Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design

Probabilistic design, such as reliability-based design and robust design, offers tools for making reliable decisions with the consideration of uncertainty associated with design variables/parameters and simulation models. Since a probabilistic optimization often involves a double-loop procedure for the overall optimization and iterative probabilistic assessment, the computational demand is extremely high. In this paper, the sequential optimization and reliability assessment (SORA) is developed to improve the efficiency of probabilistic optimization. The SORA method employs a single-loop strategy with a serial of cycles of deterministic optimization and reliability assessment. In each cycle, optimization and reliability assessment are decoupled from each other; the reliability assessment is only conducted after the deterministic optimization to verify constraint feasibility under uncertainty. The key to the proposed method is to shift the boundaries of violated constraints (with low reliability) to the feasible direction based on the reliability information obtained in the previous cycle. The design is quickly improved from cycle to cycle and the computational efficiency is improved significantly. Two engineering applications, the reliability-based design for vehicle crashworthiness of side impact and the integrated reliability and robust design of a speed reducer, are presented to demonstrate the effectiveness of the SORA method. [DOI: 10.1115/1.1649968]

1 Introduction

Probabilistic design methods have been developed and applied in engineering design. The typical probabilistic design methods include reliability-based design [1–4] and robust design [5–9]. Reliability-based design emphasizes high reliability of a design by ensuring the probabilistic constraint satisfaction at desired levels, while robust design focuses on making the design inert to the variations of system input through optimizing mean performance of the system and minimizing its variance simultaneously. One important task of a probabilistic design is uncertainty analysis, through which we understand how much the impact of the uncertainty associated with the system input is on the system output by identifying the probabilistic characteristics of system output. We then perform synthesis (optimization) under uncertainty to achieve the design objective by managing and mitigating the effects of uncertainty on system output (system performance) [10].

In spite of the benefits of probabilistic design, one of the most challenging issues for implementing probabilistic design is associated with the intensive computational demand of uncertainty analysis. To capture the probabilistic characteristics of system performance at a design point, we need to perform a number of deterministic analyses around the nominal point, either using a simulation approach (for instance, Monte Carlo simulation) or other probabilistic analysis methods (such as reliability analysis). Considerable research has been concentrating on developing practical means to make probabilistic design computationally feasible for complex engineering problems.

Our focus in this study is to develop an efficient probabilistic design approach to facilitate design optimizations that involve probabilistic constraints. Reliability-based design is such a type of probabilistic optimization problem in which design feasibility is formulated as a reliability constraint (or the probability of constraint satisfaction). The conventional approach for solving a probabilistic optimization problem is to employ a double-loop strategy; the analysis and the synthesis are nested in such a way that the synthesis loop (outer loop) performs the uncertainty analysis (inner loop for reliability assessment) iteratively for meeting the probabilistic objective and constraints. As the double-loop strategy may be computationally infeasible, various techniques have been developed to improve its efficiency. These techniques can be classified into two categories: one is through improving the efficiency of uncertainty analysis methods, for example, the methods of Fast Probability Integration [11] and Two-Point Adaptive Nonlinear Approximations [3]; the other is by modifying the formulation of probabilistic constraints, for example, the performance measure approach [12]. A comprehensive review of various feasibility modeling approaches for design under uncertainty is provided in Du and Chen [13].

Even though the improved uncertainty analysis techniques and modifications of problem formulation have lead to improved efficiency of probabilistic optimization, the improvement is quite limited due to the nature of the double loop strategy. Recent years have seen preliminary studies on a new type of method—the single loop method [14–16] that avoids the nested loops of optimization and reliability assessment. In [14], the reliability constraints are formulated as deterministic constraints that approximate the condition of the Most Probable Point (MPP) [17], a concept used for reliability assessment. Since this method does not conduct the expensive MPP search, its efficiency is very high. However, there is no guarantee that an active reliability constraint converges to its actual MPP; the optimal solution may not satisfy the required reliability. In [15], a method using “approximately equivalent deterministic constraints” was developed. The method creates a link between a probabilistic design and a safety-factor based design. With this method, only deterministic design variables are considered and the uncertainties can be only associated with (uncontrollable) design parameters. In [16], optimization and reliability assessment are decoupled in each cycle. In optimization, reliability constraints are linearized around the MPPs obtained in the reliability assessment of the previous cycle. The linearization improves the efficiency of overall optimization but may also lead to convergence difficulties. Although the single loop strategy appears promising as no nested synthesis and uncertainty analysis loops
are involved, the above methods are relatively new and their applicability to various design applications is yet to be verified.

In this paper, we present a new probabilistic design method, the Sequential Optimization and Reliability Assessment (SORA) method that we believe can significantly improve the efficiency of probabilistic optimization. Our method employs a single loop strategy which decouples optimization synthesis and uncertainty analysis. As an integral part of the proposed strategy, we employ the formulation of performance measure for the reliability constraints along with an efficient inverse MPP search algorithm. The SORA method has the capability to deal with both deterministic and random design variables with the presence of random parameters. In this paper, we will first review a few commonly used strategies of probabilistic design in Sec. 2. The review will lay the foundation for our proposed method, SORA, introduced in Sec. 3. In Sec. 4 two engineering examples are used to illustrate the effectiveness of the proposed method. Section 5 is the closure, which highlights the effectiveness of the proposed method and provides discussions on its applicability under different circumstances.

2 Probabilistic Optimization Strategies

In this section, we present two commonly used formulations under the double-loop strategy, which lays the foundation for our proposed method. Results from these two formulations are compared with those from our proposed method in case studies.

2.1 Double-Loop Strategy With Probabilistic Formulation.

A typical model of a probabilistic design is given by:

Minimize: \( f(d, X, P) \)

Design Variable: \( DV = \{d, \mu_x\} \)

Subject to: \( \text{Prob}(g_i(d, X, P) \leq R_i, i = 1, 2, \ldots, m) \geq 0 \)

where \( f \) is an objective function, \( d \) is the vector of deterministic design variables, \( X \) is the vector of random design variables, \( P \) is the vector of random design parameters, \( g_i(d, X, P) \) are constraint functions, \( R_i \) are the reliability requirements, and \( m \) is the number of constraints. The design variables are \( d \) and the means (\( \mu_x \)) of the random design variables \( X \). Note that the following rules of symbols are used to differentiate the representation of random variables, deterministic variables, and vectors. A capital letter is used for a random variable, a lower case letter for a deterministic variable or a realization of a random variable, and a bold letter is used for a vector. For example, \( X \) stands for a random variable and \( x \) for a deterministic variable or a realization of random variable \( x \); \( X \) denotes a vector of random variables while \( x \) denotes a vector of deterministic variables.

In the above probabilistic design model, the design feasibility is formulated as the probability (Prob) of constraint satisfaction \( g(d, X, P) \leq 0 \) less than or equal to a desired reliability \( R \). As shown in Fig. 1, the probability of \( g(d, X, P) \leq 0 \) is the area underneath the curve of probability density function (PDF) of \( g \) for \( g \leq 0 \), and this area should be bigger than or equal to \( R \).

The probability of constraint satisfaction is also called reliability. Analytically, the reliability is given by the integral

\[
\text{Prob}(g(d, X, P) \leq 0) = \int \cdots \int_{g(d, X, P) \leq 0} h_{X}(x, p) dx dp. \tag{2}
\]

where \( h_{X}(x, p) \) is the joint probability density function of \( X \) and \( P \), and the probability is evaluated by the multidimensional integration over the region \( g(d, X, P) \leq 0 \). It is generally difficult or even impossible to perform the multidimensional integration in Eq. (2). One alternative method to evaluate the integration is Monte Carlo simulation. However, when the reliability is very high (approaching 1.0), the computational effort of Monte Carlo simulation is prohibitively expensive [13]. Hasofer and Lind [17] proposed the concept of the Most Probable Point (MPP) in the structural reliability field to approximate the integration.

With the MPP approach, the random variables \( (X, P) \) are transformed into an independent and standardized normal space \( (U_X, U_P) \). The MPP is formally defined in the standardized normal space as the minimum distance point on the constraint boundary \( g(d, U_X, U_P) = 0 \) to the origin. The minimum distance \( \beta \) is called reliability index. When the First Order Reliability Method (FORM) [17] is used, the reliability is given by

\[
\text{Prob}(g(d, X, P) \leq 0) = \Phi(\beta), \tag{3}
\]

where \( \Phi \) is the standard normal distribution function. Finding the MPP and the reliability index is a minimization problem, which usually involves an iterative search process. Therefore, the reliability assessment itself is an optimization problem. For details about the MPP based method, refer to [18].

When the probability formulation in design model (1) is directly used to solve the problem, the method is called “double-loop method with probability formulation” (DLM_Prob) [12,19,20]. The efficiency of this type of method is usually low since it employs nested optimization loops to first evaluate the reliability of each probabilistic constraint and then to optimize the design objective subject to the reliability requirements.

2.2 Double-Loop Strategy With Percentile Formulation.

An equivalent model to (1) is given by [12,15,21]

Minimize: \( f(d, X, P) \)

Design Variable: \( DV = \{d, \mu_x\} \)

Subject to: \( g_i(d, X, P) \leq R_i, i = 1, 2, \ldots, m \),

where \( g^R \) is the \( R \)-percentile of \( g(d, X, P) \), namely,

\[
\text{Prob}(g(d, X, P) \leq g^R) = R. \tag{5}
\]

Equation (5) indicates that the probability of \( g(d, X, P) \) less than or equal to the \( R \)-percentile \( g^R \) is exactly equal to the desired reliability \( R \). The concept is demonstrated in Fig. 2. If the shaded area is equal to the desired reliability \( R \), then the left ending point \( g^R \) on the \( g \) axis is called the \( R \)-percentile of function \( g \). From Fig. 2 we see that, if \( g^R \leq 0 \), it indicates that \( \text{Prob}(g(d, X, P) \leq 0) \geq R \), i.e., the constraint is feasible. Therefore, the original
The percentile $g^k$ can be evaluated by the inverse MPP method. When using FORM, the desired reliability $R$, the reliability index $\beta$ is first calculated by

$$\beta = \Phi^{-1}(R)$$  \hspace{1cm} (6)

The inverse MPP problem is formulated as shown in the following minimization model,

$$\begin{align*}
\text{minimize} & \quad g(U) \\
\text{subject to} & \quad (U^TU)^{1/2} = \beta,
\end{align*}$$  \hspace{1cm} (7)

where $U = (U_x, U_p)$.

Using an inverse MPP search algorithm, the optimal solution MPP $u_{MPP}$ can be identified and the $R$ percentile is evaluated by

$$g^R = g(u_{MPP}) = g(x_{MPP} - p_{MPP}).$$  \hspace{1cm} (8)

To some extent, the evaluation of Eq. (8) can be viewed as deterministic by substituting the MPP values ($x_{MPP}$ and $p_{MPP}$ in the original random space) directly into the $g$ function. Since applying the inverse MPP method also involves iterative procedures, we call the method for solving model (4) “the double-loop method with percentile formulation” (DLM_Per). It is also called performance measure approach (PMA) in [12,21].

To distinguish the type of function evaluations for the probabilistic constraints (Eqs. (3) or (8)) from those for the original constraint functions $g(d, x, P)$, we call the function evaluations for the probabilities $\text{Prob}[g(d, x, P) \leq 0]$ or the $R$-percentile $g^R = g(u_{MPP}) = g(x_{MPP} - p_{MPP})$ “probabilistic function evaluations” and those for the original function $g(d, x, P)$ “the performance function evaluations” or simply “the function evaluations.”

For both DLM_Prob and DLM_Per, to fulfill the probabilistic optimization, the outer loop optimizer calls the objective function and each probabilistic constraint repeatedly. Therefore, the total number of function evaluations can be very huge. For instance, the optimization method adopted in this work the strategy of “serial single loops” [14-16] to develop a sequential optimization and reliability assessment (SORA) method. Our proposed method is different from the existing single loop methods in the way we establish equivalent deterministic constraints from the probabilistic constraints. We also employ an efficient inverse MPP search algorithm as an integral part of the proposed procedure.

### 3 Sequential Optimization and Reliability Assessment (SORA) Method

To improve the efficiency of probabilistic optimization, we adopt in this work the strategy of “serial single loops” [14-16] to develop a sequential optimization and reliability assessment (SORA) method. Our proposed method is different from the existing single loop methods in the way we establish equivalent deterministic constraints from the probabilistic constraints. We also employ an efficient inverse MPP search algorithm as an integral part of the proposed procedure.

#### 3.1 The Measures Taken in Developing the SORA Method

In developing the SORA method, several measures have been taken, including efficient evaluation of the reliability only at the desired level ($R$-percentile), using an efficient and robust inverse MPP search algorithm, and employing sequential cycles of optimization and reliability assessment.

(1) Evaluating the reliability only at the desired level ($R$-percentile)

It is noted that in probabilistic optimization, the closer the reliability $P[g(d, x, P) \leq 0]$ is to 1.0, the more computational effort is required. For using the MPP based methods, the higher reliability means larger search region in the standardized normal space to locate the MPP and it is very likely that more function evaluations are required. In probabilistic optimization with multiple constraints, some constraints may never be active and their reliabilities are extremely high (approaching 1.0). Although these constraints are the least critical, the evaluations of these reliabilities will unfortunately dominate the computational effort in the probabilistic design process if the DLM_Prob strategy (Sec. 2.1) is employed. Our solution to this problem is to perform the reliability assessment only up to the necessary level, represented by the desired reliability $R$.

To this end, we use the percentile formulation for probabilistic constraints with the SORA method. Based on Eq. (8), the design model (5) of DLM_Per is rewritten as

$$\begin{align*}
\text{Minimize:} & \quad f(d, \mu_x) \\
\text{Subject to:} & \quad g_i(d, x_{MPP} - p_{MPP}) \leq 0, \quad i = 1,2,\ldots,m.
\end{align*}$$  \hspace{1cm} (9)

This model establishes the equivalence between a probabilistic optimization and a deterministic optimization since the original constraint functions $g_i(d, x_{MPP} - p_{MPP})$ are used to evaluate design feasibility using the inverse MPPs corresponding to the desired reliabilities $R_i$. Figure 3 is used to further explain how a probabilistic constraint is converted to an equivalent deterministic constraint. In this illustrative example, only two random design variables $x_1$ and $x_2$ are involved; there are no random parameters $P$. Two coordinate systems are plotted in Fig. 3; one is the design space (the space composed of design variables $\mu_{x_1}$ and $\mu_{x_2}$), and the other is the random space ($x_1$ and $x_2$). If we do not consider any uncertainty, curve $g(\mu_{x_1}, \mu_{x_2}) = 0$ is the constraint boundary in the deterministic design. When we consider uncertainty, the constraint boundary becomes $\text{Prob}[g(x_1, x_2) \leq 0] = R$. Since in a probabilistic design, the required reliability $R$ is often much higher than the reliability achieved by a deterministic design, the constraint of a probabilistic design is stricter than a deterministic design. Geometrically, the feasible region of a probabilistic design is narrower than the one of a deterministic design; in other words, the feasible region of a probabilistic design is a reduced region in comparison with a deterministic feasible design. Determining the probabilistic constraint boundary $\text{Prob}[g(x_1, x_2) \leq 0] = R$ needs a reliability analysis. Since $\text{Prob}[g(x_1, x_2) \leq 0] = R$ is equivalent to $g(d, x_{MPP} - p_{MPP}) = 0$, where $(x_{MPP}, p_{MPP})$ is the inverse MPP point, the evaluation of a probabilistic constraint at design point $(\mu_{x_1}, \mu_{x_2})$ is equivalent to evaluating the deterministic constraint at the inverse MPP point, i.e., $g(d, x_{MPP} - p_{MPP})$. As shown in Fig. 3, to maintain the probabilistic constraint $g(d, x_{MPP} - p_{MPP}) = 0$, the inverse MPP corresponding to the design point $(\mu_{x_1}, \mu_{x_2})$ on the probabilistic constraint boundary should be exactly on the deterministic constraint boundary $g(\mu_{x_1}, \mu_{x_2}) = 0$. Therefore, to maintain the design feasibility, the inverse MPP of each probabilistic constraint should be within the deterministic feasible region.

(2) Using an efficient and robust inverse MPP search algorithm

In SORA, we employ an efficient MPP based percentile evaluation method (inverse MPP search algorithm) of which principle is detailed in [22]. This new inverse MPP search algorithm combines several techniques, such as using the steepest descent direc-

---

**Fig. 3 Probabilistic constraint**
tion as the search direction, performing an arc search if no progress is made along the steepest decent direction, and adopting the adaptive step size for numerical derivative evaluation. This search algorithm is considered robust since it is suitable for any continuous constraint functions (including non-concave and non-convex functions) and continuous distributions of uncertainty.

(3) Employing sequential cycles of optimization and reliability assessment

It is noted that in a probabilistic design, most of the computations are used for reliability assessments. Therefore, to improve the overall efficiency of probabilistic optimization we need to reduce the number of reliability assessments as much as possible. The essence is to move the design solution as quickly as possible to its optimum so as to reduce the needs for locating MPPs. To achieve this, SORA employs a serial of cycles of optimization and reliability assessment. Each cycle includes two parts, one part is the (deterministic) optimization and the other part is the reliability assessment. The reliability assessment refers to the evaluation of R-percentile corresponding to a given reliability R. In each cycle, at first we solve an equivalent deterministic optimization problem, which is formulated by the information of the inverse MPPs obtained in the last cycle. Once the design solution is updated, we then perform reliability assessment to identify the new inverse MPPs and to check if all the reliability requirements are satisfied. If not, we use the current inverse MPPs to formulate the constraint for the deterministic optimization in the next cycle in which the constraint boundary will be shifted to the feasible region by changing the locations of design variables. Using this strategy, the reliability of constraints improves progressively and the solution to a probabilistic design can be found within a few cycles, and the need for searching MPPs can be reduced significantly. Detailed flowchart and procedure are provided in Sec. 3.2.

3.2 SORA Flowchart and Procedure. The flowchart of the SORA method is provided in Fig. 4. For the deterministic optimization in the first cycle, since there is no information about the MPPs, the values of $x_{MPP}$ and $p_{MPP}$ are set as the means of the random design variables and the random parameters, respectively. The following is the deterministic optimization model in the first cycle of probabilistic optimization,

$$
\text{Minimize: } f(d, \mu_d, \mu_p)
$$

$$
DV = \{d, \mu_d\} \quad (10)
$$

Subject to: $g_i(d, \mu_d, \mu_p) \leq 0, \quad i = 1, 2, \ldots, m$

To demonstrate the strategy of separating (deterministic) optimization and reliability assessment while ensuring both segments work together to bring the design solution quickly to a feasible and optimal solution, we use the same illustrative plot (no deterministic design variables d and random parameters P) as shown in Fig. 3 for demonstration. We start our explanation for the first cycle and then extend the same principle to the 4th cycle. In the first cycle, after solving model (10) (deterministic optimization), some of the constraints may become active. For an active constraint $g_i$, the optimal point $\mu_d^{(1)} = (\mu_d^{(1)}, \mu_p^{(1)})$ is on the boundary of the deterministic constraint function $g_i(d, \mu_d, \mu_p)$. When considering the randomness of X, as seen on the graph (Fig. 5), the actual reliability (probability of constraint being feasible) is only around 0.5. After the deterministic optimization, the reliability assessment is implemented at the deterministic optimum solution $\mu_d^{(1)} = (\mu_d^{(1)}, \mu_p^{(1)})$ to locate the (inverse) MPP that corresponds to the desired R level. As one can expect, the MPP $x_{MPP}$ of constraint $g_i(\mu_d, \mu_p)$ will fall outside (to the left of) the deterministic feasible region. From our discussion in Sec. 3.1, we know that to ensure the feasibility of a probabilistic constraint, the inverse MPP corresponding to the R percentile should fall within the deterministic feasible region. Therefore, when establishing the equivalent deterministic optimization model in Cycle 2, the constraints should be modified to shift the MPP at least onto the deterministic boundary to help insure the feasibility of the probabilistic constraint. If we use $s$ to denote the shifting vector, the new constraint in the deterministic optimization of the next cycle is formulated as

$$
g(\mu_d - s) \leq 0 \quad (11)
$$

From Fig. 5, to ensure the MPP onto the deterministic boundary, we derive the shifting vector as

$$
s = (s_1, s_2) = (\mu_d^{(1)} - x_{MPP}^{(1)}, \mu_p^{(1)} - x_{MPP}^{(1)}) \quad (12)
$$

Correspondingly, Eq. (11) indicates that the location of the design variables $(\mu_d)$ in the deterministic optimization model needs to move further to the boundary of the probabilistic constraint to

---

Transactions of the ASME
ensure feasibility under uncertainty. This shifted deterministic constraint boundary is shown in Fig. 5 by the dotted curve. If there are more than one probabilistic constraints, other constraint boundaries are also shifted towards the feasible region by the distance between the optimal point $\mu^{(1)} = (\mu_1^{(1)}, \mu_2^{(2)})$ and their own (inverse) MPPs accordingly. In the second cycle of probabilistic optimization, the new constraints form a narrower feasible region in comparison with the one in the first cycle as shown in the following optimization model:

$$
\begin{align*}
\text{Minimize:} & \quad f(d, \mu) \\
DV = & \{d, \mu\} \\
\text{Subject to:} & \quad g(d, u, s^{(2)}) \leq 0,
\end{align*}
$$

where

$$
s^{(2)} = \mu^{(1)} - s^{(1)}
$$

After the optimization in Cycle 2, the reliability assessment of Cycle 2 is conducted to find the updated inverse MPPs and to check the design feasibility. The reliability of those violated probabilistic constraints in Cycle 1 should improve remarkably using the proposed MPP shifting strategy. If some probabilistic constraints are still not satisfied, we repeat the procedure cycle by cycle until the objective converges and the reliability requirement is achieved when all the shifting distances become zero.

As for the general case where deterministic design variables $d$ and random design variables $X$ as well as the random parameters $P$ exist, deterministic design variables $d$ can be considered as special random variables with zero variances and the sifting distances corresponding to $d$ is zero. Since we have no means to control the random parameters $P$ in the design, we could not use the same shifting treatment. However, considering model (9), we see that to maintain the reliability requirement, the deterministic constraint function should satisfy $g(d, X_{MPP}, d_{MPP}) \leq 0$. Therefore, for random parameters $P$ we simply use the MPP $P_{MPP}$ obtained in the previous cycle, such that

$$
\begin{align*}
\text{Minimize:} & \quad f(d, \mu, s, p) \\
DV = & \{d, \mu, s\} \\
\text{Subject to:} & \quad g(d, u, s^{(k+1)}, P_{MPP}) \leq 0, \quad i = 1, 2, \ldots, m,
\end{align*}
$$

where

$$
s^{(k+1)} = \mu^{(k)} - s^{(k)}
$$

It is noted that since each probabilistic constraint has its own (inverse) MPP, each probabilistic constraint has its own shifting vector $s_i$.

To further improve the efficiency, we also take the following measures: 1) The starting point for (inverse) MPP search in reliability assessment of the current cycle is taken as the (inverse) MPP obtained in the last cycle. Since the (inverse) MPPs of probabilistic constraints in two consecutive cycles are very close, using the (inverse) MPP of last cycle gives a good initial guess of the (inverse) MPP in the next cycle, and hence reduces the computational effort for MPP search. 2) Similarly, the starting point of the optimization of one cycle is taken as the optimum point of the previous cycle. 3) After one cycle of optimization, if the design variables included in one probabilistic constraint do not change or have very small changes compared with those in the last cycle, the MPP in the current cycle will be the same as or very close to that in the last cycle. Therefore, it is unnecessary to search the MPP again for this probabilistic constraint in the reliability assessment that follows.

The stopping criteria of the SORA method are as follows: 1) The objective approaches stable: the difference of the objective function between two consecutive cycles is small enough. 2) All the reliability requirements are satisfied.

From the procedure of the SORA method we see that the reliability analysis loop (locating the inverse MPPs) is completely decoupled from the optimization loop and that in the optimization part, equivalent deterministic forms of constraints are used, taking the same form of the original constraint functions. As a result, it is easy to code and to integrate the reliability analysis with any optimization software. We also see that the design is progressively improved (the desired reliability is progressively achieved) in the proposed probabilistic design process. This helps a designer track the design process more efficiently. Since the SORA method requires much fewer optimization iterations and reliability assessments to converge, the overall efficiency is high.

4 Applications

Two engineering design problems are used to demonstrate the effectiveness of the SORA method. These two examples include the reliability-based design for vehicle crashworthiness of side impact and the integrated reliability and robust design for the speed reducer of a small aircraft engine.

4.1 Reliability-Based Design for Vehicle Crashworthiness of Side Impact. The computational analysis of crashworthiness for vehicle impact has become a powerful and efficient tool to reduce the cost and development time for a new product that meets corporate and government crash safety requirements. Since the effects of uncertainties associated with the structure sizes, material properties, and operation conditions in the vehicle impact are considerably of importance, reliability-based design optimization for vehicle crashworthiness has been gained increasing attention and has been conducted in automotive industries [23, 24]. Typically, in a reliability-based design, the design feasibility is formulated as the reliability constraints while the design objective is related to the nominal value of the objective function. SORA is applied to the reliability-based design for vehicle crashworthiness of side impact based on global response surface models generated by Ford Motor Company.

There are nine (9) random variables $X_1 - X_9$, representing sizes of the structure, material properties ($X_8$ and $X_9$), and two (2) random parameters $P_1$ (Barrier height) and $P_2$ (Barrier hitting position).

The reliability-based design model is given in Fig. 6.

In this design model, $g$ is the weight of the structure, $F$’s are abdomen load and pubic symphysis force, $VC$’s are viscous criteria, and $D$’s are rib deflections (upper, middle, and lower).

![Fig. 6 Reliability-based design model for vehicle crashworthiness of side impact](image-url)
Table 1 Result of SORA method for vehicle side impact for case 1

<table>
<thead>
<tr>
<th>Cycle 1</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nominal Value</td>
</tr>
<tr>
<td>$F_{\text{Abdomen-Load}}$</td>
<td>0.5727</td>
</tr>
<tr>
<td>$D_{\text{low}}$</td>
<td>32.0000</td>
</tr>
<tr>
<td>$D_{\text{middle}}$</td>
<td>27.6641</td>
</tr>
<tr>
<td>$D_{\text{up}}$</td>
<td>29.3721</td>
</tr>
<tr>
<td>$V_{C1}$</td>
<td>0.2299</td>
</tr>
<tr>
<td>$V_{C2}$</td>
<td>0.2029</td>
</tr>
<tr>
<td>$V_{C3}$</td>
<td>0.2925</td>
</tr>
<tr>
<td>$F_{\text{Public}}$</td>
<td>4.0100</td>
</tr>
<tr>
<td>$V_{bp}$</td>
<td>9.3423</td>
</tr>
<tr>
<td>$V_d$</td>
<td>15.6781</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle 2</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nominal Value</td>
</tr>
<tr>
<td>$F_{\text{Abdomen-Load}}$</td>
<td>0.4839</td>
</tr>
<tr>
<td>$D_{\text{low}}$</td>
<td>31.1742</td>
</tr>
<tr>
<td>$D_{\text{middle}}$</td>
<td>27.1367</td>
</tr>
<tr>
<td>$D_{\text{up}}$</td>
<td>29.5611</td>
</tr>
<tr>
<td>$V_{C1}$</td>
<td>0.2330</td>
</tr>
<tr>
<td>$V_{C2}$</td>
<td>0.2116</td>
</tr>
<tr>
<td>$V_{C3}$</td>
<td>0.2896</td>
</tr>
<tr>
<td>$F_{\text{Public}}$</td>
<td>3.9487</td>
</tr>
<tr>
<td>$V_{bp}$</td>
<td>9.2581</td>
</tr>
<tr>
<td>$V_d$</td>
<td>15.4671</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cycle 3</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nominal Value</td>
</tr>
<tr>
<td>$F_{\text{Abdomen-Load}}$</td>
<td>0.4838</td>
</tr>
<tr>
<td>$D_{\text{low}}$</td>
<td>31.1742</td>
</tr>
<tr>
<td>$D_{\text{middle}}$</td>
<td>27.1367</td>
</tr>
<tr>
<td>$D_{\text{up}}$</td>
<td>29.5611</td>
</tr>
<tr>
<td>$V_{C1}$</td>
<td>0.2330</td>
</tr>
<tr>
<td>$V_{C2}$</td>
<td>0.2116</td>
</tr>
<tr>
<td>$V_{C3}$</td>
<td>0.2896</td>
</tr>
<tr>
<td>$F_{\text{Public}}$</td>
<td>3.9485</td>
</tr>
<tr>
<td>$V_{bp}$</td>
<td>9.2581</td>
</tr>
<tr>
<td>$V_d$</td>
<td>15.4671</td>
</tr>
</tbody>
</table>

Note: The reliability $= 1.0$ means that the reliability approaches closely but may not exactly equals to 1.0.

To verify the proposed method, in addition to the SORA method, the existing DLM_Prob and the DLM_Per strategies are also used to solve the problem. We consider two cases. In Case 1, all the desired reliabilities are set to $R = 0.9$. This is the case used by Ford Motor Company. In Case 2, we use higher reliability, $R = 0.99865$ which is equivalent to the safety index $\beta = 3$. For all the three methods, the optimization algorithm is the sequential quadratic programming (SQP) and the reliability assessment is based on FORM with the inverse MPP search algorithm developed in [22].

Case 1—Desired Reliability $= 0.9$

The SORA method uses three cycles of sequential optimizations and reliability assessment to obtain the solution. The optimization history is given in Table 1. The method starts from a conventional deterministic optimization. The result under optimization in cycle 1 in Table 1 is the optimum solution for the deterministic optimization. It is noted that the objective (weight) reduces significantly from 29.172 kg (baseline design used as starting point) to 23.5054 kg. After the deterministic optimization, the reliability analysis is performed to locate the inverse MPP for each constraint and it is noted that the reliabilities are low for some constraints such as the deflection of low rib Prob[$D_{\text{low}} \leq 32(\text{mm})$] and public force Prob[$F_{\text{Public}} \leq 4.01(\text{KN})$]. Based on the result of the deterministic optimization and the information of the inverse MPPs, the constraints boundaries are shifted as formulated in Eq. (14) and the feasible region is rearranged (reduced towards feasible directions) for the optimization in Cycle two. After Cycle 2, all the reliability requirements are satisfied. Therefore, the result of Cycle 3 is identical to that of Cycle 2. Cycle 3 is a repeated cycle for convergence purpose. From the result, we see that the desired reliability is progressively achieved and design is quickly improved.

The deterministic optimization in Cycle 1 often generates an infeasible probabilistic solution even though with a better (in minimization, lower) objective function value than the final optimal probabilistic design. The reliability assessment in Cycle 1 often shows that the reliabilities of some constraints are lower than required. In this example, after the deterministic optimization in cycle 1, the objective function value is 23.5054 kg and the worst reliability among all constraints is 0.5, lower than the desired reliability (0.9). With the progress of SORA, the feasibility of constraints improves but the objective function in deterministic optimization deteriorates. In our example, after Cycle 2, the objective function value deteriorates to 24.4897 kg while the worst reliability is improved to 0.9749. After Cycle 3, the worst reliability increases to the required level with the final objective function value of 24.4913 kg.

The convergence history of the objective (weight) is depicted in Fig. 7 where cycles distinguish from each other clearly—in each cycle, one reliability assessment follows one optimization. It is noted that most of computations are for reliability analyses. The total number of function evaluations is 491 including 74 for opti-
mizations and 341 for reliability analyses. The average number of function evaluations for reliability analysis during each cycle is 114.

Our confirmation test shows that SORA has the same accuracy as the double-loop methods (the DLM_Prob and the DLM_Per). However, the DLM_Prob and the DLM_Per require much more function evaluations as shown in Table 2. The numbers of function evaluations required by the DLM_Per and the DLM_Prob are 3324 and 26984, respectively. It is noted that the SORA method is the most efficient and the DLM_per is more efficient than the DLM_Prob.

2) Case 2—Desired Reliability = 0.99865 (β = 3)

All the three methods (the SORA method, the DLM_Prob and the DLM_Per) generate the same results as follows:

μ_q = 28.4397 kg, R_1 = R_3 = R_4 = R_5 = R_6 = R_7 = 1.0, R_8 = R_10 = 0.99865. In this case, three constraints (Drib_low, Pubic_F and, vd) are active with the exact reliability of 0.99865. With the SORA method, three sequential cycles of optimization and reliability assessments are used. Since the desired reliability is higher than that in Case 1, the reliability analysis needs more computations. The number of function evaluations for reliability is 446 (see Table 3), and the average number for each cycle is 149 which is larger than the one in Case 1. The number of function evaluations for optimization is 84 and the total number of function evaluations is 530. The numbers of function evaluations required by the DLM_Per and the DLM_Prob are 3272 and 456195, respectively. Therefore, the SORA method is still the most efficient and the DLM_per is more efficient than the DLM_Prob.

4.2 Integrated Reliability and Robust Design for the Speed Reducer. The speed reducer problem presents the design of a simple gearbox of a small aircraft engine, which allows the engine to rotate at its most efficient speed. This has been used as a testing problem for nonlinear optimization method in the literature. The original design was modeled by Golinski [25,26] as a single-level optimization, and since then many others have used it to test a variety of methods, for example, as an artificial multidisciplinary optimization problem [27–31].

Since in the design of the speed reducer there are many random variables, such as the sizes of the components (gears, shafts, etc.), material properties, and the operation environment (rotation speed, engine power etc.), it is also a good example for optimization under uncertainty. We modify this problem as a probabilistic design problem by assigning randomness to appropriate variables and parameters.

The deterministic design model of the speed reducer is given in [27]. In the probabilistic design, there are two deterministic design variables: \( d_1 \)-teeth module, and \( d_2 \)-number of pinion teeth, and five random design variables: \( X_1 \)-face width, \( X_2 \)-shaft-length 1 (between bearings), \( X_3 \)-shaft-length 2 (between bearings), \( X_4 \)-shaft diameter 1, \( X_5 \)-shaft diameter 2. There are 15 random parameters \( P_{11} \sim P_{15} \), including the material properties, the rotation speed, and the engine power, and 11 constraints among which ten \( (g_1 \sim g_{10}) \) are probabilistic constraints which are related to the bending condition, the compressive stress limitation, the transverse deflection of shafts and the substitute stress conditions, as well as one deterministic constraint \( g_{11} \). The design objective is to minimize the weight of the speed reducer.

The integrated reliability and robust design model is provided in Fig. 8.

### Table 2. Number of function evaluations

<table>
<thead>
<tr>
<th>Method</th>
<th>NFE for Reliability Assessment</th>
<th>NFE for Optimization</th>
<th>Total NFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORA</td>
<td>341</td>
<td>74</td>
<td>415</td>
</tr>
<tr>
<td>DLM_Per</td>
<td>-</td>
<td>-</td>
<td>3324</td>
</tr>
<tr>
<td>DLM_Prob</td>
<td>-</td>
<td>-</td>
<td>26984</td>
</tr>
</tbody>
</table>

NFE: Number of Function Evaluations

### Table 3. Number of function evaluations

<table>
<thead>
<tr>
<th>Method</th>
<th>NFE for Reliability Assessment</th>
<th>NFE for Optimization</th>
<th>Total NFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORA</td>
<td>446</td>
<td>84</td>
<td>530</td>
</tr>
<tr>
<td>DLM_Per</td>
<td>-</td>
<td>-</td>
<td>3272</td>
</tr>
<tr>
<td>DLM_Prob</td>
<td>-</td>
<td>-</td>
<td>456195</td>
</tr>
</tbody>
</table>
5 Discussions and Conclusion

The purpose of developing the SORA method is to improve the efficiency of probabilistic design. Different from the existing double loop methods, the SORA method employs the strategy of sequential single loops for optimization and reliability assessment, which separates the reliability assessment from the optimization loop. The measures taken by SORA include the use of the percentile formulation for probabilistic constraints instead of the reliability formulation to avoid evaluating the actual reliabilities; the use of sequential cycles of optimization and reliability assessments to reduce the total number of reliability analyses; and the use of an efficient and robust inverse MPP search algorithm to perform the reliability assessments.

The combination of these measures formulates a serial of "equivalent" deterministic optimization problems such that the optimum solution can be identified progressively and quickly. The probabilistic constraints are formulated as the deterministic constraint functions (for R percentile evaluations), which are evaluated at their inverse MPPs. If the design objective is deterministic, such as those in reliability-based design, there is no need to perform any probabilistic analysis in the optimization process. Therefore, the SORA method is extremely efficient for reliability-based optimization. As demonstrated in Example 1, the SORA method has much higher efficiency than the double loop methods. When the objective is formulated probabilistically, for example, the design objective is related to both the mean and standard deviation of the objective function for a robust design, or the design objective is the expected utility in the utility optimization, the SORA method is still applicable. However, its efficiency depends on how to evaluate the probabilistic characteristics of the objective function. If computationally expensive methods, such as the sampling method, are employed, the efficiency will decrease. If deterministically equivalent methods are used to evaluate the probabilistic objective, the efficiency of the SORA method will still be acceptable. One example of this treatment is demonstrated by the integrated reliability and robust design for the speed reducer presented in Section 4, where we employed the Taylor expansion to evaluate the mean and the standard deviation of the objective function.

Even though the SORA method is shown to be very effective with all the problems tested and there were not any convergence difficulties, one should be aware that due to the novelty of the proposed strategy, there might be a convergence problem when the objective and/or constraint functions are highly nonlinear or discontinuous. In those cases, the activities of deterministic constraints may change drastically from cycle to cycle, and the strategy of shifting the boundary of active deterministic constraints may not work. One should also be aware that if the dimensionality of the problem is huge or there are systems that are coupled (e.g., multidisciplinary systems), all existing probabilistic design methods, including the SORA method, will be very computationally expensive. The computational demand of MPP based approach is approximately proportional to the number of random variables/parameters when numerical derivative approaches are employed. DOE (Design of Experiments) can be used to screen out unimportant random variables/parameters [16] to reduce the problem size. When multidisciplinary systems are involved, special reliability analysis formulations [32] can be used to alleviate the computational expense.

Furthermore, there is a potential to improve the efficiency of the SORA method. Some of the probabilistic constraints are never active during the whole design process and their reliabilities are always above the desired levels. Therefore, it is not necessary to evaluate the percentiles of those constraints in the reliability assessment in each cycle. By investigating the method to identify the never-active probabilistic constraints can avoid unnecessary reliability assessments and hence can improve the efficiency considerably.

Acknowledgment

The supports from the National Science Foundation grants DMI-9896300 and DMI-0099775 are gratefully acknowledged. The authors would also like to acknowledge Drs. Lei Gu and Ren-Jye Yang of Ford Motor Company for providing the design model of reliability-based design for vehicle crashworthiness of side impact.

References


