Monotonicity and Active Set Strategies in Probabilistic Design Optimization

Kuei-Yuan Chan
Graduate Student
e-mail: chanky@umich.edu

Steven Skerlos
Associate Professor
e-mail: skerlos@umich.edu

Panos Y. Papalambros
Professor
e-mail: pyp@umich.edu

Department of Mechanical Engineering,
University of Michigan,
G.G. Brown Bldg.,
Ann Arbor, MI 48109
e-mail: pyp@umich.edu

Probabilistic design optimization addresses the presence of uncertainty in design problems. Extensive studies on reliability-based design optimization, i.e., problems with random variables and probabilistic constraints, have focused on improving computational efficiency of estimating values for the probabilistic functions. In the presence of many probabilistic inequality constraints, computational costs can be reduced if probabilistic values are computed only for constraints that are known to be active or likely active. This article presents an extension of monotonicity analysis concepts from deterministic problems to probabilistic ones, based on the fact that several probability metrics are monotonic transformations. These concepts can be used to construct active set strategies that reduce the computational cost associated with handling inequality constraints, similarly to the deterministic case. Such a strategy is presented as part of a sequential linear programming algorithm along with numerical examples. [DOI: 10.1115/1.2202887]

Keywords: monotonicity, active set strategy, RBDO, probabilistic design, optimization under uncertainty

1 Introduction

An important probabilistic formulation of the general nonlinear programming (NLP) problem is commonly known as the reliability-based design optimization (RBDO) problem. Design variables are mean values of the normal random variables $\mathbf{X}$ while the standard deviations $\boldsymbol{\sigma}_{\mathbf{X}}$ are fixed parameters. The objective is to minimize a function of $\boldsymbol{\mu}_{\mathbf{X}}$ subject to probabilities that the jth constraint violation is no bigger than a certain amount $P_{\mathrm{f},j}$. The RBDO problem is thus stated in negative null form as in Eq. (1), where no probabilistic equality constraints are defined

$$\begin{aligned} &\min_{\boldsymbol{\mu}_{\mathbf{X}}} f(\boldsymbol{\mu}_{\mathbf{X}}) \\ \text{subject to } &\Pr[g_j(\mathbf{X})>0]-P_{\mathrm{f},j}\le 0 \\ &\forall j\in\mathcal{K} \end{aligned} \tag{1}$$

Key challenges in RBDO are the evaluation of constraint probabilities and the integration of these probabilistic constraints into optimization algorithms. Many methods have been proposed to address these issues, including sampling and approximation techniques. Sampling techniques grounded on Monte Carlo simulation aim to provide accurate probability approximations by "smart" sampling within the design space; for example, see Refs. [1–3]. Approximation techniques use first- or second-order Taylor series expansions to approximate the original constraints. Probabilities of the design violating the approximate boundaries are calculated as the approximate probabilities to the original boundaries; such examples include the first-order second moment method (FOSM), the first-order reliability method (FORM), and the second-order reliability method (SORM) [4–9].

Satisfaction of probabilistic constraints within an optimization algorithm typically requires an auxiliary optimization process or "loop" associated with finding the most probable point (MPP), the point on the constraint boundary with minimal distance to the origin in the standard normal space. "Double loop" iterations, optimizing the design and finding the MPP, are expensive; for example, see Refs. [10–12]. Some methods combine the two loops into one by including the optimality conditions of the MPP auxiliary problem as equality constraints on the outer design optimization problem; for example, see Refs. [13–15]. Efficiency is improved without losing accuracy, but costs are still high.

Active inequalities are defined as those whose removal from the problem alters the location of the optimum [16], and usually they will be satisfied as equalities at the optimum. Active constraints manifest the design modes of failure, so they present indispensable information in design optimization. In the absence of equalities, as is the case here, the number of variables equals the degrees of freedom (DOF) of the optimal design problem. Clearly, the number of active constraints can be at most equal to the DOF of the original problem. If a constraint can be identified as active a priori the remaining degrees of freedom will be reduced by 1, making the problem easier to solve [17]. In many design applications the number of inequality constraints is significantly larger than the number of variables, and so most constraints will be inactive. Computational costs associated with evaluating inactive constraints can be very high, particularly in RBDO solution strategies.

Constraint activities are usually unknown until the optimum is found. An exception is NLP problems with functions that possess extensive monotonicity properties. Monotonicity analysis (MA) was developed to identify activities prior to computation, see Ref. [16] and references therein for MA development. Presence of global monotonicity can lead to definitive constraint activity identification and possibly obtain the optimum without iterative algorithms [16,18,19]. For design problems that are not globally monotonic, local or regional monotonicity can be used into effective active set strategies [20–24]. Several computer codes have been developed to automate MA within various optimization algorithms; for example, see Refs. [25–27].

Active set strategies are commonly employed in deterministic NLP algorithms [16,28,29]. At a given iteration point a subset of the (inequality) constraints is active satisfying the constraints as equalities, and is referred to as the (working) active set. In subsequent iterations constraints are activated or deactivated based on constraint violations and values of Lagrange multiplier estimates, respectively. The active set is updated until the correct one is

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received September 15, 2005; final manuscript received January 5, 2006. Review conducted by Shapour Azarm. Paper presented at the 2005 ASME International Mechanical Engineering Congress (IMECE2005), November 5, 2005–November 11, 2005, Orlando, Florida, USA.

confirmed at the optimum. The advantage is that only a subset of the constraints enters in the computations at each iteration, and the disadvantage is that convergence is theoretically assured if global optima are found for each active set. In practice this limitation is not too serious.

A typical algorithm employing active set strategy is a trust region sequential linear programming (SLP). Of particular interest here is an SLP algorithm devised to solve RBDO problems [15]. In this implementation the algorithm steps are as follows.

Step 0: Provide an initial design, all parameter values, and an initial adaptive trust region. This trust region is calculated as the maximum step length from the initial design to maintain feasibility.

Step 1: Create a deterministic LP subproblem of the original probabilistic NLP at the current design point $\boldsymbol{\mu}_{\mathbf{X}}^k$. Use FORM or SORM judiciously to compute a deterministic NLP local approximation to the probabilistic NLP. Choice of FORM or SORM depends on constraint activities and the relationships between local constraint curvatures, input variations, and the values of $\mathbf{P}_{\mathrm{f}}$. For convenience, let $g_j'(\boldsymbol{\mu}_{\mathbf{X}})$ be a continuous function satisfying Eq. (2)

$$\Pr[g_j(\mathbf{X}) > 0] - P_{\mathrm{f}} \equiv g_j'(\boldsymbol{\mu}_{\mathbf{X}}) \leq 0 \tag{2}$$

A constraint is called δ-active at iteration k if

$$g_j'(\boldsymbol{\mu}_{\mathbf{X}}^k) \geq -\delta, \tag{3}$$

where δ is a positive real number close to zero. To take advantage of FORM's efficiency, only δ-active constraints are approximated with SORM and the rest with FORM. For constraints identified as δ-active a further test is performed to examine their local curvature compared to the input variabilities and the $\mathbf{P}_{\mathrm{f}}$ values. FORM is applied to constraints judged locally linear, otherwise SORM is applied. The algorithm then constructs a linear subproblem at point $\boldsymbol{\mu}_{\mathbf{X}}^k$ using these deterministic equivalent constraints.

Step 2: Solve the LP subproblem using a standard LP solver to obtain a search step $\mathbf{s}^k$. A "trial" point $(\boldsymbol{\mu}_{\mathbf{X}}^k + \mathbf{s}^k)$ is checked for termination. If convergence criteria are satisfied, the optimum has been found, otherwise the algorithm continues to step 3.

Step 3: Determine if the LP solution from step 2 is acceptable by applying a filter. This filter accepts points that have improvements on either the objective function or the infeasibility, calculated as the maximum constraint violation. If the trial point is accepted by the filter, it becomes the new design point and the algorithm returns to step 1. If the trial point is not acceptable, then the algorithm proceeds to step 4.

Step 4: Reduce the trust region by half. Proceed to step 2 to re-solve the LP subproblem with the new trust region.

Efficiency of such SLP algorithms depends on the effectiveness of the employed active set strategy. This is a key motivation for the present work. Specifically, active set strategies are particularly effective in the presence o function monotonicity. In that case correct activity is identified quickly and contributes to overall algorithmic efficiency; for example in SLP and general convex approximation algorithms used for structural optimization applications with large number of constraints.

This article presents an extension of monotonicity analysis and attendant active set strategies to problems with probabilistic constraints. This extension of MA is first applied to ensure proper boundedness and allow problem complexity reduction before numerical treatment. Then the proposed active set strategy is applied to solve the problem. The goal is to reduce the computational cost associated with probabilistic constraint calculations. Previous similar ideas to reduce probabilistic constraint calculation cost include considering only the "most violated" constraint [30] or constraints known to be active a priori [14]. The approach here generalizes these ideas using MA and proposes algorithm modifications with active set strategies such that probabilistic constraints are calculated only when necessary. The material is organized as follows. Section 2 develops and extends concepts from deterministic to probabilistic problems, including global and local monotonicity analysis, dominance, and active set definitions. In Sec. 3 these concepts are used to compose an active set strategy used in conjunction with the above SLP algorithm [15]. Section 4 illustrates the approach with two analytical examples. Concluding remarks are offered in Sec. 5.

Table 1 Activity definitions

	g_j satisfied as a strict inequality at the optimum	g_j satisfied as an equality at the optimum
Removal of g_j **does not affect** the set of optimal solutions	**inactive**	**tight**
Removal of g_j **alters** the set of optimal solutions, but **does not** affect its objective function value	**weakly semi-active**	**strongly semi-active**
Removal of g_j **alters** both the optimal solution set and its objective function value	**weakly active**	**strongly active**

2 Concepts Related to Constraint Activity

In this section the concepts of constraint activity, monotonicity, active sets, and local dominance are developed for probabilistic constraints.

2.1 Activity of Probabilistic Constraints. In standard deterministic NLP an inequality constraint is active if removing the constraint changes the optimum [16]. For simplicity of presentation, assume that a unique optimum exists and let the feasible set $\mathcal{F}$ be the union of all constraint sets $\mathcal{K}_j = \{\mathbf{x} : g_j(\mathbf{x}) \leq 0\}$, $j = 1, \ldots, m$. The optimum with all inequality constraints present is defined as $\mathbf{x}^* = \arg\min f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{F}$, while the optimum with constraint g_j removed is defined as $\mathbf{x}_j^* = \arg\min f(\mathbf{x}), \mathbf{x} \in \mathcal{F} - \mathcal{K}_j$. By the definition of an optimum, $f(\mathbf{x}^*) \geq f(\mathbf{x}_j^*)$. Constraint g_j is active iff $f(\mathbf{x}^*) \neq f(\mathbf{x}_j^*)$. Pomrehn et al. [31] described various activity definitions for continuous constraints with continuous and discrete variables, shown in Table 1. These definitions are adopted and extended for probabilistic constraints. Activity of a probabilistic constraint can be defined similarly: a probabilistic constraint is active if removing the constraint changes the value of the optimum. For any design vector $\boldsymbol{\mu}_{\mathbf{X}}$, g_j' in Eq. (2) has the same feasibility information as the probabilistic constraint. Hence g_j' is a deterministic equivalent constraint function that can be used to represent the activity of the probabilistic constraint function $\Pr[g_j(\mathbf{X}) > 0] - P_{\mathrm{f},j}$. Assuming g_j' is a continuous, differentiable function, the definitions in Table 1 can be extended simply by replacing g_j by g_j'.

Algorithmically, an active set strategy identifies new constraint activities based on Lagrange multiplier estimated values $\boldsymbol{\lambda}$. However, obtaining Lagrange multipliers of probabilistic constraints is generally difficult in practice because g_j' is hard to calculate globally, if at all possible. Fortunately, in the neighborhood of a design point this transformation can be approximated locally using FORM or SORM depending on local curvatures [15]. As noted, the true $\boldsymbol{\lambda}$ is only computed at the optimum, and so the Lagrange multipliers are only estimates in the intermediate iterations. In an active set strategy, as $k \to \infty$ (or in practice as iterations proceed), $\boldsymbol{\mu}_{\mathbf{X}}^k \to \boldsymbol{\mu}_{\mathbf{X}}^*$, the multiplier estimates approach the true ones, $\boldsymbol{\lambda}^k \to \boldsymbol{\lambda}^*$, and the working set $\mathcal{G}^k$ becomes the true active set $\mathcal{G}^*$.

Although Table 1 presents several refined activity definitions, identifying these in algorithmic procedures is not practical, and algorithms using multipliers and function values will separate only strongly active constraints from inactive ones. This is sufficient for most engineering applications and is assumed to be the case here as well.

2.2 Global and Local Monotonicity Analysis. Monotonicity analysis is a formal methodology to examine if models are well constrained and to identify constraint activities. The first and the second monotonicity principles (MP1 and MP2) are as follows [16]:

MP1: In a well-constrained minimization problem every increasing variable is bounded below by at least one nonincreasing active constraint.

MP2: In a well-constrained minimization problem every nonobjective variable is bounded both below by at least one nonincreasing semiactive constraint and above by at least one nondecreasing semiactive constraint.

To apply these principles to probabilistic problems, monotonicity of functions of random variables must be studied. Three common types of functions of random variables are studied here: probabilities, expectations, and percentiles.

Let $g(x)$ be a continuous deterministic function of a single-variable x. We say $g(x^{+})$ is monotonically increasing with respect to x if

$$g(x_1) > g(x_2) \quad \forall\, x_1 > x_2.$$

Similarly, for decreasing functions indicated as $g(x^{-})$.

Proposition. Assume $g(x^{+})$ and let x become a random variable $X(\mu_X)$ with μ_X being the nominal design variable. Then, (i) the probability of constraint violation $\Pr[g(X(\mu_X))>0]$ *is monotonically increasing with respect to μ_X; (ii) the expectation $E[g(X)]$ is monotonically increasing with respect to μ_X; (iii) the percentile of $g(X)$ is monotonically increasing with respect to μ_X (similarly for monotonically decreasing functions). The proof of this proposition is as follows.*

Monotonicity of constraint violation. The monotonicity of the probabilistic function $\Pr[g(X(\mu_X))>0]$ is evaluated by its probability density function (PDF), Eq. (4)

$$\Pr[g(X(\mu_X)) > 0] = \int_0^{\infty} f_g(g)dg. \tag{4}$$

Consider X_1 and X_2 with nominal values $\mu_{X_1} \geq \mu_{X_2}$. The PDFs of $g(X_1)$ and $g(X_2)$, denoted as f_{g_1}, and f_{g_2}, respectively, can be calculated using functions of random variables [32] as

$$\begin{aligned} f_{g_i}(g)dg &= \sum_{j=1}^{nk} f_{X_i}(x^j)dx \\ &= f_{X_i}dx \quad \text{(for monotone functions)} \end{aligned}$$

$$\text{or } f_{g_i}(g) = f_{X_i}(x)\frac{dx}{dg} = \frac{f_{X_i}(x)}{dg/dx} \tag{5}$$

where $\mathbf{x}^j$ are the roots of $f_{X_i}=g$ for a given g value with nk being the total number of such roots. Comparing $\Pr[g(X_1)>0]$ and $\Pr[g(X_2)>0]$ we get

$$\begin{aligned} \int_0^{\infty} f_{g_1}(g)dg - \int_0^{\infty} f_{g_2}(g)dg &= \int_0^{\infty} \frac{f_{X_1}(x)}{dg/dx}dg - \int_0^{\infty} \frac{f_{X_2}(x)}{dg/dx}dg \\ &= \int_0^{\infty} \frac{(f_{X_1}(x) - f_{X_2}(x))}{dg/dx}dg \end{aligned} \tag{6}$$

Since $dg/dx>0$ due to $g(x^{+})$, and $\mu_{X_1} \geq \mu_{X_2}$ by construction, the difference in Eq. (6) is always positive. Thus the probability of constraint violation will have the same monotonicity as its deterministic counterpart.

Monotonicity of expectation. The expectation is defined as $E[g(X)]=\int_{-\infty}^{\infty} f_g \cdot g dg$. Let μ_{X_1} and μ_{X_2} be two nominal values with $\mu_{X_1} \geq \mu_{X_2}$; then

$$E(G_i) = E[g(X(\mu_{X_i}))] = \int_{-\infty}^{\infty} f_{g_i} g dg, \quad i=\{1,2\}.$$

Integrating by parts we obtain

$$\int_a^b g f_{g_i} dy = gF_{g_i}(g)\Big|_a^b - \int_a^b F_{g_i}(y)dy, \tag{7}$$

where F_{g_i} is the cumulative distribution function (CDF) of g_i. The first term of the right-hand side in Eq. (7) $gF_{g_i}(g)|_a^b=bF_{g_i}(b)-aF_{g_i}(a)$. The rate of $F_{g_i}(g)\to 0$ is much faster than $g\to-\infty$ and also the rate of $F_{g_i}(g)\to 1$ faster than $g\to\infty$. Thus $gF_{g_i}(g)|_a^b\to b$ when $a\to-\infty$ and $b\to\infty$. In addition, the monotonicity of $\Pr[g(X)>0]$ is the same as that of $g(X)$. Since $F_{g_i}(0)=\Pr[g(X)\leq 0]=1-\Pr[g(X)>0]$, the monotonicity of $F_{g_i}(y)$ is opposite than that of$g(X)$.

We can conclude that the monotonicity of the original function is preserved, if formulated as function expectation as follows:

$$\begin{aligned} E(G_1) - E(G_2) &= \int_{-\infty}^{\infty} f_{g_1} g dg - \int_{-\infty}^{\infty} f_{g_2} g dg \\ &= \left[yF_{Y_1}(y)\Big|_{-\infty}^{\infty} - yF_{Y_2}(y)\Big|_{-\infty}^{\infty} \right] \\ &\quad - \left[\int_{-\infty}^{\infty} F_{g_1}(g)dg - \int_{-\infty}^{\infty} F_{g_2}(g)dg \right] \\ &= -\int_{-\infty}^{\infty} [F_{g_1}(g) - F_{g_2}(g)]dg \end{aligned}$$

When $g(x^{+})$, $F_g(g)$ will be monotonically decreasing with respect to x. Equation (8) ensures that $E(G_1)-E(G_2)>0$, and the conclusion above is reached.

Monotonicity of percentiles. The pth percentile value g^p is formulated as

$$g^p: \int_{g^p}^{\infty} f_g dg = \Pr[g(X) > g^p] = p\%$$

Let g_1^p and g_2^p be the pth percentiles for $g(X_1)$ and $g(X_2)$, respectively. Then

$$F_{g_1}(g_2^p) \leq F_{g_2}(g_2^p) = p\,\% = F_{g_1}(g_1^p)$$

Since the CDF is always a monotonically increasing function, we conclude that $g_1^p \geq g_2^p$. This completes the proof of the proposition. □

With the above results, the extensions of MP1 and MP2 to probabilistic constraints is straightforward.

PMP1: In a well-constrained minimization problem if $f(\mu_{X_i}^{+})$ is increasing, there must exist at least one constraint $g_j'(\mu_{X_i}^{-})$ that bounds X_i from below.

PMP2: In a well-constrained minimization problem every nonobjective variable X_i must be bounded from below by at least one decreasing active constraint $g_k'(\mu_{X_i}^{-})$ and from above by at least one increasing active constraint $g_l'(\mu_{X_i}^{+})$.

Table 2 Monotonicity table

	x_1	x_2
f	+	−
g_1	+	−
g_2	−	−
g_3	−	+

As in the deterministic case, the MA principles hold for functions of vector variables, with monotonicities computed for each variable component. In a typical MA process active constraints will be identified by repeated application of MP1 and MP2. If exactly one constraint satisfies the above conditions that constraint will be always active (or "critical") in any active set strategy that will be algorithmically employed.

Example. Consider the example

$$\min_{\mu_{x_1},\mu_{x_2}} f(\boldsymbol{\mu}_{\mathbf{X}}) = \mu_{x_1} - \mu_{x_2}$$

$$\text{subject to } \Pr[g_1{:}2X_1 - 3X_2 - 10 > 0] \le 10\%$$

$$\Pr[g_2{:}-5X_1 - 2X_2 + 2 > 0] \le 10\%$$

$$\Pr[g_3{:}-2X_1 + 7X_2 - 8 > 0] \le 10\%$$

From the above, function monotonicity of functions will not change when placed in the probabilistic form. Therefore the monotonicity table can be easily created; see Table 2. MP1 shows g_3 to be critical to bound x_2, and g_2 critical to bound x_1. Thus we can conclude that the probabilistic optimum is [1.6659, 0.2860] without using any iterative algorithm.

2.3 Extended Active Sets in Trust Region Algorithms. As it is typical in mathematical programming, the previously defined working active set will have at most as many members as the number of degrees of freedom. However, strategies motivated by structural optimization problems employ active sets that include constraints that may become active in some subsequent iteration, and the membership of the active set is larger than the degrees of freedom. This is a heuristic that allows the algorithm to select a set of likely active constraints for the calculations of any given iteration and ignore a much larger number of constraints that are included in the problem statement. The aim is to reduce computation costs in, say, problems with large numbers of stress and deflection constraints but relatively few structural variables; see Refs. [16,33].

A similar idea presents itself when dealing with probabilistic constraints. Figure 1 illustrates the situation. At the current design point $\boldsymbol{\mu}_{\mathbf{X}}^k$, the deterministic equivalent constraints g_1' and g_2' are formed with given failure probability levels $\beta = -\Phi^{-1}(\mathbf{P}_f)$ and input variances $\boldsymbol{\sigma}_{\mathbf{X}}^2$. If a trust region algorithm is used, the trust region (or "move limit") is Δ^k and the current working set is $\mathcal{G}^k$. The linear approximations to g_1' and g_2' are $\hat{g}_1'^k$ and $\hat{g}_2'^k$. The distance from the current design point to the constraint approximation $\hat{g}_j'^k$ is d_j^k. In Fig. 1, the current design has distances $d_1^k > \Delta^k$ and $d_2^k < \Delta^k$. Since the algorithm step length $\|\mathbf{s}\|$ is restricted by the trust region, we know that constraint g_1' will not be violated at $\boldsymbol{\mu}_{\mathbf{X}}^{k+1}$, hence will not be active. Constraint g_2', on the other hand, can possibly be active. Thus an extended active set will contain only g_2'. Since only constraints in this set can be violated in the next iteration, feasibility of the next point need only be examined for these constraints, thus shaving computations.

Determining whether a constraint should be in the extended active set requires a relative scale and not an accurate calculation, so linear approximations $\hat{\mathbf{g}}'^k$ calculated via the first-order reliability method as in Eq. (8) will suffice.

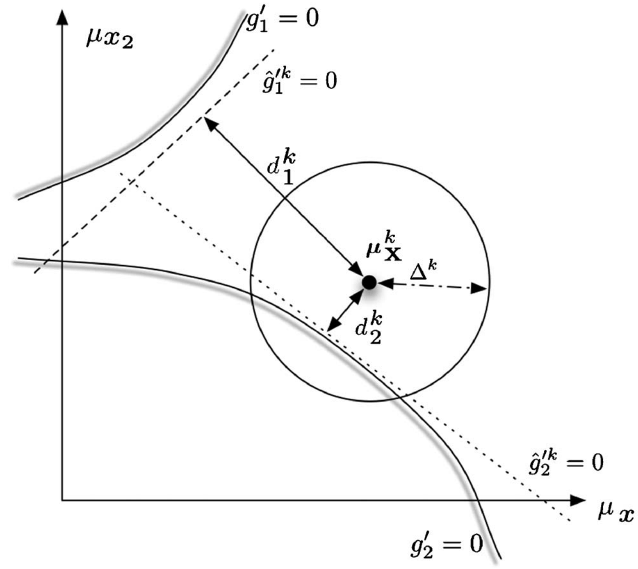

Fig. 1 Extended active set concept

$$g_j' \approx \hat{g}_j'^k \le 0 = \Phi\left(\frac{\mu_{\hat{g}_j}}{\sigma_{\hat{g}_j}}\right) - P_{f,j} \le 0$$
$$= g_j(\boldsymbol{\mu}_{\mathbf{X}}^k) + (\nabla g_j)^T \cdot (\boldsymbol{\mu}_{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{X}}^k) - (\nabla g_j)^T \cdot \boldsymbol{\sigma}_{\mathbf{X}} \Phi^{-1}(P_{f,j}) \le 0 \quad (8)$$

From the above follows that a constraint g_j' will not be active for the next iteration $(k+1)$ if $d_j^k \ge \Delta^k$. In fact, if the trust region never increases, g_j' will not be active up to the $(k+t)$th iteration where

$$t = \text{ceil}\left(\frac{d_j^k}{\Delta^k}\right) \quad (9)$$

Figure 2 illustrates this concept. Although the step $\mathbf{s}^k$ might not be orthogonal to $\hat{g}_j'$, Eq. (9) provides a conservative estimation.

At the kth iteration, the extended active set, produced using Eqs. (8) and (9), is denoted as $\mathcal{K}^k$. With the inclusion of extended and working active sets, the linearized subproblem of the original probabilistic NLP is now

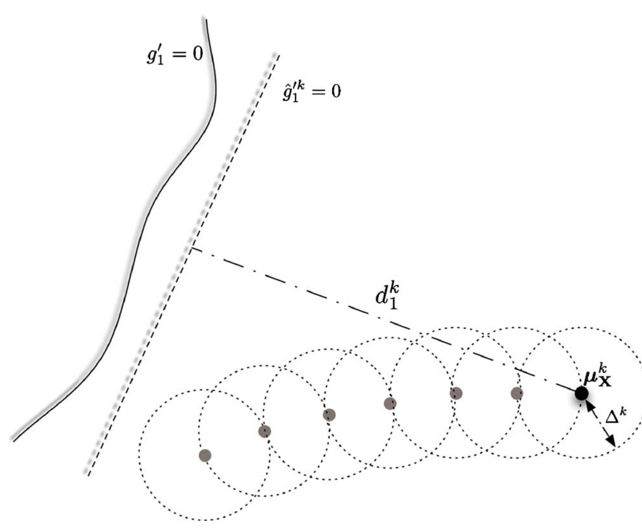

Fig. 2 Extended active set approximation

$$\min_{\mathbf{s}} \hat{f}^k(\mathbf{s})$$

$$\text{subject to } \hat{g}_j'^k(\mathbf{s}) \le 0 \quad \forall j \in \mathcal{K}^k$$

$$\hat{g}_j'^k(\mathbf{s}) = 0 \quad \forall j \in \mathcal{K}^k - \mathcal{G}^k$$

$$\|\mathbf{s}\| \le \Delta^k \tag{10}$$

2.4 Local Dominance for Active Set Updates. A deterministic constraint g_j in the negative null form dominates g_k if $g_k(\mathbf{x}) \le g_j(\mathbf{x}) \le 0,\ \forall \mathbf{x} \in \mathcal{F}$ [16]. Identifying global dominance is a rare treat, but local dominance can be effected using linear approximations. Let $\hat{g}_j'^k = A_j \boldsymbol{\mu}_{\mathbf{X}} + B_j$ be the linear approximation of $g_j'^k$. Given a unit direction $\vec{\mathbf{s}} = \mathbf{s}/\|\mathbf{s}\|$, the maximum feasible step length along direction $\vec{\mathbf{s}}$ is

$$\alpha_j^k = (\boldsymbol{\mu}_{\mathbf{X}}^k - A_j^{-1} B_j) \cdot \vec{\mathbf{s}}^{-1} \tag{11}$$

A constraint $\hat{g}_r'^k$ locally dominates another constraint $\hat{g}_t'^k$ if $\alpha_r^k \le \alpha_t^k$. The local dominance information (the α_j^k values) along with the trust region Δ^k and Eq. (9) are used to update the extended active set as follows.

- A constraint whose d_j^k value is larger than Δ^k is removed from the set.
- A constraint with negative α_j in Eq. (11) is removed from the set.
- A constraint is added to the set if the iteration counter satisfies Eq. (9).
- The progression of steps is recorded; if the iterate moves away from a constraint in the set [i.e., α_j in Eq. (11) remains negative], the constraint is removed; otherwise it remains in the set.

3 An SLP Active Set Strategy

The concepts developed in Sec. 3 can be incorporated into an active set strategy for the adaptive SLP algorithm given in Ref. [15] and outlined in the Introduction. For probabilistic constraints that are not active or that are locally linear, FORM is used; otherwise SORM is used. Constraint activity is determined using the δ-active concept. The choice of δ depends on the scale of g_j', which is usually unknown a priori. In the presence of many constraints and few degrees of freedom significant efficiencies can be gained using an extended active set. The active set strategy can update both the working and the extended sets as optimization progresses.

The flowchart of the resulting algorithm is shown in Fig. 3. At iteration k the original probabilistic NLP is approximated by a deterministic NLP. Probabilistic constraints in the extended active set are converted using FORM. Constraints in the working set require an additional local curvature check: If the local curvature is small, FORM is used; otherwise SORM is used. All other probabilistic constraints are excluded from the deterministic NLP and the conversion computations are avoided.

This deterministic NLP can be now stated as

$$\min_{\mathbf{s}} f^k(\mathbf{s})$$

$$\text{subject to } g_l'^k(\mathbf{s}) = 0 \quad \forall l \in \mathcal{G}^k$$

$$g_j'^k(\mathbf{s}) \le 0 \quad \forall j \in \mathcal{K}^k$$

$$\|\mathbf{s}\| \le \Delta^k \tag{12}$$

with constraints only those in the working and extended sets $\mathcal{G}^k$ and $\mathcal{K}^k$, respectively. Linearizing this problem at $\boldsymbol{\mu}_{\mathbf{X}}^k$ gives the deterministic LP subproblem

$$\min_{\mathbf{s}} \hat{f}^k(\mathbf{s})$$

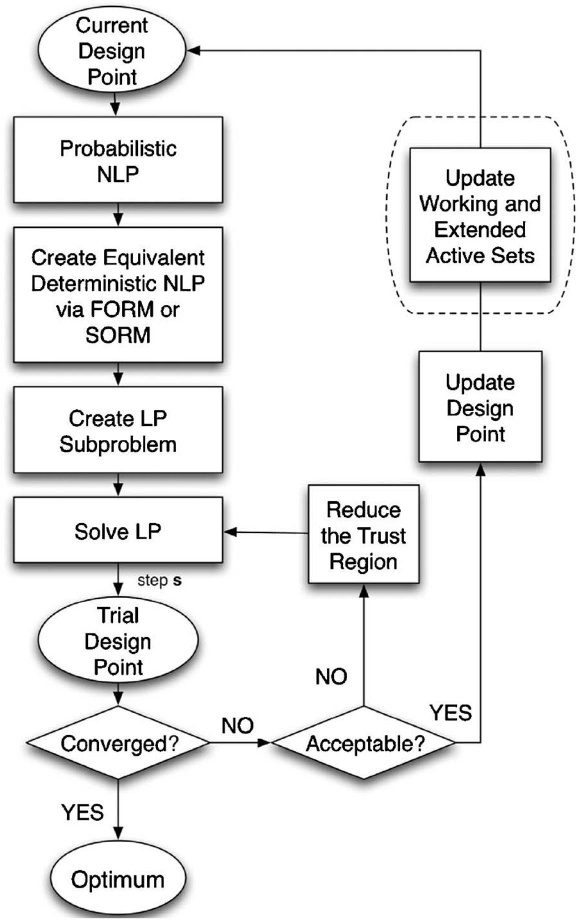

Fig. 3 Active set strategy for SLP [15] with added step in active sets update

$$\text{subject to } \hat{g}_l'^k(\mathbf{s}) = 0 \quad \forall l \in \mathcal{G}^k$$

$$\hat{g}_j'^k(\mathbf{s}) = 0 \quad \forall j \in \mathcal{K}^k$$

$$\|\mathbf{s}\| \le \Delta^k \tag{13}$$

Solution of this LP subproblem provides the iteration step $\mathbf{s}^k$ and the new design point will be $(\boldsymbol{\mu}_{\mathbf{X}}^k + \mathbf{s}^k)$. Following Refs. [15,34] the point is considered a "trial" one. Acceptance of a trial point is based on a filtering method. The objective function f^k and the infeasibility h^k are calculated using Eq. (14), and the pair $\{h^k, f^k\}$ is associated with each trial point

$$h^k = \max(0,\ \|g_l'^k(\boldsymbol{\mu}_{\mathbf{X}}^k + \mathbf{s}^k)\|,\quad g_j'^k(\boldsymbol{\mu}_{\mathbf{X}}^k + \mathbf{s}^k)). \tag{14}$$

Design k dominates design l if $h^k \le h^l$ and $f^k \le f^l$, and the kth pair is said to dominate the lth pair. A filter which contains all nondominant pairs of $\{h, f\}$ determines if the trial point is acceptable. If the trial point is rejected by the filter, the trust region is reduced by half and Eq. (13) is re-solved. The solution of the LP subproblem with the new trust region forms a new trial point and the algorithm iterates until the trial point is accepted by the filter and becomes the next design point $\boldsymbol{\mu}_{\mathbf{X}}^{k+1}$.

The dashed box in Fig. 3 illustrates how the active set strategy is applied in the SLP algorithm. The working set is updated at the new point using the typical active set rules: A constraint with negative Lagrange multiplier estimate is removed from the working set. If more than one constraint have negative multipliers, the one with the most negative value is removed. A violated constraint is added to the working set, and if more than one are violated the one violated the most is added. In summary,

Table 3 Monotonicity table of Example 1

	x_1	x_2	x_3	x_4	x_5	x_6
f	+	+	+	+	+	+
g_1	−	−	−	U	U	−
g_2	−	−	−	U	−	−
g_3		+		U	U	
g_4	U	+		−	−	U
g_5	−					
g_6	+					
g_7		−				
g_8		+				
g_9			−			
g_{10}			+			
g_{11}				−		
g_{12}				+		
g_{13}					−	
g_{14}					+	
g_{15}						−
g_{16}						+

+: function increasing
−: function decreasing
U: monotonicity undetermined

Table 4 Results with updating extended active set

Constraint	$\boldsymbol{\mu}_{\mathbf{X}}^*$	1st ite.	2nd ite.	3rd ite.	4th ite.	5th ite.	6th ite.	7th ite.	8th ite.	9th ite.
g_1	√							√	√	√
g_2		√	×	×	×	×	×	×	×	×
g_3			×	×	×	×	×	×	×	×
g_4				×	×	×	×	×	×	×
g_5							×	×	×	×
g_6			×	×	×	×	×	×	×	×
g_7	√					√	√	√	√	√
g_8			×	×	×	×	×	×	×	×
g_9	√		√	√	√	√	√	√	√	√
g_{10}			×	×	×	×	×	×	×	×
g_{11}	√				√	√	√	√	√	√
g_{12}			×	×	×	×	×	×	×	×
g_{13}						×	×	×	×	×
g_{14}	√						√	√	√	√
g_{15}					×	×	×	×	×	×
g_{16}	√			√	√	√	√	√	√	√

√ indicates activity
× indicates removal from extended active set

$$\mathcal{G}^{k+1} = \mathcal{G}^k - \mathcal{I}^k + \mathcal{E}^k \text{ where } \begin{cases} \hat{\lambda}_j^k(\boldsymbol{\mu}_{\mathbf{X}^*}^k) \le 0, & \forall\, j \in \mathcal{I}^k \\ \hat{g}_j'^k(\boldsymbol{\mu}_{\mathbf{X}^*}^k) > 0, & \forall\, j \in \mathcal{E}^k \end{cases}. \tag{15}$$

The extended active set is updated by the methods described in Sec. 3. The progression of values for $\mathbf{s}^k$ and the trust region are used to make the updates.

4 Illustrative Examples

Mathematical Example. The analytical example that follows is used to illustrate the procedures. Comprehensive engineering studies are reserved for future reporting. The example is taken from Hock and Schittkowski (#98) [35] and its deterministic formulation is shown in Eq. (16)

$$\min_{\mathbf{x}} f(\mathbf{x}) = 4.3x_1 + 31.8x_2 + 63.3x_3 + 15.8x_4 + 68.5x_5 + 4.7x_6$$

$$\text{subject to } g_1{:}-(17.1x_1 + 38.2x_2 + 204.2x_3 + 212.3x_4 + 623.4x_5 + 1495.5x_6 - 169x_1x_3 - 3580x_3x_5 - 3810x_4x_5 - 18500x_4x_6 - 24300x_5x_6) + 32.97 \le 0$$

$$g_2{:}-(17.9x_1 + 36.8x_2 + 113.9x_3 + 169.7x_4 + 337.8x_5 + 1385.2x_6 - 139x_1x_3 - 2450x_4x_5 - 16600x_4x_6 - 17200x_5x_6) + 25.12 \le 0 g_3{:}273x_2 + 70x_4 + 819x_5 - 26000x_4x_5 - 124.08 \le 0$$

$$g_4{:}-159.9x_1 + 311x_2 - 587x_4 - 391x_5 - 2918x_6 + 14000x_1x_6 - 173.02 \le 0$$

$$g_5{:}0 \le x_1 \le .31{:}g_6, \quad g_7{:}0 \le x_2 \le .046{:}g_8$$

$$g_9{:}0 \le x_3 \le .068{:}g_{10}, \quad g_{11}{:}0 \le x_4 \le .042{:}g_{12}$$

$$g_{13}{:}0 \le x_5 \le .028{:}g_{14}, \quad g_{15}{:}0 \le x_6 \le .0134{:}g_{16} \tag{16}$$

First we apply the active set strategy on the deterministic formulation using standard SLP. A probabilistic formulation of the problem is then created assuming all variables are uncorrelated normally distributed random variables. The adaptive SLP strategy is then used with the new active set strategy.

The deterministic formulation is a 6DOF problem with 16 constraints. The monotonicity table [16] is shown in Table 3. According to MP1 and MP2, no constraints can be concluded as critical. The proposed strategy converges in nine iterations and the solution is the same as in Ref. [35]. The working and extended active sets at each iteration are listed in Table 4. Constraints 1, 7, 9, 11, 14, 16 are active at the optimum. At the initial point $[0,0,0,0,0,0]$ constraint 2 is in the initial working set, and the rest in the extended set. The algorithm excludes constraints 2, 3, 6, 8, 10, 12 at the first iteration, 2, 3, 4, 6, 8, 10, 12, 15 at the second, and so on. At iteration 7, the correct active set is identified and no constraints remain in the extended set. The optimum design, $\mathbf{x}^* = [0.2686, 0, 0, 0, 0.028, 0.0134]$, the optimal objective function value, $f^* = 3.1358$, and the active set $\mathcal{G}^* = [g_1, g_7, g_9, g_{11}, g_{14}, g_{16}]$ are identical to the ones reported in Ref. [35]. The constraints involved in the optimization calculations are reduced from 16 at the 1st iteration to 6 at the 7th iteration without sacrificing accuracy.

The adaptive trust region at the initial point is 0.1. Since the objective function is linear, SLP converges to the optimum with superior efficiency without reducing the trust region in this example. Different starting points and different initial trust region choices will affect the number of iterations as well as the composition of the set, but will not affect the final optimum.

The probabilistic formulation of Eq. (16) is now given as

$$\min_{\boldsymbol{\mu}_{\mathbf{X}}} f = 4.3\mu_{X_1} + 31.8\mu_{X_2} + 63.3\mu_{X_3} + 15.8\mu_{X_4} + 68.5\mu_{X_5} + 4.7\mu_{X_6}$$

$$\text{subject to } \Pr[g_j(\mathbf{X}) > 0] \le P_{\mathrm{f},j} \text{ where } j = 1,2,3,40 \le \mu_{X_1} \le 0.31, \quad 0 \le \mu_{X_2} \le 0.046, \quad 0 \le \mu_{X_3} \le 0.0680 \le \mu_{X_4} \le 0.042, \quad 0 \le \mu_{X_5} \le 0.028, \quad 0 \le \mu_{X_6} \le 0.0134 \tag{17}$$

Using the monotonicity table for the deterministic problem we can infer that none of the probabilistic constraints in Eq. (17) are critical. Applying the adaptive SLP with the new active set strategy (Fig. 3) and with all standard deviations set at 0.001 (coefficient of variation with respect to the deterministic optimum is 0.3–7.5%) we obtain the results shown in Table 5 for different failure probability levels. Increased settings of failure probabilities lead to deteriorating optimal objective values; the optimum objective goes from 3.1358 with nearly 50% reliability (determinist solution) to 5.0625 with 90% reliability.

Note that different failure probabilities result in different active sets. This is due to the small feasible design space. Using one

Table 5 Example results with $\sigma_{\mathbf{X}}=0.001$

$\mathbf{P}_{\mathrm{f}}$	0.4	0.3	0.2	0.1
$\boldsymbol{\mu}_{\mathbf{X}}^{*}$	$\begin{bmatrix}0.3075\\0.0435\\0.0500\\0.0025\\0.0255\\0.0123\end{bmatrix}$	$\begin{bmatrix}0.3095\\0.0104\\0.0005\\0.0005\\0.0275\\0.0129\end{bmatrix}$	$\begin{bmatrix}0.3092\\0.0280\\0.0008\\0.0008\\0.0272\\0.0126\end{bmatrix}$	$\begin{bmatrix}0.3087\\0.0447\\0.0064\\0.0013\\0.0267\\0.0121\end{bmatrix}$
f^{*}	3.2579	3.6454	4.2055	5.0626
$\mathcal{G}^{*}$	g_1 g_7 g_9 g_{11} g_{14} g_{16}	1 6 9 11 14 16	1 6 9 11 14 16	1 6 8 11 14 16
$\Pr[g_j>0]$†	$\begin{bmatrix}0.4002\\0.0549\\0\\0\\0\\0\\0.3995\\0\\0.4000\\0\\0.3992\\0\\0\\0.3993\\0\\0.4001\end{bmatrix}$	$\begin{bmatrix}0.2992\\0.0252\\0\\0\\0\\0.3021\\0\\0\\0.3002\\0\\0.3007\\0\\0\\0.2996\\0\\0.3010\end{bmatrix}$	$\begin{bmatrix}0.2014\\0.0097\\0\\0\\0\\0.2006\\0\\0\\0.2006\\0\\0.1993\\0\\0\\0.2010\\0\\0.2011\end{bmatrix}$	$\begin{bmatrix}0.1003\\0.0027\\0\\0\\0\\0.9990\\0\\0.9820\\0\\0\\0.9971\\0\\0\\0.9969\\0\\0.1009\end{bmatrix}$

†: Monte Carlo simulation with 1 million samples

million Monte Carlo samples around the optimum point, one can confirm that the active set strategy successfully predicted the active constraints. Throughout the optimization process, only FORM is used because all constraints are linear or bilinear.

Ten-bar Truss Example. A ten-bar truss as shown in Fig. 4 is considered [36]. An external force is applied to node 2 with $F=10^4 kN$. All members have circular cross sections. Members $\{1,2,3,4,5,6\}$ are identical with length $l=914$ cm and members $\{7,8,9,10\}$ are also identical.

The design problem is to find the optimal radii of all bar members to minimize the overall weight of the truss system without any yielding and buckling occurring. The problem also constrains node-2 displacement to no bigger than 2 cm. The problem is summarized in Eq. (18), where r_1 and r_2 are the radii of the cross-section areas for bars 1–6 and 7–10, respectively. Yield strength E, moment of inertia I, resulting force F, and tensile stress σ all have subscript i indicating the number of the bar element. δ_2 is the node-2 displacement. Structural steel (ASTM-A36) with density $\rho=7860\ \mathrm{kg/m^3}$, modulus of elasticity $E=200$ GPa, and yield strength $Y=250$ MPa is used for this truss system.

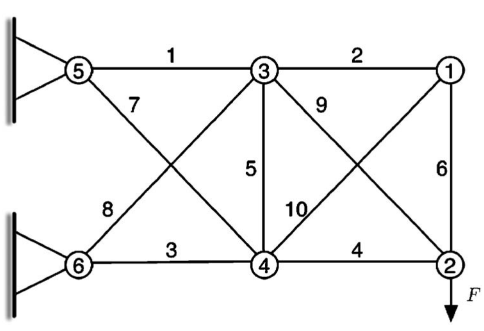

Fig. 4 Ten-bar truss

$$\min_{r_1,r_2} 6\cdot \pi r_1^2 l + 4\cdot \pi r_2^2 \sqrt{2} l$$

$$\text{subject to } F_i \le P^c = \frac{\pi^2 E_i I}{l^2} \quad \sigma_i \le Y_i \quad \delta_2 \le 2 \tag{18}$$

Finite element analysis is used to calculate the resulting forces and displacements on each member. The deterministic optimum $[r_1^*, r_2^*]=[11.4461, 12.6032]$ is obtained, with active constraints the stress on bar 9 and buckling on bar 8. Considering uncertainty from manufacturing, such that design variables r_1, r_2 are normally distributed random variables with $\boldsymbol{\sigma}=0.1$, this deterministic optimum has only a 45% reliability for satisfying bar 9 and 48% for satisfying bar-8 constraints.

A probabilistic solution with reliability of 99% is desired. At the starting point $[r_1, r_2]=[5,5]$, the active set strategy identifies that the displacement constraint, and constraints for bars $\{1,4,6,7,8,9,10\}$ are violated. Using a large enough move limit to maintain feasibility, the second iteration identifies only the constraints on bars $\{1,6,8,9\}$ as being in the working set. After five iterations, the active set contains constraints for bars $\{8,9\}$ and the problem converges to the optimum $\boldsymbol{\mu}_{\mathbf{X}}^{*}=[11.5103, 12.8542]$. Monte Carlo simulation with one million samples confirms that all constraints are satisfied with at least 99% reliability. The final truss system increases 2.67% in weight after variations are considered. Compared to using SLP without active set strategy [15], the total number of function evaluations is reduced from 168 to 51 with active set strategy. The original problem had 11 constraints and using the active set strategy the correct set was identified efficiently.

5 Concluding Remarks

The concepts of monotonicity analysis were extended to probabilistic problems in a straightforward manner. Monotonicity principles and dominance analysis can be used to identify active constraints just as in the deterministic case. Local monotonicity can be used in conjunction with a global one to create effective active set strategies. The active set strategy implemented here shows good potential for solving efficiently RBDO problems with a large number of constraints.

Some practical points of note are as follows. The working set update of removing and adding only one constraint at a time is done to avoid the zigzag effect [16,29]. All violated constraints can be included in the extended set. In fact it is suggested that the initial extended set contain all constraints not in the working set. Otherwise the algorithm will not have information to bring into the extended set constraints previously not in it. Also, an inaccurate initial working set will affect the speed of convergence. In the absence of other information, the initial working set should consist only of constraints identified as critical and those that happen to be satisfied as equalities at that point.

The proposed active set strategies strengthen the previously proposed SLP algorithm. The biggest advantage of applying an active set strategy is gained when the number of constraints is much larger than the number of design variables. In the resent implementation, the following assumptions are made:

(i) all random variables have independent Gaussian distributions;
(ii) design variables are the means of the random variables while keeping the standard deviations constant; and
(iii) local function curvature is no larger than second order.

The SLP algorithm does not require the constraint and objective functions to be linear or bilinear. By switching between FORM and SORM, the algorithm preserves the efficiency of FORM and the accuracy of SORM for nonlinear functions.

The two illustrative examples were selected such that the results are easy to replicate. From these examples, it is clear that the proposed algorithm reduces the number of function evaluations significantly. For a complex engineering application where simulations of constraints are expensive, this reduction can save a great amount of computation time.

As in the deterministic case, this type of SLP-based active set strategy will be effective in problems with extensive global monotonicity properties, such as several classes of structural optimization problems. More testing with such design problems is in order. Indeed, any algorithm that creates subproblems which maintain the monotonicity of the original functions should work well for such problems. The general class of convex approximation algorithms, including the method of moving asymptotes, would be a good candidate for extending the scope of the proposed active set strategy. The challenge will be the proper tailoring of the strategy to the logic of the particular algorithm.

Nomenclature

d_j^k = distance from kth design to constraint j
f_g = probability density function of $g(\mathbf{X})$
$f(\mathbf{x})$ = objective function to be optimized with respect to $\mathbf{x}$
$\mathcal{F}$ = feasible set
$\mathcal{G}^*$ = active set at the optimum
$\mathcal{G}^k$ = working set at kth iteration
$\mathbf{g}$ = deterministic constraints in negative null form
$\mathbf{g}'$ = equivalent deterministic constraints
$\mathcal{K}$ = constraint set
$\mathcal{K}^k$ = extended active set at the kth subproblem
k = iteration counter
m = number of constraints
n = number of design variables
$\mathbf{Pr}[\cdot]$ = probability of $\cdot$
$\mathbf{P_f}$ = failure probability levels
$\mathbf{s}$ = step vector
$\mathbf{x}$ = deterministic design variables
$\mathbf{X}$ = random design variables
α_j = the feasible step length for constraint j in $\mathbf{s}$ direction
Δ = trust region radius
δ = active constraint tolerance, positive value
$\boldsymbol{\lambda}$ = Lagrange multipliers
$\boldsymbol{\mu}_\mathbf{X}$ = mean values of random variables $\mathbf{X}$
$\boldsymbol{\sigma}_\mathbf{X}$ = standard deviations of random variables $\mathbf{X}$
$\|\cdot\|$ = Euclidean norm of $\cdot$

Acknowledgment

This research was partially supported by the Automotive Research Center, a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles at the University of Michigan. This support is gratefully acknowledged.

References

[1] Au, S. K., and Beck, J. L., 1999, "New Adaptive Importance Sampling Scheme for Reliability Calculations," Struct. Safety, **21**(2), pp. 135–158.
[2] Kalagnanam, J. R., and Diwekar, U. M., 1997, "An Efficient Sampling Technique for Off-Line Quality Control," Technometrics, **39**(3), pp. 308–319.
[3] Halder, A., and Mahadevan, S., 1999, *Probability, Reliability and Statistical Methods in Engineering Design*, John Wiley and Sons, New York.
[4] Hasofer, A. M., and Lind, N. C., 1974, "Exact and Invariant Second-Moment Code Format," J. Eng. Mech. Div., Am. Soc. Civ. Eng., **100**(EM1), pp. 111–121.
[5] Hohenbichler, M., and Rackwitz, R., 1983, "First-Order Concepts in System Reliability," Struct. Safety, **1**(3), pp. 177–188.
[6] Breitung, K., 1989, "Asymptotic Approximations for Probability Integrals," Probab. Eng. Mech., **4**(4), pp. 187–190.
[7] Ditlevsen, O., 1979, "Generalized Second Moment Reliability Index," J. Struct. Mech., **7**(4), pp. 435–451.
[8] Hohenbichler, M., and Rackwitz, R., 1981, "Non-normal Dependent Vectors in Structural Safety," J. Eng. Mech. Div., Am. Soc. Civ. Eng., **107**(6), pp. 1227–1238.
[9] Rackwitz, R., 2001, "Reliability Analysis—A Review and Some Perspective," Struct. Safety, **23**(4), pp. 365–395.
[10] Youn, B. D., Choi, K. K., and Park, Y. H., 2003, "Hybrid Analysis Method for Reliability-Based Design Optimization," ASME J. Mech. Des., **125**(2), pp. 221–232.
[11] Du, X., and Chen, W., 2004, "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," ASME J. Mech. Des., **126**(2), pp. 225–233.
[12] Choi, K. K., Tu, J., and Park, Y. H., 2001, "Extensions of Design Potential Concept for Reliability-Based Design Optimization to Nonsmooth and Extreme Cases," Struct. Multidiscip. Optim., **22**(5), pp. 335–350.
[13] Liang, J., Mourelatos, Z. P., and Tu, J., 2004, "A Single Loop Method for Reliability-Based Design Optimization," in *Proceedings of the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, Utah, September 28–October 2, DETC2004-57255.
[14] Chen, X., Hasselman, T. K., and Neill, D. J., 1997, "Reliability Based Structural Design Optimization for Practical Applications," in *Proceedings of the 38th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Part 4 (of 4)*, Vol. 4, pp. 2724–2732, Kissimmee, FL, USA, Apr. 7–10.
[15] Chan, K.-Y., Skerlos, S. J., and Papalambros, P. Y., 2006, "An Adaptive Sequential Linear Programming Algorithm for Optimal Design Problems with Probabilistic Constraints," to appear J. Mech. Des. (November).
[16] Papalambros, P. Y., and Wilde, D. J., 2000, *Principles of Optimal Design*, Cambridge University Press, New York, 2nd edition.
[17] Williams, B. C., and Cagan, J., 1996, "Active Analysis: Simplifying Optimal Design Problems Through Qualitative Partitioning," Eng. Optimiz., **27**(2), pp. 109–137.
[18] Hansen, P., Jaumard, B., and Lu, S. H., 1989, "Some Further Results on Monotonicity in Globally Optimal Design," ASME J. Mech., Transm., Autom. Des., **111**(3), pp. 345–352.
[19] Li, B., and Chen, Y., 1986, "Improved Optimum Design Technique Using Global Monotonicity Analysis," in *ASME Design Engineering Technical Conference*, p. 8, Columbus, OH.
[20] Papalambros, P. Y., and Wilde, D. J., 1980, "Regional Monotonicity in Optimum Design," ASME J. Mech. Des., **102**(3), pp. 497–500.
[21] Zhou, J., and Mayne, R. W., 1984, "Monotonicity Analysis and Recursive Quadratic Programming in Constrained Optimization," in *ASME Design Engineering Technology Conference*, p. 6, Cambridge, MA.
[22] Zhou, J., and Mayne, R. W., 1984, "Monotonicity Analysis and the Reduced Gradient Method in Constrained Optimization," ASME J. Mech., Transm., Autom. Des., **106**(1), pp. 90–94.
[23] Li, H. L., and Papalambros, P. Y., 1988, "Combined Local-Global Active Set Strategy for Nonlinear Design Optimization," ASME J. Mech., Transm., Autom. Des., **110**(4), pp. 464–471.
[24] Li, H. L., and Papalambros, P. Y., 1988, "Interior Linear Programming Algorithm Using Local and Global Knowledge," ASME J. Mech., Transm., Autom. Des., **110**(1), pp. 58–64.
[25] Azarm, S., and Papalambros, P. Y., 1984, "Automated Procedure for Local Monotonicity Analysis," ASME J. Mech., Transm., Autom. Des., **106**(1), pp. 82–89.
[26] Rao, J. R., and Papalambros, P. Y., 1991, "PRIMA. a production-based implicit elimination system for monotonicity analysis of optimal design models," ASME J. Mech. Des., **113**(4), pp. 408–415.
[27] Choy, J. K., and Agogino, A. M., 1986, SYMON: Automated symbolic monotonicity analysis system for qualitative design optimization. In *Proceedings of the ASME International Computers in Engineering Conference and Exhibition*, Vol. 1, 207–212, Chicago, IL, USA.
[28] Gill, E. P., Murray, W., and Wright, H. M., 1981, *Practical Optimization*, Academic Press, New York.
[29] Fletcher, R., 1987, *Practical Methods of Optimization*. John Wiley and Sons, New York, 2nd edition.
[30] Lee, J. J., and Lee, B. C., 2005, "Efficient evaluation of probabilistic constraints using an envelope function," Eng. Optimiz., **37**(2), pp. 185–200.
[31] Pomrehn, L. P., and Papalambros, P. Y., 1994, "Global and discrete constraint activity," ASME J. Mech. Des., **116**(3), pp. 745–748.
[32] Stark, H., and Woods, J. W., 2002, *Probability, Random Processes, and Estimation Theory for Engineers*, Prentice-Hall, New Jersey, USA, 3rd edition.
[33] Papalambros, P. Y., 1990, A knowledge-based framework for constraint activity identification in optimal design of aircraft structures. In M. Klieber, editor, *Artifical Intelligence: Applications in Computational Engineering*, pages 209–226, Ellis Horwood Ltd, New York.
[34] Fletcher, R., Leyffer, S., and Toint, Ph.-L., 1998, "On the global convergence of an SLP-filter algorithm," University of Dundee Report, **98**(13), pp. 1–11.
[35] Hock, W., and Schittkowski, K., 1980, *Test Examples for Nonlinear Programming Code*, Springer-Verlag, New York.
[36] Venkayya, V. B., 1971, "Design of optimum structures," Comput. Struct., **1**(1-2), pp. 265–309.