Reliability-Based Optimization Using Evolutionary Algorithms

Reliability-Based Optimization Using Evolutionary Algorithms

Kalyanmoy Deb, Shubham Gupta, David Daum, Jürgen Branke, Abhishek Kumar Mall and Dhanesh Padmanabhan

***Abstract*— Uncertainties in design variables and problem parameters are often inevitable and must be considered in an optimization task if reliable optimal solutions are sought. Besides a number of sampling techniques, there exist several mathematical approximations of a solution's reliability. These techniques are coupled in various ways with optimization in the classical reliability-based optimization field. This paper demonstrates how classical reliability-based concepts can be borrowed and modified and, with integrated single and multiobjective evolutionary algorithms, used to enhance their scope in handling uncertainties involved among decision variables and problem parameters. Three different optimization tasks are discussed in which classical reliability-based optimization procedures usually have difficulties, namely 1) reliability-based optimization problems having multiple local optima, 2) finding and revealing reliable solutions for different reliability indices simultaneously by means of a bi-criterion optimization approach, and 3) multiobjective optimization with uncertainty and specified system or component reliability values. Each of these optimization tasks is illustrated by solving a number of test problems and a well-studied automobile design problem. Results are also compared with a classical reliability-based methodology.**

***Index Terms*— Ditlevsen's bound, evolutionary multiobjective optimization, most probable point, Pareto-optimal front, reliability-based optimization, reliable front, system reliability.**

I. Introduction

FOR PRACTICAL optimization studies, reliability-based techniques are getting increasingly popular, due to the uncertainties involved in realizing design variables and stochasticities involved in various problem parameters. For a canonical deterministic optimization task, the optimum solution usually lies on a constraint surface or at the intersection of more than one constraint surface. However, if the design variables or some system parameters cannot be achieved exactly and are uncertain with a known probability distribution, the deterministic optimum (lying on one or more constraint surfaces) will fail to remain feasible on many occasions. In such scenarios, a stochastic optimization problem is usually formed and solved, in which the constraints are converted into probabilistic (or chance) constraints, meaning that the probability of failure (of being an infeasible solution) is limited to a prespecified value (say $(1 - R)$) [1], [2], where R is the specified reliability of the design.

Existing reliability-based optimization techniques differ in the manner they handle the probabilistic constraints. One simple approach is to use a Monte Carlo simulation technique to create a number of samples following the probability distribution to represent uncertainties and stochastitices in the design variables and problem parameters and evaluate them to compute the probability of failure [3]–[5]. However, such a technique becomes computationally expensive when the desired probability of failure is very small (say, one in a million).

Recently, optimization-based methodologies, instead of sampling methods, are suggested to take care of the probabilistic constraints. In these methods, stochastic variables and parameters are transformed into the standard normal variate space, and a separate optimization problem is formulated to compute the probability of failure and equate it with the desired value $(1 - R)$. At least three different concepts—double-loop methods, single-loop methods, and decoupled methods—exist. In this paper, we extend the double-loop method to be used with an evolutionary optimization technique. To handle multiple constraints, we borrow the system reliability concepts through the use of Ditlevsen's bounds to compute a more accurate probability of failure. Furthermore, we propose and use a computationally faster technique to compute the reliability estimate of a design. We apply the proposed methodology to three different types of optimization problems and demonstrate by solving test problems and an automobile design problem that the evolutionary optimization techniques are good candidates for reliability-based design. Results are compared with a couple of classical methods, and the advantages and disadvantages of them are discussed. This paper clearly brings out problem domains in which reliability-based evolutionary algorithms will have an edge over their classical counterparts and should encourage more such studies.

Manuscript received October 16, 2007; revised March 7, 2008, October 25, 2008, and January 6, 2009; accepted January 8, 2009. First version published August 7, 2009; current version published September 30, 2009. This work was supported in part by the India Science Laboratory, General Motors Research and Development, Bangalore. The work of K. Deb was supported by the Academy of Finland and Foundation of Helsinki School of Economics under Grant 118319.

K. Deb is with the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208016, India and also with the Helsinki School of Economics, Helsinki, Finland (e-mail: deb@iitk.ac.in).

S. Gupta is with the Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: shubhamg@mit.edu).

D. Daum is with the Solar Energy and Building Physics Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland (e-mail: david.daum@epfl.ch).

J. Branke is with the Warwick Business School, The University of Warwick Coventry, CV4 7AL, U.K. (e-mail: juergen.branke@wbs.ac.uk).

A. K. Mall is with Deutsche Bank, Mumbai, India (e-mail: akmall.iitk@gmail.com).

D. Padmanabhan is with the Decision Support and Analytic Services Division, Hewlett Packard, Bangalore, India (e-mail: dhanesh.padmanabhan@hp.com).

Digital Object Identifier 10.1109/TEVC.2009.2014361

The paper is structured as follows. Section II introduces the reliability-based optimization problem and describes currently used classical reliability-based methodologies. Further related work, in particular in the area of evolutionary computation, is surveyed in Section III. Then, three possible scenarios for reliability-based optimization are described in Section IV. Our evolutionary approach describing the computationally faster technique is presented in Section V. Then, Sections VI to VIII report empirical results of our approach on the aforementioned three scenarios with a comparison to classical approaches. The paper concludes with a summary and some ideas for future work in Section IX.

II. Problem Definition and Classical Reliability-Based Methodologies

A. Problem Definition

Let us consider here a reliability-based single-objective optimization problem of the following type:

$$\begin{array}{ll} \underset{(\mathbf{x},\mathbf{d})}{\text{Minimize}} & f(\mathbf{x},\mathbf{d},\mathbf{p}) \\ \text{subject to} & g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0, \quad j=1,2,\ldots,J \\ & h_k(\mathbf{d}) \geq 0, \quad k=1,2,\ldots,K \\ & \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \\ & \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}. \end{array} \tag{1}$$

Here, $\mathbf{x}$ is a set of design variables which are uncertain. That is, for a particular vector $\mu_{\mathbf{x}}$ considered in the optimization, the realized value is distributed with a probability distribution. In our discussion here, we shall assume a normal distribution $N(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$ with mean $\mu_{\mathbf{x}}$ and a covariance matrix $\sigma_{\mathbf{x}}$, which is dependent on the variable vector value $\mu_{\mathbf{x}}$. Appropriate transformation techniques are available to consider other probability distributions as well [4]. Similarly, $\mathbf{p}$ is a set of uncertain parameters (which are not design variables) and follow a probability distribution $N(\mu_{\mathbf{p}}, \sigma_{\mathbf{p}})$ representing the uncertainty. However, $\mathbf{d}$ is a set of deterministic design variables, which are not uncertain and can be realized as they are specified exactly. Thus, the stochasticity in the optimization problem comes from two sets of parameters: $\mathbf{x}$ and $\mathbf{p}$. However, although the above problem is written in a way to mean that $\mathbf{x}$ and $\mathbf{d}$ are decision variable vectors to the optimization problem, in reality, $\mu_{\mathbf{x}}$ and $\mathbf{d}$ are decision variable vectors. In most cases, fixed covariance vectors are used for $\mathbf{x}$ and $\mathbf{p}$, or covariances as known functions of $\mathbf{x}$ and $\mathbf{p}$ are assumed.

Here, we only consider inequality constraints. This is because if an equality constraint involves $\mathbf{x}$ or $\mathbf{p}$, there may not exist a solution for any arbitrary desired reliability against failure. All inequality constraints can be classified into two categories: 1) stochastic constraints g_j involving at least one stochastic quantity ($\mathbf{x}$, $\mathbf{p}$ or both) and 2) h_k involving no stochastic quantity.

Fig. 1 shows a hypothetical problem with two stochastic inequality constraints. Typically, the deterministic optimal solution [the solution to the problem given in (1) without any uncertainty in $\mathbf{x}$ or $\mathbf{p}$] lies on a particular constraint boundary or at the intersection of more than one constraints, as shown in the figure. In the event of uncertainties in design variables, as shown in the figure with a probability distribution

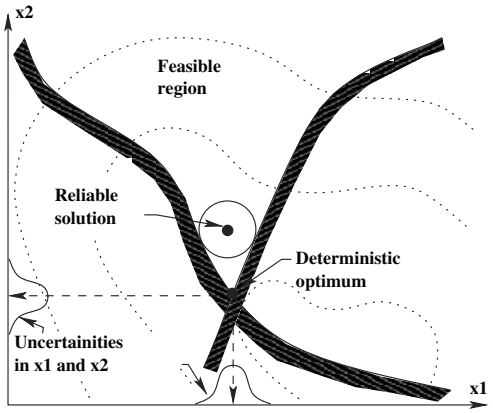

Fig. 1. Concept of reliability-based optimization procedure.

around the optimal solution, in many instances, such a solution will be infeasible. In order to find a solution that is more reliable (meaning that there is a small probability of resulting in an infeasible solution), the true optimal solution must be sacrificed, and a solution interior to the feasible region may be chosen. For a desired reliability measure R, it is then desired to find that feasible solution that will ensure that the probability of having an infeasible solution instance created through uncertainties from this solution is at most $(1-R)$. To arrive at such a solution, the above optimization problem can be converted into a new optimization problem. Since the objective function f and constraints g_j are probabilistic due to the randomness in variable set $\mathbf{x}$ and parameter set $\mathbf{p}$, the following deterministic formulation can be made:

$$\begin{array}{ll} \underset{(\mu_{\mathbf{x}},\mathbf{d})}{\text{Minimize}} & f(\mu_{\mathbf{x}},\mathbf{d},\mu_{\mathbf{p}}) \\ \text{subject to} & P(\bigwedge_{j=1}^{J}(g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0)) \geq R \\ & h_k(\mathbf{d}) \geq 0, \quad k=1,2,\ldots,K \\ & \mathbf{x}^{(L)} \leq \mu_{\mathbf{x}} \leq \mathbf{x}^{(U)}, \\ & \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)} \end{array} \tag{2}$$

where $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{p}}$ denote the mean of variables $\mathbf{x}$ and $\mathbf{p}$, respectively. The term $P()$ signifies the joint probability of the solution $\mathbf{x}$ being feasible from all J constraints under the uncertainty assumption. The quantity R is the desired reliability (within [0, 1]) for satisfying all the constraints. The conversion of the constraint $g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0$ into a probabilistic constraint with the introduction of a reliability term is a standard technique and the transformed probabilistic constraint is also known as a *chance* constraint. However, finding the joint probability of a solution being feasible from multiple constraints is a difficult mathematical proposition and approximate methods are used to make an estimate of the above probability. We discuss some of the commonly used procedures in Section II-B and shall discuss a couple of ways of handling the joint probability term for multiple constraints later in Section II-D. Many reliability-based optimization studies simply break the above probability constraint into J chance constraints as follows:

$$P(g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0) \geq R_j, \quad j=1,2,\ldots,J \tag{3}$$

where R_j is the desired probability of constraint satisfaction of the jth constraint. Of course, this requires the definition

of sensible reliabilities for each individual constraint, while a decision maker is usually only interested in the overall system reliability R.

A nice matter about the optimization problem given in (2) with the joint probability term or the individual probability term as given in (3) is that constraints associated with the optimization problem are now deterministic, and hence, any existing optimization methodology can be used to solve the problem. What remains to be done is a computational procedure to estimate the probability $P()$. As we see next, this is not an easy matter, and most of the remainder of this paper is devoted to estimating this probability in an accurate and computationally efficient manner.

B. Determining a Solution's Reliability

Ideally, the reliability of a solution must be determined by checking whether the solution is adequately safe against all constraints. Since this is mathematically and computationally a challenging task [6], we defer the discussion on simultaneous consideration of all constraints till Section II-D and first discuss the procedures for computing reliability of a solution against a single constraint (3).

Mathematically, the probability of a solution $\mathbf{x}$ being safe against the jth constraint (or $P(g_j(\mathbf{x}, \mathbf{d}, \mathbf{p}) \geq 0)$) can be written as $(1 - P_j)$, with

$$P_j(\mu_{\mathbf{x}}, \mathbf{d}, \mu_{\mathbf{p}}) = \int_{g_j(\mathbf{x},\mathbf{d},\mathbf{p})<0} \varphi_{\mu_{\mathbf{x}},\mu_{\mathbf{p}}}(\mathbf{x}, \mathbf{p}) d\mathbf{x} d\mathbf{p} \quad (4)$$

where P_j is the failure probability, and $\varphi_{\mu_{\mathbf{x}},\mu_{\mathbf{p}}}$ is the joint probability density function of $(\mathbf{x}, \mathbf{p})$. However, it is usually impossible to find an analytical expression for the above integral for any arbitrary constraint function which forces researchers to follow one of the following two approximate procedures: 1) statistical approximation by sampling or 2) optimization-based procedures by estimating a *distance* of the solution from the constraint. We discuss both of these approaches one by one.

1) Sampling-Based Reliability Measures: In this procedure, N different sample solutions are created by following the known joint probability distribution of variation of $\mathbf{x}$ and $\mathbf{p}$. Thereafter, for each sample, the constraint g_j can be evaluated and checked for its violation. If r_j cases (of N) do not satisfy the constraint, $P_j = (r_j/N)$ and the probabilistic constraint $P(g_j(\mathbf{x}, \mathbf{d}, \mathbf{p}) \geq 0)$ can be substituted by a deterministic constraint as follows:

$$1 - \frac{r_j}{N} \geq R_j. \quad (5)$$

An advantage of this approach is that it can be used to handle multiple constraints by simply checking the feasibility of samples on all constraints. Such a method is simple and works well if the desired reliability R_j is not too close to one [7], [8]. However, a major bottleneck of this approach is that the sample size N needed for finding the quantity r_j must be of the order of at least $O(1/(1 - R_j))$, such that at least one infeasible case is present in the sample. For a very stringent reliability requirement, such as for a limiting failure probability $(1 - R_j)$ of $O(10^{-6})$, a large sample size $(N \sim O(10^6))$ is required to compute r_j. This may be computationally too expensive to be of any practical use.

The number of necessary samples can be somewhat reduced by using a more systematic sampling, e.g., Latin hypercube sampling [5], importance sampling [9], or directional sampling [10] (see also [4]). Wang *et al.* [11] proposed a combination of sampling and meta-modeling. Their approach applies a discriminative sampling strategy, which generates more points close to the constraint function. Then, in the neighborhood of the constraint function, a kriging model is built, and the reliability analysis is performed based on this model.

However, even these improvements may not be sufficient to render the approach applicable if the desired reliability is large.

2) Optimization-Based Reliability Measures: The underlying idea of this class of reliability measures is to determine a point on the constraint boundary which is closest to the solution. This point is usually called the "most probable point" (MPP) of failure [12]. Assuming a single constraint, and approximating it as being linear in the vicinity of the MPP, a solution's reliability can then be calculated. Because of the assumption of linearity, these methods are also known as first-order reliability methods (FORMs).

To do so, we first convert the $\mathbf{X}$ coordinate system into an independent standard normal coordinate system $\mathbf{U}$, through the Rosenblatt transformation [13]. The standard normal random variables are characterized by zero mean and unit variance. In this space, we approximate the hyper-surface $(g_j(\mathbf{x}, \mathbf{d}, \mathbf{p}) = 0$ or equivalently $G_j(\mathbf{U}) = 0)$ by a first-order approximation at the MPP. In other words, the MPP corresponds to a reliability index β_j, which makes a first-order approximation of $P_j = \Phi(-\beta_j)$, where $\Phi()$ is the standard normal density function.

The remainder of this section discusses some alternatives to calculate the MPP.

a) Performance measure approach (PMA): To find the MPP in the PMA approach, the following optimization problem is solved [4]:

$$\begin{array}{ll} \text{Minimize} & G_j(\mathbf{U}) \\ \text{subject to} & \|U\| = \beta_j^r \end{array} \quad (6)$$

where β_j^r is the required reliability index computed from the required reliability R_j as $\beta_j^r = \Phi^{-1}(R_j)$. The above formulation finds a $\mathbf{U}^*$ point which lies on a circle of radius β_j^r and minimizes $G_j(\mathbf{U})$. The original probability constraint is replaced by

$$G_j(\mathbf{U}^*) \geq 0. \quad (7)$$

Fig. 2 illustrates this approach on a hypothetical problem.

The figure shows a probabilistic constraint g_j in the $\mathbf{U}$-space (for ease of illustration, two variables are considered here). The corresponding constraint $G_j(u_1, u_2)$ and the feasible region are shown. The circle represents solutions that correspond to a reliability index of β_j^r. Thus, the PMA approach finds a point $\mathbf{U}^*$ on the circle for which the function $G_j(\mathbf{U})$ takes the minimum value. Then, if the corresponding constraint function value is non-negative (or, $G_j(\mathbf{U}^*) \geq 0$), the probabilistic constraint $P(g_j(\mathbf{x}, \mathbf{d}, \mathbf{p}) \geq 0) \geq R_j$ is considered to have been satisfied.

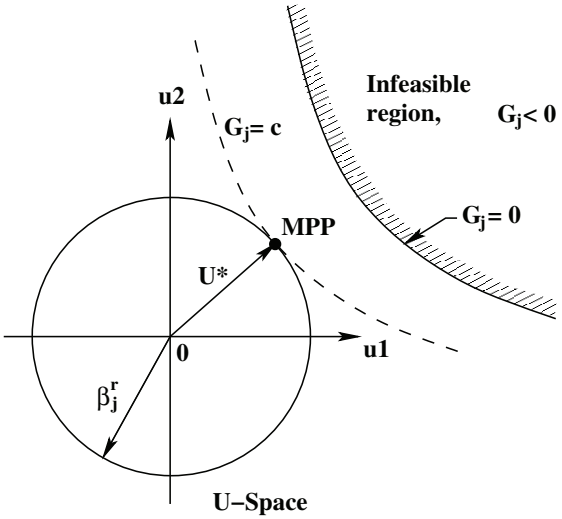

Fig. 2. PMA approach.

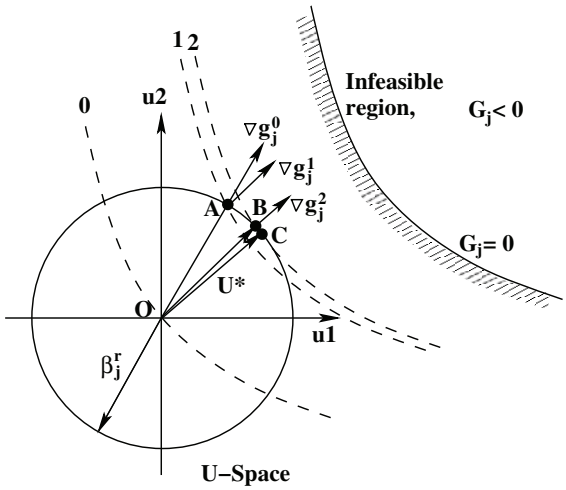

Fig. 3. Fast approach for solving the PMA problem.

Although the above optimization problem involves an equality constraint, a customized optimization procedure can be employed to consider solutions only on the $\|U\| = \beta_j^r$ hypersurface, thereby making every solution a feasible solution. Such a customized algorithm will make the search process comparatively faster.

b) Fast performance measure approach (FastPMA): A faster variant of the PMA approach is suggested in [14] and is illustrated in Fig. 3. To speed up PMA, a gradient vector ∇g_j^0 of each probabilistic constraint g_j is first computed at the origin of the **U**-space. Its intersection (point A) with a circle of radius β_j^r is computed and a new gradient (∇g_j^1) is recomputed at this point (A). Thereafter, the intersection (point B) of this new gradient direction from the origin with the circle is recomputed and a new gradient vector (∇g_j^2) is computed at B. This procedure is continued till a convergence of the norm of two consecutive gradient vectors with a predefined tolerance (ϵ_{PMA}) or a fixed number of iterations η_{PMA} is met. This point $(\mathbf{U}^*)$ is an estimate of the MPP of the original PMA problem.

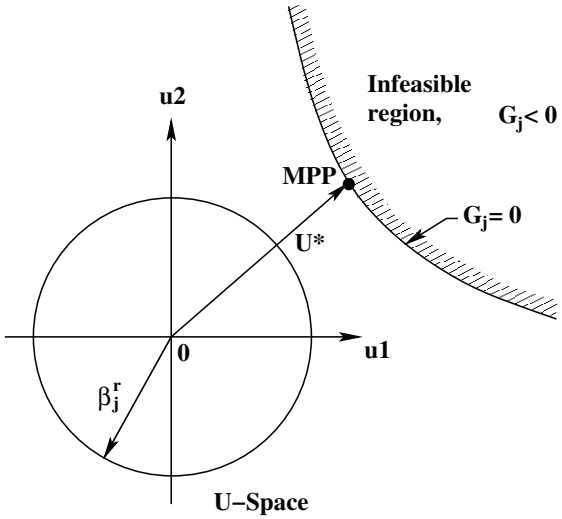

Fig. 4. RIA approach.

c) Reliability index approach (RIA): In this method, the following optimization problem is solved:

$$\begin{aligned} &\text{Minimize} \quad \|\mathbf{U}\| \\ &\text{subject to} \quad G_j(\mathbf{U}) = 0. \end{aligned} \tag{8}$$

Here, the MPP is calculated by finding a point which is on the constraint curve in the **U**-space and is closest to the origin. The optimum point $\mathbf{U}^*$ is used to replace the original probability constraint as follows:

$$\|\mathbf{U}\| \geq \beta_j^r. \tag{9}$$

Fig. 4 illustrates the procedure. During the optimization procedure, the desired reliability index β_j^r is ignored, and the minimum **U**-vector on the constraint boundary is found. Thereafter, the minimal $\mathbf{U}^*$ is compared with β_j^r.

This approach also involves an equality constraint. Although this method is computationally more expensive than the PMA approach, a nice aspect is that the optimization problem directly returns the *distance* of the solution from the constraint (which is directly related to the reliability against a violation of the constraint). The PMA approach, on the other hand, only determines whether a solution is reliable or not against constraint satisfaction with respect to a specified reliability index.

d) Fast reliability index approach (FastRIA): There can also be a relatively fast yet less-accurate variant of RIA, which we propose here. First, we find an intermediate MPP point $(\mathbf{U}_{PMA}^*)$ on a *unit circle* (assuming $\beta_j^r = 1$) based on the above FastPMA approach. As discussed, this operation is computationally fast. Thereafter, we perform a unidirectional search along $\mathbf{U}_{PMA}^*$ and locate the point for which $G_j(\mathbf{U}) = 0$. We employ the Newton–Raphson approach for performing the unidirectional search [15]. Due to the conversion of the original multivariable problem to a single-variable problem, the computation is usually fast, requiring only a numerical derivative of the constraint function in the **U**-space. However, it is worth mentioning here that the MPP point obtained by this dual procedure is an approximation to the exact MPP, particularly for highly nonlinear constraints.

To find a better search direction $\mathbf{U}^*_{PMA}$, we also suggest another procedure in which we compute the MPP on a circle of radius β^r_j (computed from the supplied reliability index R_j), instead of a unit circle. Since the MPP computation is performed directly on the circle of interest, this approach is expected to produce solutions with a better accuracy than the previous approach.

With this background, we are now ready to describe the essential procedures for reliability-based optimization.

C. Reliability-Based Optimization

The above-mentioned methods to measure a solution's reliability have been integrated into an optimization algorithm in several ways. Some of them are described in the following subsections.

1) Double-Loop Methods: In the so-called double-loop methods [16], a nested optimization is used. The outer optimization problem (usually referred as a "loop") optimizes the original problem given in (2) and uses ($\mathbf{x}$, $\mathbf{d}$) as decision variable vectors. For each solution considered in the outer loop, the chance constraint is computed by solving another optimization problem (called the "inner loop"), using either the PMA or the RIA approach described above. Because of the nested nature of the overall optimization task, the double-loop methods are computationally expensive.

2) Single-Loop Methods: The single-loop methods [3] combine both optimization tasks together by not exactly finding the optimum of the inner-level optimization task, thereby constituting an approximate task of finding the true MPP point. For example, in [3] the following replacement of the original probabilistic constraint is suggested:

$$g_j(\bar{\mathbf{x}}, \bar{\mathbf{p}}, \mathbf{d}) \geq 0 \tag{10}$$

where $\mathbf{x}$ and $\mathbf{p}$ are computed from the derivatives of g_j with respect to $\mathbf{x}$ and $\mathbf{p}$ at the means, respectively, as follows:

$$\bar{\mathbf{x}} = \mu_{\mathbf{x}} - \beta^r_j \sigma \frac{\nabla_{\mathbf{x}} g_j}{\sqrt{\|\nabla_{\mathbf{x}} g_j\|^2 + \|\nabla_{\mathbf{p}} g_j\|^2}}, \tag{11}$$

$$\bar{\mathbf{p}} = \mu_{\mathbf{p}} - \beta^r_j \sigma \frac{\nabla_{\mathbf{p}} g_j}{\sqrt{\|\nabla_{\mathbf{x}} g_j\|^2 + \|\nabla_{\mathbf{p}} g_j\|^2}}. \tag{12}$$

Since the above is only an approximation to the double-loop procedure, the single-loop methods often cannot produce accurate results but are computationally faster than the double-loop methods. A study [17] compares a number of single-loop approximate ideas against double-loop methods.

3) Decoupled Methods: In the decoupled methods, two optimization (outer-level and inner-level) approaches are applied one after the other. Decoupled methods have been shown to be a good compromise between the two approaches mentioned above [18], [19]. These methods are started by first finding the deterministic optimal solution in the search space (without considering any uncertainty on design variables $\mathbf{x}$ or parameters $\mathbf{p}$ and using the mean of $\mathbf{x}$ as decision variables). Thereafter, the most probable point (MPP) for each constraint g_j is found using the PMA or RIA approach. Then, in the next iteration, each constraint is shifted according to their MPP points found in the last inner-level optimization, and a deterministic optimization to the shifted constraint problem is solved. This dual optimization continues in turn until no further improvement in the current solution is achieved. The outcome of such a strategy is sketched in Fig. 5. From the deterministic optimum (A), both constraints are considered (one at a time) to find the corresponding MPP points (A1 and A2). Thereafter, the corresponding constraints are shifted at these points and a new optimization problem is solved to find a new point B. The procedure continues (by finding B1 and B2 for both constraints) until convergence. Fig. 6 sketches a particular approach [sequential optimization and reliability assessment (SORA) method] suggested elsewhere [18], in which the PMA approach is used to determine the MPP in the second optimization problem.

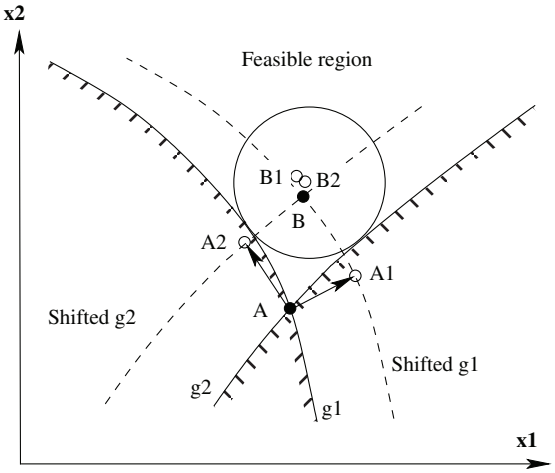

Fig. 5. Working principle of SORA.

D. Handling Multiple Constraints

Ideally, the reliability of a solution should be computed by considering a cumulative effect of all constraints, as presented by the probability term in (2). However, the above PMA and RIA methods assume a single constraint in their approaches and compute an MPP for a particular constraint at a time. There are basically two ways to extend the approaches to multiple constraints.

a) Closest Constraint: The simplest way to consider multiple constraints is to determine the failure probability P_j for each constraint individually and then to calculate the following bounds on the overall failure probability P_F:

$$\max_j P_j \leq P_F \leq \min(1, \sum_j P_j). \tag{13}$$

Intuitively, usually, the larger the failure probability P_j of a constraint, the closer the constraint to the solution. Thus, the above lower bound signifies simply the failure probability of the closest constraint and can be used as a crude estimate (usually an underestimation) of the failure probability against all constraints. The upper bound of P_F holds if the constraints have no overlapping regions and, in other cases, is an overestimation of the overall failure probability. The only reason for

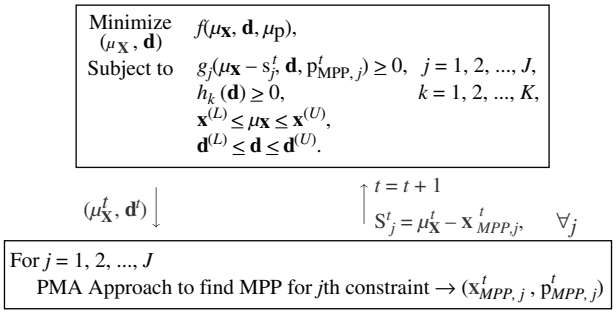

Fig. 6. Specific decoupled method (SORA) [18]. Initial value of s_j is set equal to zero for all j.

using such simple estimates is the low computational burden, since no joint failure probability estimate need be computed by this approach.

b) Ditlevsen's bound: Much closer bounds on the overall failure probability are given elsewhere [2], [20], in which first, the failure probability P_j of each constraint is computed, and then, the overall failure probability P_F of the solution from all constraints is bounded as follows:

$$P_1 + \sum_{i=1}^{J} \max\left\{0, \left(P_i - \sum_{j=1}^{i-1} P_{ji}\right)\right\} \leq P_F \leq \sum_{i=1}^{J} P_i - \sum_{i=2}^{J} \max_{j|j<i} P_{ji}. \quad (14)$$

The formula depends on the exact ordering of the failure modes considered in the study. Usually, the failure modes are ordered according to decreasing values of P_i [21]. Thus, P_1 and P_J correspond to the largest and smallest failure probabilities, respectively. The joint probability P_{ji} of failure against both ith and jth constraints is given by the cumulative distribution function (CDF) of the bivariate normal distribution

$$P_{ji} = \Phi(-\beta_j, -\beta_i, \rho_{ji}) \quad (15)$$

and the correlation coefficient ρ_{ji} is given as follows [20]:

$$\rho_{ji} = \frac{\left\langle \mathbf{u}_j^*, \mathbf{u}_i^* \right\rangle}{\left\| \mathbf{u}_j^* \right\| \left\| \mathbf{u}_i^* \right\|} \quad (16)$$

where $\mathbf{u}_j^*$ is the MPP point in the **U**-space for the jth constraint alone computed for solution $\mathbf{x}$. The cosine-inverse of ρ_{ji} indicates the angle between the two $\mathbf{u}^*$ vectors. Fig. 7 explains this procedure. The FORM approach makes a linear approximation of each constraint at MPP, meaning that the MPP point is the point where the linear approximation is a tangent to the original constraint. Once these MPP points are found for all constraints, the correlation coefficient ρ_{ji} is the cosine of the angle formed by the MPP vectors of jth and ith constraints.

The above Ditlevsen's bounds are much tighter than the closest constraint bound but involve computation of the pairwise joint failure probabilities. To be conservative, we shall consider the upper Ditlevsen's bound here and replace all J chance constraints in (2) with a single constraint of comparing

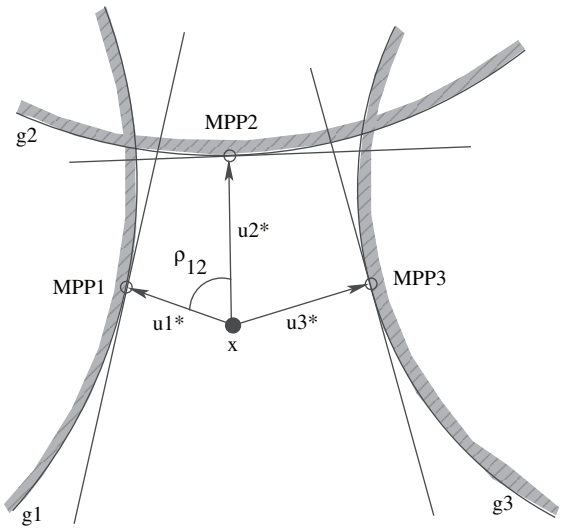

Fig. 7. Computation of the correlation coefficient.

the overall survival probability $(1 - P_F)$ with the desired reliability R

$$1 - \left(\sum_{i=1}^{J} P_i - \sum_{i=2}^{J} \max_{j|j<i} P_{ji}\right) \geq R. \quad (17)$$

The difference between considering only the closest constraint approach [lower bound in (13)] and using the Ditlevsen's upper bound on failure probability can be explained using Fig. 8. If there is only one constraint (the leftmost panel), it divides the space into a feasible area (grey) and an infeasible area (white). Overall failure probability becomes identical to P_1 and both methods estimate the failure probability without any error. For the case of two constraints (the middle panel), if only the closest constraint (C_1 for the solution marked) is considered, feasibility will be overestimated by not computing the failures arising from the area denoted as A_2. However, using the Ditlevsen's upper bound, both constraints are handled accurately, as P_F is now computed as $(P_1 + P_2 - P_{12})$, which corresponds to failure arising from areas A_2, A_3, and A_4. For more than two constraints, the Ditlevsen's upper bound may no longer provide an exact estimate of the failure probability. For the scenario in the rightmost panel in Fig. 8, true failure probability should arise from cumulative areas marked A_2 to A_7. However, the Ditlevsen's upper bound will estimate it to be $(P_1 + P_2 + P_3 - P_{12} - \max(P_{13}, P_{23}))$. If $P_{23} \geq P_{13}$, the failure arising from area A_3 will be considered twice; otherwise, the failure arising from A_8 will be considered twice. Thus, the Ditlevsen's upper bound may overestimate the actual failure probability for a problem having more than two constraints, i.e., the true reliability will be larger than the estimated reliability.

Nevertheless, the Ditlevsen's upper bound is usually tight. In principle, it would be possible to improve the bounds by also using higher order intersections, but this involves much more numerical effort with a little gain in accuracy of the result [22].

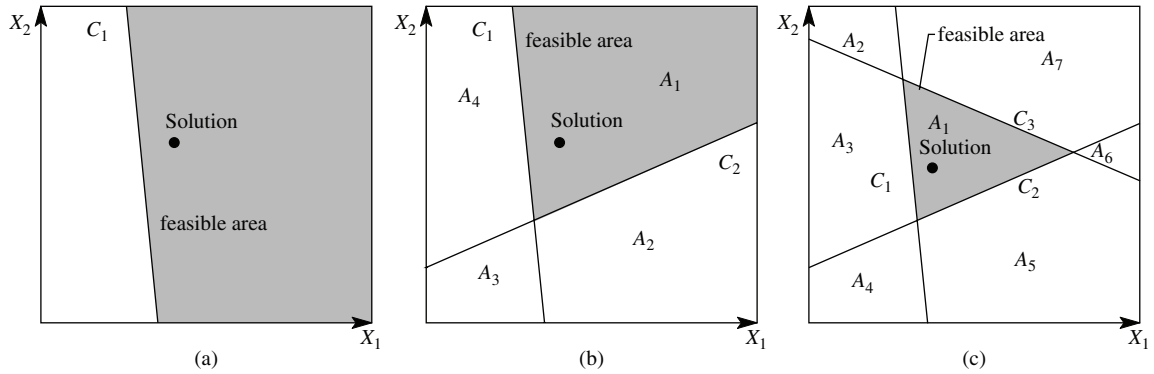

Fig. 8. Area considered (grey) feasible and (white) infeasible for (a) one constraint, (b) two constraints, (c) three constraints.

III. Overview of Other Related Studies

There are two closely related aspects when optimizing in the presence of uncertainty: reliability and robustness. The terms are not uniquely defined in the literature, and we will make the following distinction in this paper under variable and parameter uncertainty. Reliability-based design (which is the focus of this paper) aims at finding the best solution that satisfies the constraints with a specified probability. Robust design is usually concerned with the solution quality and not the constraints. There are many possible notions of robustness, including a good expected performance, a good worst-case performance, a low variability in performance, or a large range of disturbances still leading to acceptable performance (see also [23, p. 127]).

As the classical reliability-based optimization has already been discussed in depth above, the following survey focuses on reliability and robustness in combination with evolutionary optimization. In recent years, there has been a growing interest in applying evolutionary computation to optimization problems involving uncertainty, and a recent survey on this field can be found in [24].

Most research in the EA community so far has focused on the robustness of solutions, in particular the *expected* fitness given a probability distribution of the uncertain variable. From the point of view of the optimization approach, this reduces the fitness distribution to a single value: the *expected* fitness. Thus, in principle, standard evolutionary algorithms could be used with the expected fitness as the driving force. Unfortunately, it is usually not possible to *calculate* the expected fitness analytically; it has to be *estimated*. This, in turn, raises the question how to estimate the expected fitness efficiently, which is the topic of many studies of robustness within EAs. In [25], it was shown that for the case of an infinite population size and proportional selection, adding random perturbations to the design variables in each generation is equivalent to optimizing on the expected fitness function. For finite population sizes, *explicit averaging* (e.g., [23], [26]) or the use of metamodels (e.g., [27]) may be successful.

Robustness based on expected fitness has also been studied for the case of multiobjective problems [28]–[30]. These approaches rely on multiple sampling for estimation. Then, a standard EMO algorithm is used to work with these expected fitnesses. Reference [29] thereby extends [28] by additionally taking into account robustness with respect to constraint violations.

In contrast to searching the solution with the best expected fitness, the *worst-case* cannot usually be obtained by sampling. Instead, finding the worst-case for a particular solution may itself be a complex optimization problem. In [31], this is solved by running an embedded optimizer searching for the worst-case for each individual (called anti-optimization in [31]). Similarly, in [32] a simplified meta-model around a solution is constructed and a simple embedded local hill climber to search for the worst-case is used. In [33], the maximum disturbance range that guarantees fitness above a certain threshold is used. Again, this is determined by an embedded search algorithm.[1] In [34], a coevolutionary approach for a scheduling problem, co-evolving solutions, and worst-case disturbances are used. Others simply calculate some bounds on the worst-case behavior (e.g., [35]). In [36], a multiobjective evolutionary algorithm is used to evolve the tradeoff between Pareto optimality and worst normalized variation among all objectives due to uncertainty.

A few papers treat robustness as an additional criterion to be optimized. Robustness is measured, e.g., as variance [27], [37], [38], as maximal range in parameter variation that still leads to an acceptable solution [33], [39] or as the probability to violate a constraint [40], [41]. This allows the decision maker to analyze the possible tradeoff between solution quality and robustness/reliability. The challenges and approaches are quite similar to the single objective optimization in determining the performance measures.

An excellent and comprehensive survey on robustness optimization, which also discusses the connection to reliability optimization and the role of evolutionary computation in this area, can be found in [42].

With respect to evolutionary reliability optimization, several studies [5], [43] have used Monte Carlo simulation with Latin hypercube sampling (LHS) within an EA to estimate reliability. Reference [44] uses a Taguchi approach to analyze

[1]These approaches are similar to what is proposed below in the sense that they use an embedded optimizer to evaluate a solution. However, they consider robustness, while we consider reliability, and we use techniques specifically designed to calculate a solution's robustness.

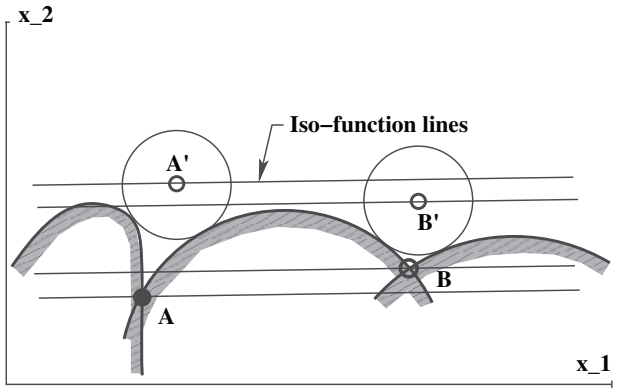

Fig. 9. Sketch of two optima A and B and their corresponding reliable solutions (A′ and B′) for a fixed reliability index.

the sensitivity of engineering designs found by an evolutionary algorithm. An overview of the reliability-based design optimization methods for automotive structures is given by [45], which also includes sampling techniques, nonlinear response surface methodologies, robust assessment, and robust design formulation. None of the above sampling-based approaches is applicable if very high levels of reliability are desired. Another study used interval arithmetic approach with an evolutionary algorithm to find reliable solutions [46].

This paper summarizes and extends the approaches presented in [40], [41], where it has first been suggested to integrate classical methods to calculate a solution's reliability within an EMO algorithm.

IV. Three Reliability-Based Optimization Cases

In this section, we present three different aspects of reliability-based optimization problems which may be difficult to solve using the classical optimization techniques mentioned in Section II-C but for which evolutionary algorithms (EAs, search heuristics mimicking the natural evolutionary principles [47]–[49]) may be suitable.

A. Single-Objective, Multimodal Reliability-Based Optimization

Many single-objective optimization problems involve multiple global and local optima. Most classical methods start with a deterministic optimum and then search for a close reliable solution. However, in some problems the deterministic global minimum is highly constrained, and the closest reliable solution is far away. On the other hand, a different local optimum may be much less constrained, and a close reliable solution might actually be better than the reliable solution close to the global optimum.

This is illustrated in Fig. 9. In a sufficiently nonlinear problem, the reliable minimum (point A′) corresponding to the global deterministic minimum (point A) need not be the best solution and the reliable minimum (point B′) corresponding to a local deterministic minimum (point B) may be better. The classical serial procedure of first finding the deterministic global optimum (solution A) and then finding the reliable solution (solution A′) may not be a good idea in such problems. Evolutionary optimization methods are population-based approaches and do not need to start their search from a deterministic optimum. They can be directly used to solve the reliability-based optimization problem (2). Moreover, due to their population approach, they are more likely to avoid the locally optimal reliable solution and converge to the true reliable solution.

It is worth mentioning here that although we discussed the multimodality issue in the context of single-objective optimization, such a scenario may very well exist in the case of a multiobjective optimization problem. In such a scenario, a classical method may find it difficult to converge to the globally reliable frontier and may instead get stuck in a locally Pareto-optimal frontier.

B. Optimization for Seeking Multiple Solutions for Different Reliability Values

In most reliability-based design optimization (RBDO) studies, the aim is to find the reliable optimum corresponding to a given failure probability (or a given reliability index). However, in the context of design optimization, it would be educative to learn how the reliable solutions change with different levels of reliability index, as shown in Fig. 10. When reliability is not considered, the deterministic optimum is the desired solution. As discussed earlier, when the optimization is performed for a particular reliability (say $R = 0.9$), a solution in the interior to the feasible region becomes the corresponding reliable solution. As the desired reliability value is increased, the resulting solution will move further away from the constraint and inside the feasible region. That is, if we can locate the reliable optimum for small (say 80%) to large value (say 99.999%, meaning a failure of one in a thousand) of reliability, the trace of solutions will reveal important insights about how to change decision variables to make the corresponding solutions more and more reliable. Fig. 10 shows such a trace on the decision variable space for a hypothetical problem. Such multiple reliable solutions can be found simultaneously by treating the problem as a two-objective optimization problem of optimizing the original objective and, in addition, maximizing the reliability index (R or β), as well as by locating a number of tradeoff optimal solutions using an evolutionary multiobjective optimization (EMO) strategy to this bi-objective optimization problem

$$\begin{aligned} &\underset{(\mu_{\mathbf{X}},\mathbf{d})}{\text{Minimize}} \;\; f(\mu_{\mathbf{X}}, \mathbf{d}, \mu_{\mathbf{p}}) \\ &\underset{(\mu_{\mathbf{X}},\mathbf{d})}{\text{maximize}} \;\; R(\mu_{\mathbf{X}}, \mathbf{d}, \mu_{\mathbf{p}}) \\ &\text{subject to } h_k(\mathbf{d}) \geq 0, \qquad k = 1, 2, \ldots, K \\ &\qquad\qquad \mathbf{x}^{(L)} \leq \mu_{\mathbf{X}} \leq \mathbf{x}^{(U)}, \\ &\qquad\qquad \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)} \end{aligned} \tag{18}$$

where

$$R(\mu_{\mathbf{X}}, \mathbf{d}, \mu_{\mathbf{p}}) = 1 - P_F.$$

The overall failure probability P_F can be computed by computing individual failure probabilities P_j involving inequality constraints g_j. The procedure for computing P_F was discussed in Section II-D.

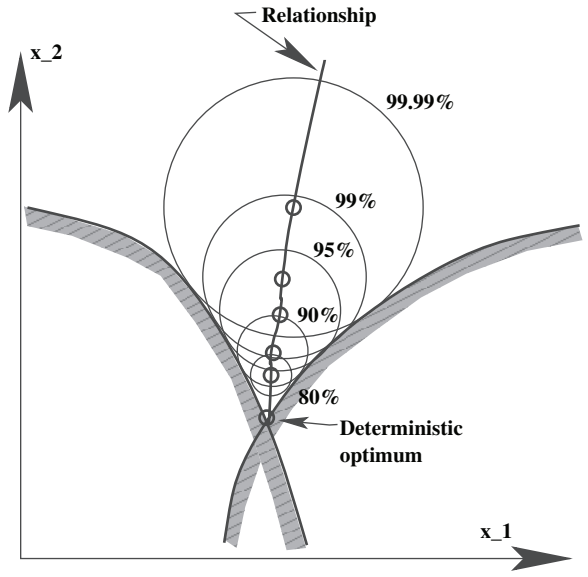

Fig. 10. Different reliability indexes may result in an interesting relationship among reliable solutions. Circles show a solution's distance to the constraints (not a quantile of the probability density function).

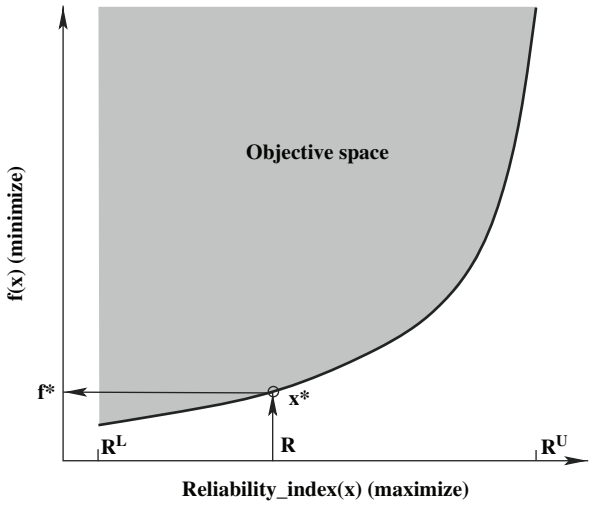

Fig. 11. Bi-objective formulation of minimizing objective function $f(\mathbf{x})$ and maximizing reliability index $R(\mathbf{x})$.

The scenario is depicted in Fig. 11. The shaded region marks the objective space, but the solid front at the bottom-right part of the shaded region marks the optimal solutions corresponding to different reliability index values within a prespecified range $[R^L, R^U]$. The EMO procedure is capable of finding multiple Pareto-optimal solutions for solving such a bi-objective optimization problem, thereby finding multiple reliable solutions corresponding to differing reliability values. Such a study will help to analyze the effect of the reliability index on the quality of solutions (both in objective value and in decision parameter values) and may help to determine a suitable reliability index for a particular application. It is worth mentioning here that instead of finding the complete front using an EMO, a number of reliability index values (such as R illustrated in the figure) can be chosen; for each case, the objective function $f(\mathbf{x})$ can be optimized with the consideration of constraints, variable bounds, and uncertainty, and a corresponding reliable solution $\mathbf{x}^*$ can be found. Such multiple independent applications of *a posteriori* multiple criterion decision making (MCDM) method (such as the ϵ-constraint method [50]) works in a similar principle as an EMO and can also be used for this purpose. However, a recent study [51] has discussed the difficulties of using *a posteriori* MCDM methods, particularly in handling difficult optimization problems. Also, sequential methods are usually computationally more expensive than an EMO procedure, searching for several Pareto-optimal solutions concurrently.

C. Multiobjective Reliability-Based Optimization

The concept of reliability-based optimization methods can also be applied to solve multiobjective reliability-based optimization problems

$$\begin{aligned} \underset{(\mathbf{x},\mathbf{d})}{\text{Minimize}} \ & (f_1(\mathbf{x},\mathbf{d},\mathbf{p}),\ldots,f_M(\mathbf{x},\mathbf{d},\mathbf{p})) \\ \text{subject to} \ & g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0, \quad j=1,2,\ldots,J \\ & h_k(\mathbf{d}) \geq 0, \quad k=1,2,\ldots,K \\ & \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \\ & \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}. \end{aligned} \tag{19}$$

In such cases, instead of a single reliable solution, a reliable frontier is the target, as shown in Fig. 12. When reliability aspects are considered, the corresponding reliable front may be different from the original front and will, in general, be placed inside the feasible objective space. As the reliability index is increased (to get more reliable solutions), the front is expected to move further inside the feasible objective space. To solve multiobjective optimization problems, EMO procedures can be applied directly on the following deterministic optimization problem:

$$\begin{aligned} \underset{(\mu_{\mathbf{x}},\mathbf{d})}{\text{Minimize}} \ & (f_1(\mu_{\mathbf{x}},\mathbf{d},\mu_{\mathbf{p}}),\ldots,f_M(\mu_{\mathbf{x}},\mathbf{d},\mu_{\mathbf{p}})) \\ \text{subject to} \ & P(g_j(\mathbf{x},\mathbf{d},\mathbf{p}) \geq 0) \geq R_j, \quad j=1,2,\ldots,J \\ & h_k(\mathbf{d}) \geq 0, \quad k=1,2,\ldots,K \\ & \mathbf{x}^{(L)} \leq \mu_{\mathbf{x}} \leq \mathbf{x}^{(U)}, \\ & \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}. \end{aligned} \tag{20}$$

The probability constraint $P()$ can be computed as before by using any of the four methods discussed earlier. The advantage of finding the complete reliable frontier is that the relative sensitivity of different regions of the frontier can be established with respect to the uncertainties in design variables and parameters. This information will be useful to the designers and decision makers in choosing a solution from a relatively insensitive region of the tradeoff frontier.

There is a fourth problem scenario involving M conflicting objectives, in which an $(M+1)$-dimensional tradeoff frontier can be attempted to be found by including an additional objective of maximizing derived reliability R, as considered in Section IV-B for a single-objective optimization problem. This will provide a plethora of information about the nature of change of the original M-dimensional tradeoff frontier with the required reliability value. In this paper, we do not explicitly add such a reliability objective for multiobjective optimization

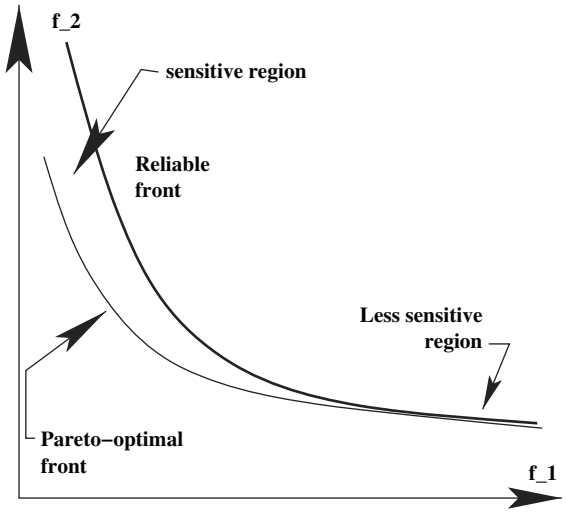

Fig. 12. Reliable front in a multiobjective reliability-based optimization problem.

problems but perform multiple independent M-objective runs with different fixed reliability (R) values and show the effect of R on the obtained frontier.

V. Proposed Evolutionary Approaches

A. General Setup

We suggest here reliability-based optimization procedures based on evolutionary optimization algorithms to handle all three problem classes described above.

For the problems described in Sections IV-A and C, we suggest to use the FastPMA approach of computing the MPP, because it suffices to determine whether a solution fulfills the specified reliability, and the method is fast. To handle problems described in Section IV-B, the RIA-based approach is needed because, for each solution, the corresponding reliability value has to be determined, as this value corresponds to one of the objectives which needs to be maximized during the optimization task. Since computational efficiency is still an issue, we use the FastRIA variant in this case.

For evolution, we use a real-parameter GA with a penalty-parameterless constraint handling approach [52] to handle all deterministic constraints in the case of the single-objective scenario of Section IV-A. For multiobjective optimization problems, we employ the constrained tournament concept with the elitist non-dominated sorting GA or NSGA-II [53]. We use tournament selection with tournament size of 2, and the simulated binary crossover (SBX) operator [54] to create two blended offspring solutions. The crossover probability is 0.9, meaning that 90% of the pairs are recombined to create offspring solutions, and the remaining 10% parents are simply chosen. The SBX operator involves a distribution index controlling the spread of obtained solutions. We have used a value of 2, which is recommended in the original study [54]. Finally, a polynomial mutation operator [49] is used to perturb the offspring solutions in their neighborhood. A mutation probability of $1/n$ is used so that on average one of the design variables are mutated per offspring solution. A distribution index of 50 is used for mutation. For details of these operators, see a description given elsewhere [49]. A C-code implementing the above-mentioned GA is available at http://www.iitk.ac.in/kangal/soft.htm.

But before we discuss the simulation results, we suggest a procedure of identifying redundant constraints for the purpose of computing the overall probability of failure so that overall computational time is further reduced.

B. Identifying Redundant Constraints

Determining the MPP for every solution and every constraint can be computationally demanding, particularly when dealing with a large number of constraints and population members. The use of FastPMA and FastRIA variants discussed earlier for MPP computations alleviates the problem to some extent. Here, we propose a procedure to make a further reduction in computation of overall failure probability by identifying constraints which either are far away or do not contribute much to the overall failure probability. To understand these cases, we first sort the constraints from the largest failure probability to the smallest failure probability.

After sorting, we have $P_i \geq P_j$ for $i < j$. The Ditlevsen's upper bound can be computed in the following manner. The overall failure probability P_F can be computed by adding terms $(P_i - \max_{j|j<i} P_{ji})$ (for $i > 1$) one by one to P_1. It is interesting to note that this term (within brackets) is always non-negative. As the terms are included one by one, one of two scenarios can happen. The value of the term becomes so small that the inclusion of it in P_F computation does not affect the failure probability value significantly (say, the term has a value less than a threshold η, which is set much smaller than $(1-R)$). In this case, the constraint i can be said to be redundant for the P_F computation. For example, consider a two-constraint scenario shown in Fig. 13 for which two constraints are almost parallel to each other with respect to the current solution $\mathbf{x}$ and that the solution $\mathbf{x}$ makes the second constraint almost redundant. Using our check for identifying redundant constraints stated above, we realize that $P_1 > P_2$ and $P_{12} \approx P_2$, thereby making the above-specified term $(P_2 - P_{12})$ almost equal to zero. Thus, our proposed η-threshold check will then declare the second constraint as a redundant one for the current solution.

Second, the failure probability P_i of the ith constraint itself can be so small (say, less than η) that it is not worth including in the P_F computation. In this case, all other constraints placed beyond ith constraint in the sorted list can also be termed as redundant. Fig. 14 shows such a scenario. Constraint g_1 is so close to the current solution compared to other constraints that other constraints will cause a negligible failure probability (P_2 and so on) compared to P_1. Hence, constraints g_2 and other far away constraints can be declared redundant for the current solution.

The inclusion of the above-mentioned terms $(P_i - \max_{j|j<i} P_{ji})$ one by one has another advantage in dealing with problems described in Sections IV-A and C, in which a desired reliability (R) is supplied by the user. Recall that the

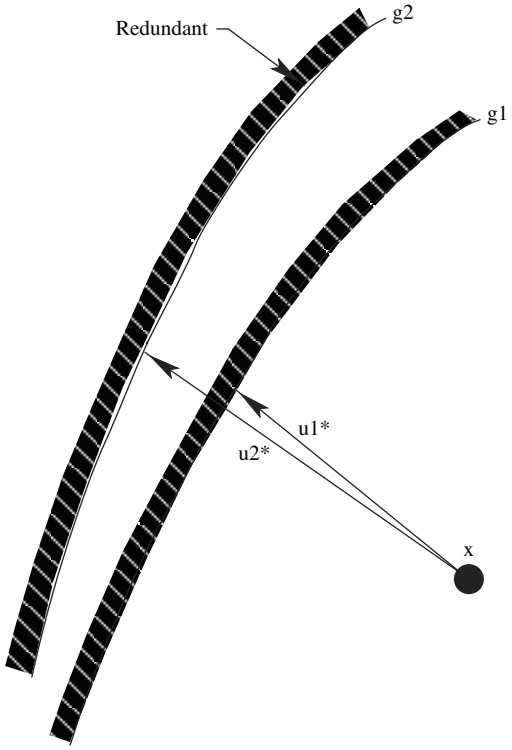

Fig. 13. Almost parallel constraints making one constraint redundant.

chance constraints require that the probability of a solution being feasible is at least R (or, $P_F \leq 1 - R$). While considering the term for the kth constraint in the ordered list, the Ditlevsen's upper bound can be overestimated by $P_1 + \sum_{i=2}^{k-1}(P_i - \max_{j|j<i} P_{ji}) + (J-k)P_k$ and the following condition can be tested:

$$P_k \leq \frac{1}{J-k}\left((1-R) - P_1 - \sum_{i=2}^{k-1}(P_i - \max_{j|j<i} P_{ji})\right). \quad (21)$$

If the above condition is true, there is no need to proceed with kth and all other constraints thereafter in the ordered list. The above test can be made for including each constraint starting from $k = 2$.

It is interesting to realize that the redundancy check suggested above has a local property. That is, constraints which are found redundant by a solution $\mathbf{x}$ may also be found redundant for most other near-by solutions (say $\mathbf{y}$, provided $\|\mathbf{y} - \mathbf{x}\| \leq \epsilon$). In all simulations in this paper, we have used $\epsilon = 0.01$ in the $\mathbf{U}$-space and $\eta = 9(10^{-7})$. This principle can lead us to saving expensive MPP computations in the following way. From the initial generation, we maintain a database storing a solution $\mathbf{x}$ and a linked list of constraints which are found redundant for the solution. In later generations, when a solution $\mathbf{y}$ close (with the above-mentioned ϵ-neighborhood check) to $\mathbf{x}$ is found, we do not need to compute the MPP for the redundant constraints of $\mathbf{y}$, the information of which is taken from the database for $\mathbf{x}$. This procedure will save *constraint calls* for computing MPP vectors, thereby saving computational time.

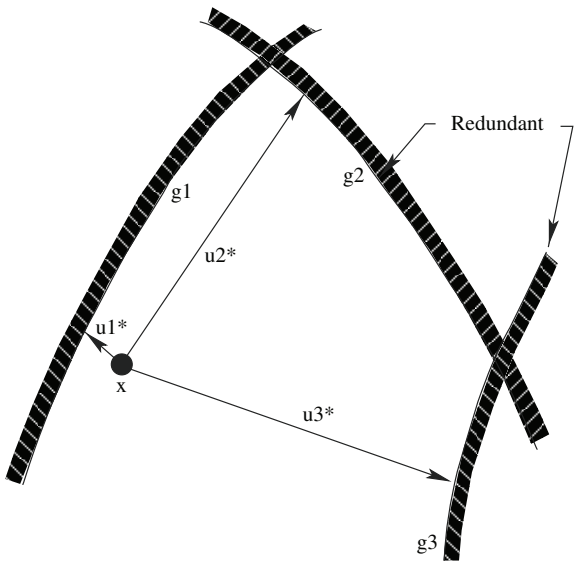

Fig. 14. Far away constraints which can be redundant.

VI. Simulation Results on Multimodal Reliability-Based Optimization Problems

In a real-world optimization problem, there often exist multiple local optima, irrespective of whether reliability is considered or not. Often, the reliable local optima are located close to some deterministic local optima. But the global optimum when taking reliability into account may be close to a local deterministic optimum. This is a problem for methods like SORA, which first compute the deterministic optimum, and search for a reliable solution from there.

In this section, we compare EAs with two classical reliability-based optimization methods, and show that EAs do not suffer from this problem.

Let us consider the following two-variable test problem:

$$\begin{aligned} &\text{Maximize } y \\ &\text{subject to } x^2 - 1000y \geq 0, \\ &\qquad y - x + 200 \geq 0, \\ &\qquad x - 3y + 400 \geq 0, \\ &\qquad -400 \leq x, y \leq 300. \end{aligned} \quad (22)$$

In this problem, $\mathbf{x} = (x, y)$ are uncertain variables, and there exists no deterministic variable d. Also, there does not exist any uncertain problem parameter (p). We assume independent and normally distributed uncertainties with $\sigma_x = \sigma_y = 10$ and a desired reliability index of $\beta^r = 4$.

First, we present results obtained with a real-coded genetic algorithm with the proposed FastPMA to check reliability of a solution. The proposed EA uses the simulated binary crossover and the polynomial mutation operators [49], and a population size of 20. The GA is terminated after 60 generations have elapsed. Recall that the FastPMA approach (discussed in Section II-B2b) begins with a guess of a MPP direction using the derivative vector of the underlying constraint function and then iterates to converge on a particular direction. We terminate the MPP direction finding strategy when the difference between two consecutive direction vectors is $\epsilon_{PMA} = 0.001$ or

TABLE I

COMPARISON OF PERFORMANCES OF DOUBLE-LOOP METHOD, SORA, AND PROPOSED GA. IN EACH CASE, A TOTAL OF 2500 RUNS ARE PERFORMED

Method	# success	Proportion success	Solution Evaluations		
			Best	Average	Worst
GA ($\epsilon_{PMA} = 0.001$)	2500	100%	9483	9756.2	9768
GA ($\eta_{PMA} = 2$)	2500	100%	3663	3663.0	3663
SORA	1630	~65%	491	1394.0	15679
Double-loop	1219	~49%	2910	147994.7	413040

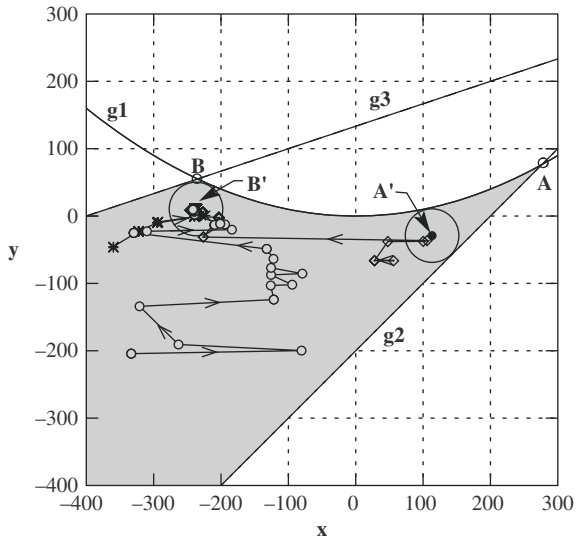

Fig. 15. Proceedings of three GA simulations showing how population-best solution can progress toward the global maximum (B′).

less. We perform 2500 runs with different initial populations to check the robustness of the proposed procedure. The first row in Table I shows the performance of the proposed GA. We observe that in all 2500 runs, the GA is able to find a solution near the correct reliable optimum solution $(237.908, 11.820)^T$, with a function value of 11.820. Fig. 15 shows how three different GA simulations starting at different regions in the search space (considering the population best) avoid the local maximum (A′) and converge to the global maxima (B′). One of the runs has its best located near A′ in the initial population and even then the GA with the proposed FastPMA approach can advance and converge near the globally optimal reliable solution. Fig. 16 shows all 2500 points obtained by the 2500 GA applications.

To investigate the effect of the extent of iterations on finding the MPP direction on the final outcome of the algorithm, next we terminate the MPP direction finding strategy only after two ($\eta_{PMA} = 2$) iterations. The second row in Table I shows the performance of the modified GA. Again, we obtain 100% successful result but this time with far fewer overall solution evaluations. A plot of the obtained solutions in the decision variable space produces a similar plot as in Fig. 16 and is not presented here for brevity. In this problem, the choice of the gradient direction on β^r-circle as an initial guess of the MPP direction is close enough to the true MPP direction, and two iterations were enough to locate a near MPP point for this problem. This reduces the solution evaluations drastically without degrading the performance of the GA procedure.

Now, let us compare the performance of the EA with two classical methods. First, we have implemented the classical SORA approach [18] discussed in Section II-C3 to solve the above problem. The MATLAB code `fmincon` is used to optimize both the PMA and the overall optimization tasks. We terminate each of the two optimization tasks when the tolerances in variable vector (`TolX`), function value (`TolFun`) and constraint value (`TolCon`) are 10^{-8}. We performed a number of simulations using larger tolerance values in order to find the smallest number of solution evaluations for a successful application of SORA, but most runs resulted in nonoptimal solutions, other than solutions A′ or B′. We argue that a larger tolerance value causes intermediate unidirectional search iterations of `fmincon` to prematurely terminate to nonoptimal solutions, thereby not allowing the overall algorithm to advance to the true optimal solutions. Fig. 17 shows the final obtained solutions from each of 2500 simulations. It is clearly visible that, even with a tolerance value of 10^{-8}, not all runs converge to the true reliable optimum near B′, but many runs find their way to a solution near the deterministic optimum A′. The fact that the approach sometimes finds the true reliable optimum is surprising, and may probably be attributed to the fact that SORA first stage, which searches for the deterministic global optimum, gets stuck in the local optimum B′. This situation happens particularly when the initial starting point is chosen near the local optimum B. Combined solution evaluations for 2500 runs are recorded and presented in Table I. Although overall function evaluations are much smaller than that required with our proposed EA, the SORA method is found to be successful in only about 65% of the simulations.

Next, we implement the double-loop method using MATLABs `fmincon` code for both upper and lower level optimization tasks. After some trials using different tolerance values, we observe that an identical tolerance value (10^{-8}) in each optimization task as that needed in SORA is needed to obtain near-optimal results. Fig. 18 plots the solutions obtained by 2500 runs. Interestingly, nonoptimal solutions are found in many runs. It seems that the outcome of the procedure strongly depends on the chosen initial points. For some initial points, the MPP solution obtained by the lower level search cannot be improved by the upper level search and the combined algorithm gets stuck to a point parallel to a critical constraint boundary and requires in a huge number of function evaluations.

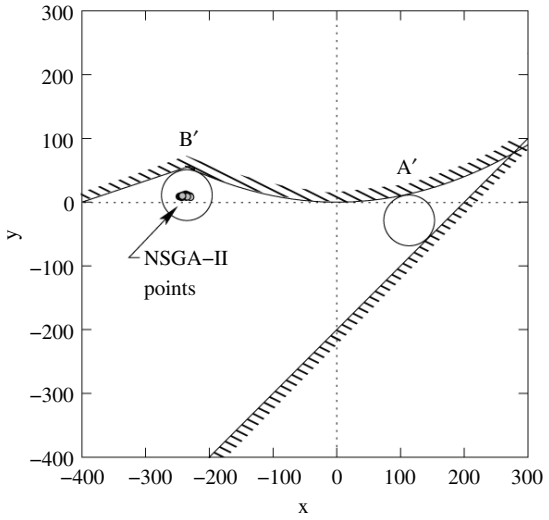

Fig. 16. GA results.

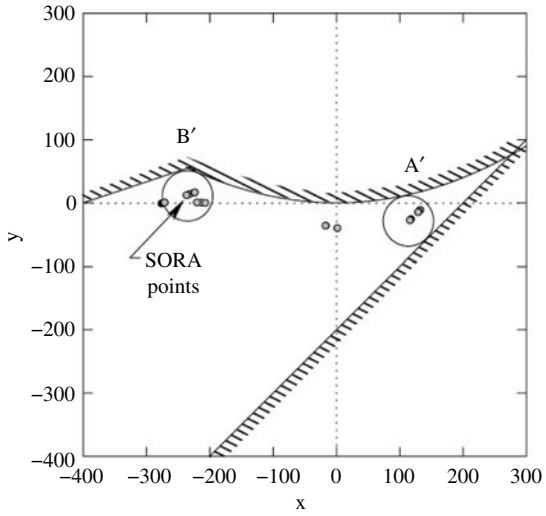

Fig. 17. SORA results.

Results of all three approaches for 2500 independent runs are compared in Table I. From these extensive computer results, we conclude the following.

1) The proposed GA can find the correct reliable optimum in 100% of all runs, compared with 65% for SORA and 49% for the double-loop method.
2) SORA is the fastest approach in terms of function evaluations. The double-loop method is the second fastest (in terms of the best algorithm performance) but with some extremely long runs. However, the GA performs second in terms of average required function evaluations but does best in terms of worst function evaluations in 2500 simulations.
3) The double-loop method is not as accurate and also requires more solution evaluations than SORA.
4) The performance of the proposed GA approach is consistent and more reliable than both SORA and the double-loop method.

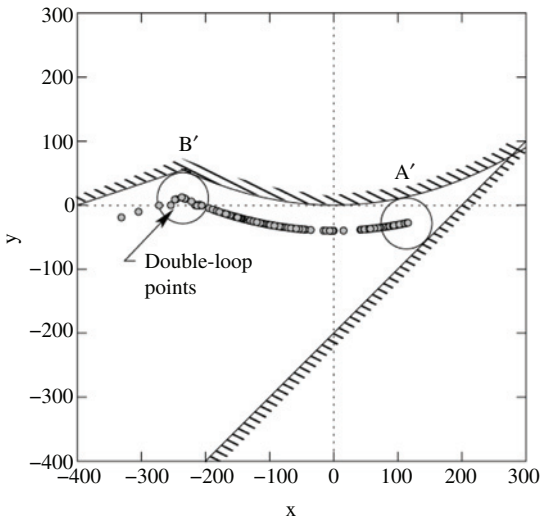

Fig. 18. Double-loop results.

5) The double-loop method is prone to get attracted to sub-optimal solutions due to complex interactions between upper and lower level optimization tasks.

This paper clearly indicates the importance of EA-based approaches to difficult reliability-based optimization problems.

VII. Simulation Results on Finding Multiple Reliable Solutions

Here, we consider two problems—the two-variable problem considered in the previous section and an automobile car side-impact problem.

A. Test Problem Revisited

We now consider an additional objective of maximizing the reliability index. To handle two objectives, we employ the NSGA-II algorithm, in which every population member is checked with the RIA optimization approach to find the corresponding reliability index of the solution. Here, we employ the fastRIA approach described in Section II-B2d. The reliability index is restricted to lie within 0.05 and 5.0, corresponding to 51.98388% to 99.99997% reliability values. We use a population size of 40 and run NSGA-II for 80 generations. The resulting population members are shown in Fig. 19. It is clear that as the reliability index is increased, the corresponding optimal function value gets worse (reduced here). There seems to be two different patterns of variation of the optimal function value. Up until about a reliability index of 0.7 (meaning about 75.8% reliability), the drop in optimal function value is more rapid, but thereafter, the rate is slow. To illustrate, a number of intermediate solutions with their associated reliability index is marked on the figure with a diamond. A plot of these solutions in the decision variable space (see Fig. 20) reveals that up until $\beta^r \leq 0.7$, a solution near the global optimum (solution A) is still the reliable optimum. However, with a larger reliability requirement, the reliable optimum moves near the local optimum (solution B).

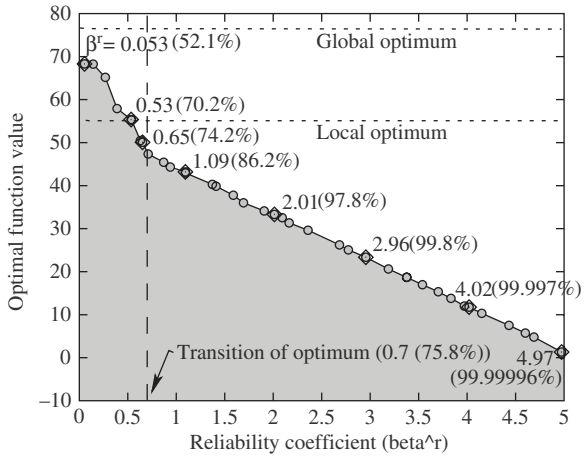

Fig. 19. Optimal objective value reduced with desired reliability index.

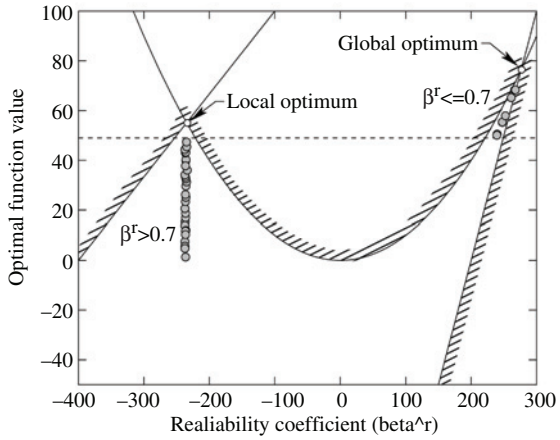

Fig. 20. Location of optimal solutions with desired reliability index.

Fig. 19 also depicts that the worsening of optimal objective value is almost linear in the increase in reliability index β^r. Fig. 20 shows how the optimal solution starting near the global optimum (for a small reliability requirement) moves inside the feasible search space with an increased demand in reliability, then moves near the local optimum, and finally moves further interior to the search space with an increase in reliability index. Such information provides a good understanding of how the optimal solution varies depending on the desired reliability and is extremely valuable to designers and practitioners in solving real-world optimization problems.

We now consider an engineering design problem and employ both the closest constraint and multiple constraint strategies to find and analyze the solutions to decipher more meaningful design principles associated with reliable solutions.

B. Car Side-Impact Problem

A car is subjected to a side-impact based on European Enhanced Vehicle-Safety Committee (EEVC) procedures. The effect of the side-impact on a dummy in terms of head injury (HIC), load in abdomen, pubic symphysis force, viscous criterion ($V*C$), and rib deflections at the upper, middle, and lower rib locations are considered. The effect on the car are considered in terms of the velocity of the B-Pillar at the middle point and the velocity of the front door at the B-Pillar. An increase in dimension of the car parameters may improve the performance on the dummy but with a burden of increased weight of the car, which may have an adverse effect on the fuel economy. Thus, there is a need to find a design balancing the weight and the safety performance. The optimization problem formulated elsewhere [55] included the minimization of the weight of the car subject to EEVC restrictions on safety performance. There are 11 design variables $\mathbf{x}$ which can be grouped into two sets: uncertain decision variables $\mathbf{x} = (x_1, \ldots, x_7)$ and uncertain parameters $\mathbf{p} = (x_8, \ldots, x_{11})$. All variables/parameters (in millimeters) are assumed to be stochastic with standard deviations (in millimeters) given below. Problem parameters x_8 to x_{11} are assumed to take a particular distribution with a fixed mean of 0.345, 0.192, 0, and 0 mm, respectively. Thus, the stochastic optimization problem involves seven decision variables and three stochastic parameters which all vary with a normal distribution. Their description and the standard deviation of their variations are given in the Appendix.

We use a population of size of 100 and run NSGA-II to optimize two objectives $f(x)$ (minimize weight function) and R (maximize reliability index) for 100 generations. Fig. 21 shows the tradeoff, nondominated front obtained using three methodologies: 1) the approach which uses only the closest constraint to compute MPP (direction for MPP is computed at a unit circle); 2) the approach which uses Ditlevsen's upper bound to compute reliability; and 3) the Ditlevsen's approach which does not consider redundant constraints to compute reliability.

We make a few interesting observations from this figure. First, the shape of the tradeoff front suggests that till up to a reliability index near 1.5, the worsening of optimal weight with an increased reliability requirement is less compared to that for solutions beyond a reliability index of 1.5. This means that larger sacrifice in weight is needed compared to the gain in reliability index for achieving a solution having such a large reliability requirement. Thus, unless a very large reliability is needed, it may not be wise to unnecessarily set a high reliability demand.

Second, the nondominated front obtained using multiple constraint consideration is located inside the feasible objective space relative to the nondominated front obtained using a single-constraint case. This is due to the fact that a single-constraint (albeit closest) consideration overestimates the probability of feasibility, thereby resulting in a front which appears to be better. To illustrate this fact, we have computed the overall reliability index value using the Ditlevsen's bound for each of the solutions obtained using the closest constraint strategy and plotted them against the reported reliability index in Fig. 22. It is clear from the figure that each solution obtained using the closest constraint strategy corresponds to a smaller overall reliability index than that obtained with respect to closest constraint alone. Thus, when all constraints are considered using the Ditlevsen's bound, each of these solutions

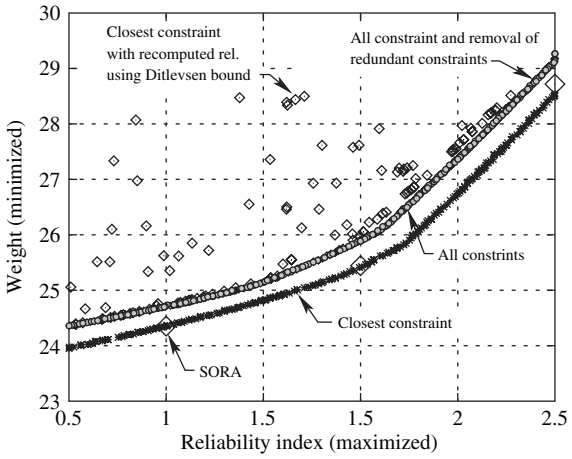

Fig. 21. Tradeoff frontier between f^* (weight) and reliability index β^r for the car side-impact problem. Solutions shown in diamonds are obtained using SORA.

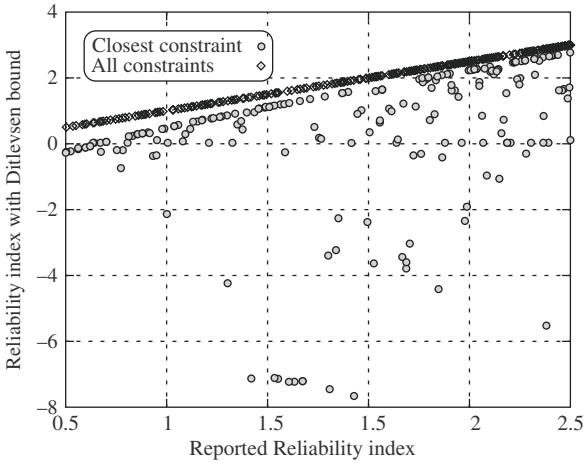

Fig. 22. Closest constraint solutions corresponding to a smaller overall reliability index than reported for the car side-impact problem.

will be infeasible with respect their specified reliability index. When these solutions are re-plotted with the overall reliability index computed using the Ditlevsen's bound in Fig. 21, they come out to be dominated by the solutions obtained by the all-constraint strategy. Fig. 22 also shows that solutions from the all-constraint strategy have identical reliability index values to their reported values. As discussed earlier and evident from this paper, the closest constraint consideration may not produce accurate computation of the tradeoff frontier, particularly in the case of multiple critical constraints.

Third, an interesting feature is that both multiple constraint considerations (with and without redundant constraints checks) produce identical fronts. This is expected, since the check for redundant constraints is suggested in this paper to reduce the computational overhead and not to compromise on the accuracy of the obtained solutions. The redundant constraint consideration strategy requires only 33% (on average 329.825 constraint calls for every 1000 calls) of the total constraint calls compared to the all-constraint approach. This is a substantial reduction in computation. To understand this aspect better, we compute the distance of the MPP from the current solution in the **U**-space for all 10 constraints and plot the distance values in the obtained solutions in Fig. 23. In the figure, constraints g_1, g_6, and g_{10} produce a distance value more than 10.0 and hence are not shown. A closer examination of the plot reveals that only two (out of 10) constraints (g_2 and g_8) are critical for most reliable solutions. Thus, a consideration of redundant constraints in this problem becomes an efficient strategy in reducing the computational effort yet producing an almost identical level of accuracy. Moreover, since only two constraints are dominant in this problem, the handling of combined failure probability using Ditlevsen's upper bound is accurate (as discussed in Section II-D), thereby providing confidence on the accuracy of the obtained frontier using "All constraints" strategy in Fig. 21.

Next, we compare our obtained solutions with an existing decoupled method (SORA). For three different reliability indices, SORA solutions are marked with diamonds in Fig. 21. Since SORA uses one constraint at a time, the algorithm finds solutions similar to those obtained by our closest constraint approach and fails to find more accurate solutions.

Finally, we make an attempt to analyze the obtained solutions by analyzing the changes in the variable values, as we move from the minimum-weight solution to the maximum-reliability solution. Fig. 24 shows how all seven variables vary with each solution's reliability coefficient (recall that a small reliability index corresponds to a small weight solution). We gather the following important information about these solutions.

Interestingly, x_5, x_6, and x_7 remain fixed for all reliable solutions (over a wide range of reliability indices [0.5, 3.0]). Variables x_5 and x_7 are fixed at their lower bounds and x_6 gets fixed at its upper bound. In the context of the car side-impact problem, to ensure an optimal weight with prefixed reliability values, the thickness of door beam (x_5) and roof rail (x_7) must be chosen as small as possible and the door beltline reinforcement (x_6) must be chosen as large as possible.

Furthermore, for solutions up to around a reliability index of $\beta^r = 2$ (corresponding to about 97.725% reliability), x_1 and x_3 must be kept fixed to their lower bounds and thereafter they must be increased for a larger reliability solution. These variables represent the thickness of B-Pillar inner and floor side inner, respectively. On the other hand, till about this critical reliability requirement, x_2 (thickness of B-Pillar reinforcement) and x_4 (thickness of cross members) must take larger values with an increase in reliability. Around this critical reliability index, they must be set to their upper limit values.

Thus, overall it seems that a good recipe to obtain a minimum weight solution of the car side-impact problem under uncertainty in its decision variables and parameters is to make the reinforcements stronger while compromising the weight by using thinner members of other components. The figure seems to suggest that if no upper bound were used for these variables, the optimal strategy would have been to use a monotonically increased dimension of x_2 and x_4 with increased reliability requirement. Since upper limits were imposed, when the variables reach their upper limits at high reliability values, the optimal strategy must change. To still

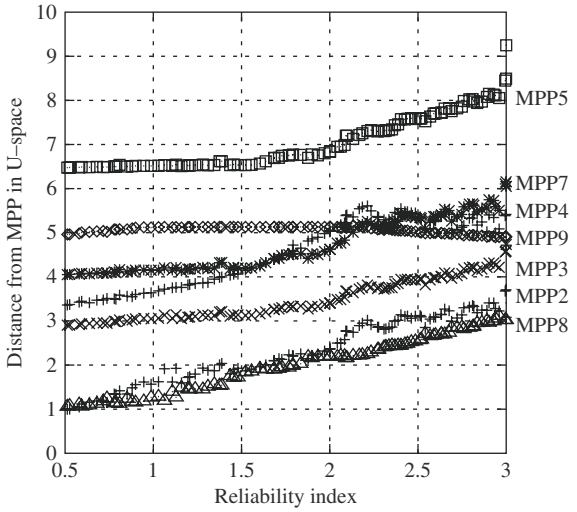

Fig. 23. Distance of MPP of constraints from each of the obtained tradeoff solutions.

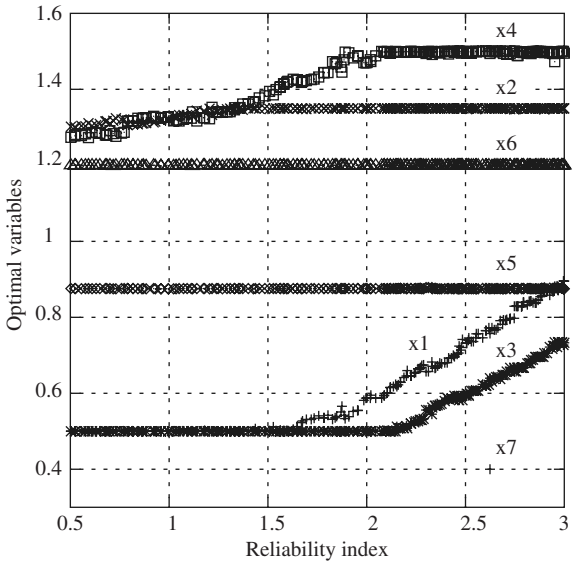

Fig. 24. Variable values changing with different reliability requirement.

achieve a minimum weight solution with a large reliability, focus has to be changed to variables x_1 and x_3. Thicker members for B-Pillar inner (x_1) and floor side inner (x_3) must be used. Such information about the nature of solutions and their interactions with the desired reliability index are interesting and provide valuable knowledge about the problem to a design engineer. Such an analysis procedure for finding useful information about a problem via a multiobjective optimization task has been termed *innovization* task in a recent study [56]. Here, we show a higher level concept of the innovization task in which salient relationships among *reliable* tradeoff solutions are revealed. We strongly recommend pursuing such a task to other engineering design tasks for the sake of unveiling important problem knowledge.

VIII. Simulation Results on Multiobjective Reliability-Based Optimization Problems

Finally, we consider a couple of two-objective optimization problems to illustrate the effect of considering reliability in multiobjective optimization.

A. Test Problem

First, we solve a two-variable, two-objective test problem [49]

$$\begin{aligned} &\text{Minimize } f_1 = x \\ &\text{minimize } f_2 = \tfrac{1+y}{x} \\ &\text{subject to } y + 9x - 6 \geq 0, \\ &\qquad -y + 9x - 1 \geq 0, \\ &\qquad 0.1 \leq x \leq 1, \quad 0 \leq y \leq 5. \end{aligned} \tag{23}$$

Both variables are uncertain: $\mathbf{x} = (x, y)$ with $\sigma = 0.03$. We use a population of size 50 and run NSGA-II for 50 generations. Fig. 25 shows the deterministic front and three reliable frontiers with β^r equal to 1.28 (90%), 2.0 (97.725%), and 3.0 (99.875%), respectively.

To demonstrate the principle of using a specified reliability index for a multiobjective optimization problem, in this problem, we employ the closest constraint strategy alone. In the next problem, we shall use the multiple constraint strategies.

A part of the deterministic optimal front lies on constraint g_1; however, the minimum f_1 solution lies on constraint g_2 as well, as shown in Fig. 26. Fig. 25 shows how the reliable tradeoff frontier moves inside the feasible objective space as β^r increases. For $\beta^r > 0$, both constraints govern the location of the reliable tradeoff frontier. The theoretical change in the minimum f_1 solution is marked ("Boundary") in Fig. 25. The figure indicates that optimal solutions for small f_2 are more reliable and less vulnerable to change due to reliability consideration than the small f_1 solutions.

Fig. 26 supports this argument. The figure shows how the solutions get inside the feasible region with an increase in β^r. To be safe from both constraints, the minimum f_1 solution must be moved equally away from both constraints, as shown in the inset figure. The circle indicates that the case in which the $\beta^r = 2$ variation boundary touches both constraints. Thus, in the presence of uncertainties in decision variables, a part of the deterministic optimal frontier is sensitive and a new frontier becomes an optimal choice. This paper demonstrates that if the user is interested in finding an optimal frontier which is insensitive to variable uncertainties with a particular reliability index, NSGA-II with the handling of chance constraints described in this paper remains as a viable approach for the task. We reconsider the car side-impact problem and attempt to explain the importance of this task better.

B. Car Side-Impact Problem Revisited

We use the car side-impact problem discussed earlier, but now use an additional objective of minimizing the average rib deflection, which is calculated by taking the average of three deflections $g_5(\mathbf{x})$, $g_6(\mathbf{x})$, and $g_7(\mathbf{x})$. All 10 constraints are considered. Fig. 27 shows the reliable front as a function of β^r using the closest constraint strategy.

Once again, with an increase in the reliability index, the optimal frontier gets worse. We observe the following features from the figure.

1) The figure indicates the rate at which the front deteriorates. In this problem, the rate of deterioration

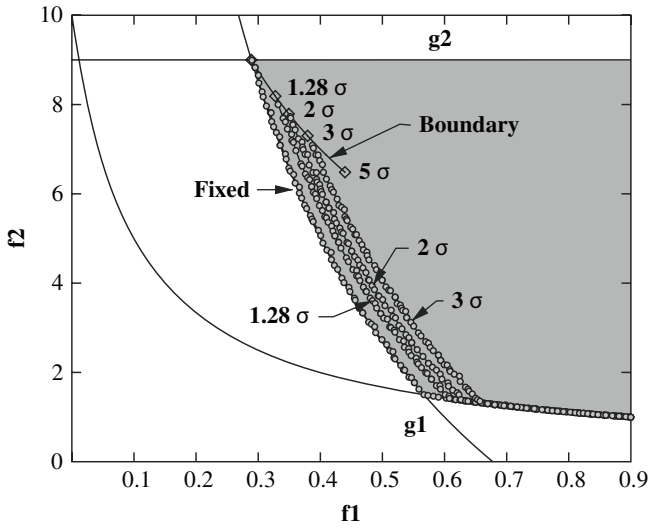

Fig. 25. Tradeoff frontiers between f_1 and f_2 for different β^r for the two-constraint test problem. The closest constraint strategy is used here.

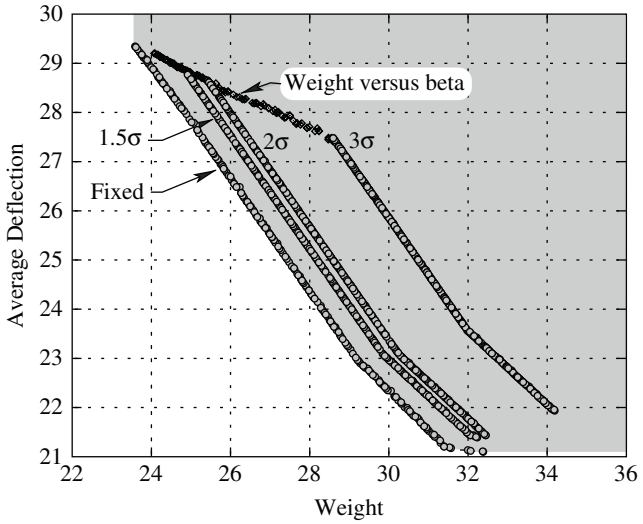

Fig. 27. Tradeoff frontiers between f_1 and f_2 for different β^r for the car side-impact problem using the closest constraint strategy.

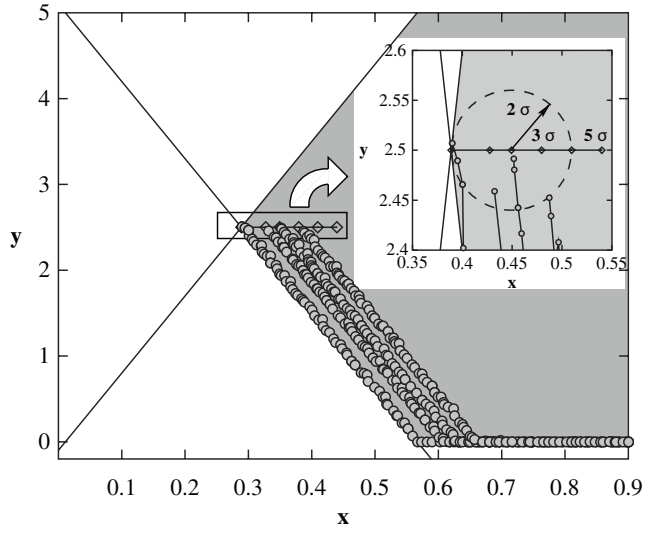

Fig. 26. Corresponding reliable solutions for the two-constraint test problem.

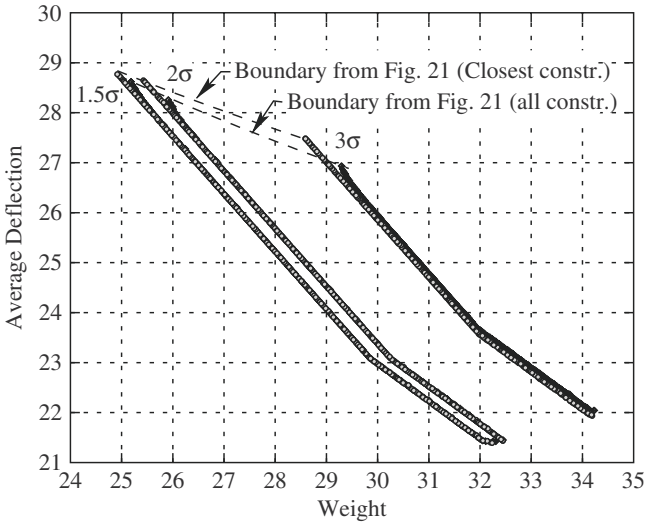

Fig. 28. Tradeoff frontiers using the closest and multiple constraint strategies.

seems to be faster than linear, as was also discussed in Section VII-B. Thus, an unnecessary large reliability index corresponds to solutions that are far from being optimum. Designers must carefully set a reliability index to make a good compromise of optimality and reliability of solutions.

2) An interesting fact about this problem is that the front moves inside the feasible objective space parallel to each other, indicating that the whole front is uniformly sensitive to a change in the reliability index.
3) The near minimum-weight solutions are found to be more sensitive to the chosen reliability index. The optimal solutions obtained in Fig. 21 in Section VII-B are also plotted in Fig. 27 (marked as "Weight versus beta"). Interestingly, these solutions mark the boundary to the obtained NSGA-II solutions of this section. The support of the optimization results obtained in this section by those obtained from a different optimization task on the same problem provides confidence as to the accuracy of the obtained solutions and efficiency of the proposed procedure.

Fig. 28 shows a comparison of tradeoff frontiers obtained using the closest constraint strategy and the multiple constraints strategy (with all constraints considered) for three different reliability values. In all three cases, the difference occurs near the minimum-weight region of the tradeoff frontiers. The minimum-weight bondaries for both closest and all-constraint strategies are plotted on the figure using dashed lines taken from Fig. 21. Recall from Fig. 21 that for minimum-weight solutions, the computation of reliability index using closest constraint strategy is different from that for multiple constraints using the Ditlevsen's bound. However, surprisingly, for most other regions of the tradeoff frontier, both strategies find almost identical solutions, except that for $\beta^r = 3$, there is slight visible difference between the obtained fronts. To understand this behavior better, we compute the distance of

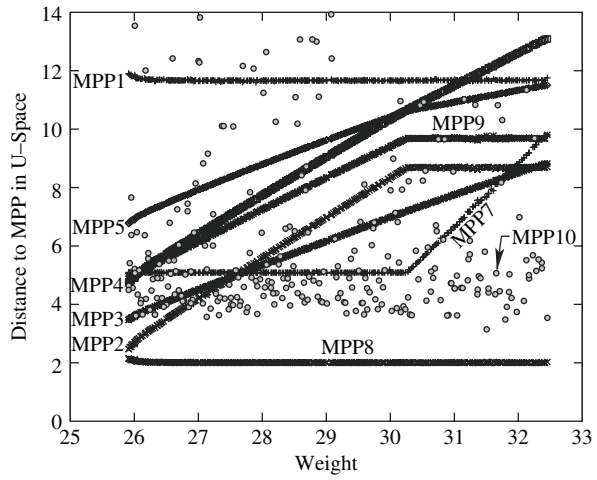

Fig. 29. Distance to MPP points (with $\beta^r = 2$), revealing that only one constraint is critical.

MPP from the current solution for each obtained tradeoff solution for $\beta^r = 2$ and plot them in Fig. 29. Here, the MPP points for constraint g_6 fall outside the range of the figure and are not shown. It is clear that only one constraint (g_8) is critical for all tradeoff solutions and lies at a distance of 2 (thereby corresponding to a reliability index of 2). Since only one constraint is critical, there is not much of a difference between the closest constraint and multiple constraint strategies observed in Fig. 28. However, near the minimum-weight region, the MPP for constraint g_2 is at a similar distance to that of constraint g_8, and the presence of two critical constraints make the reliability computation using closest constraint strategy erroneous, thereby causing a shift in the obtained frontier near the minimum-weight region. It is also important to report that the multiple constraint strategy with the redundant constraint consideration finds a frontier which is indistinguishable from the one shown in Fig. 28 with all constraints and is not plotted in the figure for brevity.

IX. Conclusion

In this paper, we have reviewed the recent classical methods for handling uncertainties in arriving at reliable solutions, instead of deterministic optimal solutions, in an optimization problem involving uncertainty. By reviewing these methodologies, we have identified at least three different problem domains in which the proposed evolutionary reliability-based optimization approaches have an edge over their classical counterparts. The problems have complexities—multimodality and multiobjectiveness—which are difficult to handle using a classical point-by-point approach. Here, we have developed a couple of evolutionary optimization-based approaches for handling probabilistic constraints under uncertainties in decision variables and/or problem parameters to solve the problems to satisfaction. The suggested methodology has considered both accuracy of obtained solutions and computational overhead by using a system reliability approach and by identifying redundant constraints. On a number of test problems and an automobile design problem, the proposed procedures have shown their efficacy in quickly (about 67% savings in constraint computations) finding the desired reliable solution(s). In the car side-impact design problem, a number of interesting properties about the reliable solutions have been revealed. The proposed evolutionary methods are compared with a state-of-the-art classical methodology, and the niche of the former in single and multiobjective reliability-based optimization has been clearly demonstrated.

We have also compared two reliability computation strategies—the closest constraint and all-constraint strategies. The simulation results clearly show that when a single constraint determines the location of the reliable optimum, both methods perform identically. However, if multiple critical constraints exist near the optimum, the closest constraint strategy overestimates the optimum, but the use of all-constraint strategy is recommended.

This paper should encourage researchers and practitioners in the area of classical reliability-based design optimization to pay more attention to EA-based search and optimization procedures and vice versa, a process which may lead to the development of more such hybrid evolutionary-classical RBDO approaches in the coming years.

Appendix

Description of the Car Side-Impact Problem

Seven decision variables (x_1 to x_7) and four stochastic parameters x_8 to x_{11} are described as follows:

x_1: Thickness of B-Pillar inner (0.03);
x_2: Thickness of B-Pillar reinforcement (0.03);
x_3: Thickness of floor side inner (0.03);
x_4: Thickness of cross members (0.03);
x_5: Thickness of door beam (0.05);
x_6: Thickness of door beltline reinforcement (0.03);
x_7: Thickness of roof rail (0.03);
x_8: Material of B-Pillar inner (0.006);
x_9: Material of floor side inner (0.006);
x_{10}: Barrier height (10);
x_{11}: Barrier hitting position (10).

The quantity in bracket shows the standard deviation of stochastic variation of each variables. The optimization problem formulation is as follows:

$$\begin{aligned}
&\underset{(x_1,\ldots,x_7)}{\text{Min.}} && f(\mathbf{x}) = \text{Weight} \\
&\text{s.t.} && g_1(\mathbf{x}) \equiv \text{Abdomen load} \le 1 \text{ kN}; \\
& && g_2(\mathbf{x}) \equiv V * C_u \le 0.32 \text{ m/s}; \\
& && g_3(\mathbf{x}) \equiv V * C_m \le 0.32 \text{ m/s}; \\
& && g_4(\mathbf{x}) \equiv V * C_l \le 0.32 \text{ m/s}; \\
& && g_5(\mathbf{x}) \equiv \text{upper rib deflection} \le 32 \text{ mm}; \\
& && g_6(\mathbf{x}) \equiv \text{middle rib deflection} \le 32 \text{ mm}; \\
& && g_7(\mathbf{x}) \equiv \text{lower rib deflection} \le 32 \text{ mm}; \\
& && g_8(\mathbf{x}) \equiv \text{Pubic force} \le 4 \text{ kN}; \\
& && g_9(\mathbf{x}) \equiv \text{Vel. of V-Pillar at mid. pt.} \le 9.9 \text{ mm/ms}; \\
& && g_{10}(\mathbf{x}) \equiv \text{Front door vel. at V-Pillar} \le 15.7 \text{ mm/ms}; \\
& && 0.5 \le x_1 \le 1.5, \quad 0.45 \le x_2 \le 1.35; \\
& && 0.5 \le x_3 \le 1.5, \quad 0.5 \le x_4 \le 1.5; \\
& && 0.875 \le x_5 \le 2.625, \quad 0.4 \le x_6 \le 1.2; \\
& && 0.4 \le x_7 \le 1.2.
\end{aligned} \tag{24}$$

The functional forms of the objective function and constraints are given below:

$$\begin{aligned} f(\mathbf{x}) &= 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 \\ &\quad + 4.01x_4 + 1.78x_5 + 0.00001x_6 \\ &\quad + 2.73x_7, \end{aligned} \tag{25}$$

$$\begin{aligned} g_1(\mathbf{x}) &= 1.16 - 0.3717x_2x_4 \\ &\quad - 0.00931x_2x_{10} - 0.484x_3x_9 \\ &\quad + 0.01343x_6x_{10}, \end{aligned} \tag{26}$$

$$\begin{aligned} g_2(\mathbf{x}) &= 0.261 - 0.0159x_1x_2 \\ &\quad - 0.188x_1x_8 - 0.019x_2x_7 \\ &\quad + 0.0144x_3x_5 + 0.87570.001x_5x_{10} \\ &\quad + 0.08045x_6x_9 + 0.00139x_8x_{11} \\ &\quad + 0.00001575x_{10}x_{11}, \end{aligned} \tag{27}$$

$$\begin{aligned} g_3(\mathbf{x}) &= 0.214 + 0.00817x_5 \\ &\quad - 0.131x_1x_8 - 0.0704x_1x_9 \\ &\quad + 0.03099x_2x_6 - 0.018x_2x_7 \\ &\quad + 0.0208x_3x_8 + 0.121x_3x_9 \\ &\quad - 0.00364x_5x_6 + 0.0007715x_5x_{10} \\ &\quad - 0.0005354x_6x_{10} + 0.00121x_8x_{11} \\ &\quad + 0.00184x_9x_{10} - 0.018x_2x_2, \end{aligned} \tag{28}$$

$$\begin{aligned} g_4(\mathbf{x}) &= 0.74 - 0.61x_2 - 0.163x_3x_8 \\ &\quad + 0.001232x_3x_{10} - 0.166x_7x_9 \\ &\quad + 0.227x_2x_2, \end{aligned} \tag{29}$$

$$\begin{aligned} g_5(\mathbf{x}) &= 28.98 + 3.818x_3 \\ &\quad - 4.2x_1x_2 + 0.0207x_5x_{10} \\ &\quad + 6.63x_6x_9 - 7.77x_7x_8 \\ &\quad + 0.32x_9x_{10}, \end{aligned} \tag{30}$$

$$\begin{aligned} g_6(\mathbf{x}) &= 33.86 + 2.95x_3 \\ &\quad + 0.1792x_{10} - 5.057x_1x_2 \\ &\quad - 11x_2x_8 - 0.0215x_5x_{10} \\ &\quad - 9.98x_7x_8 + 22x_8x_9, \end{aligned} \tag{31}$$

$$\begin{aligned} g_7(\mathbf{x}) &= 46.36 - 9.9x_2 - 12.9x_1x_8 \\ &\quad + 0.1107x_3x_{10}, \end{aligned} \tag{32}$$

$$\begin{aligned} g_8(\mathbf{x}) &= 4.72 - 0.5x_4 - 0.19x_2x_3 \\ &\quad - 0.0122x_4x_{10} + 0.009325x_6x_{10} \\ &\quad + 0.000191x_{11}x_{11}, \end{aligned} \tag{33}$$

$$\begin{aligned} g_9(\mathbf{x}) &= 10.58 - 0.674x_1x_2 \\ &\quad - 1.95x_2x_8 + 0.02054x_3x_{10} \\ &\quad - 0.0198x_4x_{10} + 0.028x_6x_{10}, \end{aligned} \tag{34}$$

$$\begin{aligned} g_{10}(\mathbf{x}) &= 16.45 - 0.489x_3x_7 \\ &\quad - 0.843x_5x_6 + 0.0432x_9x_{10} \\ &\quad - 0.0556x_9x_{11} - 0.000786x_{11}x_{11}. \end{aligned} \tag{35}$$

REFERENCES

[1] S. S. Rao, "Genetic algorithmic approach for multiobjective optimization of structures," in *Proc. ASME Annu. Winter Meeting Structures Controls Optimization*, vol. 38. 1993, pp. 29–38.

[2] O. Ditlevsen and H. O. Madsen, *Structural Reliability Methods*. New York: Wiley, 1996.

[3] J. Liang, Z. Mourelatos, and J. Tu, "A single loop method for reliability-based design optimization," in *Proc. ASME Design Eng. Tech. Conf.*, 2004.

[4] T. R. Cruse, *Reliability-Based Mechanical Design*. New York: Marcel Dekker, 1997.

[5] D. H. Loughlin and S. R. Ranjithan, "Chance-constrained genetic algorithms," in *Proc. Genetic Evol. Comput. Conf.*, San Francisco, CA: Morgan Kaufmann, 1999, pp. 369–376.

[6] H. C. Gea and K. Oza, "Two-level approximation method for reliability-based design optimisation," *Int. J. Materials Product Technol.*, vol. 25, no. 1–3, pp. 99–111, 2006.

[7] K. Deb and P. Chakroborty, "Time scheduling of transit systems with transfer considerations using genetic algorithms," *Evol. Comput. J.*, vol. 6, no. 1, pp. 1–24, 1998.

[8] R. T. F. King, H. C. S. Rughooputh, and K. Deb, "Evolutionary multiobjective environmental/economic dispatch: Stochastic versus deterministic approaches," in *Proc. 3rd Int. Conf. Evol. Multicriterion Optimization*, LNCS vol. 3410, 2005, pp. 677–691.

[9] A. Harbitz, "An efficient sampling method for probability of failure calculation," *Structural Safety*, vol. 3, pp. 109–115, 1986.

[10] O. Ditlevsen and P. Bjerager, "Plastic reliability analysis by directional simulation," *ASCE J. Eng. Mechanics*, vol. 115, no. 6, pp. 1347–1362, 1989.

[11] G. Wang, L. Wang, and S. Shan, "Reliability assessment using discriminative sampling and metamodeling," *SAE Trans., J. Passenger Cars–Mech. Syst.*, vol. 114, pp. 291–300, 2005.

[12] A. M. Hasofer and N. C. Lind, "Exact and invariant second-moment code format," *ASCE J. Eng. Mechanics Devision*, vol. 100, no. EM1, pp. 111–121, 1974.

[13] M. Rosenblatt, "Remarks on a multivariate transformation," *Ann. Math. Statist.*, vol. 23, pp. 470–472, 1952.

[14] X. Du and W. Chen, "A most probable point based method for uncertainty analysis," *J. Design Manuf. Automat.*, vol. 4, no. 1, pp. 47–66, 2001.

[15] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization Methods and Applications*. New York: Wiley, 1983.

[16] H. Agarwal, "Reliability-based design optimization: Formulations and methodologies," Ph.D. dissertation, Dept. Mech. Eng., Univ. Notre Dame, Notre Dame, IN, 2004.

[17] R. Yang, C. Chuang, L. Gu, and G. Li, "Experience with approximate reliability-based optimization methods II: An exhaust system roblem," *Structural and Multidisciplinary Optimization*, vol. 6, no. 29, pp. 488–497, 2005.

[18] X. Du and W. Chen, "Sequential optimization and reliability assessment method for efficient probabilistic design," *ASME Trans. J. Mech. Design*, vol. 126, no. 2, pp. 225–233, 2004.

[19] H. Agarwal, J. Renaud, J. Lee, and L. Watson, "A unilevel method for reliability-based design optimization," in *Proc. 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dyn., Materials Conf.*, Palm Springs, CA, 2004, paper ID: AIAA-2004–2029.

[20] O. Ditlevsen, "English Narrow reliability bounds for structural system," *Eng. J. Structural Mech.*, vol. 4, no. 1, pp. 431–439, 1979.

[21] C. Cornell, "Bounds on the reliability of structural systems," *ASCE J. Struct. Div.*, vol. 93, pp. 171–200, 1967.

[22] K. Ramachandran and M. Baker, "New reliability bound for series systems," in *Proc. Int. Conf. Structural Safety Reliability*, 1985, pp. 157–169.

[23] J. Branke, "Reducing the sampling variance when searching for robust solutions," in *Proc. Genetic Evol. Comput. Conf.*, San Mateo, CA: Morgan Kaufmann, 2001, pp. 235–242.

[24] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments–A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[25] S. Tsutsui and A. Ghosh, "Genetic algorithms with a robust solution searching scheme," *IEEE Trans. Evol. Comput.*, vol. 1, no. 3, pp. 201–208, Jun. 1997.

[26] D. Wiesmann, U. Hammel, and T. Bäck, "Robust design of multilayer optical coatings by means of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 162–167, Nov. 1998.

[27] I. Paenke, J. Branke, and Y. Jin, "Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 405–420, Aug. 2006.

[28] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," *Evol. Comput. J.*, vol. 14, no. 4, pp. 463–494, 2006.

[29] K. Deb and H. Gupta, "Handling constraints in robust multiobjective optimization," in *Proc. Congr. Evol. Comput.* Piscataway, NJ: IEEE Press, 2005, pp. 450–457.

[30] B. Forouraghi, *A Genetic Algorithm for Multiobjective Robust Design*. New York: Springer-Verlag, 2000.

[31] I. Elishakoff, R. T. Haftka, and J. Fang, "Structural design under bounded uncertainty-optimization with anti-optimization," *Comput. Structures*, vol. 53, pp. 1401–1405, 1994.

[32] Y.-S. Ong, P. B. Nair, and K. Y. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 392–404, Aug. 2006.

[33] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B.-S. Lee, "Inverse multi-objective robust evolutionary design," *Genetic Programming Evolvable Mach.*, vol. 7, no. 4, pp. 383–404, 2006.
[34] M. Tjornfelt-Jensen and T. K. Hansen, "Robust solutions to job shop problems," in *Proc. Congr. Evol. Comput.*, vol. 2. Piscataway, NJ: IEEE Press, 1999, pp. 1138–1144.
[35] C. P. Pantelides and S. Ganzerli, "Design of trusses under uncertain loads using convex models," *J. Structural Eng.*, vol. 124, no. 3, pp. 318–329, 1998.
[36] C. K. Goh and K. C. Tan, "Evolving the tradeoffs between pareto-optimality and robustness in multiobjective evolutionary algorithms," in *Proc. Evol. Comput. Dynamic Uncertain Environments*, New York: Springer-Verlag, 2007, pp. 457–478.
[37] I. Das, "Robustness optimization for constrained nonlinear programming problems," *Eng. Optimization*, vol. 32, no. 5, pp. 585–618, 2000.
[38] Y. Jin and B. Sendhoff, "Tradeoff between performance and robustness: An evolutionary multiobjective approach," in *Proc. Evol. Multicriterion Optimization*, ser. LNCS 2632. New York: Springer-Verlag, 2003, pp. 237–251.
[39] M. Li, A. Azarm, and V. Aute, "A multiobjective genetic algorithm for robust design optimization," in *Proc. Genetic Evol. Comput. Conf.*, New York: ACM, 2005, pp. 771–778.
[40] K. Deb, D. Padmanabhan, S. Gupta, and A. K. Mall, "Reliability-based multiobjective optimization using evolutionary algorithms," in *Proc. Evol. Multicriterion Optimization*, ser. LNCS vol. 4403. New York: Springer-Verlag, 2007, pp. 66–80.
[41] D. Daum, K. Deb, and J. Branke, "Reliability-based optimization for multiple constraint with evolutionary algorithms," in *Proc. Congr. Evol. Comput.*, Piscataway, NJ: IEEE Press, 2007, pp. 911–918.
[42] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Comput. Methods Appl. Mechanics Eng.*, vol. 196, no. 33–34, pp. 3190–3218, 2007.
[43] N. D. Lagaros, V. Plevris, and M. Papadrakakis, "Multiobjective design optimization using cascade evolutionary computations," *Comput. Methods Appl. Mechanics Eng.*, vol. 194, no. 30–33, pp. 3496–3515, 2005.
[44] R. Roy, I. C. Parmee, and G. Purchase, "Sensitivity analysis of engineering designs using taguchi's methodology," in *Proc. ASME DETC-Design Automation Conf.*, 1996, paper Number 96-DETC/DAC-1455, CD-ROM.
[45] L. Gu and R. Yang, "On reliability-based optimisation methods for automotive structures," *Int. J. Materials Product Technol.*, vol. 25, no. 1–3, pp. 3-–26, 2006.
[46] D. E. Salazar and C. M. Rocco, "Solving advanced multiobjective robust designs by means of multiple objective evolutionary algorithms (MOEA): A reliability application," *Reliability Eng. Syst. Safety*, vol. 92, no. 6, pp. 697–706, 2007.
[47] J. H. Holland, *Adaptation Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
[48] D. E. Goldberg, *Genetic Algorithms for Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[49] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
[50] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. New York: North-Holland, 1983.
[51] P. Shukla and K. Deb, "On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1630–1652, 2007.
[52] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mechanics Eng.*, vol. 186, no. 2–4, pp. 311–338, 2000.
[53] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[54] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[55] L. Gu, R. J. Yang, C. H. Tho, L. Makowski, O. Faruque, and Y. Li, "Optimization and robustness for crashworthiness of side-impact," *Int. J. Vehicle Design*, vol. 26, no. 4, pp. 348–360, 2001.
[56] K. Deb and A. Srinivasan, "Innovization: Innovating design principles through optimization." in *Proc. Genetic Evol. Comput. Conf.*, New York: ACM, 2006, pp. 1629–1636.

Kalyanmoy Deb received the B.S. degree in mechanical engineering from the Indian Institute of Technology (IIT), Kharagpur, India, in 1985 and the M.S. and Ph.D. degrees from the University of Alabama, Tuscaloosa, in 1989 and 1991, respectively.

He currently holds the Deva Raj Chair Professor position of Mechanical Engineering at IIT Kanpur, India and is also a Finland Distinguished Professor at Helsinki School of Economics, Finland. His main research interests are in the areas of computational optimization, modeling and design, and evolutionary algorithms. He has written two text books on optimization and published or presented more than 235 papers in international journals and conferences. He has pioneered and is a leader in the field of evolutionary multiobjective optimization. He is on the editorial boards of a number of major international journals.

Prof. Deb is a Fellow of the Indian National Academy of Engineering, the Indian National Academy of Sciences, and the International Society of Genetic and Evolutionary Computation. He is the recipient of the prestigious Shanti Swarup Bhatnagar Prize in Engineering Sciences in India for 2005. He has also received the Thomson Citation Laureate Award from Thompson Scientific for having highest number of citations in computer science during 1996–2005 in India. He received the Fredrick Wilhelm Bessel Research award from the Alexander von Humboldt Foundation, Germany, in 2003.

Shubham Gupta was born in Jhansi, India, on April 5, 1986. He received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Kanpur, India, in 2007. He is currently pursuing the Ph.D. degree in operations research with the Massachusetts Institute of Technology, Cambridge.

His current research area is air traffic flow management.

David Daum received the Diploma in industrial engineering from the University of Karlsruhe, Karlsruhe, Germany, in 2007. He is currently pursuing the Ph.D. degree with the Swiss Federal Institute of Technology, Lausanne, Switzerland, in the Solar Energy and Building Physics Laboratory.

His research interest include optimization and self-adaptation of intelligent home automation to receive both energy efficiency and higher comfort levels.

Jürgen Branke received the Diploma in industrial engineering and management science and the Ph.D. degree from the University of Karlsruhe, Karlsruhe, Germany, in 1994 and 2000, respectively.

He is currently an Associate Professor with the Warwick Business School, The University of Warwick, Coventry, U.K. He has been active in the area of nature-inspired optimization since 1994 and is a leading expert on optimization in the presence of uncertainties, including noisy or dynamically changing environments. Further research interests include complex system optimization, organic computing, multiobjective optimization, and simulation optimization. He has co-authored over 100 peer-reviewed papers in international journals, conferences, and workshops. He has also applied nature-inspired optimization techniques to several real-world problems as part of industry projects.

Abhishek Kumar Mall was born on April 24, 1985. He received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Kanpur, India, in 2007.

He is currently working at Deutsche Bank, Mumbai, India.

Dhanesh Padmanabhan received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology, Madras, India, in 1998, and the Ph.D. degree in mechanical engineering from the University of Notre Dame, Notre Dame, IN, in 2003.

He is currently a manager of a pricing analytics group at the Decision Support and Analytic Services Division, Hewlett Packard, Bangalore, India. Earlier, he was a senior statistician at the analytics consulting firm Marketics, working on the application of statistics and data mining to B2B marketing. Prior to this, he was a researcher at General Motors Research and Development, Bangalore, where he worked on advanced vehicles development and specialized in areas of optimization and reliability analysis. His interests include optimization, statistics, data mining, and econometrics.