



Decoupled reliability-based design optimization with a double-step modified adaptive chaos control approach

Majid Ilchi Ghazaan¹ · Fatemeh Saadatmand¹

Received: 26 May 2022 / Revised: 26 August 2022 / Accepted: 31 August 2022 / Published online: 21 September 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Reliability analysis using first-order reliability methods (FORM) has been widely used in reliability-based design optimization (RBDO) due to their simplicity and efficiency. The performance of the RBDO is highly dependent on how it deals with the loops of deterministic optimization and reliability analysis as well as the process of reliability assessment. In this paper, sequential optimization and reliability analysis (SORA) is employed to reduce the computational cost of RBDO. Moreover, a double-step modified adaptive chaos control method (DS-MACC) based on an improved adaptive chaos control approach is developed to speed up the reliability analysis loop. In the method presented here, two sets of novel criteria are introduced within two steps to distinguish the condition of the iterative process, compute and modify the step size. The efficiency and robustness of the proposed method is shown with eight inverse reliability problems and five RBDO examples and is compared with some methods developed recently. The results illustrate that the proposed method is more efficient with a competitive convergence rate.

Keywords Reliability-based design optimization · Reliability analysis · Performance measure approach · Chaos control method · Sequential optimization and reliability assessment method

1 Introduction

In structural design, safety and cost are the two main contradictory design criteria that have to be met simultaneously by designers. While cost savings can reduce the safety level of structures, improving safety may result in increasing costs (Melchers and Beck 2018). Moreover, uncertainty is the intrinsic characteristic of all engineering systems in real world and can be associated with material properties and physical quantities such as dimensions, manufacturing tolerances, and external loads. Although traditional deterministic design optimization methods can reduce the design costs, they ignore the effects of uncertainty which can result in unreliable structures and even catastrophic failures. Accordingly, reliability-based design optimization (RBDO) methods have been introduced that use structural reliability theory to take the effect of uncertainty into account (Shayanfar et al. 2018). The main objective

of this theory is to find the failure probability p_f based on Limit State Functions (LSF) or performance functions, and can be evaluated as (Liu and Der Kiureghian 1991):

$$P_f = \int \dots \int_{G(\mathbf{X}) \leq 0} f_{\mathbf{X}}(x_1, x_2, \dots, x_n) dx_1 \dots dx_n = \int_{G(\mathbf{X}) \leq 0} f_{\mathbf{X}}(\mathbf{X}) d\mathbf{X}, \quad (1)$$

where $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$ is the vector of random variables, $f_{\mathbf{X}}(\mathbf{X})$ is the joint probability density function (JPDF) of random variables vector, $G(\mathbf{X})$ is the LSF, and $G(\mathbf{X}) \leq 0$ indicates failure region. The JPDF of random variables is usually rarely known and evaluating the above-mentioned multi-dimensional integral is also a very daunting task, especially in the case of dealing with highly complicated implicit LSFs. Therefore, the aforementioned computational challenges have led to the development of alternative ways to estimate P_f . These alternatives can be grouped into two main methods: (1) simulation methods such as Monte Carlo simulation (MCS), importance sampling (IS), and Latin hypercube (LHC). (2) approximation methods including first-order reliability methods (FORM) and second-order reliability methods (SORM). In comparison with FORM, SORM has more accuracy, but more computational burden since it requires to calculate second derivatives. So, FORM is

Responsible Editor: Lei Wang

✉ Majid Ilchi Ghazaan
ilchi@iust.ac.ir

¹ School of Civil Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran

widely used in RBDO for its effectiveness and less effort. There are two different first-order reliability assessment approaches including the reliability index approach (RIA) and performance measure approach (PMA). While in the former, the algorithm seeks for the most probable failure point (MPFP) on the LSF surface with the minimum distance from the origin, in the latter, the algorithm searches for the minimum performance target point (MPTP) on the target reliability index hypersphere.

In the RIA, the most commonly used algorithm is HL-RF which was first introduced by Hasofer and Lind (Hasofer and Lind 1974) and was further developed by Rackwitz and Flessler with the inclusion of distribution information of random variables (Rackwitz and Flessler 1978). Although this method has a fast convergence rate for linear and some moderately nonlinear LSFs, it can produce periodic oscillation or fail to converge for some simple and highly nonlinear LSFs, which can result in unstable solutions. Therefore, ample studies have been done to circumvent these problems and increase the accuracy and speed of the method. A modified HL-RF that uses a merit function to monitor the convergence of the method was proposed in (Liu and Der Kiureghian 1991). Different merit functions were also developed to determine the step length (Zhang 1994; Santos et al. 2012). Stability transformation method (STM) of chaos feedback control was employed in the chaos control (CC) method for convergence control of FORM (Yang 2010). Despite its higher robustness compared with the HL-RF, CC method is computationally expensive for highly nonlinear problems and converges very slowly due to its small control factor. To alleviate its computational cost, various improved CC methods such as enhanced HL-RF (EHL-RF) (Kang et al. 2011), adaptive chaos control (ACC) (Li et al. 2015), enhanced chaos control (ECC) (Hao et al. 2017), directional stability transformation method (DSTM) (Meng et al. 2017) were proposed in which the control parameter is adaptively chosen between 0 and 1 based on various criteria. Finite-step-length (FSL) iterative algorithm was introduced in (Gong and Yi 2011) that uses a new step length in the direction of the gradient vector. Roudak et al. proposed a generalization of HL-RF and FSL that uses two parameters to eliminate the numerical instability of HL-RF (Roudak et al. 2017). Different methods were introduced to calculate the optimum search direction in each iteration. Keshtegar and Miri adopted the conjugate gradient approach and Wolfe conditions to compute the search direction and step size, respectively, in line (Keshtegar and Miri 2014). Conjugate stability transformation (CSTM) (Keshtegar 2016a, b), chaotic conjugate control (CCC) (Keshtegar 2016a, b), conjugate finite-step-length (CFSL) (Keshtegar 2017a, b) were also developed by Keshtegar et al. to improve the robustness of FORM. Several studies have shown that RIA is less efficient than PMA due to its low numerical efficiency (Lee et al. 2002).

In the PMA, the main idea is the fact that minimizing a complicated function under simple constraints is much easier than

minimizing a simple function under complicated constraints (Aoues and Chateaneuf 2010). So, the target reliability level is defined in advance and MPTP is sought on this level. Advanced mean value (AMV) (Wu et al. 1990) uses the steepest decent direction to find MPTP and is the fastest approach for convex problems. However, it cannot perform well for concave and highly nonlinear LSFs, so unstable solutions such as oscillation, bifurcation, and slow convergence may occur. Thus, conjugate mean value (CMV) and hybrid mean value (HMV) were proposed (Youn et al. 2003). Despite the good convergence of the HMV method for convex and slightly nonlinear problems, it fails to converge for highly nonlinear concave LSFs. Enhanced hybrid mean value (EHMV) was proposed further by Youn to increase numerical efficiency (Youn et al. 2005). Meng et al. reduced the computational cost of the CC method by extending the iterative points to the β -hypersphere known as MCC method (Meng et al. 2015). The efficiency of the PMA was improved by the studies of Keshtegar et al. in relaxed PMA (Keshtegar et al. 2016c), modified advanced mean value (MMV) (Keshtegar 2017a, b), and Self-adaptive modified chaos control (SMCC) (Keshtegar et al. 2017c). Hybridized conjugate mean value (HCMV) (Zhu et al. 2021), Dynamical accelerated chaos control (DACC) (Keshtegar et al. 2018a), and Augmented step size adjustment (ASSA) (Hao et al. 2019) are other methods that improved stability and robustness of PMA.

In recent years, various methods have been proposed for solving RBDO problems (Shayanfar et al. 2018). A traditional strategy is a nested two-level approach called the double-loop approach in which, the outer loop is responsible for deterministic constrained optimization (DO), whereas the inner loop, which is responsible for reliability analysis, assesses the value of random variables by computing the failure probability based on predefined limit state functions. Accordingly, either PMA or RIA can be used in this loop. Despite the simplicity of the classic RBDO, the computational costs can be very high since a large number of function evaluations may be needed. Therefore, single-loop and decoupled methods have been proposed for efficiency improvement (Jeong and Park 2017). In single-loop approaches, the reliability analysis is removed by replacing probabilistic constraints with equivalent deterministic constraints. In decoupled methods, reliability analysis is separated from deterministic optimization. Sequential optimization and reliability assessment (SORA) is one of the robust decoupled methods proposed by (Du and Chen 2004) which will be discussed further. It is also worth mentioning that machine learning-based RBDO methods have been developed recently (Li and Wang 2020, 2022). The Kriging method has been employed in various studies to develop surrogate models (Wang and Wang 2013). Surrogate modeling is a special case of supervised machine learning that is trained using a data-driven approach and the idea behind this method is to replace the original computationally expensive model with a cheap easy-to-analysis approximation model to reduce both time and

computer runs and increase efficiency. Moreover, time-variant RBDO framework is another area which has been investigated in several studies like (Wang et al. 2021, 2022).

One of the recently developed methods that uses adaptive chaos control factor is proposed by (Roudak et al. 2018). This method that employs the RIA with two internal parameters and a criterion to alter the control parameter during the computation process, exhibits numerical stability and small iteration numbers to reach MPP. However, it suffers from a high number of function evaluations in each iteration to achieve the proper control parameter. Therefore, in this study, it has been tried to reduce the computational burden of this algorithm by introducing novel criteria through two stages in DS-MACC to distinguish the type of function and modify the step length. Furthermore, to improve the numerical efficiency, the idea of SORA is employed in this study to expedite the RBDO process. The proposed algorithm is compared with AMV, HMV, MCC, improved adaptive CC (IACC), and some other recent methods for mathematical and structural reliability analysis and RBDO problems.

The organization of this paper is as follows. RBDO formulation as well as RIA and PMA approaches are illustrated in Sect. 2. IACC method is explained in detail within Sect. 3. In the next two sections, the proposed method is introduced and formulated. Illustrative examples are used to compare aforementioned methods in Sect. 6. Eventually, the conclusion is drawn in last Section.

2 Formulation of the reliability-based design optimization (RBDO)

The mathematical model of the classic two-level RBDO is generally formulated as follows:

$$\begin{aligned} & \text{Find } \mathbf{d}, \boldsymbol{\mu}_X \\ & \text{Minimize } f(\mathbf{d}, \mathbf{X}) \\ & \text{s.t. } P_f(G_i(\mathbf{d}, \mathbf{X}) \leq 0) \leq P_{f,i}^t, \quad (2) \\ & \text{Where } P_{f,i}^t = \Phi(-\beta_i^t), i = 1, 2, \dots, m \\ & \mathbf{d}^L \leq \mathbf{d} \leq \mathbf{d}^U \end{aligned}$$

where \mathbf{d} represents design variables vector with the lower and upper bounds of \mathbf{d}^L and \mathbf{d}^U for each variable, respectively. \mathbf{X} is the random variables vector and $\boldsymbol{\mu}_X$ contains the mean value of these variables. $f(\cdot)$ is the objective function, $G_i(\cdot)$ is the i th performance function (constraint function) and m is the number of probabilistic constraints. $P_f(\cdot)$ and $P_{f,i}^t$ are the probability of failure and allowable failure probability, respectively. β_i^t is the target reliability index of the i th performance function, and $\Phi(\cdot)$ stands for the standard normal cumulative distribution function. As mentioned before, to reduce the computational cost of the nested two-level RBDO, sequential

optimization and reliability assessment (SORA) was proposed (Du and Chen 2004). In this method, deterministic optimization (DO) and reliability analysis are performed separately as follows. During the first cycle, constrained DO is carried out first to obtain the value of the design variables. Afterward reliability analysis is performed. At this stage, the feasibility of the constraint is checked, and the MPTP corresponding to each constraint is computed. This is the end of the first cycle. To start the next cycle, a shifting vector is needed to move the violated constraints to the feasible zone, and the DO in the next cycle is performed with shifted constraints. This shifting vector is computed based on the MPTP points obtained in the previous cycle as follows:

$$s_i^k = \boldsymbol{\mu}_X^{k-1} - \mathbf{X}_{MPTP_i}^{k-1}, \quad i = 1, 2, \dots, m \quad (3)$$

In which, k is the number of the current cycle, and $\boldsymbol{\mu}_X^{k-1}$ is the vector of mean values of random design variables drawn during constrained DO of previous cycle $k - 1$. $\mathbf{X}_{MPTP_i}^{k-1}$ is the MPTP computed for the i th constraint in reliability analysis of previous cycle $k - 1$, and m is the number of constraints. This should be noted that this vector for unviolated constraints is 0. The formulation of SORA is mathematically expressed as:

$$\begin{aligned} & \text{Find } \mathbf{d}, \boldsymbol{\mu}_X \\ & \text{Minimize } f(\mathbf{d}, \boldsymbol{\mu}_X) \\ & \text{s.t. } G_i(\mathbf{d}, \boldsymbol{\mu}_X - s_i^k) \geq 0, \quad i = 1, 2, \dots, m \end{aligned} \quad (4)$$

Thus, as the aforementioned procedure goes on, the reliability of the optimum design is augmented and the probability of the failure is reduced. Eventually, when the probabilistic constraints are feasible and the convergence condition is met, the algorithm stops. The schematic view of the SORA process in the first and last cycles are illustrated in Fig. 1.

The probabilistic constraints of the RBDO can be assessed by two alternative approaches of RIA and PMA which have been introduced briefly before. In RIA, the reliability index is obtained by solving the following constrained optimization problem:

$$\begin{aligned} & \min_u ||\mathbf{U}|| \\ & \text{s.t. } G_i(\mathbf{U}) \leq 0 \end{aligned} \quad (5)$$

where \mathbf{U} is the vector of random variables transformed from X -space to standard normal space (U -space). In Fig. 2a, the schematic view of this approach for a two-variable LSF can be easily observed. As it is obvious, in RIA, the algorithm tries to find the minimum distance from the origin to the limit state $G(\mathbf{U}) = 0$. Therefore, the radius of the β -circle varies as the iterations go on until it touches the limit state surface. The distance is called reliability index. Whereas, in PMA, as can be seen in Fig. 2b, the radius of β -circle is

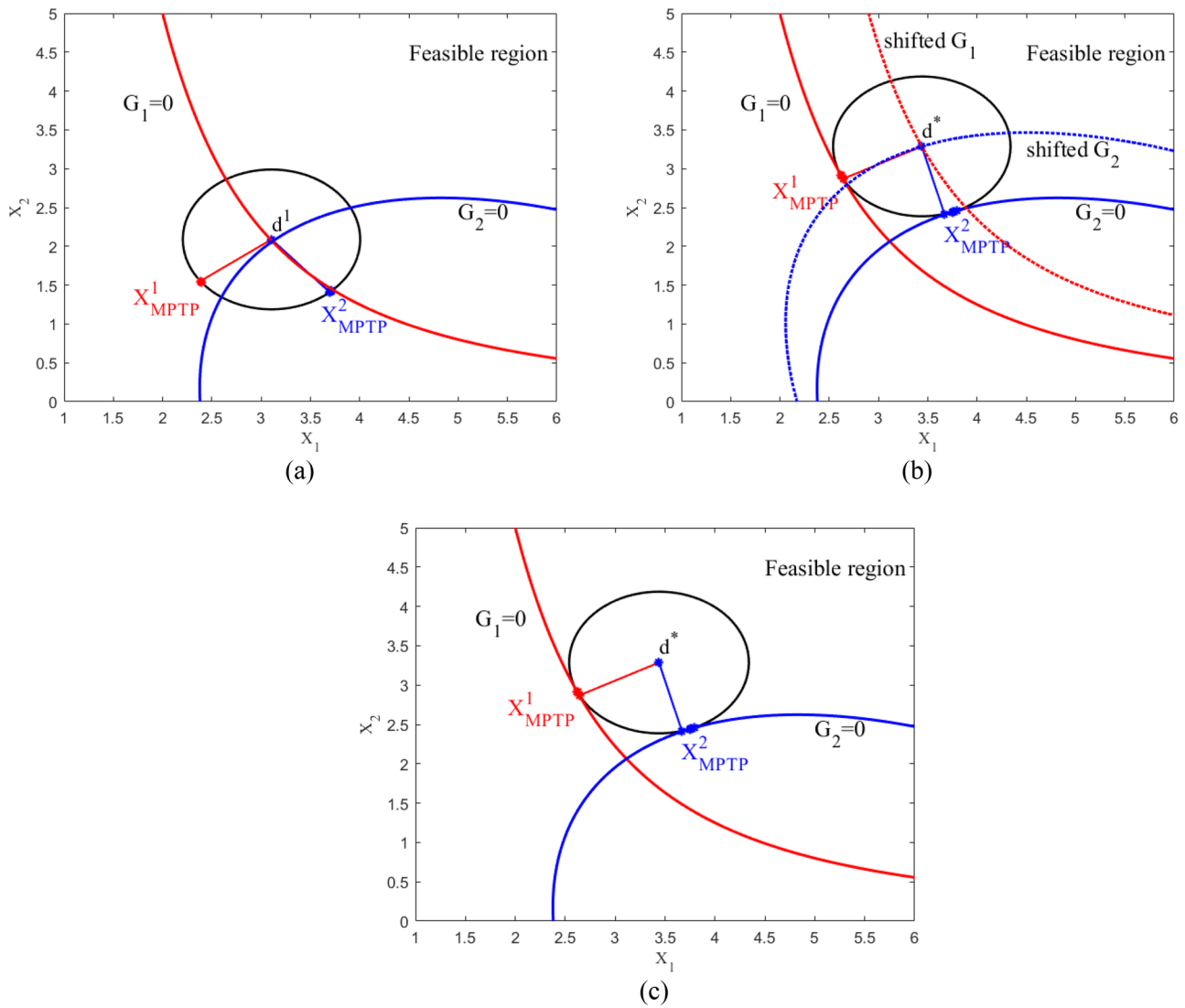


Fig. 1 Schematic view of SORA procedure **a** DO in the first cycle, **b** Shifting constraints before the last cycle, **c** Finding the reliable optimum design in the last cycle

constant and equal to the target reliability index β^T , and the algorithm searches for MPTP over this circle. PMA can be mathematically expressed as:

$$\begin{aligned} \min_u G_i(\mathbf{U}) \\ \text{s.t. } \|\mathbf{U}\| = \beta_i^T \end{aligned} \quad (6)$$

3 Improved FORM method using adaptive chaos control (IACC)

The process of chaos control method is formulated as follows (Yang and Yi 2009):

$$(\mathbf{U}_{k+1}^{CC}) = \mathbf{U}_k^{CC} + \lambda \mathbf{C}(f(\mathbf{U}_k) - \mathbf{U}_k^{CC}) \quad 0 < \lambda < 1, \quad (7)$$

where \mathbf{C} is the $n \times n$ dimensional involutory matrix with merely one element in each row. The value of elements in each column is 1 or -1 and the other ones are 0 in this matrix. So, there are $2^n n!$ Involutory matrixes can be selected. However, the unit matrix \mathbf{I} is usually picked as matrix \mathbf{C} . λ is the control factor and is determined based on the eigenvalues of the original system's Jacobian matrix. $f(\mathbf{U}_k)$ is the next iterative point calculated using steepest descent direction. While a small value for control factor λ can lead to a more robust iterative process, especially for highly nonlinear problems, the computational cost can become very exorbitant. Conversely, a large value of λ can be very efficient, however, the possibility of oscillation or divergence may rise. In the CC method usually, a fixed and relatively small value of λ is selected for the iterative

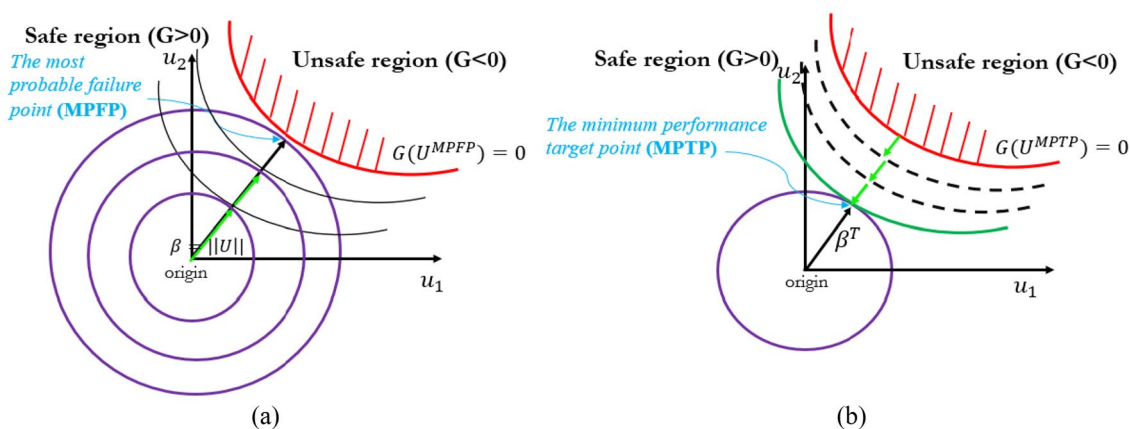


Fig. 2 Illustration of **a** RIA and **b** PMA approaches

process, which can result in a large number of iterations to reach MPFP. Therefore, efficiency is obviously sacrificed for accuracy especially when a larger value for λ may work. Roudak et al. proposed a method that is RIA-based and employs a varying control parameter λ to improve the efficiency of the CC method (Roudak et al. 2018). In the IACC method, λ is proportional to the rate of convergence and is reduced based on the nonlinearity degree in a specific part of an LSF by considering the change in the direction of the two consecutive gradient vectors. At the onset of the search process of this method, λ is equal to 1 and its reduction is controlled with three parameters c_1, c_2 , and b , all of which are between 0 to 1. In the following, the function of these parameters is explained briefly.

Consider this algorithm is performing the k th iteration and wants to find the next iterative point (U_{k+1}). To do so, the last value of λ is first chosen to compute U_{k+1} . Then, $\nabla G(U_{k+1})$ is calculated at that point. Here, a criterion is utilized for checking the direction of the two gradient vectors in current and prospective iterative points; thus, the cosine of the angle between the aforementioned vectors is computed by the following relation:

$$\cos \theta_k = \frac{\nabla G(U_k) \nabla G(U_{k+1})}{|\nabla G(U_k)| |\nabla G(U_{k+1})|} \tag{8}$$

If $|\cos \theta_k|$ is larger than b , U_{k+1} is accepted and λ is decreased very slightly as $\lambda = c_2 \lambda$. If not, $\lambda = c_1 \lambda$ and U_{k+1} will be re-calculated using the new reduced value of λ and this continues until the relation expressed in Eq. 8 is larger than b . In the aforementioned study, values of c_1, c_2 , and b are suggested to be 0.9, 0.99, and 0.95, respectively. c_1 is the local sensitivity controller; the closer the value of c_1 is to 1, the more conservative the iterative process will be, since λ is reduced very slowly. When the nonlinearity degree of the LSF is high, this is favored, however, this can lead to a high number of function evaluations. When

the angle between two consecutive gradient vectors is sufficiently small, to prevent oscillation, the value of λ is not let to be constant, so c_2 which is very close to unity is used to reduce the control parameter slightly. In addition, the value of b determines the severity of Eq. 8. A larger value of b indicates a more limited accepted range for the angle between two successive gradient vectors, consequently, the convergence rate may reduce. Whereas, by a smaller value for b , the process is faster, but the risk of divergence increases.

The above-mentioned method has been successful in reducing the number of iterations and is robust during the iterative process, which will be illustrated in Sect. 6. However, the high number of function evaluations is the main deficiency of this algorithm. Considering a constant fixed value such as c_1 or c_2 for control parameter reduction, especially in first iterations, the computational cost is still high which can reduce the efficiency of the algorithm. Moreover, RIA-based algorithms typically require much more function evaluations. What is more, considering the absolute value of the $\cos \theta_k$ is not efficient, because in case of highly nonlinear LSFs, this can lead to slow convergence of the algorithm i.e., if $\cos \theta_k$ is -0.97 , its absolute value is 0.97 which is greater than $b = 0.95$. This means the algorithm should modify the step size with c_2 parameter which is too slow. In Sect. 6, example 7 is a good illustration of this problem which will be discussed further. For this reason, in the next section, these problems are tackled to improve the efficiency of this algorithm.

4 Proposed method

In reliability methods using chaos control, all use a control parameter to prevent chaotic solutions. The important thing is what value should be chosen for this control parameter to reach a balance between the number of

function calls and convergence. For instance, a very small value of control parameter λ may be able to converge to MPP but needs thousands of iterations to do so and vice versa. Thus, the value of λ is very effective. Some methods like the MCC method use small and fixed values for this parameter which cannot guarantee convergence and may lead to high computational cost. Some methods use varying values of λ during the process which is more reasonable. In our study, we tried to use the available information at each iteration to find an appropriate value of λ and modify it at each iteration to reach high efficiency and fast convergence. In this section, double-step modified adaptive chaos control (DS-MACC) method is proposed to alleviate the high computational cost of the IACC method. Besides, the proposed method is coupled with SORA to enhance the efficiency and convergence speed of the RBDO. In the first step, two new criteria are introduced to find the intervals in which MPTP may exist. Information required to check the conditions are demonstrated in the following relations:

$$\cos \theta_1 = \frac{U_{k-1}U_k}{|U_{k-1}||U_k|}, \cos \theta_2 = \frac{U_{k+1}^{AMV}U_k}{|U_{k+1}^{AMV}||U_k|}, \tag{9}$$

where θ_1 is the angle between the current (U_k) and previous iterative points (U_{k-1}) which can be seen in Fig. 3. The next relation is the same as Eq. 8 as follows:

$$\cos \theta_k = \frac{\nabla G(U_k)\nabla G(U_{k+1})}{|\nabla G(U_k)||\nabla G(U_{k+1})|} \tag{10}$$

To predict the probable location of MPTP, there are 3 different conditions:

1. Steepest descent direction is the direction through which, the value of the performance function decreases at its fastest rate and it also points to the minimum of the performance function. Thus, if the cosine of the angle between the current and previous iterative points is positive ($\cos \theta_k \geq 0$), it means that MPTP is located out of the

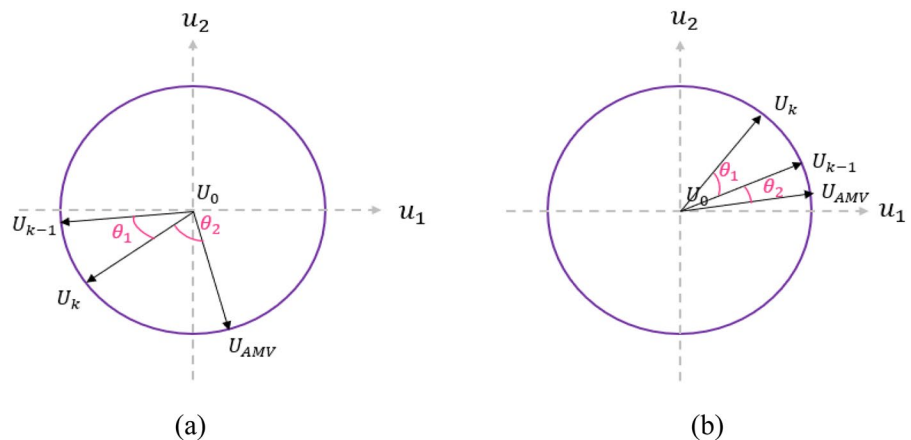
interval of U_{k-1} and U_k . In addition, U_{k+1}^{AMV} is utilized to find out if the steepest descent direction can be used as U_{k+1} or not. In other words, if $|\cos \theta_1| \leq |\cos \theta_2|$ or $(\theta_2 < \theta_1)$ defined in Eq. 9, steepest descent direction is the fastest and most efficient search direction; therefore, AMV method is adopted to calculate the next iterative point using Eq. 11 and the algorithm goes for the next iteration; otherwise, another approach is employed which will be explained further in this section.

$$U_{k+1} = U_{k+1}^{AMV} = -\beta_T \frac{\nabla G(U_k)}{\|\nabla G(U_k)\|} \tag{11}$$

To better understand this, consider the following example: $G_1(X) = -\exp(x_1 - 7) - x_2 + 10$ is a convex function with two normally distributed random variables $x_{1,2} \sim N(6, 0.8)$, and the target reliability index is set as 3. This performance function is depicted in Fig. 4a. The first two iterative points that are calculated utilizing the steepest descent direction are shown with numbers 1 and 2. At each of them, negative normalized gradient vectors are illustrated with black arrows, and as can be observed, they point out the interval beyond U_1 and U_2 . So, the next probable iterative point is obtained using the AMV method (Eq. 11). At this stage, the second criterion is checked. It can be easily seen that $|\cos \theta_1| \leq |\cos \theta_2|$, as a result, AMV method is chosen to calculate U_{k+1} .

2. If $\cos \theta_k \leq 0$, MPTP is located between U_k and U_{k-1} and there is no need to check the other condition. Consider $G_6(X) = 0.2Ln[\exp[5(1 + x_1 - x_2)] + \exp[5(5 - 5x_1 - x_2)]]$ as an example, which is a concave function having two random variables with normal distributions: $x_{1,2} \sim N(0, 1)$, and β_t is 3. This $G_6(x)$ is shown in Fig. 4b. Regarding the direction of the negative normalized gradient vectors, it is obvious that MPTP is between U_1 and U_2 . When this occurs, the step length is calculated using U_{k-1} and U_k . For doing so, the augmented step size adjustment (ASSA) method (Hao et al. 2019) is employed which offers one the most efficient step length adjustment processes. ASSA method

Fig. 3 Illustration of θ_1 and θ_2 for different types of LSFs



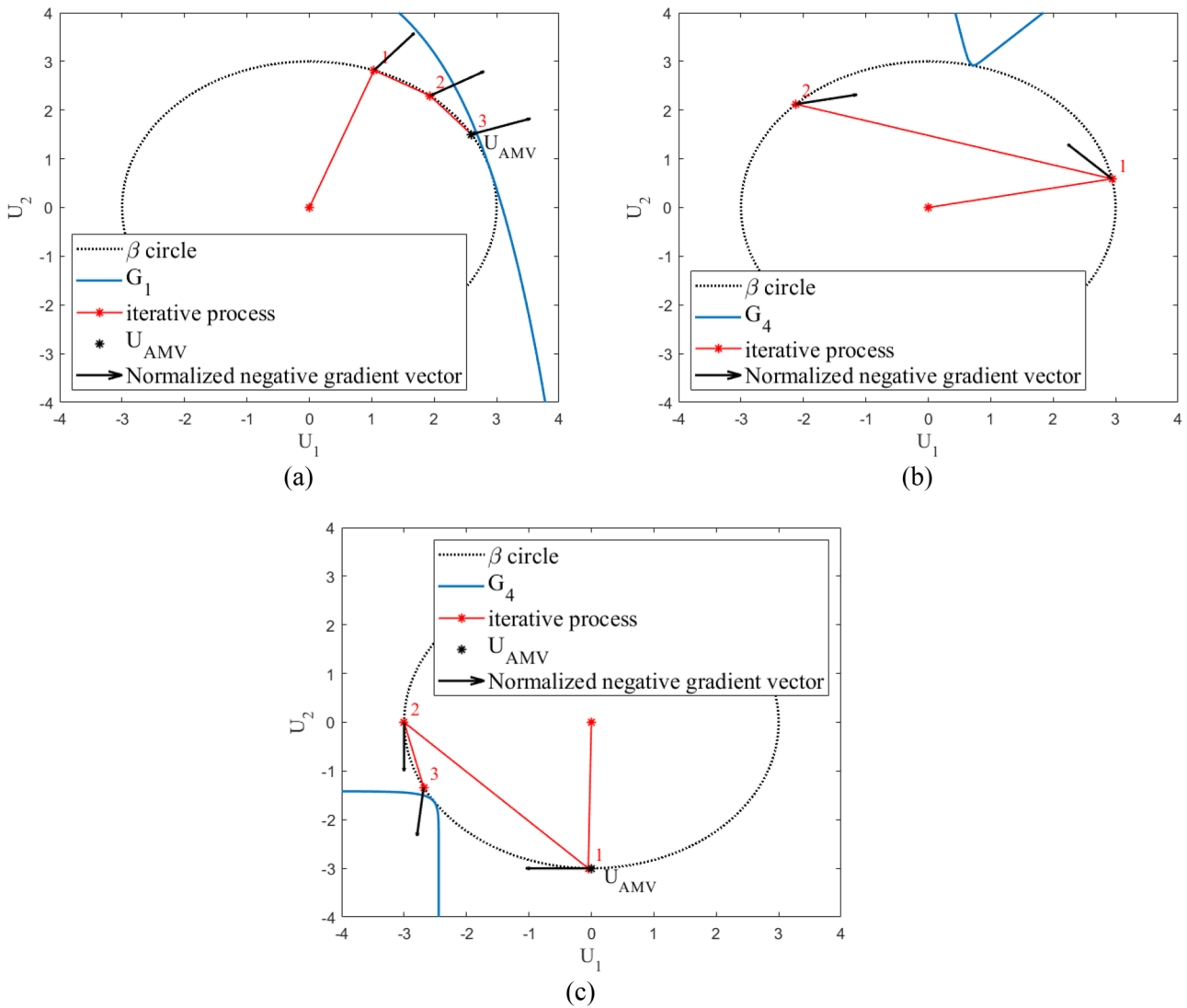


Fig. 4 Three different states of iterative histories a 1st state, b 2nd state, c 3rd state

uses a vector that is depicted in Fig. 5 to calculate the step length λ . This vector which is expressed as \vec{ab}^k is calculated by the following relation (Hao et al. 2019):

$$\vec{ab}^k = -\frac{\nabla G(\mathbf{U}_k)}{\|\nabla G(\mathbf{U}_k)\|} - \frac{\mathbf{U}_k}{\|\mathbf{U}_k\|} \tag{12}$$

As shown in Fig. 5, \vec{ab}^k links the endpoint of the normalized direction vector a to the endpoint of the normalized negative gradient vector b in the k th iteration. The step length λ^k is then computed as follows:

$$\lambda^k = \frac{\|\vec{ab}^k\|}{\|\vec{ab}^{k-1}\| + \|\vec{ab}^k\|} \|\mathbf{U}_k - \mathbf{U}_{k-1}\| \tag{13}$$

The next iterative point can be obtained by the following relations:

$$\mathbf{U}_{k+1} = \beta^t \frac{n(\mathbf{U}_{k+1})}{\|n(\mathbf{U}_{k+1})\|}, n(\mathbf{U}_{k+1}) = \mathbf{U}_k + \lambda^k \frac{\mathbf{U}_{k+1}^{AMV} - \mathbf{U}_k}{\|\mathbf{U}_{k+1}^{AMV} - \mathbf{U}_k\|} \tag{14}$$

3. As mentioned before, if $\cos\theta_k \geq 0$, MPTP is located out of the interval \mathbf{U}_{k-1} to \mathbf{U}_k . Additionally, if $|\cos\theta_1| > |\cos\theta_2|$, the steepest descent direction may lead to divergence. This can be readily observed via the next example, a highly nonlinear performance function $G_4(X) = \exp(1.5x_1^{1.5} - 5) + \exp(1.2x_2^2 - 15) - 15$ (Keshtegar et al. 2018c) which is illustrated in Fig. 4c. The random variables are $x_{1,2} \sim N(5, 0.8)$ with $\beta_t = 3$. In the corresponding figure, the iterative point U_3 depicts the current iteration

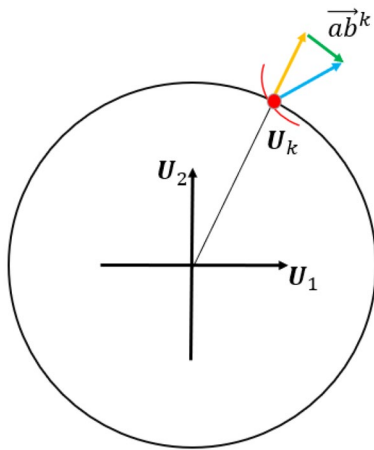


Fig. 5 Illustration of \vec{ab} vector

and the algorithm wants to find the interval in which, MPTP is located. The negative normalized gradient vectors point out of U_2 and U_3 . Furthermore, the location of the iterative point obtained by the AMV method cannot satisfy the second condition and this means the AMV method here, can lead to divergence or bifurcation, so it cannot be employed. Moreover, the direction of the steepest descent directions in points U_1 and U_3 , illustrates the MPTP is somewhere between U_1 to U_3 . Generally speaking, λ^k in this state, is computed as:

$$\lambda^k = \frac{\|\vec{ab}^k\|}{\|\vec{ab}^{k-2}\| + \|\vec{ab}^k\|} \|U_k - U_{k-2}\| \quad (15)$$

when either of the second or third states happens, this is the end of the first step in which a trial U_{k+1} is obtained and needed to be modified in the second step. Based on the criterion introduced in Eq. 16 the cosine of the angle between $\nabla G(U_k)$ and $\nabla G(U_{k+1})$ should be greater than b which is considered to be 0.7 in this study. If this condition is met, U_{k+1} is accepted and the next iteration will be performed; otherwise, λ^k is reduced exponentially by Eq. 17 until the aforementioned criterion is satisfied. In Eq. 17, t is the number of times that the second step is performed until the k th iteration. So, t changes only when the second step is adopted. This prevents the step length reduction rate from being either too slow or too fast. If the LSF is highly nonlinear, during the step size modification in initial iterations, the t parameter causes the step size to reduce slightly, thereby searching the sensitive part of the performance function more efficiently. During the last iterations, the step length is usually very small and thus, less sensitive to the value of t .

$$\cos\theta_k = \frac{\nabla G(U_k)\nabla G(U_{k+1})}{|\nabla G(U_k)||\nabla G(U_{k+1})|} > b = 0.7 \quad (16)$$

$$\lambda^k = \exp\left(-\frac{t}{10}\right) \|U_{k+1} - U_k\| \quad (17)$$

5 The iterative procedure of DS-MACC

Step 1: Define a performance function i.e., $G(d, X)$, target reliability index β_t , μ and σ of random variables. Set $k = 0$, stopping criterion ϵ , and the control parameter b . (here, b is set as 0.7).

Step 2: Transform random variables from X-space to U-space.

Step 3: If $k \leq 3$, then compute the new point based on the AMV method formula in Eq. 11.

Else; start the first step:

- If $|\cos\theta_1| \leq |\cos\theta_2|$ (defined in Eq. 9) and $\cos\theta_k \geq 0$ (defined in Eq. 10), then compute the new point based on AMV method formula in Eq. 11. Afterward, go to **Step 5**
- If $\cos\theta_k \leq 0$, then Compute the trial point U_{k+1} using Eqs. 12, 13, and 14. Afterward, go to the next step.
- If $|\cos\theta_1| > |\cos\theta_2|$, and $\cos\theta_k \geq 0$, then Compute the trial point U_{k+1} using Eqs. 14 and 15. Afterward, go to the next step.

Step 4: checking the criterion introduced in Eq. 16 in the second step:

- While $\cos\theta_k < b$, modify λ^k using Eq. 17 and recalculate the U_{k+1} until this criterion is satisfied.

Step 5: Check the convergence criterion: If $\|U_{k+1} - U_k\| / \|U_k\| < \epsilon$, then stop and finish the process. Otherwise, set $k = k + 1$ and go to **Step 3**.

The flowchart of the proposed method is shown in Fig 6.

6 Illustrative examples and numerical results

In this section, several numerical and structural examples are solved for verifying the proposed method. The numerical results are compared with other methods including

AMV, H MV, MCC, IACC, and some other recent methods in terms of the number of iterations and function evaluations for investigating their accuracy and efficiency. For doing so, two sets of examples are considered. The first set deals with seven mathematical and one structural performance functions for reliability analysis. The implementation of the above-mentioned methods in RBDO are illustrated via the second set including five mathematical and structural problems. All methods are coded with MATLAB. For the MCC method, the control parameter λ is set both 0.1 and 0.5. It has to be mentioned that the value of λ will also be different for these methods in RBDO

examples. Moreover, $C = I$ for all methods employing the concepts of the CC method. To have a fair comparison, the PMA approach of IACC is also implemented. The convergence criterion for the first set is considered as $\|U_{k+1} - U_k\| / \|U_k\| < 10^{-5}$, and for the second set is $\|U_{k+1} - U_k\| / \|U_k\| < 10^{-6}$.

6.1 Set I: mathematical and structural examples for reliability analysis

Seven mathematical nonlinear limit state functions are used for comparison and listed in Table 1. The value of the

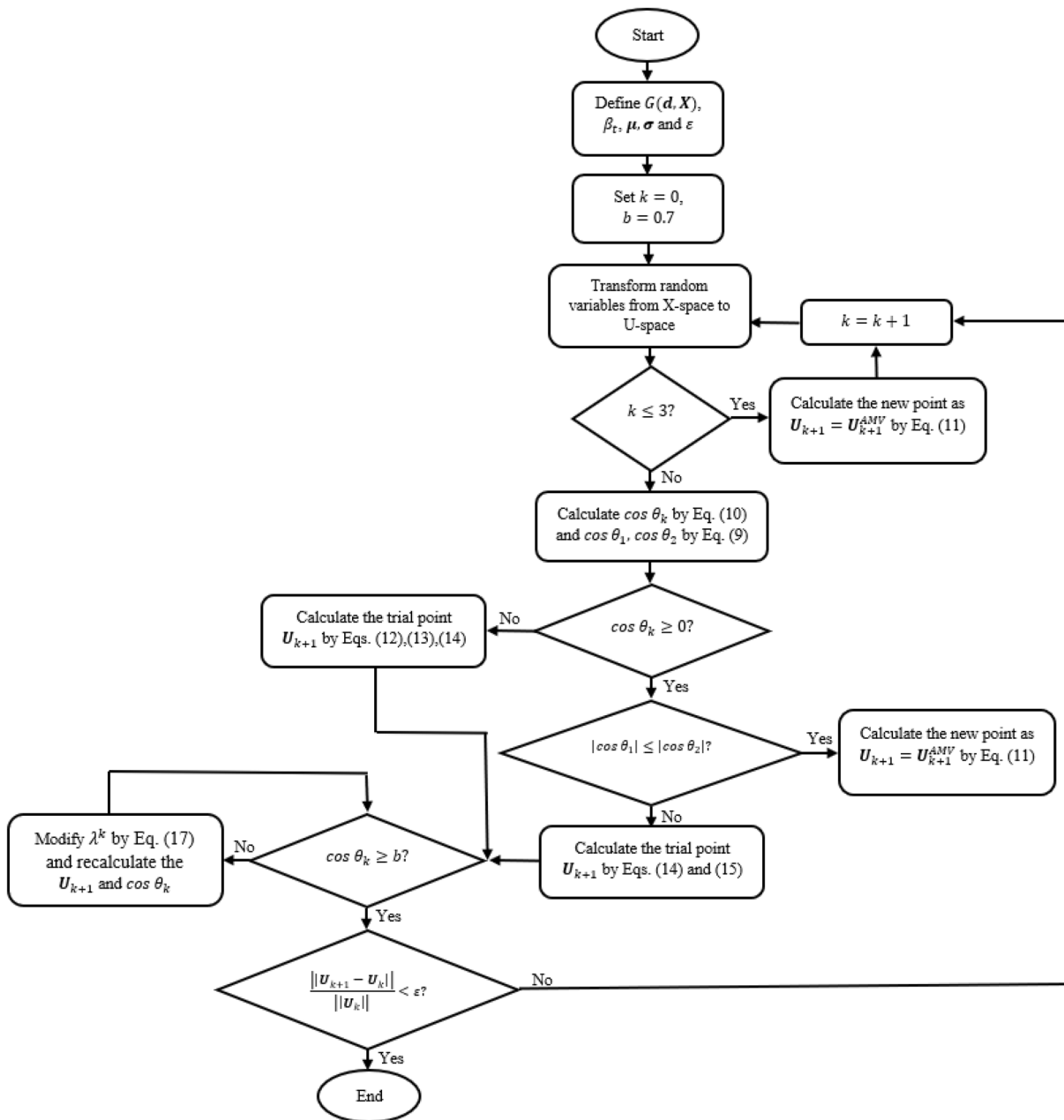


Fig. 6 The flowchart of the DS-MACC method

performance functions ($G(X^*)$) and the number of function evaluations, as well as iterations, are provided in Table 2. From this table, it is readily apparent that IACC and DS-MACC can converge to MPTP for all LSFs. However, the number of function evaluations of the proposed method is much less than that of IACC. MCC with $\lambda = 0.1$ has shown to be a robust method in comparison with the other four methods. AMV is the least efficient method among all since it can only reach to MPTP for merely one LSF. MCC with $\lambda = 0.5$ and HMV methods usually have greater computational effort compared with other iterative processes.

The first performance function, as mentioned before, is a convex function which is weakly nonlinear. AMV, HMV, and the proposed method converge to X^* with the same number of function evaluations because based on the criteria utilized in these methods, they all use the steepest descent direction for the process and act like the AMV method. Although the IACC

shows fewer function evaluations compared with the MCC method, its computational cost is still higher since it modifies λ unnecessarily. In the MCC method, the small fixed step size has increased the number of iterations especially when λ is 0.1. The convergence history of example 2 is plotted in Fig. 7. This is a moderately nonlinear performance function, and all the methods except for AMV can find the MPTP. As can be seen from Table 2, DS-MACC method is the most efficient approach for MPTP search and is five times faster than the IACC method. In this example, a small control parameter for the MCC method is more desirable because of the high non-linearity of LSF near MPTP. For this reason, the MCC method with $\lambda = 0.5$ and HMV are very slow to converge.

The third performance function has a large reliability index that augments the challenge of seeking MPTP. From Fig. 8, it is apparent that oscillation occurs in AMV iterative procedure. DS-MACC method provides the best numerical

Table 1 Limit state functions with their corresponding distribution information

#	Limit state function	Distribution	β_t
1	$G_1(X) = -\exp(x_1 - 7) - x_2 + 10$ (Hao et al. 2019)	$x_1 \sim N(6, 0.8)$ $x_2 \sim N(6, 0.8)$	3
2	$G_2(X) = x_1^3 + x_2^3 - 18$ (Hao et al. 2019)	$x_1 \sim N(10, 5)$ $x_2 \sim N(9.9, 5)$	3
3	$G_3(X) = 0.3x_1^2x_2 - x_2 + 0.8x_1 + 1$ (Hao et al. 2019)	$x_1 \sim N(1.2, 0.42)$ $x_2 \sim N(1, 0.42)$	6
4	$G_4(X) = \exp(1.5x_1^{1.5} - 5) + \exp(1.2x_2^2 - 15) - 15$ (Keshtegar et al. 2018c)	$x_1 \sim N(5, 0.8)$ $x_2 \sim N(5, 0.8)$	3
5	$G_5(X) = -0.75 + 0.489x_1x_4 + 0.843x_2x_3 - 0.0432x_5x_6 + 0.0556x_5x_7 + 0.000786x_7^2$ (Hao et al. 2019)	$i = 1 - 4x_i \sim N(1, 0.05)$ $i = 5x_i \sim N(0.3, 0.006)$ $i = 6, 7x_i \sim N(0, 10)$	3
6	$G_6(X) = 0.2Ln[\exp[5(1 + x_1 - x_2)] + \exp[5(5 - 5x_1 - x_2)]]$ (Hao et al. 2019)	$x_1 \sim N(0, 1)$ $x_2 \sim N(0, 1)$	3
7	$G_7(X) = 0.3x_1^2x_2 - x_2 + 0.8x_1 + 1$ (Keshtegar et al. 2018b)	$x_1 \sim N(0, 0.55)$ $x_2 \sim N(6, 0.55)$	3

Table 2 Numerical results of MPTP search for different methods (iterations/function evaluations)

	AMV	HMV	MCC ($\lambda = 0.1$)	MCC ($\lambda = 0.5$)	IACC	DS-MACC
$G_1(X^*)$	-0.3579 (9/9)	-0.3579 (9/9)	-0.3579 (97/97)	-0.3579 (21/21)	-0.3579 (13/16)	-0.3579 (9/9)
$G_2(X^*)$	-	-31.0665 (165/165)	-31.0665 (26/26)	-31.0665 (89/89)	-31.0665 (42/68)	-31.0665 (13/13)
$G_3(X^*)$	-	-2.2293 (594/594)	-2.2293 (25/25)	-2.2293 (160/160)	-2.2293 (20/41)	-2.2293 (14/15)
$G_4(X^*)$	-	-	4.44695 (17/17)	-	4.44695 (8/30)	4.44695 (12/19)
$G_5(X^*)$	-	0.075283 (551/551)	0.075283 (42/42)	0.075283 (169/169)	0.075283 (36/56)	0.075283 (14/15)
$G_6(X^*)$	-	-	-	-	-1.1566 (11/56)	-1.1566 (10/21)
$G_7(X^*)$	-	-	-6.71162 (11/11)	-	-6.71162 (68/89)	-6.71162 (12/15)

results with 15 function evaluations to find MPTP. This is mainly because it uses the steepest descent direction when it is needed based on the conditions introduced in this study, which expedites the process. In addition, the step length expressed in Eq. 17 can sufficiently narrow the search interval for the second step of the algorithm and this prevents it from unnecessary function calls in the IACC method. IACC has fewer iterations than MCC with $\lambda = 0.1$, whereas, its number of function evaluations is relatively twofold. MCC ($\lambda = 0.5$) and HMV also require many iterations to satisfy the convergence criterion and thus, are less efficient.

For example 4, iterations are plotted in Fig. 9. It is highly nonlinear with a great curvature around MPTP; a difficult problem for AMV, HMV, MCC methods. It can be viewed that the IACC method can converge to MPTP after 8 iterations, which is the least among all. However, its computational burden is larger than others with 30 function evaluations. MCC ($\lambda = 0.1$) and the proposed method exhibit acceptable efficiency. Example 5 is a multi-dimensional highly nonlinear LSF with seven random variables. DS-MACC method can accurately reach MPTP with only 15 function calls which is approximately less than one-third of that of the next best result obtained by MCC with 42 function calls. IACC has been able to reduce the number of iterations of the MCC method, but its computational cost is higher. MCC ($\lambda = 0.5$) and HMV found the MPTP after 169 and 551 iterations, respectively, showing their ineffectiveness. AMV encounters oscillation and fail to converge.

Example 6 and 7 are two strictly nonlinear performance functions and their iterative procedures are depicted in

Fig. 10 and Fig. 11, respectively. For example 6, only the IACC and DS-MACC method can find MPTP and are the most robust and efficient methods compared to others. Even the small control parameter of MCC ($\lambda = 0.1$) could not help it to reach the minimum target point. The proposed method has the least number of function calls which is around half of the IACC method's function calls. This illustrates that DS-MACC method has been successful to improve the numerical instability and efficiency of the IACC method. This happens thanks to the criterion introduced to narrow the search interval iteration by iteration letting the algorithm reduce the step length fewer times than the IACC method. In terms of example 7, both the MCC ($\lambda = 0.1$) and proposed method demonstrate good efficiency in MPTP search. Conversely, IACC though converge to MPTP, as can be seen in Fig. 11, it shows instabilities and λ reduction is done insufficiently resulting in the high number of function evaluations. This is mainly because it considers the absolute value of the $\cos\theta_k$ which is not correct and can lead to this kind of iteration history. This problem is successfully tackled in the proposed method. Thus, it can be concluded from the examples above that DS-MACC method is the most efficient and accurate one to find MPTP. MCC ($\lambda=0.1$) is also an efficient algorithm for highly nonlinear LSFs due to its small control parameter. However, it may fail to converge for LSFs with severe nonlinearity like example 7. Also, it shows less efficiency for moderately nonlinear or convex performance functions. The IACC is also a robust method that can solve problems with different levels of curvature, but its computational cost can be high.

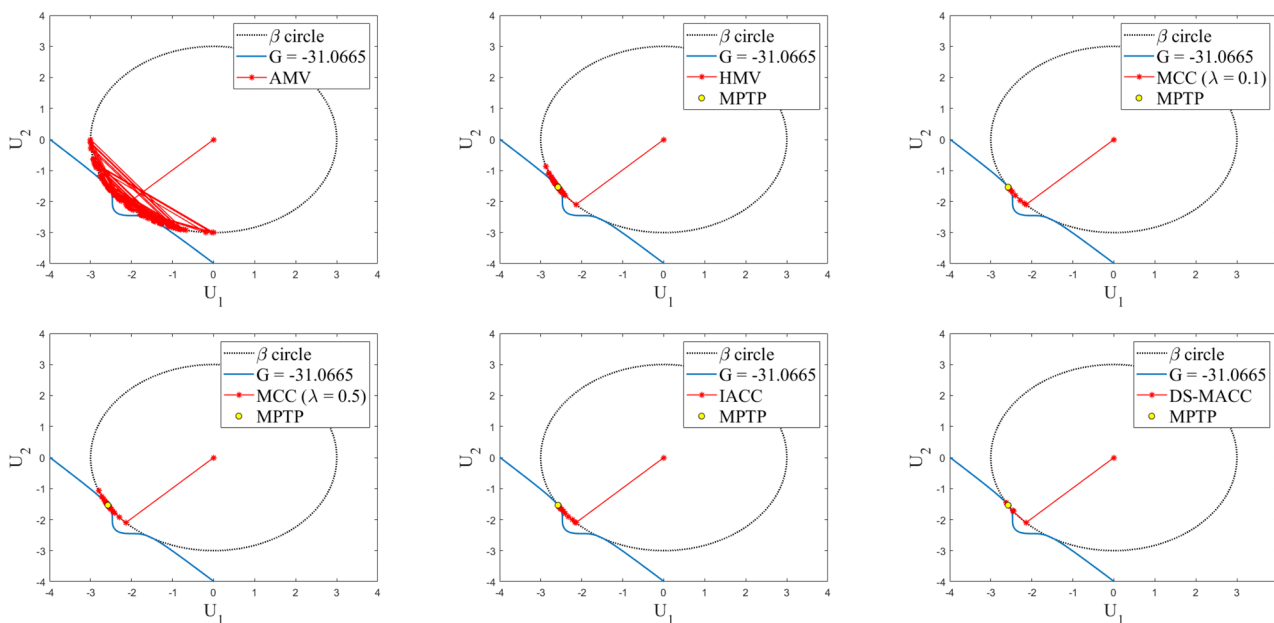


Fig. 7 Iterative histories of MPTP search for Example 2

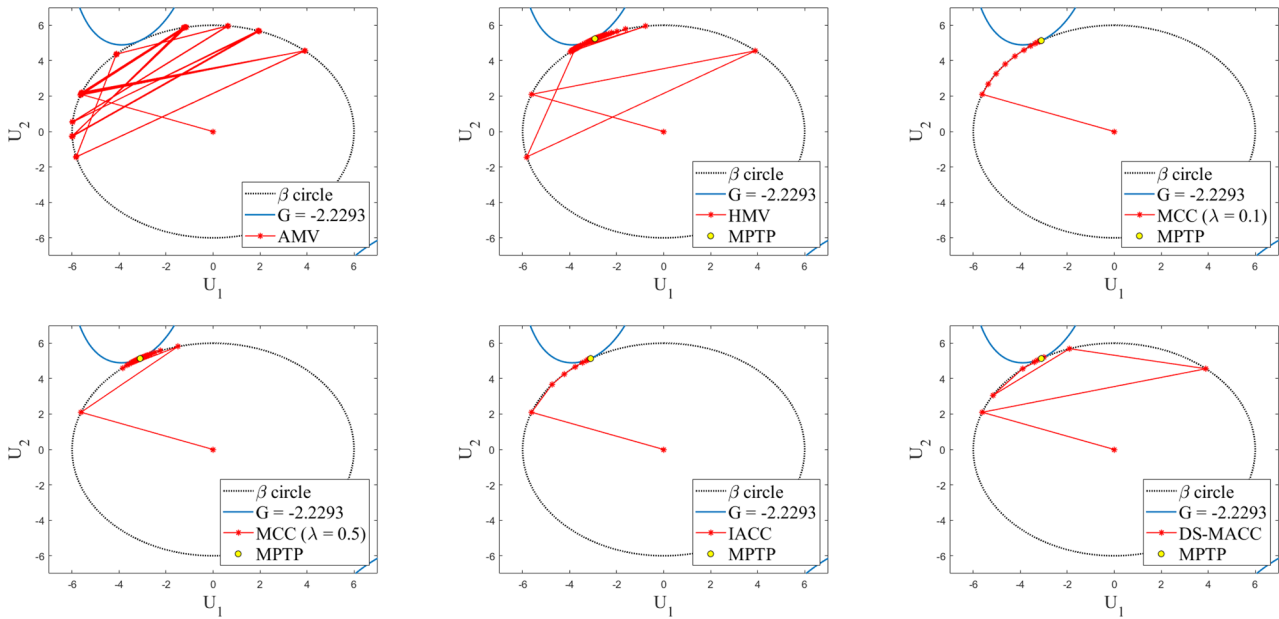


Fig. 8 Iterative histories of MPTP search for Example 3

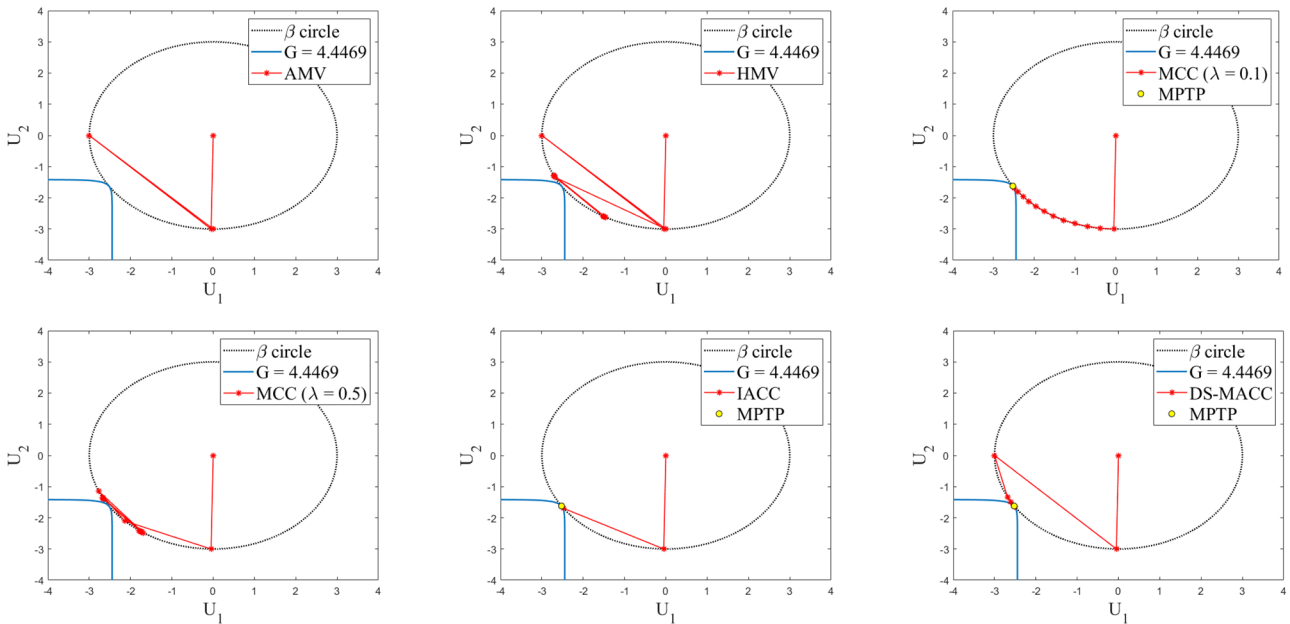


Fig. 9 Iterative histories of MPTP search for Example 4

Example 8: To illustrate the efficiency of the proposed method, an explicit performance function with a large number of random variables is considered here which is introduced based on the displacement of the node under load $P_1(\Delta_{p1}^z)$ at the z-direction in the space truss with 24 elements shown in Fig. 12. Finite element method can be used to

calculate Δ_{p1}^z which should be less than the maximum value of 0.01 m (Keshtegar et al. 2018a).

$$G_8(X^*) = 0.01 - |\Delta_{p1}^z| \tag{18}$$

There are 32 independent normal and non-normal random variables in this example. Their description and

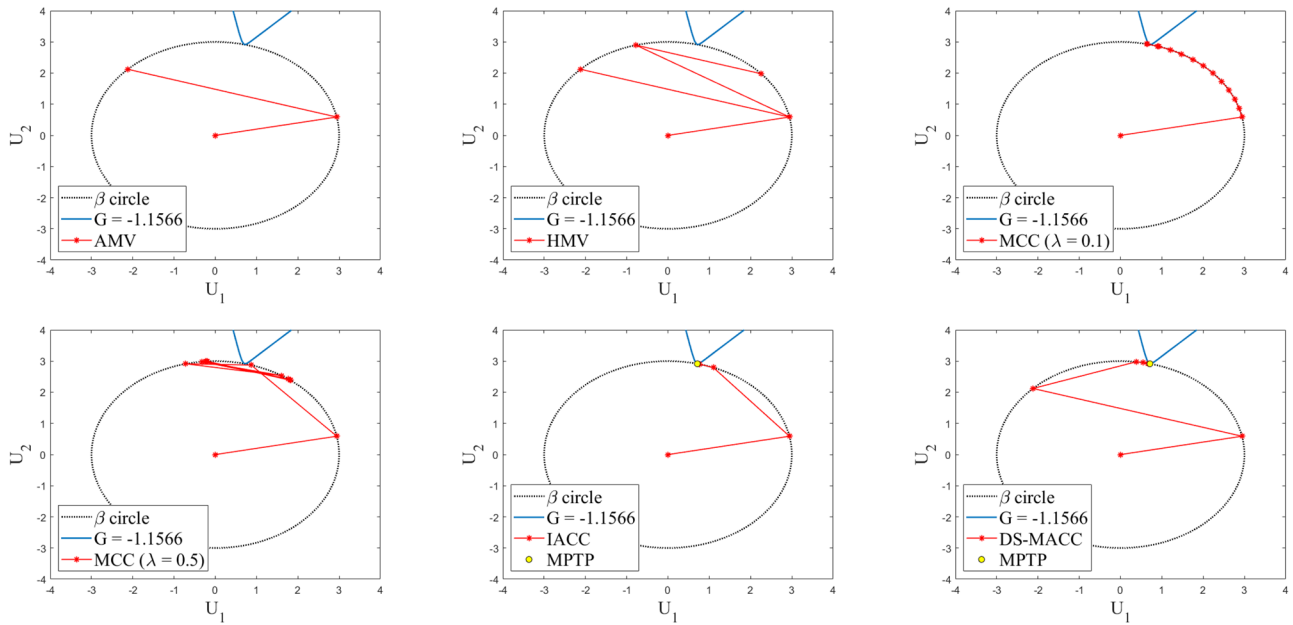


Fig. 10 Iterative histories of MPTP search for Example 6

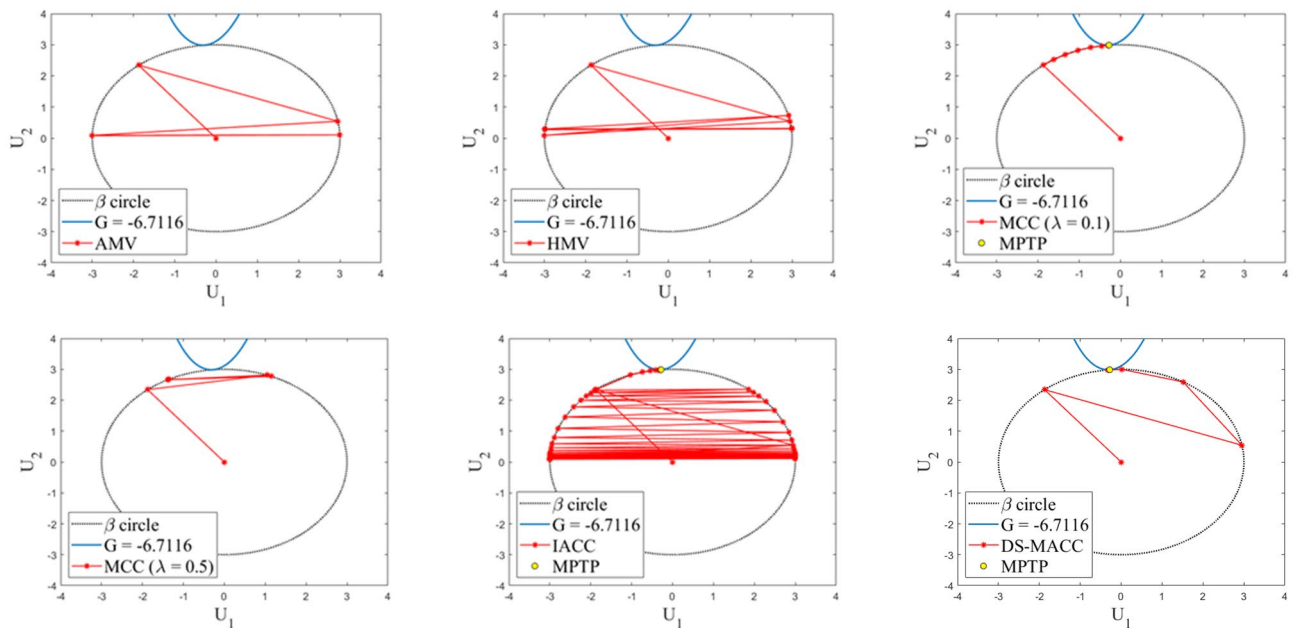


Fig. 11 Iterative histories of MPTP search for Example 7

statistical properties of them are listed in Table 3. The target reliability index is 3.0. From Table 4, it is apparent that the proposed method has superior performance above others including the IACC method. While the Ds-MACC method requires only 650 function calls to converge to MPTP, the IACC and SMCC methods need 1170 and 1366

function evaluations to do so. In addition, HMV, HCC, and MCC methods could not reach the solution because of their instability and inefficiency. This example vividly shows the robustness of the proposed method to reduce the computational burden of structural reliability analysis.

Fig. 12 The space truss structure of Example 8

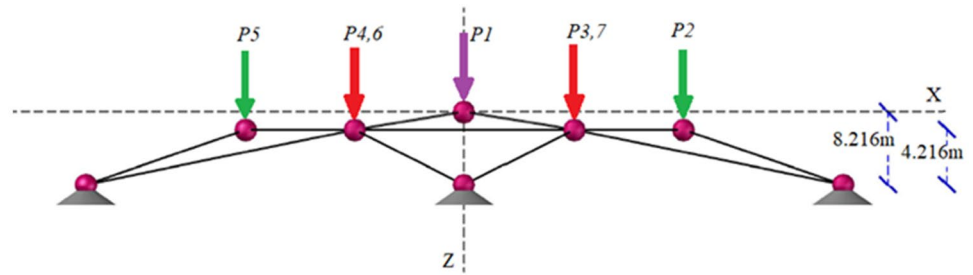


Table 3 Statistics of random variables in Example 8

Variables	Distribution	Mean	Standard deviation	Description
$A_1 - A_6(m^2)$	Normal	0.013	0.1	Cross-sectional area
$A_7 - A_{12}(m^2)$	Normal	0.01	0.1	Cross-sectional area
$A_{13} - A_{24}(m^2)$	Normal	0.016	0.1	Cross-sectional area
$E(GPa)$	Normal	205	0.12	Young's modulus
$P_1(kN)$	Gumbel	20	0.15	Point load
$P_2 - P_7(kN)$	Gumbel	10	0.12	Point loads

6.2 Set II: mathematical and structural examples for reliability-based design optimization

Example 9: A weakly nonlinear RBDO example is given as follows (Hao et al. 2019):

$$\begin{aligned}
 \text{find } \mathbf{d} &= [d_1, d_2]^T \\
 \text{min } f(\mathbf{d}) &= d_1 + d_2 \\
 \text{s.t. } P_f[G_j(\mathbf{X}) > 0] &\leq \Phi(-\beta_j^i), j = 1, 2, 3 \\
 G_1 &= 1 - \frac{X_1^2 X_2}{20} \\
 G_2 &= 1 - \frac{(X_1 + X_2 - 5)^2}{30} - \frac{(X_1 - X_2 - 12)^2}{120} \\
 G_3 &= 1 - \frac{80}{X_1^2 + 8X_2 + 5} \\
 0 \leq d_i \leq 10, X_i &= N(d_i, 0.3) \text{ for } i = 1, 2 \\
 \mathbf{d}^0 &= [5, 5]^T, \beta_1^1 = \beta_2^2 = \beta_3^3 = 3.0
 \end{aligned}
 \tag{19}$$

Table 4 Comparison of different methods for Example 8

	HMV	HCC	MCC	SMCC (Keshtegar et al. 2018a)	IACC	DS-MACC
X_{MPTP}^*						
$A_1 - A_6(m^2)$	-	-	-	128.4612	128.5791	128.5791
$A_7 - A_{12}(m^2)$	-	-	-	99.6559	99.6588	99.6588
$A_{13} - A_{24}(m^2)$	-	-	-	159.7680	159.7923	159.7923
$E(GPa)$	-	-	-	174.0782	168.8693	168.8693
$P_1(kN)$	-	-	-	31.9374	31.1967	31.1967
$P_2 - P_7(kN)$	-	-	-	9.7826	9.7681	9.7681
$g(X_{MPTP})(cm)$	Chaos	Chaos	Chaos	- 0.46428	- 0.4679	- 0.4679
Iterations	-	-	-	21	18	10
Function calls	-	-	-	1366	1170	650

This example involves two independent random variables with normal distribution and three probabilistic constraints. The optimum results obtained by AMV, HMV, MCC, HCC, and ASSA are extracted from literature (Hao et al. 2019) and are shown in Table 5 for comparison. These methods use a double-loop RBDO algorithm to find the optimum. The MCS method with 10^6 samples is utilized to validate the reliability level of each probabilistic constraint. Results indicate all methods can find the optima, however, DS-MACC and IACC methods that use the SORA procedure are faster and more efficient to find the optimum design variables and can converge to that point after 212 and 219 function calls, respectively, which is about two third of that of needed for the ASSA method. Moreover, the MCC method with $\lambda = 0.1$ is the most computationally expensive method because of using a very small step size in each iteration. The design variables obtained after the first, second, and last cycles can be seen in Fig. 13. It can be seen that the third constraint is inactive during the process.

Example 10 A highly nonlinear RBDO example is considered as follows (Keshtegar et al. 2018a):

Table 5 Comparison of different methods for Example 9

	f	d	Iter/FE	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}
AMV	6.7257	(3.4391, 3.2866)	-/360	2.98	3.05	Inf
HMV	6.7257	(3.4391, 3.2866)	-/369	2.98	3.05	Inf
HCC	6.7257	(3.4391, 3.2866)	-/426	2.98	3.05	Inf
MCC ($\lambda = 0.1$)	6.7257	(3.4391, 3.2866)	-/2355	2.98	3.05	Inf
ASSA	6.7257	(3.4391, 3.2866)	-/318	2.98	3.05	Inf
IACC	6.7257	(3.4391, 3.2866)	4/219	2.97	3.05	Inf
DS-MACC	6.7257	(3.4391, 3.2866)	4/212	2.97	3.05	Inf

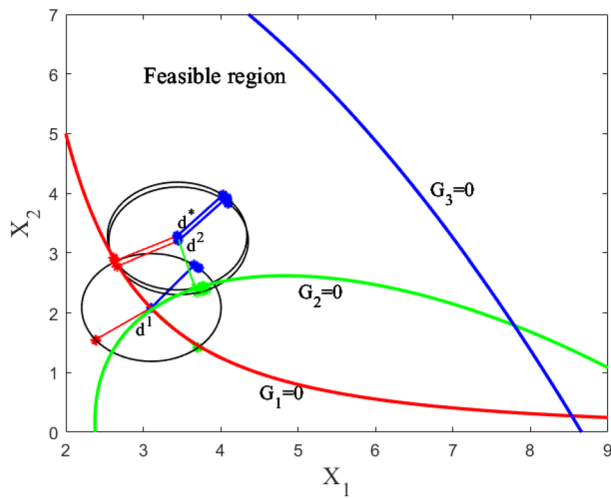


Fig. 13 Iterative history of Example 9

$$\begin{aligned}
 \mathit{find} \mathit{d} &= [d_1, d_2]^T \\
 \min f(\mathit{d}) &= -\frac{(d_1 + d_2 - 10)^2}{30} - \frac{(d_1 - d_2 + 10)^2}{120} \\
 \mathit{S.t.} P_f[G_j(\mathbf{X}) > 0] &\leq \Phi(-\beta_j^i), j = 1, 2, 3 \\
 G_1 &= 1 - \frac{X_1^2 X_2}{20} \\
 G_2 &= -1 + (Y - 6)^2 + (Y - 6)^3 - 0.6(Y - 6)^4 + Z \\
 Y &= 0.9063X_1 + 0.4226X_2, Z = 0.4226X_1 - 0.9063X_2 \\
 G_3 &= 1 - \frac{80}{X_1^2 + 8X_2 + 5} \\
 0 \leq d_i \leq 10, X_i &= N(d_i, 0.3) \text{ for } i = 1, 2 \\
 \mathit{d}^0 &= [5, 5]^T, \beta_1^1 = \beta_2^1 = \beta_3^1 = 3.0
 \end{aligned}
 \tag{20}$$

In this example, the two random variables have normal distribution. Three probabilistic constraints are defined herein. The second constraint is highly nonlinear which is a difficult challenge for methods to deal with. For the aim of comparison, the results of the iterative algorithms are extracted from literature (Keshtegar et al. 2018a). For this example, the MCS method with 10^7 samples is employed.

The RBDO results for this example are presented in Table 6. Moreover, to compare the performance of the proposed method with kriging-based RBDO methods, the results for hybrid adaptive kriging-based single-loop (HAK-SLA) method are also extracted from literature (Yang et al. 2021) which considered the stopping criterion of $\epsilon = 10^{-3}$ for reliability analysis. Therefore, we also solved this example with the same stopping criterion which is shown in the last row of Table 6. As it is evident, the proposed method shows the fastest convergence rate as it can reach the optimum after 5 cycles and 348 function evaluations. Whereas, the DCC method requires 10 iterations and 567 function calls to do so. The IACC is also more efficient than the DCC method with approximately 100 more function calls than the proposed method. The HCC and MCC need more than 16,000 and 7000 function evaluations, respectively, to converge which is very exorbitant. The AMV and HMV methods cannot make it to satisfy the convergence criterion. The DS-MACC method also works slightly better than the HAK-SLA method with only 5 cycles and 299 function calls while HAK-SLA requires 36 iterations and 308 function evaluations which shows the better performance of the proposed method in comparison with the kriging-based method. In Fig. 14, the first, second, and last cycles of the proposed method’s performance for this example are illustrated. It shows that the first and second probabilistic constraints are the active ones. The high nonlinearity of the second constraint is easily handled by the proposed method with the least number of function calls. This certifies the superior performance of DS-MACC method coupled with the SORA over other methods with nested double-loop RBDO procedure.

Example 11 The structural diagram of a roof truss subjected to uniform loads is illustrated in Fig. 15 in which, the top and compression bars are made of reinforced concrete members, bottom and tension bars are made of steel. Regarding the structural mechanics’ requirements, the perpendicular deflection Δ_C should be less than 0.03. This example is considered in the RBDO problem as follows (Keshtegar et al. 2018a):

Table 6 Comparison of different methods for Example 10

	f	d	Iter/FE	β_1^{MCS}	β_2^{MCS}	β_3^{MCS}
AMV	–	–	–	–	–	–
HMV	–	–	–	–	–	–
HCC (Keshtegar et al. 2018a)	– 1.7247	(4.5581, 1.9645)	10/16299	2.95	–	Inf
MCC ($\lambda = 0.1$) (Keshtegar 2017a, b)	– 1.7247	(4.5581, 1.9645)	10/7260	–	–	Inf
DCC (Keshtegar et al. 2018a)	– 1.7247	(4.5581, 1.9645)	10/567	2.95	–	Inf
IACC	– 1.7247	(4.5581, 1.9645)	5/412	2.95	3.21	Inf
DS-MACC	– 1.7247	(4.5581, 1.9645)	5/348	2.95	3.21	Inf
HAK-SLA ($\epsilon = 10^{-3}$) (Yang et al. 2021)	– 1.7247	(4.5581, 1.9645)	36/308	2.95	3.21	Inf
DS-MACC ($\epsilon = 10^{-3}$)	– 1.7247	(4.5581, 1.9645)	5/299	2.95	3.21	Inf

$$\begin{aligned}
 & \text{Find } \mathbf{d} = [A_c, A_s]^T \\
 & \text{min } f(\mathbf{d}) = 20224A_s + 364A_c \\
 & \text{S.t. } P_f \left[G(X) = 0.03 - \left(\frac{ql^2}{2} \right) \left(\frac{3.81}{A_c E_c} + \frac{1.13}{A_s E_s} \right) \leq 0 \right] \leq \Phi(-\beta_i) \\
 & \text{where } 0.0006 \leq A_s \leq 0.0012, 0.018 \leq A_c \leq 0.063 \\
 & [A_s, A_c]^0 = [0.001, 0.042], \beta_i = 3.0
 \end{aligned} \tag{21}$$

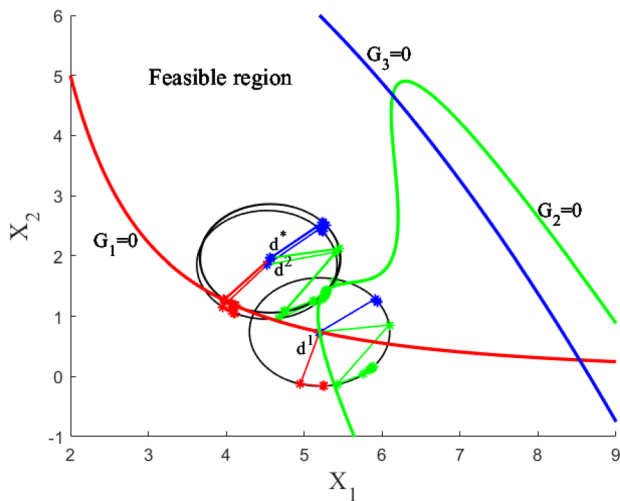


Fig. 14 Iterative history of Example 10

where l is the length of truss, A_c , and A_s are the cross-sectional areas of reinforced concrete and steel bars, respectively. E_c and E_s represent the corresponding elastic modulus. This example involves two random design variables, namely A_s and A_c with four random variables. The statistical parameters of the random variables can be found in Table 7. The results for the RBDO model of this example are solved by HCC, HDMV, and DCC methods and extracted from (Keshtegar et al. 2018b) and (Keshtegar et al. 2018a) to compare with the IACC and proposed method. In Table 8, the results are indicated. Furthermore, the reliability index for the LSF of this example is computed with 10^7 samples using the MCS method. It is observed that all methods are converged accurately to the optimum design point with a high-reliability level with a good agreement with results presented in (Rashki et al. 2014). However, the number of function calls for the proposed method is about thirteen times and three times faster than HCC, HDMV, and DCC methods, respectively. Hence, the IACC is as efficient as DS-MACC method but still requires more function evaluations. It should be mentioned that the DCC method has the highest reliability

Fig. 15 Schematic of the roof truss

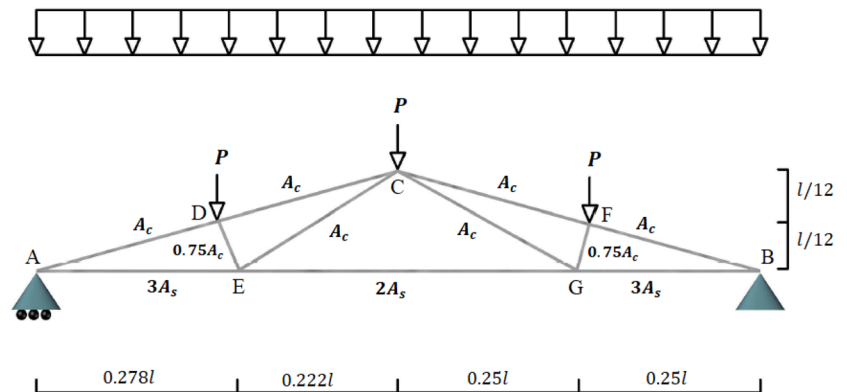


Table 7 Statistics of random variables in Example 11

Variables	Distribution	Mean	Standard deviation
$q(N/m)$	Normal	20,000	1400
$l(m)$	Normal	12	0.12
$A_s(m^2)$	Normal	Design variable	5.9852×10^{-5}
$A_c(m^2)$	Normal	Design variable	0.0048
$E_s(Pa)$	Normal	10^{11}	6×10^9
$E_c(Pa)$	Normal	2×10^{10}	1.2×10^9

level among all, while its computational effort is about two times that of the proposed method.

Table 8 Comparison of different methods for Example 11

	HCC ($\lambda = 0.1$) (Keshtegar et al. 2018b)	HDMV (Keshtegar et al. 2018b)	DCC (Keshtegar et al. 2018b)	IACC	DS-MACC	Lit (Rashki et al. 2014)
$A_s \times 10^{-3} (m^2)$	1.1202	1.0886	1.0886	1.0883	1.0883	1.068
$A_c \times 10^{-2} (m^2)$	3.6397	3.8358	3.8358	3.8371	3.8371	4.046
Optimum	35.9045	35.9781	35.9781	35.9785	35.9785	36.24
X_{MPTP}^*						
$q^* (N/m)$	22,696.129	22,522.635	22,606.49	22,504.6882	22,504.6882	
$l^* (m)$	12.0746	12.0693	12.0721	12.0686	12.0686	
$A_s^* \times 10^{-2} (m^2)$	10.5455	10.2597	10.323	10.2515	10.2515	
$A_c^* \times 10^{-2} (m^2)$	3.0575	3.0917	3.117	3.0833	3.0833	
$E_s^* \times 10^{10} (Pa)$	9.218	9.267	9.234	9.2753	9.2753	
$E_c^* \times 10^{10} (Pa)$	1.916	1.924	1.923	1.9245	1.9245	
Iterations	45	9	12	12	12	
Function calls	6929	1405	1102	576	576	40,000
β_{MCS}	2.862	2.917	3.003	2.91	2.91	3.01

Example 12 A welded beam problem (Cho and Lee 2011) shown in Fig. 16 is given. This problem has four independent normally distributed random variables and five probabilistic constraints. Detailed explanation of this example can be accessed via (Cho and Lee 2011). The RBDO model is defined as follows:

$$\begin{aligned}
 & \text{Find } \mathbf{d} = [d_1, d_2, d_3, d_4]^T \\
 & \text{min } f(\mathbf{d}, \mathbf{z}) = c_1 d_1^2 d_2 + c_2 d_3 d_4 (z_2 + d_2) \\
 & \text{s.t. } P_f [G_j(\mathbf{X}) > 0] \leq \Phi(-\beta_j^i), j = 1, 2, \dots, 5
 \end{aligned}
 \tag{22}$$

The RBDO results for AMV, HVM, MCC, HCC, and HCMV methods using double-loop strategy are extracted from from (Keshtegar et al. 2018c) and (Zhu et al. 2021), respectively, and summarized in Table 9. All methods can

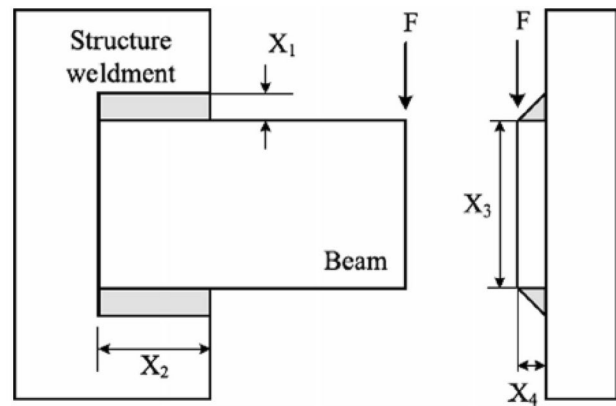


Fig. 16 Schematic of the welded beam structure

successfully obtain the optimum as 2.5913. The performance of AMV, HVM, and HCC are the same as each other with 1350 function calls. The MCC method requires the highest number of function evaluations to reach the optimal design point. The HCMV method could save about %50 of the computational cost of the HCC method, however, it is still two times and three times slower than the IACC and proposed method, respectively. DS-MACC method could also successfully reduce the number of function calls of the IACC method from 409 to 389. In addition, the SORA procedure needs only four cycles to reach the solution, while the double-loop RBDO needs more than 10 iterations to do so, which shows the high convergence rate of the proposed method.

Example 13 Speed reducer illustrated in Fig. 17 is utilized for engine and propeller rotation with efficient velocity in the light plane (Keshtegar et al. 2018c). This design problem includes seven normal and independent random variables

Table 9 Comparison of different methods for Example 12

	AMV	HMV	MCC ($\lambda = 0.15$)	HCC ($\lambda = 0.15$)	HCMV	IACC	DS-MACC
Optimum design point X^*							
d_1^*	5.7300	5.7300	5.7300	5.7300	5.73	5.7300	5.7300
d_2^*	200.8981	200.8981	200.8981	200.8981	200.9	200.8982	200.8982
d_3^*	210.5977	210.5977	210.5977	210.5977	210.6	210.5977	210.5977
d_4^*	6.238936	6.238936	6.238936	6.238936	6.239	6.2389	6.2389
Optimum	2.591318	2.591318	2.591318	2.591318	2.591	2.591319	2.591319
Iterations	15	15	12	15	12	4	4
Function calls	1350	1350	13,378	1350	854	409	389

plus eleven probabilistic constraints. The random design variables are gear width (X_1), gear module (X_2), the number of pinion teeth (X_3), distance between bearings (X_4, X_5), and diameter of each shaft (X_6, X_7). The RBDO model for this example is expressed in Eq. (22), and more details is accessible in (Keshtegar et al. 2018c):

$$\begin{aligned}
 & \text{Find } \mathbf{d} = [d_1, d_2, d_3, d_4, d_5, d_6, d_7]^T \\
 & \min f(\mathbf{d}) = 0.7854d_1d_2^2(3.3333d_3^2 + 14.9334d_3 - 43.0934) - 1.508d_1 + 7.477(d_6^3 + d_7^3) + 0.7854(d_4d_6^2 + d_5d_7^2) \quad (23) \\
 & \text{S.t. } P_f[G_j(\mathbf{X}) > 0] \leq \Phi(-\beta_t^j), j = 1, 2, \dots, 11
 \end{aligned}$$

Table 10 presents the results of the RBDO approaches for this example. Results for the double-loop based AMV, HMV, MCC, HCC, and HCMV are reported from (Zhu et al. 2021). It should be noted that the convergence criterion is considered as 10^{-4} for this example. From Table 8, while all methods can find the optima, it can be seen that only

the HCMV, IACC, and DS-MACC methods can obtain the optimum design variables with less than 1000 function calls. Both the IACC and proposed method are more efficient than others with 184 and 154 function calls, respectively. Moreover, AMV, HMV, and HCC methods have the same efficiency as each other, which shows the constraints are almost convex types. Therefore, DS-MACC method coupled with

the double-loop strategy may yield the same result as the AMV method. But, incorporating the SORA approach could reduce the number of function calls from 1147 to 154 for this example. The HCMV method also reduces the function calls by combining their proposed method with sufficient conditions with AMV.

7 Conclusion

In this work, a new double-step PMA based on the improved adaptive chaos control (IACC) method was proposed and integrated with the SORA procedure to alleviate the computational cost of RBDO problems. Usually, the reliability analysis loop is responsible for the largest percentage of computations. So, making this process more efficient is a key to reduce the computational burden. The nonlinearity degree of LSFs can be easily found via the direction of the normalized gradient vector which is utilized in the IACC method. However, the strategy used for step length reduction is not efficient since it does not consider any information related to the type of the LSF. Thus, the computational cost can rise in the case of convex and highly nonlinear LSFs. Regarding this defect, double-step modified adaptive chaos control (DS-MACC) is introduced. In this method, to detect the function type, two novel criteria are considered in the

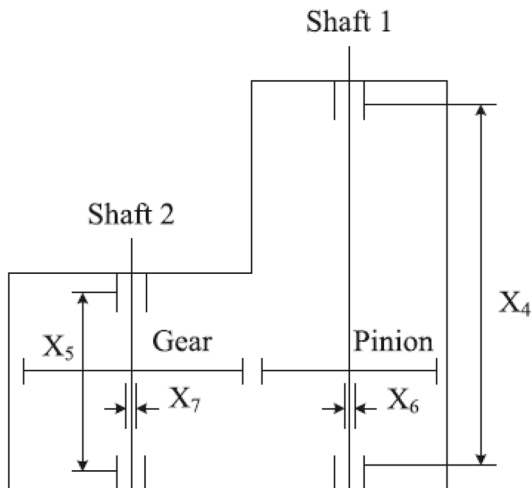


Fig. 17 Schematic of the speed reducer

Table 10 Comparison of different methods for Example 13

	AMV	HMV	MCC ($\lambda = 0.15$)	HCC ($\lambda = 0.15$)	HCMV	IACC	DS-MACC
Optimum design point X^*							
d_1^*	3.576485	3.576485	3.576485	3.576485	3.576485	3.576485	3.576485
d_2^*	0.7	0.7	0.7	0.7	0.7	0.7	0.7
d_3^*	17	17	17	17	17	17	17
d_4^*	7.3	7.3	7.3	7.3	7.3	7.3	7.3
d_5^*	7.753833	7.753833	7.753833	7.753833	7.523829	7.754134	7.754134
d_6^*	3.365218	3.365218	3.365218	3.365218	3.365218	3.365218	3.365218
d_7^*	5.301668	5.301668	5.301668	5.301668	5.301668	5.301668	5.301668
Optimum	3038.5967	3038.5967	3038.5967	3038.5967	3038.5966	3038.6125	3038.6125
Iterations	3	3	3	3	3	3	3
Function calls	1147	1147	1147	1147	961	187	154

first step. based on the different conditions of these criteria, three states of iterations can happen for each of which, a particular relation for initial step size calculation is proposed which makes the whole process faster because of choosing an efficient step size in advance. During the second step, the initial step length is adaptively modified until the angle between two consecutive steepest descent directions is smaller than a particular value. In addition, to prevent a large number of function evaluations of dealing with highly nonlinear constraints in the double-loop RBDO strategy, the DS-MACC method is linked with SORA.

To evaluate the capability of the proposed method, two sets of problems comprised of mathematical and structural reliability analysis as well as RBDO problems are considered. Results indicate that DS-MACC has been successful to address the inefficiency of IACC and the high computational cost of traditional double-loop approaches. DS-MACC is capable of dealing with highly nonlinear LSFs with a fast convergence rate. It also has a high accuracy to find the MPTP thanks to the double-step strategy integrated in this method besides the criteria introduced to find the intervals in which MPTP may exist. To conclude, the robustness and less computational burden of the proposed method certifies its competitive and reliable performance, especially in the case of dealing with highly nonlinear performance functions.

Funding No funding was received for conducting this study.

Declarations

Competing interest The authors declare that they have no conflict of interest.

Replication of results The MATLAB codes for the current study are available from the corresponding author on reasonable request.

References

- Aoues Y, Chateaneuf A (2010) Benchmark study of numerical methods for reliability-based design optimization. *Struct Multidiscip Optim* 41(2):277–294
- Du X, Chen W (2004) Sequential optimization and reliability assessment method for efficient probabilistic design. *J Mech Des* 126(2):225–233
- Gong J-X, Yi P (2011) A robust iterative algorithm for structural reliability analysis. *Struct Multidiscip Optim* 43(4):519–527
- Hao P, Wang Y, Liu C, Wang B, Wu H (2017) A novel non-probabilistic reliability-based design optimization algorithm using enhanced chaos control method. *Comput Methods Appl Mech Eng* 318:572–593
- Hao P, Ma R, Wang Y, Feng S, Wang B, Li G, Xing H, Yang F (2019) An augmented step size adjustment method for the performance measure approach: toward general structural reliability-based design optimization. *Struct Saf* 80:32–45
- Hasofer AM, Lind N (1974) Exact and invariant second-moment code format. *J Eng Mech-Asce* 100:111–121
- Jeong S-B, Park GJ (2017) Single loop single vector approach using the conjugate gradient in reliability based design optimization. *Struct Multidiscip Optim* 55(4):1329–1344
- Kang Z, Luo Y, Li A (2011) On non-probabilistic reliability-based design optimization of structures with uncertain-but-bounded parameters. *Struct Saf* 33(3):196–205
- Keshtegar B (2016a) Chaotic conjugate stability transformation method for structural reliability analysis. *Comput Methods Appl Mech Eng* 310:866–885
- Keshtegar B (2016b) Stability iterative method for structural reliability analysis using a chaotic conjugate map. *Nonlinear Dyn* 84(4):2161–2174
- Keshtegar B (2017a) A hybrid conjugate finite-step length method for robust and efficient reliability analysis. *Appl Math Model* 45:226–237
- Keshtegar B (2017b) A modified mean value of performance measure approach for reliability-based design optimization. *Arab J Sci Eng* 42(3):1093–1101
- Keshtegar B, Chakraborty S (2018a) Dynamical accelerated performance measure approach for efficient reliability-based design optimization with highly nonlinear probabilistic constraints. *Reliab Eng Syst Saf* 178:69–83
- Keshtegar B, Hao P (2018b) A hybrid descent mean value for accurate and efficient performance measure approach of

- reliability-based design optimization. *Comput Methods Appl Mech Eng* 336:237–259
- Keshtegar B, Lee I (2016c) Relaxed performance measure approach for reliability-based design optimization. *Struct Multidiscip Optim* 54(6):1439–1454
- Keshtegar B, Miri M (2014) Introducing conjugate gradient optimization for modified HL-RF method. *Eng Comput* 31:775
- Keshtegar B, Hao P, Meng Z (2017c) A self-adaptive modified chaos control method for reliability-based design optimization. *Struct Multidiscip Optim* 55(1):63–75
- Keshtegar B, Baharom S, El-Shafie A (2018c) Self-adaptive conjugate method for a robust and efficient performance measure approach for reliability-based design optimization. *Eng Comput* 34(1):187–202
- Lee JO, Yang YS, Ruy WS (2002) A comparative study on reliability-index and target-performance-based probabilistic structural design optimization. *Comput Struct* 80(3–4):257–269
- Li M, Wang Z (2020) Deep learning for high-dimensional reliability analysis. *Mech Syst Signal Proc* 139:106399
- Li M, Wang Z (2022) Deep reliability learning with latent adaptation for design optimization under uncertainty. *Comput Methods Appl Mech Eng* 397:115130
- Li G, Meng Z, Hu H (2015) An adaptive hybrid approach for reliability-based design optimization. *Struct Multidiscip Optim* 51:1051–1065
- Liu P-L, Der Kiureghian A (1991) Optimization algorithms for structural reliability. *Struct Saf* 9(3):161–177
- Melchers RE, Beck AT (2018) *Structural reliability analysis and prediction*. John Wiley & sons, New York
- Meng Z, Li G, Wang BP, Hao P (2015) A hybrid chaos control approach of the performance measure functions for reliability-based design optimization. *Comput Struct* 146:32–43
- Meng Z, Li G, Yang D, Zhan L (2017) A new directional stability transformation method of chaos control for first order reliability analysis. *Struct Multidiscip Optim* 55(2):601–612
- Rackwitz R, Flessler B (1978) Structural reliability under combined random load sequences. *Comput Struct* 9(5):489–494
- Rashki M, Miri M, Moghaddam MA (2014) A simulation-based method for reliability based design optimization problems with highly nonlinear constraints. *Autom Constr* 47:24–36
- Roudak MA, Shayanfar MA, Barkhordari MA, Karamloo M (2017) A robust approximation method for nonlinear cases of structural reliability analysis. *Int J Mech Sci* 133:11–20
- Roudak MA, Shayanfar MA, Karamloo M (2018) Improvement in first-order reliability method using an adaptive chaos control factor. *Structures* 16:150–156
- Santos S, Matioli L, Beck A (2012) New optimization algorithms for structural reliability analysis. *Comput Model Eng Sci* 83(1):23–55
- Shayanfar MA, Barkhordari MA, Roudak MA (2018) A new effective approach for computation of reliability index in nonlinear problems of reliability analysis. *Commun Nonlinear Sci Numer Simul* 60:184–202
- Wang Z, Wang P (2013) A maximum confidence enhancement based sequential sampling scheme for simulation-based design. *J Mech Design* 10(1115/1):4026033
- Wang L, Liu Y, Liu D, Zhangming Wu (2021) A novel dynamic reliability-based topology optimization (DRBTO) framework for continuum structures via interval-process collocation and the first-passage theories. *Comput Methods Appl Mech Eng* 386:114107
- Wang L, Liu Y, Li M (2022) Time-dependent reliability-based optimization for structural-topological configuration design under convex-bounded uncertain modeling. *Reliab Eng Syst Saf* 221:108361
- Wu Y-T, Millwater H, Cruse T (1990) Advanced probabilistic structural analysis method for implicit performance functions. *AIAA J* 28(9):1663–1669
- Yang D (2010) Chaos control for numerical instability of first order reliability method. *Commun Nonlinear Sci Numer Simul* 15(10):3131–3141
- Yang D, Yi P (2009) Chaos control of performance measure approach for evaluation of probabilistic constraints. *Struct Multidiscip Optim* 38(1):83–92
- Yang M, Zhang D, Jiang C, Han X, Li Q (2021) A hybrid adaptive kriging-based single loop approach for complex reliability-based design optimization problems. *Reliab Eng Syst Safe* 215:107736
- Youn BD, Choi KK, Park YH (2003) Hybrid analysis method for reliability-based design optimization. *J Mech Des* 125(2):221–232
- Youn BD, Choi K, Du L (2005) Adaptive probability analysis using an enhanced hybrid mean value method. *Struct Multidiscip Optim* 29(2):134–148
- Zhang Y (1994) *Finite element reliability methods for inelastic structures*. University of California, Berkeley
- Zhu S-P, Keshtegar B, Trung N-T, Yaseen ZM, Bui DT (2021) Reliability-based structural design optimization: hybridized conjugate mean value approach. *Eng Comput* 37(1):381–394

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.