}$ be denoted $\bar{\mathbf{u}}$, i.e. $\mathbf{u}=(u_1,\bar{\mathbf{u}})$. We denote the solution of Eq. (4a) by $h(\mathbf{x},\bar{\mathbf{u}})$. Hence, we have that

$$g(\mathbf{x},(h(\mathbf{x},\bar{\mathbf{u}}),\bar{\mathbf{u}}))=0. \tag{4b}$$

For a given design $\mathbf{x}$, suppose Eq. (4b) holds for all $\bar{\mathbf{u}}\in\mathbb{R}^{m-1}$. Then, we can rewrite the failure probability in the following way:

$$\begin{aligned}p(\mathbf{x})&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}I_{\mathcal{F}(\mathbf{x})}(\mathbf{u})\varphi_m(\mathbf{u})\mathrm{d}u_1\mathrm{d}u_2\cdots\mathrm{d}u_m\\&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{-\infty}^{h(\mathbf{x},\bar{\mathbf{u}})}\varphi_m(\mathbf{u})\mathrm{d}u_1\mathrm{d}u_2\cdots\mathrm{d}u_m\\&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\Phi(h(\mathbf{x},\bar{\mathbf{u}}))\varphi_{m-1}(\bar{\mathbf{u}})\mathrm{d}u_2\cdots\mathrm{d}u_m,\end{aligned} \tag{5a}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, i.e.

$$\Phi(\alpha)=\int_{-\infty}^{\alpha}\varphi_1(a)\mathrm{d}a. \tag{5b}$$

In the derivation of Eq. (5a), we assumed that the failure domain is located in the negative direction of u_1, see Fig. 1. If the failure domain is located in the positive direction of u_1, as in Fig. 2, then we obtain

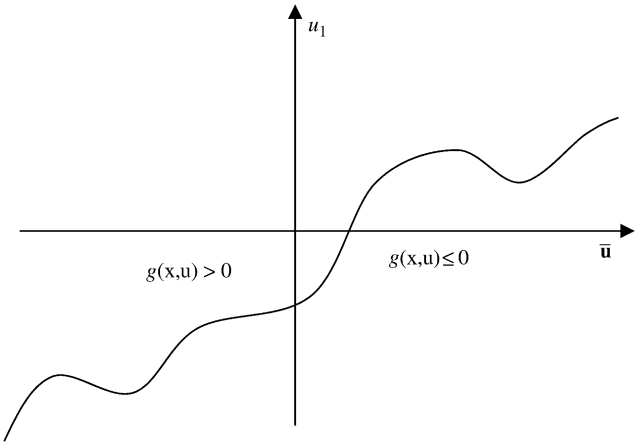

Fig. 1. Failure domain in negative u_1 direction.

$$\begin{aligned}p(\mathbf{x})&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}I_{\mathcal{F}(\mathbf{x})}(\mathbf{u})\varphi_m(\mathbf{u})\mathrm{d}u_1\mathrm{d}u_2\cdots\mathrm{d}u_m\\&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{h(\mathbf{x},\bar{\mathbf{u}})}^{\infty}\varphi_m(\mathbf{u})\mathrm{d}u_1\mathrm{d}u_2\cdots\mathrm{d}u_m\\&=\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\Phi(-h(\mathbf{x},\bar{\mathbf{u}}))\varphi_{m-1}(\bar{\mathbf{u}})\mathrm{d}u_2\cdots\mathrm{d}u_m.\end{aligned} \tag{5c}$$

In the derivations of Eqs. (5a) and (5c), we assumed that, given an $\mathbf{x}$, for each $\bar{\mathbf{u}}$ we could find a unique function $h(\mathbf{x},\bar{\mathbf{u}})$ such that Eq. (4b) holds for all $\bar{\mathbf{u}}\in\mathbb{R}^{m-1}$. Clearly, this may not always be the case as Fig. 3 illustrates. In Fig. 3, it is not possible to find a component (either u_1 or u_2) for which to solve Eq. (4a). However, we expect that for many practical cases we can find at least one component of $\mathbf{u}$ for which to solve Eq. (4a) uniquely, at least for $\bar{\mathbf{u}}$ in a sufficiently large ball around the origin. Due to the rapid decay of the standard normal probability density function, points further away from the origin will not contribute significantly to the integral in Eqs. (5a) and (5c) and can be ignored in an approximating approach. In the case of a limit-state function of the form $g(\mathbf{x},\mathbf{u})=\tilde{g}(\mathbf{x},\bar{\mathbf{u}})+f(\mathbf{x})u_1$, with

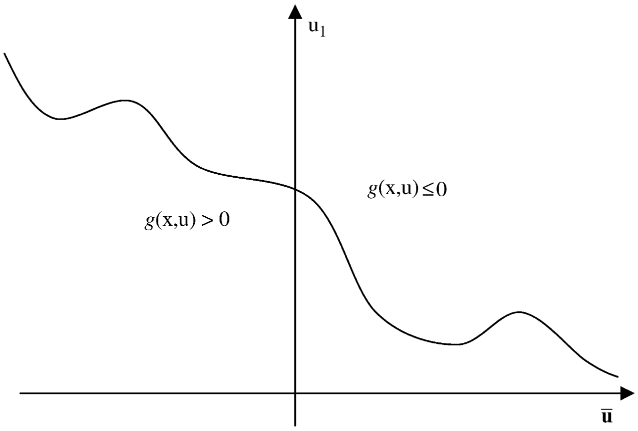

Fig. 2. Failure domain in positive u_1 direction.

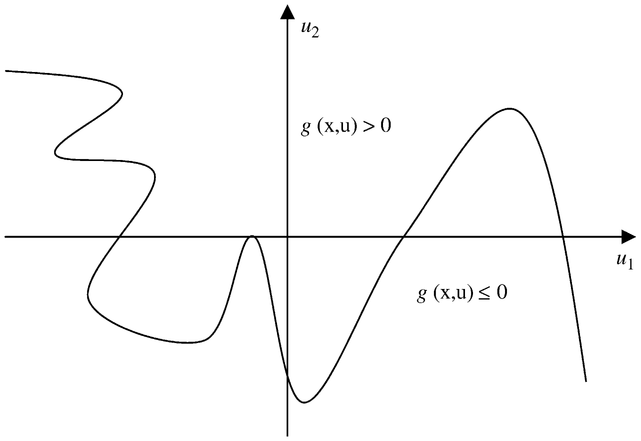

Fig. 3. Irregular failure domain.

$f(\mathbf{x}) > 0$, i.e. linear in one of the components of $\mathbf{u}$, it is particularly trivial to see that we can solve for a component. In this case, $h(\mathbf{x}, \bar{\mathbf{u}}) = -\tilde{g}(\mathbf{x}, \bar{\mathbf{u}})/f(\mathbf{x})$. Note that even if there is no component of $\mathbf{u}$ for which we can solve Eq. (4a), there may exist a rotational transformation in $\mathbb{R}^m$ (i.e. the $\mathbf{u}$-space) that leads to a new limit-state function with the desired properties. Rotational transformations do not change the problem due to the rotationally invariant property of the standard normal probability density function. If a suitable limit-state function cannot be obtained by means of a rotational transformation, which is the case in Fig. 3, the following approximating approach can always be used. Let $\epsilon > 0$ be a constant. We define a new limit-state function

$$g^*(\mathbf{x}, (\mathbf{u}, u_{m+1})) = g(\mathbf{x}, \mathbf{u}) + \epsilon u_{m+1}, \tag{5d}$$

where $g(\mathbf{x}, \mathbf{u})$ is the original limit-state function and u_{m+1} is an auxiliary variable. Then, we can solve $g^*(\mathbf{x}, (\mathbf{u}, u_{m+1})) = 0$ for u_{m+1} for all $\mathbf{x}$ and $\mathbf{u}$. The auxiliary variable u_{m+1} can be interpreted as a realization of a standard normal random variable U_{m+1}, with ϵU_{m+1} being the 'random error' in the limit-state function $g(\mathbf{x}, \mathbf{u})$. In applications, it is difficult to determine a reasonable joint distribution function for $\mathbf{V}$ when additional randomness is introduced by U_{m+1}, as in Eq. (5d). Hence, this approach is usually approximate. For a reasonably accurate approximation, ϵ must be scaled properly. We recommend setting $\epsilon \ll \sigma_g$, where σ_g^2 is the first-order approximate variance of $g(\mathbf{x}_1, \mathbf{U})$ at some initial design $\mathbf{x}_1$ and realization $\mathbf{u}^*$, i.e. $\sigma_g^2 = \mathrm{Var}[g(\mathbf{x}_1, \mathbf{u}^*) + \langle \nabla_{\mathbf{u}} g(\mathbf{x}_1, \mathbf{u}^*), \mathbf{U} - \mathbf{u}^* \rangle]$. The realization $\mathbf{u}^*$ can be set equal to the closest point to the origin on the surface $\{\mathbf{u} | g(\mathbf{x}_1, \mathbf{u}) = 0\}$.

In view of Eqs. (5a) and (5c), we see that the gradient of the failure probability exists under fairly general conditions and is given by

$$\nabla p(\mathbf{x}) = -\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \varphi_1(h(\mathbf{x}, \bar{\mathbf{u}})) \times \frac{\nabla_{\mathbf{x}} g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))}{\partial g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))/\partial u_1} \varphi_{m-1}(\bar{\mathbf{u}}) \mathrm{d}u_2 \cdots \mathrm{d}u_m, \tag{6a}$$

when the failure domain is located in the negative direction of u_1, see Fig. 1, and

$$\nabla p(\mathbf{x}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \varphi_1(h(\mathbf{x}, \bar{\mathbf{u}})) \frac{\nabla_{\mathbf{x}} g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))}{\partial g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))/\partial u_1} \times \varphi_{m-1}(\bar{\mathbf{u}}) \mathrm{d}u_2 \cdots \mathrm{d}u_m, \tag{6b}$$

when the failure domain is located in the positive direction of u_1, see Fig. 2. In Eqs. (6a) and (6b), we have used the fact that

$$\nabla_{\mathbf{x}} h(\mathbf{x}, \bar{\mathbf{u}}) = -\frac{\nabla_{\mathbf{x}} g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))}{\partial g(\mathbf{x}, (h(\mathbf{x}, \bar{\mathbf{u}}), \bar{\mathbf{u}}))/\partial u_1}, \tag{6c}$$

and that differentiation and integration operators can be interchanged. Clearly, we must assume that the limit-state function is differentiable with respect to $\mathbf{x}$ and u_1.

In Ref. [43], we find similar expressions for the gradient of the failure probability for the case with bounded random variables $\mathbf{V}$. Since unbounded random variables can be approximated by bounded random variables, the result in Ref. [43] implies that a lower bound for any failure probability is differentiable. The result in Ref. [43] also holds for parallel systems, i.e. the case where the failure domain is given by $\mathscr{F}(\mathbf{x}) = \bigcap_{k \in \mathbf{K}} \{\mathbf{u} \in \mathbb{R}^m | g_k(\mathbf{x}, \mathbf{u}) \leq 0\}$.

In the following, we assume without loss of generality that the failure domain is located in the negative direction of u_1, i.e., that Eqs. (5a) and (6a) hold.

5. Approximation results

In this section, we use sampling techniques to estimate the integrals (5a) and (6a). This gives rise to approximating problems, which are increasingly accurate as the number of sample points increases.

The Monte Carlo estimates of Eqs. (5a) and (6a) are given by the expressions

$$p_{k,N}(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} \Phi(h_k(\mathbf{x}, \bar{\mathbf{u}}_j)), \tag{7a}$$

$$\nabla p_{k,N}(\mathbf{x}) = -\frac{1}{N} \sum_{j=1}^{N} \varphi_1(h_k(\mathbf{x}, \bar{\mathbf{u}}_j)) \frac{\nabla_{\mathbf{x}} g_k(\mathbf{x}, (h_k(\mathbf{x}, \bar{\mathbf{u}}_j), \bar{\mathbf{u}}_j))}{\partial g_k(\mathbf{x}, (h_k(\mathbf{x}, \bar{\mathbf{u}}_j), \bar{\mathbf{u}}_j))/\partial u_1}, \tag{7b}$$

where $\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \ldots, \bar{\mathbf{u}}_N$ are realizations of a collection of independent standard normal $(m-1)$-dimensional random vectors $\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \ldots, \bar{\mathbf{U}}_N$, N is the number of sample points, and the subscript k is reintroduced to indicate the component. Hence, $h_k(\mathbf{x}, \bar{\mathbf{u}})$ is the solution with respect to u_1 of the equation $g_k(\mathbf{x}, \mathbf{u}) = 0$.

Instead of generating sample points according to a standard normal distribution, as in Eqs. (7a) and (7b), we can sample according to other probability distributions. This approach leads to importance sampling, which tends to improve the failure probability estimates by concentrating the samples in the most relevant region. A typical selection of sampling distribution is a normal distribution. In this paper, we use a normal random vector with mean vector $\mu_k \in \mathbb{R}^{m-1}$ and variance–covariance matrix $\sigma_k \mathbf{I}$, $\sigma_k > 0$, where $\mathbf{I}$ is the $(m-1)\times(m-1)$ unit matrix. The importance sampling estimates of Eqs. (5a) and (6a) are given by the expressions

$$p_{k,N}(\mathbf{x};\mu_k,\sigma_k) = \frac{1}{N}\sum_{j=1}^{N} \Phi(h_k(\mathbf{x},\sigma_k\bar{\mathbf{u}}_j+\mu_k))\frac{\varphi_{m-1}(\sigma_k\bar{\mathbf{u}}_j+\mu_k)}{\varphi_{m-1}(\bar{\mathbf{u}}_j)/\sigma_k^{m-1}}, \tag{8a}$$

$$\begin{aligned}\nabla p_{k,N}(\mathbf{x};\mu_k,\sigma_k) = &-\frac{1}{N}\sum_{j=1}^{N}\varphi_1(h_k(\mathbf{x},\sigma_k\bar{\mathbf{u}}_j+\mu_k))\\ &\times\frac{\nabla_{\mathbf{x}} g_k(\mathbf{x},(h_k(\mathbf{x},\sigma_k\bar{\mathbf{u}}_j+\mu_k),\sigma_k\bar{\mathbf{u}}_j+\mu_k))}{\partial g_k(\mathbf{x},(h_k(\mathbf{x},\sigma_k\bar{\mathbf{u}}_j+\mu_k),\sigma_k\bar{\mathbf{u}}_j+\mu_k))/\partial u_1}\\ &\times\frac{\varphi_{m-1}(\sigma_k\bar{\mathbf{u}}_j+\mu_k)}{\varphi_{m-1}(\bar{\mathbf{u}}_j)/\sigma_k^{m-1}},\end{aligned} \tag{8b}$$

where we have used the fact that $\sigma_k\bar{\mathbf{u}}+\mu_k$ is a normal random vector with mean vector μ_k and variance–covariance matrix $\sigma_k\mathbf{I}$. Note that for $\sigma_k=1$ and $\mu_k=\mathbf{0}$, Eqs. (8a) and (8b) simplify to Eqs. (7a) and (7b). For $\mu_k\neq\mathbf{0}$, the samples are not centered at the origin, but, hopefully, in a more relevant region. The selection of the sampling parameters σ_k and μ_k is discussed in Section 6.

The expressions in Eqs. (7b) and (8b) give estimates of the sensitivity of the failure probability with respect to certain parameters $\mathbf{x}$. These formulas are of significant interest in structural reliability analysis. It is clear that the standard estimate [5] $\sum_{j=1}^{N} I_{\mathscr{F}_k(\mathbf{x})}(\mathbf{u}_j)/N$ of $p_k(\mathbf{x})$ cannot lead to similar sensitivity results because of the nonsmoothness of the indicator function.

The convergence of the Monte Carlo simulation and importance sampling estimates to the failure probability and its gradient, as $N\rightarrow\infty$, is well-known [32,34]. To be able to prove convergence of the algorithm below, we need a uniform bound on the rate of convergence. In Ref. [36], we show by means of Ref. [34] that such a uniform bound exists under the assumption that the limit-state functions are sufficiently 'nice'. The precise mathematical statement of this assumption follows:

Assumption 1. We assume that for each component $k\in\mathbf{K}$

(i) there exists a unique real-valued function $h_k(\mathbf{x},\bar{\mathbf{u}})$ such that for all $\mathbf{x}\in\mathbb{R}^n$ and $\bar{\mathbf{u}}\in\mathbb{R}^{m-1}$, $g_k(\mathbf{x},(h_k(\mathbf{x},\bar{\mathbf{u}}),\bar{\mathbf{u}}))=0$,

(ii) the limit-state function $g_k(\mathbf{x},\mathbf{u})$ is continuously differentiable with respect to $\mathbf{x}$ and u_1,

(iii) for every bounded set $\mathbf{S}\subset\mathbb{R}^n$ there exist constants $C_1,C_2\in(0,\infty)$ such that $\|\nabla_{\mathbf{x}} g_k(\mathbf{x},(h_k(\mathbf{x},\bar{\mathbf{u}}),\bar{\mathbf{u}}))\|\le C_1$ and $|\partial g_k(\mathbf{x},(h_k(\mathbf{x},\bar{\mathbf{u}}),\bar{\mathbf{u}}))/\partial u_1|\ge C_2$ for all $\mathbf{x}\in\mathbf{S}$ and $\bar{\mathbf{u}}\in\mathbb{R}^{m-1}$.

In Assumption 1(i), we assume that Eq. (4a) is solvable for a unique u_1, which we discussed above. Item (ii) states that the limit-state functions are assumed to be sufficiently smooth, which is often the case in practical applications. If a particular model results in nonsmooth limit-state functions, a re-modelling is sometimes possible. See Ref. [12] for a comprehensive discussion of gradients of limit-state functions. In Assumption 1(iii), we assume that the gradients with respect to $\mathbf{x}$ of the limit-state functions are bounded and that the partial derivatives with respect to u_1 are bounded away from zero. The latter is motivated by Eq. (6c). Assumption 1(iii) can be difficult to check in practice. However, it is mostly of theoretical importance and it does not impose significant computational consequences. Special cases such as $g(\mathbf{x},\mathbf{u})=\tilde{g}(\mathbf{x},\bar{\mathbf{u}})+f(\mathbf{x})u_1$, with $f(\mathbf{x})>0$ for all $\mathbf{x}\in\mathbb{R}^n$ and bounded random variables $\mathbf{V}$, can easily be shown to satisfy Assumption 1(iii).

In Assumptions 1(i,ii), the statements are required to hold for all $\mathbf{x}\in\mathbb{R}^n$. However, it is sufficient that these statements hold on a sufficiently large subset of the design space containing all relevant designs. Since the characterization of such a subset is application dependent, for generality, we have adopted stronger assumptions than typically needed in practice.

In Ref. [36], we show that under Assumption 1 the error in the Monte Carlo simulation and importance sampling estimates is bounded by a function of N, which is independent of $\mathbf{x}$ and the generated samples. The bound is not tight compared to *statistical* bounds [5], but statistical bounds only state that the error is smaller than a given number with a certain probability. In contrast, the bound below is valid with probability one, which is needed in the proof of convergence of our optimization algorithm.

Theorem 1 (Proof in Ref. [36]). *Suppose that Assumption* 1 *holds, and that the sampling parameters in Eqs.* (8a) and (8b) *are either* $\mu_k=\mathbf{0}$ *and* $\sigma_k=1$ (*Monte Carlo*), *or* $\mu_k\in\mathbb{R}^{m-1}$ *and* $\sigma_k>1$ (*importance sampling*). *Let* $p_{k,N}(\mathbf{x};\mu_k,\sigma_k)$ *and* $\nabla p_{k,N}(\mathbf{x};\mu_k,\sigma_k)$ *be defined in terms of a generated sequence of realizations* $\bar{\mathbf{u}}_1,\bar{\mathbf{u}}_2,\bar{\mathbf{u}}_3,\ldots$ *of independent standard normal* $(m-1)$*-dimensional random vectors* $\bar{\mathbf{U}}_1$, $\bar{\mathbf{U}}_2,\bar{\mathbf{U}}_3,\ldots$
Then, for every constant $\kappa>0$ *and bounded set* $\mathbf{S}\subset\mathbb{R}^n$, *there exists a constant* $C\in(0,\infty)$ *such that*

$$|p_{k,N}(\mathbf{x};\mu_k,\sigma_k)-p_k(\mathbf{x})|\le C\frac{(\log N)^{\kappa+1/2}}{\sqrt{N}}, \tag{9a}$$

$$\|\nabla p_{k,N}(\mathbf{x};\mu_k,\sigma_k)-\nabla p_k(\mathbf{x})\|\le C\frac{(\log N)^{\kappa+1/2}}{\sqrt{N}} \tag{9b}$$

for all $\mathbf{x}\in\mathbf{S}$, $N\in\{1,2,3,\ldots\}$, *and* $k\in\mathbf{K}$.

Observe that the right-hand sides of Eqs. (9a) and (9b) vanish as $N\rightarrow\infty$. In Theorem 1, for technical reasons, we needed to assume that $\sigma_k>1$ whenever importance sampling is used. This is not a severe restriction because the usual strategy in importance sampling is to shift the ‘center’ of the sampling distribution to a relevant region, i.e. selecting an appropriate μ_k, and to keep $\sigma_k = 1$. Hence, by setting σ_k equal to, e.g. 1.01, we satisfy the theoretical assumption with practically no change in the sampling efficiency.

We can define a class of approximations to **P** in terms of the sampling estimate in Eq. (8a). Let, for any $N \in \{1, 2, 3, \ldots\}$, $\mu_k \in \mathbb{R}^{m-1}$, and $\sigma_k > 0$, $k \in \mathbf{K}$, the approximating problems $\mathbf{P}_N$ be given by

$$\mathbf{P}_N : \min_{\mathbf{x}\in\mathbb{R}^n} \left\{ c_0(\mathbf{x}) + \sum_{k=1}^{K} c_k(\mathbf{x}) p_{k,N}(\mathbf{x}; \mu_k, \sigma_k) | p_{k,N}(\mathbf{x}; \mu_k, \sigma_k) \le \hat{p}_k, k \in \mathbf{K}, \mathbf{x} \in \mathbf{X} \right\}. \tag{10}$$

Note that $\mathbf{P}_N$ is not well-defined before a sequence $\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \ldots, \bar{\mathbf{u}}_N$ of realizations of independent standard normal $(m-1)$-dimensional random vectors $\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \ldots, \bar{\mathbf{U}}_N$ is determined.

6. Algorithm

We present an algorithm for solving **P**, which makes use of a nonlinear optimization subroutine for solving the approximating problems $\mathbf{P}_N$. The mathematical proof of convergence of the algorithm can be found in Ref. [36].

The algorithm is based on the principle of ‘moving targets’, which can best be explained by means of Fig. 4. Consider the sequence of approximating problems $\{\mathbf{P}_N\}_{N\in\mathcal{N}}$, where $\mathcal{N}$ is an infinite sequence of strictly increasing positive integers. For the sake of the explanation, suppose that $\mathcal{N} = \{N', N'', N''', \ldots\}$. The algorithm starts from an initial design $\mathbf{x}_1$, marked with a dot in Fig. 4, and applies the nonlinear optimization subroutine to the problem $\mathbf{P}_{N'}$, whose solution is denoted $\hat{\mathbf{x}}_{N'}$. As indicated by a solid arrow from $\mathbf{x}_1$ in Fig. 4, the subroutine generates iterates that gradually get closer to $\hat{\mathbf{x}}_{N'}$. When the current iterate is sufficiently close to $\hat{\mathbf{x}}_{N'}$, as determined by a precision-adjustment rule described below, the number of sample points is increased from N' to $N'' > N'$. The subroutine then continues by computing iterates that approach a solution of $\mathbf{P}_{N''}$ until the rule again determines that the number of sample points must be increased to N'''. The solution of $\mathbf{P}_{N''}$ is denoted $\hat{\mathbf{x}}_{N''}$ in Fig. 4. This process continues and the iterates generated by the algorithm converge to the solution $\hat{\mathbf{x}}$ of **P**. The last iterate of the previous approximation level is used as a ‘warm start’ for the next approximation level. The iterates generated by the algorithm gradually get closer and closer to a solution of the current approximating problem before the number of sample points is increased. Effectively, the algorithm computes approximating solutions to a sequence of approximating problems $\{\mathbf{P}_N\}_{N\in\mathcal{N}}$ with higher and higher precision as the number of iterations increases.

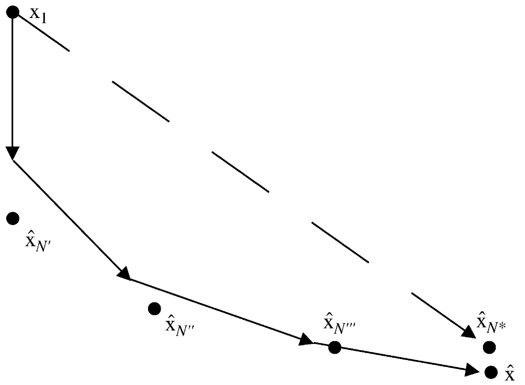

Fig. 4. Moving targets.

The ‘moving target’ scheme is much more efficient than choosing a large number of sample points N^* and solving the corresponding problem $\mathbf{P}_{N^*}$. This strategy is illustrated by the dashed arrow in Fig. 4. The dashed arrow is shorter than the solid path, but each iteration is computationally costly due to the large number of sample points N^*. In comparison, each iteration in our algorithm is relatively inexpensive until the iterates are close to a solution of **P**.

To be able to prove convergence of our algorithm, we impose a requirement on the nonlinear optimization subroutine. The precise mathematical statement of this requirement can be found in Ref. [36]. The requirement essentially ensures that a sequence generated by the subroutine, when applied to $\mathbf{P}_N$, has two properties: (i) The sequence converges to a Karush–Kuhn–Tucker point for $\mathbf{P}_N$. (ii) There is a guaranteed minimum progress towards a Karush–Kuhn–Tucker point, which, for sufficiently large N, is independent of N. In Appendix A, we present one well-tested nonlinear optimization algorithm that satisfies this requirement.

Before we describe the algorithm, we need to establish some notation. Let

$$F_N(\mathbf{x}', \mathbf{x}''; \mu, \sigma) = \max\{f_{0,N}(\mathbf{x}''; \mu, \sigma) - f_{0,N}(\mathbf{x}'; \mu, \sigma) - \gamma\psi_N(\mathbf{x}'; \mu, \sigma)_+, \psi_N(\mathbf{x}''; \mu, \sigma) - \psi_N(\mathbf{x}'; \mu, \sigma)_+\}, \tag{11a}$$

where $\gamma > 0$ is a constant, $\mu = (\mu_1, \ldots, \mu_K)$, $\sigma = (\sigma_1, \ldots, \sigma_K)$, and

$$f_{0,N}(\mathbf{x}; \mu, \sigma) = c_0(\mathbf{x}) + \sum_{k=1}^{K} c_k(\mathbf{x}) p_{k,N}(\mathbf{x}; \mu_k, \sigma_k), \tag{11b}$$

$$\psi_N(\mathbf{x}; \mu, \sigma) = \max\left\{ \max_{k\in\mathbf{K}} \{p_{k,N}(\mathbf{x}; \mu_k, \sigma_k) - \hat{p}_k\}, \max_{j\in\mathbf{J}} f_j(\mathbf{x}) \right\}, \tag{11c}$$

and $\psi_N(\mathbf{x}; \mu, \sigma)_+ = \max\{0, \psi_N(\mathbf{x}; \mu, \sigma)\}$. Note that $f_{0,N}(\mathbf{x}; \mu, \sigma)$ and $\psi_N(\mathbf{x}; \mu, \sigma)$ are the objective function and the aggregated constraint function in $\mathbf{P}_N$, respectively.

The function $F_N(\mathbf{x}', \mathbf{x}''; \mu, \sigma)$ measures how much 'better' the design $\mathbf{x}''$ is compared to $\mathbf{x}'$. Suppose $\mathbf{x}'$ is a feasible design for $\mathbf{P}_N$. Then, $\psi_N(\mathbf{x}'; \mu, \sigma) \leq 0$ and, hence

$$F_N(\mathbf{x}', \mathbf{x}''; \mu, \sigma) = \max\{f_{0,N}(\mathbf{x}''; \mu, \sigma) - f_{0,N}(\mathbf{x}'; \mu, \sigma), \psi_N(\mathbf{x}''; \mu, \sigma)\}. \tag{11d}$$

We see that if $F_N(\mathbf{x}', \mathbf{x}''; \mu, \sigma) \leq -\omega$, with ω being some positive number, then the objective function in $\mathbf{P}_N$ at $\mathbf{x}''$ is reduced with at least the amount ω compared to the value at $\mathbf{x}'$. Additionally, $\mathbf{x}''$ is feasible for $\mathbf{P}_N$ because $\psi_N(\mathbf{x}''; \mu, \sigma) \leq -\omega$.

Suppose that $\mathbf{x}'$ is not a feasible design for $\mathbf{P}_N$. Then, $\psi_N(\mathbf{x}'; \mu, \sigma) > 0$. When $F_N(\mathbf{x}', \mathbf{x}''; \mu, \sigma) \leq -\omega$, the constraint violation for $\mathbf{P}_N$ at $\mathbf{x}''$ is reduced with at least the amount ω compared to the value at $\mathbf{x}'$ because $\psi_N(\mathbf{x}''; \mu, \sigma) - \psi_N(\mathbf{x}'; \mu, \sigma) \leq -\omega$.

Our algorithm for solving $\mathbf{P}$ takes the following form.

Algorithm for solving $\mathbf{P}$.

Parameters. Select $\kappa > 0$, $\eta > 0$, $\gamma > 0$, and either $(\mu_k = \mathbf{0}, \sigma_k = 1)$ or $(\mu_k \in \mathbb{R}^{m-1}, \sigma_k > 1)$ for all $k \in \mathbf{K}$.

Data. An initial design $\mathbf{x}_1 \in \mathbb{R}^n$, an infinite sequence $\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \bar{\mathbf{u}}_3, \ldots$ of generated realizations of independent standard normal $(m-1)$-dimensional random vectors $\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3, \ldots$, and an infinite set $\mathcal{N}$ of strictly increasing positive integers.

Step 0. Set $i = 1$ and N equal to the smallest number in $\mathcal{N}$.

Step 1. Starting from $\mathbf{x}_i$, compute $\mathbf{x}^*$ by performing one iteration of a nonlinear optimization subroutine applied to $\mathbf{P}_N$.

Step 2. If

$$F_N(\mathbf{x}_i, \mathbf{x}^*; \mu, \sigma) \leq -\frac{\eta(\log N)^{\kappa+1/2}}{\sqrt{N}}, \tag{12}$$

then set $\mathbf{x}_{i+1} = \mathbf{x}^*$, $N_i = N$, and go to **Step 3**.

Else, augment N to the smallest number in $\mathcal{N}$ larger than N, and go to **Step 1**.

Step 3. Replace i by $i+1$, and go to **Step 1**.

In Ref. [36], we show that when using a suitable subroutine (e.g. the one given in Appendix A), the algorithm for solving $\mathbf{P}$ generates a sequence of designs $\{\mathbf{x}_i\}$ converging to a Karush–Kuhn–Tucker point. In fact, the algorithm is shown to also converge to more general F. John points [33]. It should be noted that the number of sample points is driven to infinity as the algorithm progresses, i.e. $N_i \longrightarrow \infty$, as $i \longrightarrow \infty$.

In the absence of convexity, algorithms for solving nonlinear optimization problems can typically only guarantee convergence to Karush–Kuhn–Tucker points, or points satisfying some other first-order necessary optimality condition. Karush–Kuhn–Tucker points are usually local minimizers of the problem at hand.

In applications, the algorithm for solving $\mathbf{P}$ is always terminated after a finite number of iterations. Various stopping rules, such as "stop when the allocated time is consumed" and "stop when $N_i > 10^3$," can be used. Advanced stopping rules and techniques for evaluating the quality of a candidate solution can be found in Refs. [38,39], and for the case with deterministic constraints, in Ref. [25].

The one-dimensional root finding problems in the evaluation of $p_{k,N}(\mathbf{x}; \mu_k, \sigma_k)$ and $\nabla p_{k,N}(\mathbf{x}; \mu_k, \sigma_k)$ usually cannot be solved exactly in finite computing time. One possibility is to introduce an additional precision parameter that ensures a gradually better accuracy in the root finding as the algorithm progresses. Alternatively, we can prescribe a rule saying that the root finding algorithm (e.g. the secant method) should terminate after cN_i iterations, with c some constant. These alternatives lead to an implementable algorithm with similar behavior as the algorithm described above. For simplicity of the presentation, we have not included the issue of root finding in the discussion. In fact, this issue is not problematic in practice. One-dimensional root finding problems can be solved in a few iterations with close to floating-point accuracy using standard algorithms. Hence, the root finding problem can be solved with a fixed precision for all iterations in the algorithm for solving $\mathbf{P}$ giving a negligible error compared to the one caused by the sampling technique.

In realistic design examples, evaluations of the limit-state functions and their gradients typically involve computationally costly steps such as the solutions of boundary value problems. It is therefore crucial to reduce the number of sample points used in the algorithm for solving $\mathbf{P}$. Such reduction can be obtained by selecting the sampling parameters μ_k and σ_k (see Eqs. (8a) and (8b)) so that the samples are concentrated in the regions with highest contributions to the estimates of the integrals. To identify such regions can be hard, but one possibility is to center the sampling density at the point $\bar{\mathbf{u}}_k^*(\mathbf{x})$, where $\mathbf{u}_k^*(\mathbf{x}) = ((u_1)_k^*(\mathbf{x}), \bar{\mathbf{u}}_k^*(\mathbf{x}))$ is the closest point to the origin on the limit-state surface $\{\mathbf{u} | g_k(\mathbf{x}, \mathbf{u}) = 0\}$, i.e. $\mathbf{u}_k^*(\mathbf{x})$ is the design point for $g_k(\mathbf{x}, \mathbf{u})$ [5]. Hence, $\bar{\mathbf{u}}_k^*(\mathbf{x})$ is the vector containing all the components, except the first one, of the design point for $g_k(\mathbf{x}, \mathbf{u})$.

A preliminary study of the accuracy of the estimation technique (8a), with $\mu_k = \bar{\mathbf{u}}_k^*(\mathbf{x})$ is presented in Table 1. The number of sample points needed to compute an estimate of the failure probability with coefficient of variation of 0.05 using standard, crude Monte Carlo simulation (MC), i.e. $\sum_{j=1}^{N} I_{\mathcal{F}_k(\mathbf{x})}(\mathbf{u}_j)/N$, standard importance sampling centered at the design point (IP), our crude technique (7a), and our focused technique (8a) are reported. In Eq. (8a), we select $\mu_k = \bar{\mathbf{u}}_k^*(\mathbf{x})$ and $\sigma_k = 1$. Two examples are considered. The first example (Linear) uses the limit-state function $g_k(\mathbf{x}, \mathbf{u}) = d - \sum_{i=1}^{m} u_i$ with varying parameters d and m. The second example (Initial Girder and Optimal

Table 1
Number of samples to estimate the failure probability with c.o.v 0.05

	MC	IP	(7a)	(8a)
Linear $d = 7.3566$, $m = 10$	44 000	980	20 000	330
Linear $d = 11.7606$, $m = 10$	4 000 000	1600	1 400 000	380
Linear $d = 16.4498$, $m = 50$	38 000	1100	30 000	620
Linear $d = 26.2974$, $m = 50$	3 700 000	1600	2 600 000	820
Initial girder	27 000	1200	18 000	470
Optimal girder	290 000	1500	190 000	550

Girder) used the nonlinear limit-state function associated with flexure in the reinforced concrete girder described in Section 7. 'Initial' and 'Optimal' refer to the initial and optimal design, respectively, of the girder as found in Section 7. It is seen from Table 1 that Eq. (8a), with $\mu_k = \bar{\mathbf{u}}_k^*(\mathbf{x})$, requires significantly fewer sample points than importance sampling (IP) to obtain the same accuracy. However, it should be noted that Eq. (8a) generally requires the solution of N one-dimensional root-finding problems, with additional evaluations of the limit-state function. Typically, it takes only a few additional evaluations of the limit-state function to solve the root-finding problem with sufficient accuracy. Even when this is taken into account, Eq. (8a), with $\mu_k = \bar{\mathbf{u}}_k^*(\mathbf{x})$, appears to be a reasonably efficient technique for estimating the failure probability.

It should be noted that in Ref. [18], a rotational transformation is used in conjunction with Eq. (7a) to obtain a sampling technique which appears to be better than importance sampling in many cases. The application of this sampling technique in conjunction with the algorithm for solving **P** appears to be problematic because the rotational transformations may lead to violations of Assumption 1. Hence, that sampling technique is not considered in this paper.

As the algorithm for solving **P** progresses, the region with the highest contribution to the integrals (8a) and (8b), and the design points $\mathbf{u}_k^*(\mathbf{x}_i)$ may vary. Hence, the design points should be re-estimated multiple times. Any scheme involving a finite number of changes in the sampling parameters μ_k can be used without affecting the convergence properties of the algorithm for solving **P**. Other more advanced sampling techniques, as found in Refs. [23,24,32,40], can potentially be used in conjunction with the algorithm for solving **P**.

7. Design of reinforced concrete girder

Consider a highway bridge with reinforced concrete girders of the type shown in Figs. 5 and 6. In this example, we design one such girder using the material and load data from Refs. [8,20]. The design variables are collected in the vector

$$\mathbf{x} = (A_s, b, h_f, b_w, h_w, A_v, S_1, S_2, S_3) \in \mathbb{R}^9, \tag{13}$$

where A_s is the area of the tension steel reinforcement, b is the width of the flange, h_f is the thickness of the flange, b_w is the width of the web, h_w is the height of the web, A_v is the area of the shear reinforcement (twice the cross-section area of a stirrup), and S_1, S_2 and S_3 are the spacings of shear reinforcements in intervals 1, 2, and 3, respectively, see Fig. 6. The random variables describing the loading and material properties are collected in the vector

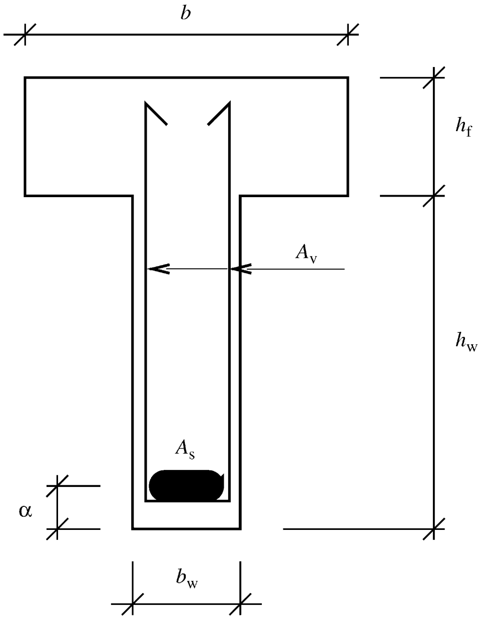

Fig. 5. Cross-section of reinforced concrete girder.

$$\mathbf{V} = (f_y, f'_c, P_D, M_L, P_{S1}, P_{S2}, P_{S3}, W) \in \mathbb{R}^8, \tag{14}$$

where f_y is the yield strength of the reinforcement, f'_c is the compressive strength of concrete, P_D is the dead load excluding the weight of the girder, M_L is the live load moment, P_{S1}, P_{S2} and P_{S3} are the live load shear forces in intervals 1, 2, and 3, respectively, see Fig. 6, and W is the unit weight of concrete. Following Ref. [20], all the random variables are considered to be independent and normally distributed with the means and coefficients of variation as listed in Table 2. Let the girder length be $L_g = 18.30$ m, and the distance from the bottom fiber to the centroid of the tension reinforcement be $\alpha = 0.1$ m, see Fig. 5.

The objective is to design the girder according to the specifications in Ref. [1]. However, these specifications do not lead to well-defined optimization problems for two

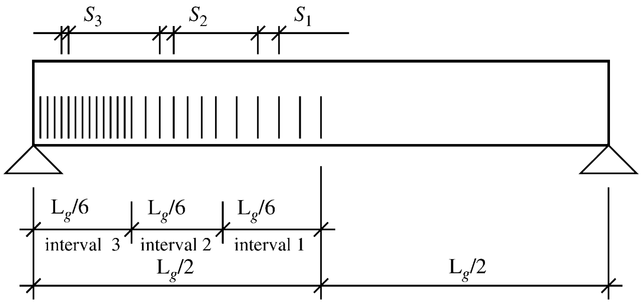

Fig. 6. Reinforced concrete girder with shear reinforcement.

Table 2
Statistics of normal random variables

Variable	Description	Mean	c.o.v.
f_y	Yield strength of reinforcement	413.4×10^6 Pa	0.15
f'_c	Compressive strength of concrete	27.56×10^6 Pa	0.15
P_D	Dead load excluding girder	13.57×10^3 N/m	0.20
M_L	Live load moment	929×10^3 N m	0.243
P_{S1}	Live load shear in interval 1	138.31×10^3 N	0.243
P_{S2}	Live load shear in interval 2	183.39×10^3 N	0.243
P_{S3}	Live load shear in interval 3	228.51×10^3 N	0.243
W	Unit weight of concrete	22.74×10^3 N/m^3	0.10

reasons. First, some of the constraints specified by American Association of State Highway and Transportation Officials [1] are not continuous functions, but of the form $f(\mathbf{x}) \leq 1$ whenever $h(\mathbf{x}) \leq 0$ and otherwise $f(\mathbf{x}) \leq 2$, where $f(\mathbf{x})$ and $h(\mathbf{x})$ are continuous functions. Second, $h(\mathbf{x})$ may also depend on the random variables of the problem. In the following, the first difficulty is overcome by considering different cases. For example, Case 1 has the constraints $f(\mathbf{x}) \leq 1$ and $h(\mathbf{x}) \leq 0$, while Case 2 has the constraints $f(\mathbf{x}) \leq 2$ and $h(\mathbf{x}) \geq 0$. The optimal design for each case is found independently, and the design with the smallest value of the objective function is our solution. The second difficulty is overcome by replacing any random variables in the definition of $h(\mathbf{x})$ by their mean values. The four cases corresponding to the different specifications in Ref. [1] are defined in Ref. [37].

Suppose that the objective is to minimize the initial cost of the reinforced concrete girder, subject to constraints on the failure probabilities and deterministic constraints according to Ref. [1]. Let $C_s = 50$ and $C_c = 1$ be the unit costs of the steel reinforcement and concrete per cubic meter, respectively. As in Ref. [20], we define the initial cost to be

$$c_0(\mathbf{x}) = 0.75C_sL_gA_s + C_sn_SA_v(h_f + h_w - \alpha + 0.5b_w) + C_cL_g(bh_f + b_wh_w), \tag{15}$$

where $n_S = L_g(1/S_1 + 1/S_2 + 1/S_3)/3$ is the total number of stirrups. In Eq. (15), the first term represents the cost of the bending reinforcement. The factor 0.75 appears due to the assumption that the total amount of bending reinforcement is placed only within a length $L_g/2$ centered at the middle point of the girder, and the remaining part is reinforced with $0.5A_s$. The second and third terms in Eq. (15) represent the costs of shear reinforcement and concrete, respectively. Since we only consider the initial cost, we set $c_k(\mathbf{x}) = 0$ for $k \in \mathbf{K}$ in Eq. (3a).

We assume that the girder can fail in four different modes corresponding to bending stress in mid-span and shear stress in intervals 1, 2, and 3. Conditions ensuring that the failure probabilities in each mode are less than or equal to 0.001350 are included as constraints. The details about the corresponding limit-state functions and 23 other deterministic constraint can be found in Ref. [37]. It should be noted that the limit-state functions are nonlinear, but given by explicit expressions. The resulting reliability-based optimal design problem is solved using the algorithm presented above, with the nonlinear optimization subroutine given in Appendix A. We select to solve the equations $g_k(\mathbf{x}, \mathbf{u}) = 0$ for the standardized random variable corresponding to P_D. In the algorithm for solving $\mathbf{P}$, we use the parameters $\kappa = 0.0001$, $\eta = 0.01$, and $\gamma = 2$. Note that smaller η implies that the precision-adjustment rule in Eq. (12) becomes easier to pass. Hence, smaller η results in an initially slower increase in the number of sample points. The parameters in the subroutine (see Appendix A) were selected to be $\alpha_a = 0.5$, $\beta_a = 0.8$, and $\delta_a = 1$. The selected values of γ, α_a, β_a, and δ_a are standard for this type of algorithms.

The set $\mathcal{N} = \{40, 200, 1000, 5000, 25\,000, \ldots\}$. Hence, the algorithm uses initially 40 sample points, before the number is increased to 200, 1000, etc. We do not specify $\mathcal{N}$ beyond 25 000 because we plan to terminate the calculations when the number of sample points is increased beyond 25 000.

Initially, the sampling parameters μ_k are determined by performing five iterations of the iHLRF algorithm [44] for finding the design point of $g_k(\mathbf{x}, \mathbf{u})$ and setting μ_k equal to the last iterate. The sampling parameters μ_k are updated for each 25 iterations of the algorithm for solving $\mathbf{P}$ by

Table 3
Optimal design of reinforced concrete girder

	i					
	1	368	425	472	495	562
A_s (m^2)	0.010000	0.008916	0.008914	0.008935	0.008942	0.008954
b (m)	0.500	0.444	0.412	0.396	0.392	0.384
h_f (m)	0.500	0.355	0.382	0.399	0.403	0.411
b_w (m)	0.500	0.211	0.204	0.200	0.199	0.197
h_w (m)	0.500	0.842	0.816	0.800	0.797	0.789
A_v (m^2)	0.0005000	0.0001586	0.0001644	0.0001659	0.0001668	0.0001685
S_1 (m)	0.500	0.539	0.537	0.536	0.536	0.535
S_2 (m)	0.500	0.221	0.227	0.229	0.229	0.230
S_3 (m)	0.500	0.141	0.142	0.141	0.142	0.143
$c_0(\mathbf{x}_i)$	17.065	13.033	12.850	12.760	12.741	12.696

Table 4
Algorithm performance

	i					
	1	368	425	472	495	562
N_i	40	40	200	1000	5000	25 000
CPU (s)	0	203	310	689	1790	14 600
$p_{1,N_i}(\mathbf{x}_i;\mu_1,\sigma_1)$	0.015412	0.001336	0.001342	0.001346	0.001347	0.001349
$p_{2,N_i}(\mathbf{x}_i;\mu_2,\sigma_2)$	0.000000	0.001336	0.001343	0.001346	0.001347	0.001349
$p_{3,N_i}(\mathbf{x}_i;\mu_3,\sigma_3)$	0.000000	0.001336	0.001342	0.001346	0.001347	0.001349
$p_{4,N_i}(\mathbf{x}_i;\mu_4,\sigma_4)$	0.000237	0.001336	0.001342	0.001346	0.001347	0.001349
c.o.v. p_{k,N_i}	0.2	0.2	0.1	0.04	0.02	0.008
$p_{1,N^*}(\mathbf{x}_i;\mu_1,\sigma_1)$	0.014171	0.001266	0.001342	0.001362	0.001351	0.001349
$p_{2,N^*}(\mathbf{x}_i;\mu_2,\sigma_2)$	0.000000	0.001161	0.001176	0.001307	0.001315	0.001349
$p_{3,N^*}(\mathbf{x}_i;\mu_3,\sigma_3)$	0.000000	0.001147	0.001216	0.001354	0.001330	0.001349
$p_{4,N^*}(\mathbf{x}_i;\mu_4,\sigma_4)$	0.000277	0.001767	0.001402	0.001308	0.001358	0.001349

performing five iterations of the iHLRF algorithm. The parameters $\sigma_k = 1.01$ are kept constant for all iterations and limit-state functions.

The algorithm for solving **P** is implemented using Matlab [28], with the QP-solver 'quadprog', and the example is run on a 1.7 GHz laptop. The calculations are terminated after 562 iterations of the algorithm, when the algorithm is about to increase the number of sample points beyond 25 000. The design and corresponding cost after various number of iterations are summarized in Table 3. The iterates $i = 368$, 425, 472, and 495 correspond to the last iterates with $N_i =$ 40, 200, 1000, and 5000, respectively.

Table 4 shows the computing time (CPU) needed to reach the various iterations and the estimates of the failure probabilities using both N_i and $N* = 25\,000$ number of sample points with corresponding estimates of the coefficients of variation (c.o.v.). The convergence of the iterates $\mathbf{x}_i$ and the objective function $c_0(\mathbf{x}_i)$, and the increase in the number of sample points N_i are shown in Figs. 7–9, respectively.

We see from Table 4 and Figs. 7–9 that the number of sample points N_i is initially small but increases as less progress is made towards the solution of the approximating problems. In fact, most iterations are performed on problems involving only 40 sample points. This low number of sample points does not give an accurate estimate of the failure probabilities, but it is sufficient to direct the search towards an optimal design. This illustrates a significant advantage of the algorithm: Coarse estimates of the failure probabilities can be used until a reasonably good design is obtained. Using this design as a 'warm start', it is necessary to perform only a few iterations with high-precision, computationally expensive estimates of the failure probabilities to obtain a nearly optimal design.

In this example, we let the algorithm continue for 67 iterations using accurate failure probability estimates (c.o.v. of 0.008) until a highly accurate estimate of the optimal design was obtained. In practice, the additional computing time needed to obtain a highly accurate estimate of the optimal design may not be available or the additional effort may not be necessary. At iterations 472 and 495, the balance between accuracy and computing time is suitable for practical calculations. As seen from Table 4, those iterations

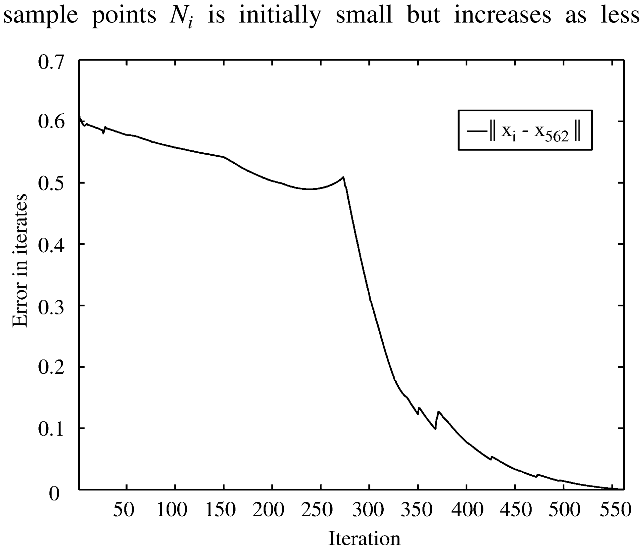

Fig. 7. Convergence in iterates $\mathbf{x}_i$.

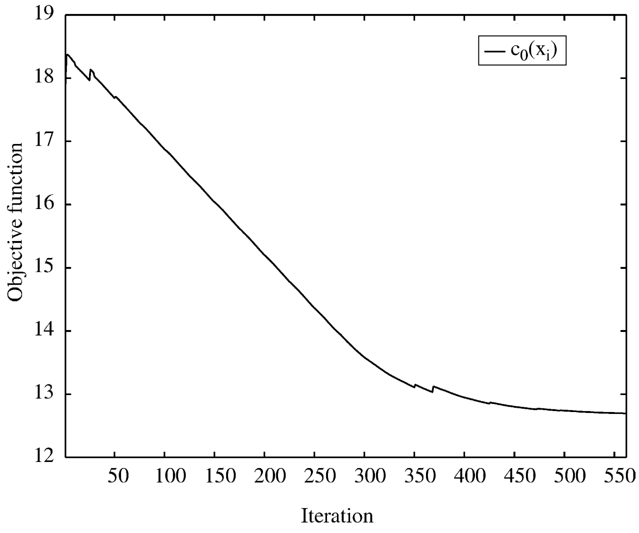

Fig. 8. Convergence in objective $c_0(\mathbf{x}_i)$.

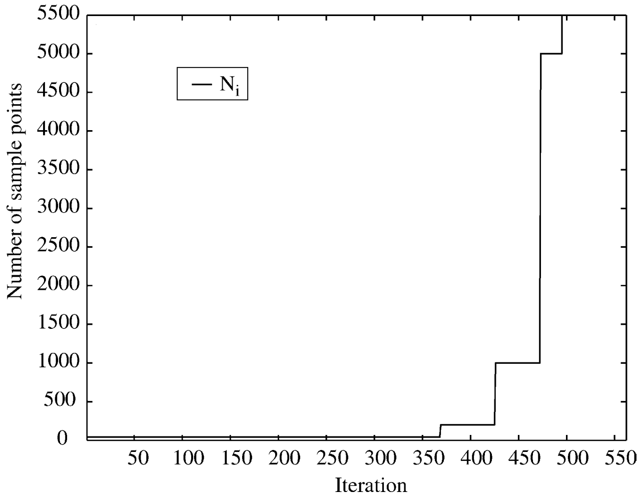

Fig. 9. Increase in number of sample points N_i.

are reached only after 11 and 30 min. The corresponding accuracy is quite good with coefficients of variation of 0.04 and 0.02 for the failure probabilities. As see from Table 4, the designs after iterations 472 and 495 may slightly violate the failure probability constraints $p_k(\mathbf{x}) \le 0.001\,350$. However, the estimates of the failure probabilities using N_{472} and N_{495} sample points are less than the bound 0.001 350, which in practice is considered sufficient.

8. Conclusions

We have developed an implementable algorithm for the solution of reliability-based optimal design problems based on Monte Carlo simulation and importance sampling. The algorithm is illustrated by an example from the area of highway bridge design. Contrary to existing algorithms, our algorithm is proven to converge to a solution of the problem with probability 1 under fairly general conditions. The algorithm, with its use of sampling techniques, yields a more accurate estimate of the optimal design than algorithms based on first-order reliability approximations.

The algorithm uses an adaptive scheme to control the precision of the failure probability estimates, which reduces the computing time significantly. Initially, only a small number of sample points is used to estimate the failure probability. This low number results in inaccurate estimates of the failure probability, but it is sufficient to direct the search towards an optimal design. As the algorithm progresses to a solution, the number of sample points is increased to obtain a high-quality solution. The derivations in this paper also led to a new sensitivity formula for the failure probability. This result is of importance in structural reliability analysis.

Our algorithm may require a large number of evaluations of the limit-state functions and their gradients. Hence, the algorithm is not applicable to problems with limit-state functions that are computationally costly to evaluate. In such cases, different types of approximations must be introduced [42]. For computationally inexpensive limit-state functions, the algorithm appears to be efficient, particularly when the importance sampling option is utilized. In view of the increasing speed of computers and the possibility for parallel processing, we expect the algorithm to be applicable to problems with moderately costly limit-state functions in the near future.

Acknowledgements

This work is sponsored by the Research Associateship Program at the National Research Council, the Taisei Chair in Civil Engineering at UC Berkeley, and the National Science Foundation under grant No. ECS-9900985. The authors are grateful for the valuable comments from Professor Armen Der Kiureghian, Mr Terje Haukaas, and Mr Junho Song at UC Berkeley.

Appendix A

The Polak-He algorithm, see section 2.6 in Ref. [33], can be used as a nonlinear optimization subroutine in the algorithm for solving $\mathbf{P}$. In Ref. [36], we show that the Polak-He algorithm satisfies the necessary requirements for the use as a subroutine. For completeness, we describe the Polak-He algorithm in the specialized form needed when solving $\mathbf{P}_N$.

For any positive integer N, sampling parameters $\mu_k \in \mathbb{R}^{m-1}$ and $\sigma_k > 0$, and sequence of realizations $\bar{\mathbf{u}}_1, \bar{\mathbf{u}}_2, \ldots, \bar{\mathbf{u}}_N$ of independent standard normal $(m-1)$-dimensional random vectors $\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \ldots, \bar{\mathbf{U}}_N$, we define one iteration of the Polak-He algorithm starting from $\mathbf{x}_i$ by the formula:

$$\mathbf{x}^* = \mathbf{x}_i + \lambda(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_i), \tag{A1}$$

where the Armijo step-size is given by

$$\lambda(\mathbf{x}_i) = \max_{k\in\{1,2,3,\ldots\}} \{\beta_a^k | F_N(\mathbf{x}_i, \mathbf{x}_i + \beta_a^k \mathbf{h}(\mathbf{x}_i); \mu, \sigma) \le \beta_a^k \alpha_a \theta(\mathbf{x}_i)\}, \tag{A2}$$

with $F_N(\mathbf{x}_i, \mathbf{x}_i + \beta_a^k \mathbf{h}(\mathbf{x}_i); \mu, \sigma)$ as in Eq. (11a), parameters $\alpha_a \in (0,1]$ and $\beta_a \in (0,1)$, and, for some parameters γ, $\delta_a > 0$

$$\theta(\mathbf{x}_i) = -\min_{\mathbf{z}\in\mathbf{Z}} \left\{ \mathbf{z}^{\mathrm{T}}\mathbf{b}(\mathbf{x}_i) + \frac{1}{2\delta_a}\mathbf{z}^{\mathrm{T}}\mathbf{A}(\mathbf{x}_i)^{\mathrm{T}}\mathbf{A}(\mathbf{x}_i)\mathbf{z} \right\}, \tag{A3}$$

with

$$\mathbf{Z} = \left\{ \mathbf{z} \in \mathbb{R}^{K+J+1} \,\middle|\, \sum_{l=1}^{K+J+1} z_l = 1,\ z_l \ge 0,\ l = 1, \ldots, K+J+1 \right\}, \tag{A4}$$

and (see Eq. (11c) for notation)

$$\mathbf{b}(\mathbf{x}_i) = \begin{pmatrix} \gamma\psi_N(\mathbf{x}_i;\mu,\sigma)_+ \\ \psi_N(\mathbf{x}_i;\mu,\sigma)_+ - p_{1,N}(\mathbf{x}_i;\mu_1,\sigma_1) + \hat{p}_1 \\ \vdots \\ \psi_N(\mathbf{x}_i;\mu,\sigma)_+ - p_{K,N}(\mathbf{x}_i;\mu_K,\sigma_K) + \hat{p}_K \\ \psi_N(\mathbf{x}_i;\mu,\sigma)_+ - f_1(\mathbf{x}_i) \\ \vdots \\ \psi_N(\mathbf{x}_i;\mu,\sigma)_+ - f_J(\mathbf{x}_i) \end{pmatrix} \tag{A5}$$

$$\mathbf{A}(\mathbf{x}_i)^{\mathrm{T}} = \begin{pmatrix} \nabla f_{0,N}(\mathbf{x}_i;\mu,\sigma)^{\mathrm{T}} \\ \nabla p_{1,N}(\mathbf{x}_i;\mu_1,\sigma_1)^{\mathrm{T}} \\ \vdots \\ \nabla p_{K,N}(\mathbf{x}_i;\mu_K,\sigma_K)^{\mathrm{T}} \\ \nabla f_1(\mathbf{x}_i)^{\mathrm{T}} \\ \vdots \\ \nabla f_J(\mathbf{x}_i)^{\mathrm{T}} \end{pmatrix}. \tag{A6}$$

Note that $\mathbf{b}(\mathbf{x}_i)$ is a $(K+J+1)$-vector and $\mathbf{A}(\mathbf{x}_i)$ is a $n\times(K+J+1)$ matrix.

Finally, the search direction

$$\mathbf{h}(\mathbf{x}_i) = -\frac{1}{\delta_a}\mathbf{A}(\mathbf{x}_i)\hat{\mathbf{z}}, \tag{A7}$$

where $\hat{\mathbf{z}}$ is any solution of Eq. (A5). The parameter γ in Eq. (A3) should be set equal to the value of the parameter γ in the algorithm for solving **P**. Note that the optimization problem in Eq. (A3) is with respect to $\mathbf{z}$ with $\mathbf{x}_i$ being fixed. The problem in Eq. (A3) is a quadratic optimization problem in $\mathbf{z}$ with positivity and one linear constraints. Hence, it can be solved in a finite number of iterations by a standard QP-solver such as 'quadprog' [28] or 'issol' [11].

References

[1] American Association of State Highway and Transportation Officials, Standard Specifications for Highway Bridges, Washington, DC, 15th edition; 1992.

[2] Beck JL, Chan E, Irfanoglu A, Papadimitriou C. Multi-criteria optimal structural design under uncertainty. Earthquake Eng Struct Dyn 1999; 28(7):741–61.

[3] Benveniste A, Metivier M, Priouret P. Adaptive algorithms and stochastic approximations. New York: Springer; 1990.

[4] Cheng FY, Ang AHS. Cost-efficient optimization for aseismic design criteria of RC buildings. In: Frangopol DM, editor. Case studies in optimal design and maintenance planning if civil infrastructure systems. Reston: ASCE; 1999.

[5] Ditlevsen O, Madsen HO. Structural reliability methods. New York: Wiley; 1996.

[6] Enevoldsen I, Sorensen JD. Reliability-based optimization in structural engineering. Struct Safety 1994;15(3):169–96.

[7] Ermoliev Y. Stochastic quasigradient methods. In: Ermoliev Y, Wets RJB, editors. Numerical techniques for stochastic optimization. New York: Springer; 1988. p. 141–86.

[8] Frangopol DM, Lin KY, Estes AC. Life-cycle cost design of deteriorating structures. Struct Eng 1997;123(10):1390–401.

[9] Gasser M, Schueller GI. Some basic principles in reliability-based optimization (RBO) of structures and mechanical components. In: Marti K, Kall P, editors. Stochastic programming methods and technical applications. Berlin: Springer; 1998.

[10] Gaivoronski A. Implementation of stochastic quasigradient methods. In: Ermoliev Y, Wets RJB, editors. Numerical techniques for stochastic optimization. New York: Springer; 1988. p. 313–52.

[11] Gill PE, Hammarling SJ, Murray W, Saunders MA, Wright MH. LSSOL 1.0 User's guide. SOL Report, Stanford, California: System Optimization Laboratory, Stanford University, No. 86-1; 1986

[12] Haukaas T. Finite Element sensitivity and reliability analysis of hysteretic degrading structures. PhD Dissertation, Berkeley: Departmant of Civil and Environmental Engineering, University of California; 2003.

[13] Higle JL, Sen S. Stochastic decomposition: an algorithm for two-stage linear programs with recourse. Math Oper Res 1991;16:650–69.

[14] Infanger G. Planning under uncertainty. Solving large scale stochastic linear programs, Boston: Boyd & Fraser; 1994.

[15] Itoh Y, Liu C. Multiobjective optimization of bridge deck maintenance. In: Frangopol DM, editor. Case studies in optimal design and maintenance planning if civil infrastructure systems. Reston: ASCE; 1999.

[16] King AJ, Rockafellar RT. Asymptotic theory for solutions in statistical estimation and stochastic programming. Math Oper Res 1993;18:148–62.

[17] King AJ, Wets RJB. Epi-convergence of convex stochastic programs. Stochastics 1991;34:83–91.

[18] Koo H. FORM, SORM and simulation techniques for nonlinear random vibration. PhD Dissertation, Berkeley: Department of Civil and Environmental Engineering, University of California; 2003.

[19] Kuschel N, Rackwitz R. Optimal design under time-variant reliability constraints. Struct Safety 2000;22(2):113–27.

[20] Lin KY, Frangopol DM. Reliability-based optimum design of reinforced concrete girders. Struct Safety 1996;18(2–3):239–58.

[21] Liu YW, Moses F. Truss optimization including reserve and residual reliability constraints. Comput Struct 1992;42(3):355–63.

[22] Mahadevan S. Probabilistic optimum design of framed structures. Comput Struct 1992;42(3):365–74.

[23] Mahadevan S, Raghothamachar P. Adaptive simulation for system reliability analysis of large structures. Comput Struct 2000;77: 725–34.

[24] Melchers R. Importance sampling in structural systems. Struct Safety 1989;6:3–10.

[25] Mak WK, Morton DP, Wood RK. Monte Carlo bounding techniques for determining solution quality in stochastic programs. Oper Res Lett 1999;24:47–56.

[26] Marti K. Robust Optimal design: a stochastic optimization problem. In: Marti K, editor. Stochastic optimization techniques. New York: Springer; 2002.

[27] Marti K. Solving stochastic structural optimization problems by RSM-based stochastic approximation methods—gradient estimation in case of intermediate variables. Math Meth Oper Res 1997;46: 409–34.

[28] MathWorks Inc. Matlab Version 6.5, Optimization Toolbox 2.2, Natick, MA; 2002. http://www.mathworks.com

[29] Mori Y, Ellingwood BR. Maintaining reliability of concrete structures. I: role of inspection/repair. Struct Eng 1994;120(3):825–45.

[30] Mori Y, Ellingwood BR. Maintaining reliability of concrete structures. II: optimum inspection/repair. Struct Eng 1994;120(3): 846–62.

[31] Nakamura H, Miyamoto A, Kawamura K. Optimization of bridge maintenance strategies using GA and IA techniques. In: Nowak AS, Szerszen MM, editors. Reliability and optimization of structural systems. Ann Arbor: Univeristy of Michigan; 2000.

[32] Niederreiter H. Random number generation and quasi-Monte Carlo methods. Philadelphia: SIAM; 1992.

[33] Polak E. Optimization. Algorithms and consistent approximations, New York: Springer; 1997.

[34] Pollard D. Convergence of stochastic processes. New York: Springer; 1984.

[35] Royset JO, Der Kiureghian A, Polak E. Reliability-based optimal structural design by the decoupling approach. RESS 2001;73: 213–21.

[36] Royset JO, Polak E. Implementable algorithm for stochastic programs using sample average approximations. Optimizat Theory Appl 2004; 122(1).

[37] Royset JO, Der Kiureghian A, Polak E. Reliability-based design optimization of series structural systems. UCB/SEMM Report, Berkeley: Department of Civil and Environmental Engineering, University of California, No. 15; 2002.

[38] Shapiro A. Stochastic programming by Monte Carlo simulation methods. Stochastic programming E-Print series; 2000. http://dochost.rz.hu-berlin.de/speps/

[39] Shapiro A, Homem-de-Mello T. A simulation-based approach to two-stage stochastic programming with recourse. Math Program 1998;81: 301–25.

[40] Schueller GI, Spanos PD. Monte Carlo simulation. Prob Eng Mech 2001;16:263–397.

[41] Thoft-Christensen P. On reliability-based structural optimization. In: Der Kiureghian A, Thoft-Christensen P, editors. Reliability and optimization of structural systems. Lecture notes in engineering, vol. 61. New York: Springer; 1991.

[42] Torczon V, Trosset MW. Using approximations to accelerate engineering design optimization. Proceedings of the Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 98-4800, St. Louis, Missouri; 1998.

[43] Tretiakov G. Stochastic quasi-gradient algorithms for maximization of the probability function. In: Marti K, editor. A new formula for the gradient of the probability function. Stochastic optimization techniques, New York: Springer; 2002.

[44] Zhang Y, Der Kiureghian A. Finite element reliability methods for inelastic structures. UCB/SEMM Report, Berkeley: Department of Civil and Environmental Engineering, University of California, Berkeley, no. 5; 1997.