

Breadth-First Search Multi-Dimensional Binary Search Tree based Algorithms for Structural Reliability Estimations

Xuyong Chen¹, Jia-Liang Le², Zhifeng Xu^{1*}, Qiaoyun Wu¹

1. School of Civil Engineering and Architecture, Wuhan Institute of Technology

2. Department of Civil, Environmental and Geo-Engineering, University of Minnesota, Twin-Cities

Abstract: This research proposes a set of novel algorithms for structural reliability estimation based on multi-dimensional binary search tree and breadth-first search, namely the reliability accuracy supervised searching algorithm, the limit-state surface resolution supervised searching algorithm and the reliability index precision supervised fast searching algorithm. The proposed algorithms have the following strengths: 1, all the proposed algorithms have satisfactory computational efficiency by reducing redundant samplings; 2, their computational costs are stable and computable; 3, performance functions of high non-linearity can be well handled; 4, the reliability accuracy supervised searching algorithm can adapt its computational cost according to a prescribed accuracy; 5, the limit-state surface resolution supervised searching algorithm is able to probe sharp changes on limit-state surfaces; 6, the reliability index precision supervised fast searching algorithm computes the reliability index with sufficient precision in a fast way.

Keywords: Structural reliability estimation; Performance function; Limit state surface; Reliability index; Binary search tree; Breadth-first search

1. Introduction

Since uncertainty cannot be eliminated completely, reliability-based analysis has penetrated to the design of various engineering structures. One main challenge in structural reliability analysis is how to efficiently estimate structural failure probability which is typically on the order of 10^{-4} [1], which has continuously attracted a great number of scholars' interests for the past decades.

Solutions for the abovementioned problem can be roughly classified into the following two classes: probabilistic modeling-based methods and Monte-Carlo simulation (MCS)-based methods [1, 2]. Probabilistic modeling-based methods, such as the bundle model with Gaussian distribution for ductile structures [3], the weakest-link models for brittle and quasi-brittle structures [4-6], the fishnet model for lamellar materials [7], etc., resort to mechanics-based probabilistic models to predict structural failure probability, in which the model parameters can be prescribed through theoretical analysis or obtained by optimal fittings with experimental data. These methods are powerful for cases of relatively simple structural configurations and failure mechanisms. Recently, Xu and Le proposed a random field-based model for structural reliability estimation [8, 9], which can effectively describe failure statistics of structures with complex geometries. Yet the failure mechanism employed by the model is simple, and the computational cost is relatively expensive.

On the other hand, in spite of higher computational cost than that of probabilistic modeling-based methods, MCS-based methods plays a prevailing role in structural reliability analyses for their advantage in dealing with sophisticated structural configurations and diverse failure mechanisms [10]. Since direct MCS is cumbersome, numerous approaches aiming at improving the balance between computational cost and estimation accuracy have been developed [1]. Most of improvements on MCS -based methods follow the following four clues: sampling methods, reliability approximations, surrogate models and searching algorithms. Among various improvements on sampling methods, conditional probability-based methods (e.g., importance sampling [11], Markov Chain-based subset sampling [12], Bayesian updating sampling [13], etc.) are widely used for its ability to dramatically decrease the sample size. However, the error of these methods would propagate as the sample size decreases. Other improvements on Monte-Carlo (MC) sampling methods, such as Latin hypercube sampling [14], stratified sampling [15], Latinized stratified sampling [16] and quasi-MC sampling [17], would alleviate the contradiction between sampling's uniformity and randomness, but they are not very useful in reducing the sample size. Among various improvements emphasizing on approximation methods, the most famous one should be design point-based methods [18], which use statistical information of some designated points, called design points, to predict structural reliability. First-order reliability method (FORM) [19] and second-order reliability method (SORM) [19] employ the closest point to the origin on the limit-state surface (LSS) as the design point for structural reliability estimations, in which FORM only utilizes information of the first moments, while SORM uses both first and second moments. Hereafter, a series of methods using multiple design points and higher order moments have emerged [20-24]. Nevertheless, the design point-based methods often face poor accuracy if the failure probability ignored is comparable with the required accuracy, etc. Surrogate model-based methods attempt to replace the original performance function (PF) with some computational cheaper models, which are recently heated along with the soaring of machine learnings [25]. Nevertheless, due to the lack of solid mechanical bases, most of the surrogate models, including conventional response surface methods [21], statistical learnings [26], Kriging interpolations [27], support vector machines [28], artificial neuron networks [29] and their hybrids or variations [30-33], would appear to be too crude when high accuracy of reliability estimation is required, which would be even deteriorated when sharp changes on LSS exist. Compared to methods following the three aforementioned three clues, methods emphasize on searching algorithms seem more efficient and accurate, which are realized by regulating the sampling process strategically through some algorithms. The family

of searching algorithms for structural reliability is board and varied, in which some of the heuristic ones are directional divisions [34-37], swarm algorithms [38, 39], genetic algorithms [40-42], gradient or difference-based algorithms [43, 44], etc. Yet, further improvements on searching algorithms with respect to the following aspects are always welcomed: 1, maximizing computational efficiency by minimizing the number of redundant samplings; 2, increasing robustness by adapting to a board type range of PFs; 3, stabilization of the convergence rate; 4, customization for different engineering cases according to the requirement of accuracy.

Aiming at improving structural reliability estimations, this research proposes a set of three algorithms, which employ multi-dimensional Binary Search Trees (k -D Trees) as the data structure and breadth-first search (BFS) as the searching strategy. The proposed algorithm set consists of three algorithms, which are namely the reliability accuracy supervised searching algorithm, the limit-state surface resolution supervised searching algorithm and the reliability index precision supervised fast searching algorithm. The core idea of the proposed algorithms is minimizing the number of sampling points away from LSSs, which are superior for the following aspects: 1, they have satisfactory computational efficiency by significantly reducing redundant samplings; 2, their computational costs are stable and computable; 3, PFs with high non-linearity can be properly handled; 4, the reliability accuracy supervised searching algorithm is adaptive to a prescribed accuracy; 5, the limit-state surface resolution supervised searching algorithm is able to probe sharp changes on LSSs; 6, the reliability index precision supervised fast searching algorithm can fast compute the reliability index (RI). Moreover, the efficiency and practicality of this algorithm set are demonstrated numerically. The rest of this paper is structured as follows: Section 2 presents necessary fundamentals on structural reliability estimations; Section 3 introduces the algorithm set with detailed illustrations on the theory and the implementations; Section 4 provides engineering examples of the proposed algorithms for structural reliability estimations.

2. Fundamentals on structural reliability analysis

Given a structure subjected to k number of random variables denoted by a k -dimensional (k -D) random vector $\mathbf{x} = \{x_1, x_2, \dots, x_k\}$, a nominal response g , referred to as the PF, is chosen as the measure of the structure's failure state. It is noted that the random inputs $\mathbf{x}$ can be a combination of both applied loads and material resistances. By treating $\mathbf{x}$ and g as the inputs and output respectively, we can write $g = g(\mathbf{x})$, in which $g > 0$ or $g < 0$ represent the structure is in the safe state or the failure state respectively.

The limit state between safety and failure is represented by the LSS, i.e.:

$$\mathbf{u} = \{\mathbf{x} \mid g(\mathbf{x}) = 0\} \quad (1)$$

And the structural reliability is computed by

$$R = \int_{g(\mathbf{x}) > 0} f_{\mathbf{x}}(x_1, x_2, \dots, x_k) dx_1 dx_2 \dots dx_k \quad (2)$$

where $f_{\mathbf{x}}(x_1, x_2, \dots, x_k)$ is the joint probability density function (PDF) of each x_i .

In practice, it is convenience to transfer $\mathbf{x}$ into the corresponding zero-mean standard normal vector $\mathbf{X}$ through Nataf transformation, i.e.:

$$F_i(x_i) = \Phi(X_i) \quad (3)$$

where F_i is the cumulative distribution function (CDF) of x_i , and Φ is the standard normal CDF. It is noted that the dependency between any x_i and x_j should also be translated into the corresponding dependency between X_i and X_j [45].

Accordingly, the PF with respect to $\mathbf{X}$ becomes

$$G(\mathbf{X}) = g\{\Phi^{-1}[F(\mathbf{X})]\} \quad (4)$$

where Φ^{-1} is the inverse function of Φ .

The LSS with respect to $\mathbf{X}$ is

$$\mathbf{U} = \{\mathbf{X} \mid G(\mathbf{X}) = 0\} \quad (5)$$

In design point-based methods such as FORM and SORM, the RI is defined as the closest distance from the origin to the LSS, which is also referred to as Hasofer-Lind index [46], i.e.:

$$\beta = \min[|\mathbf{X}|, \mathbf{X} \in \mathbf{U}] \quad (6)$$

where $|\mathbf{X}| = \sqrt{\mathbf{X}^T \mathbf{X}}$ is the norm.

Consequently, the probability within the hyper-sphere of radius β centered at the origin can serve as a conservative approximation of R , i.e.:

$$R \geq R_s = \int_{|\mathbf{X}| < \beta} \Phi_{\mathbf{x}}(X_1, X_2, \dots, X_k) dX_1 dX_2 \dots dX_k \quad (7)$$

where $\Phi_{\mathbf{x}}$ is the joint PDF of each X_i .

If each x_i is independent, then each X_i becomes an independent and identically distributed (i.i.d) standard normal variable, so that β should obey Chi-square distribution with k degree of freedom, i.e.:

$$R \geq R_s = \chi^2(\beta^2, k) \approx \Phi\left(\frac{\beta^2 - k}{\sqrt{2k}}\right) \quad (8)$$

where χ^2 is the Chi-square CDF, and the approximation on the RHS is valid when k is sufficient.

3. Breadth-first search multi-dimensional binary search tree based searching algorithm set for structural reliability estimations

This section presents the essence of BFS and k -D Tree as well as the set of three algorithms for structural reliability estimations, namely the reliability accuracy supervised searching algorithm, the limit-state surface resolution supervised searching algorithm and the reliability index precision supervised fast searching algorithm.

3.1 Method background

The problem for the algorithm to design is stated as follows: given an output g subjected to an input $\mathbf{x} = [x_1, x_2, \dots, x_k]^T$ that obeys the joint PDF $f_{\mathbf{x}}$, and each x_i is considered to be bounded between $[lb_i, ub_i]$, the objective of the algorithm is to efficiently compute the structural reliability represented by the integral in Eq. (2) whose boundary is $\mathbf{u}(\mathbf{x}) = \{\mathbf{x} | g(\mathbf{x}) = 0\}$. Figs. 2 (a) shows the configuration of the problem statement in a two-dimensional (2-D) case.

The core idea of the BFS k -D Tree-based algorithms is hatching important geometrical information for the computation of Eq. (2), such as the volume representing structural reliability, the LSS or the RI, by cooperating topological characteristics, k -D Tree and BFS. Given a k -D cubic space denoted by $\mathbf{C}^k$ whose side-length is C , and a binary search tree whose node has a maximum of 2^l child nodes, where l is level number. The tree and the space $\mathbf{C}^k$ can be linked by the following binary partition construction: 1, each node represents a hyper-cube storing information of the 2^k vertices; 2, the side length of a hyper-cube from the l^{th} level is equal to $2^{-l+1}C$; 3, a hyper-cube can form its 2^k child cubes through a binary spacing portition on each dimension; 4, the hyper-cube represented by the root node is just the space $\mathbf{C}^k$. Traversing of a k -D Tree can be realized by BFS, i.e., iterate each node level by level in an ascending manner. Fig. 1 illustrates the formation of a k -D tree and its traversing using BFS.

Serving as guidances for the search of structural reliability, two types of topological characteristics of a LSS embedded in a mesh grid defined by a level of the k -D Tree are defined, which are grids and nodes (shown in Fig. 2 (b)). A node is called to be positive or negative according to the sign of its PF. Accordingly, grids can be classified into the following three major types: 1, safe grid (SG), which is a grid belongs to the safe domain but does not touch any boundaries, and all its nodes are positive; 2,

failure grid (FG), which is a grid belongs to the failure domain, and all its nodes are negative; 3, limit-state grid (LSG), which is a grid intersects with the LSS or the boundaries. It is evident that the reliability portion of a FG is 0 and the reliability portion of a SG is $F_X(\mathbf{ub}_G) - F_X(\mathbf{lb}_G)$, where $\mathbf{lb}_G$ and $\mathbf{ub}_G$ are the vectors of lower bounds and upper bounds of the grid. The reliability portion of a LSG is bounded between 0 to $F_X(\mathbf{ub}_G) - F_X(\mathbf{lb}_G)$. In addition, LSGs can be further classified into the following four minor types: 1, Type-1 LSG, which contains both positive and negative nodes; 2, Type-2 LSG, which contains only positive nodes and touches the boundaries; 3, Type-3 LSG, which only contains positive nodes and does not touch the boundaries; 4, Type-4 LSG, which only contains negative nodes. It should be noted that judging the major type of a grid according to the signs of its nodes is insufficient, since both SGs and Type-3 LSGs only have positive nodes, while both failure grids and Type-4 LSGs only have negative nodes. However, for in most practical engineering cases, curvatures on LSSs are usually moderate, resulting in the occurrences of Type-3 and Type-4 LSGs so low that whose reliability portions can be ignored. Therefore, a grid with uniformly positive or negative nodes can be postulated to a SG or FG for a primary search of structural reliability, after which a delicate search considering the error caused by Type-3 and Type-4 LSGs can be carried out if required by the accuracy. If a grid is judged as a SG or FG, no further search for LSS is need within that grid, which considerably saves unnecessary calling of PFs and increases the computational efficiency.

3.2 Algorithm presentation

This subsection presents the three algorithms for structural reliability estimations emphasizing on the following three different considerations: securing the accuracy of reliability, ensuring the resolution of LSS, and fast computing the RI. In order to be in accordance with the engineering conventions, all the inputs are translated into the corresponding standard normal inputs through Eq. (3) before computation.

The reliability accuracy supervised searching algorithm

The reliability accuracy supervised searching algorithm is realized by progressively identifying and updating the valid reliability portions from SGs or Type-2 LSGs level by level in the construction of the k -D Tree through BFS. To be specific, in the construction of a level, only Type-1 LSGs will generate child grids for the search of the next level, and the reliability portions of SGs and Type-2 LSGs are added to the total reliability. The stopping criterion is that: the algorithm stops when the maximum error of reliability estimation is less than the reliability accuracy requirement. Before stopping, the reliability portions obtained by interpolations of Type-1 LSGs on the last level are added to the total reliability. The

maximum error is computed as the reliability portion of the total hyper-volume of all Type-1 LSGs in the current level. For instance, if the current level has p number of LSGs and q number of SGs, then the estimated reliability is bounded between $\sum_{i=1}^q F_x(\mathbf{ub}_{SG-i}) - F_x(\mathbf{lb}_{SG-i})$ to $\sum_{i=1}^q F_x(\mathbf{ub}_{SG-i}) - F_x(\mathbf{lb}_{SG-i}) + \sum_{j=1}^p F_x(\mathbf{ub}_{LSG-j}) - F_x(\mathbf{lb}_{LSG-j})$, where $\mathbf{ub}_{SG-i}$ and $\mathbf{lb}_{SG-i}$ are the lower bounds and upper bounds of the i^{th} SG, and $\mathbf{ub}_{LSG-j}$ and $\mathbf{lb}_{LSG-j}$ are the lower bounds and upper bounds of the j^{th} LSG. Evidently, the algorithm is supervised by a prescribed reliability accuracy, which is the source of its name. Fig. 3 (a) shows the algorithm's flowchart, the corresponding pseudo code is presented in the appendix. Though this algorithm would miscount reliability portions from Type-3 and Type-4 LSGs, the reliability estimation is expected to have satisfactory accuracy since the occurrence of Type-3 and Type-4 LSGs with large reliability portion is rare in most engineering cases. Besides, the interpolation process on the last level would dramatically increase the accuracy. At this point, the following formula is recommended to compute the reliability portion of a Type-1 LSG:

$$R_{\text{intp}} = \frac{V_{\text{safe}}}{V_{\text{grid}} \Omega_{\text{grid}}} \int_{\Omega_{\text{grid}}} f_x(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (9)$$

where Ω_{grid} is the grid's domain, V_{grid} is the grid's total hyper-volume, V_{safe} is the hyper-volume of the hyper-polytope in safe domain. Since majority of the computational cost is spent to the evaluations of PF, it is reasonable to use the time complexity measured by the total number of PF calls as the measure of computational cost. In the calculation of the time complexity, the representative LSS is chosen to be the total surface of the input space, while the time complexity of an arbitrary case should be on the same order as that of the representative case. For the representative case, the hyper-cube representing the m^{th} level has $2k$ faces and each face are embedded with approximately $2^{(m-1)(k-1)}$ child grids. Consequently, the total number of grids is $\sum_{m=1}^n 2k \cdot 2^{(m-1)(k-1)} \approx 2k \cdot 2^{n(k-1)}$, where n is the total number of levels. Assuming the number of PF calls a constant per grid construction, the time complexity is obtained as $O(k \cdot 2^{n(k-1)})$.

By further defining the accuracy $\varepsilon_R = \text{Const} \cdot 2^{-n}$, the time complexity becomes $O(k \cdot \varepsilon_R^{-k+1})$.

The limit-state surface resolution supervised searching algorithm

The limit-state surface resolution supervised searching algorithm is anchored by computing the structural reliability from the reliability portion insides the closed LSS with sufficient resolution, in which the LSS resolution (LSSR) is defined as the number of grids along the edge of the input space. In the

algorithm, LSSR should be set to 2^{n-1} , where n is the maximum number of levels. Hence, LSSR and the smallest grid side length Δ can be related by $\text{LSSR} = 2^{n-1} \geq C/\Delta$, where C is the side length of the input space. The workflow of this algorithm is similar to that of the reliability accuracy supervised searching algorithm, yet the last level has an enhanced search securing the LSS is closed, and the reliability is estimated from the domain insides the closed LSS. Evidently, such an enhanced search would take into account the influence of Type-3 and Type-4 LSGs, which improves the accuracy of structural reliability estimations. The stopping criterion is that: the algorithm stops when the maximum level determined by LSSR is reached. This algorithm is name after its ability of securing the resolution of the estimated LSS. The enhanced search for LSS is realized by tracing the discontinues of Type-1 and Type-2 LSGs on the last level, and the identification of domain within the LSS is realized by a recursive function. Fig. 3 (b) shows the algorithm's flowchart and the appendix presents its pseudo code, which contain the realization details of forming a closed LSS and computing the reliability portion insides a closed LSS. Similar to that of the reliability accuracy supervised searching algorithm, the computational cost of this algorithm is also around $O(k \cdot 2^{n(k-1)})$ or $O(k \cdot (2\text{LSSR})^{k-1})$, which is expected to be slightly higher than that of the reliability accuracy supervised searching algorithm. Despite that it cannot directly guarantee a prescribed accuracy, this algorithm would adapt its accuracy of structural reliability estimation through altering the LSSR. Moreover, since it forces the estimated LSS to be closed, sharp changes on the LSS will be detected, which contributes to higher accuracy.

The reliability index precision supervised fast searching algorithm

Different from the above-presented two algorithms, the reliability index precision supervised fast searching algorithm measures the RI through an accelerated searching strategy first, and then compute the structural reliability through the measured RI. The acceleration process is realized by culling out grids that absolutely do not contain the design point before constructing a new level. For instance, there are two LSGs denoted by LSG1 and LSG2, for LSG1, its max distance to the origin is l_1 , for LSG2, its min distance to the origin is l_2 . If $l_1 < l_2$, then LSG2 cannot contain the design point. Following this logic, before constructing a new level, we can first compute the LSG whose max distance to the origin is minimal of all and denoted it as the index grid (IG), then LSGs whose distances to the origin are greater than LSG whose max distance to the origin can be safely excluded. Fig.3 (c) and the appendix show the algorithm's flowchart and pseudo code respectively. The stop criterion is: the algorithm stops when the

maximum level determined by the precision of the RI is reached. The precision of the RI, denoted by ε_p , is linked to the maximum level by $2^{n-1} \geq 1/\varepsilon_p$. It should be noted that the influence of sharp changes on LSS is ignored, which is in a sense reasonable since the objective is to gauge the structural reliability than ensuring the measure RI to be the minimal distance between the origin to the LSS, and the structural reliability estimation using the minimal distance would be too conservative if there exist deep thorns on the LSS. Besides, this algorithm is attractive for its low computational cost. By assuming each level has a maximum of D number of grids to search, the total number of grids searched is at most $D \cdot n$, and the number of PF calls per grid construction is proportional to 2^k . Thus, the time complexity of this algorithm is around $O(2^k \cdot n)$ or $O(2^k \cdot \log_2(2/\varepsilon_p))$.

3.3 Numerical validation

In order to demonstrate the validity, the proposed algorithms are applied to the reliability computation of two exemplary cases, in which the two independent inputs x_1 and x_2 obey standard normal distribution, the bounds of x_1 and x_2 are respectively $[-6, 6]$ and $[-4, 4]$, the PFs are $g_1(x_1, x_2)$ and $g_2(x_1, x_2)$ for Case 1 and Case 2 respectively. $g_1(x_1, x_2)$ is the following modified version of the series system reliability problem with high nonlinearity [31]:

$$\begin{aligned} g_1 &= \min\{g_m^1, g_m^2, g_m^3, g_m^4\} \\ g_m^i &= g^i + 2\sin(g^i) \\ g^1 &= 3 + 0.1(x_1 - x_2)^2 - (x_1 + x_2)/\sqrt{2} \\ g^2 &= 3 + 0.1(x_1 - x_2)^2 + (x_1 + x_2)/\sqrt{2} \\ g^3 &= x_1 - x_2 + 7/\sqrt{2} \\ g^4 &= x_2 - x_1 + 7/\sqrt{2} \end{aligned} \quad (10)$$

To verify the proposed algorithms' ability to deal with shape changes on PFs, $g_2(x_1, x_2)$, a modified version of $g_1(x_1, x_2)$ with a deep crack, is also employed in this validation, i.e.:

$$g_2 = \begin{cases} g_2 - 5, & 12(x_1 - 0.2) > (x_1 + 2.1) \text{ and } 12(x_1 - 0.2) < -(x_1 + 2.1) \\ g_2, & \text{else} \end{cases} \quad (11)$$

Figs. 4 (a) and (b) show $g_1(x_1, x_2)$ and $g_2(x_1, x_2)$ respectively, in which the black solid curves represent the corresponding LSSs, and the non-linearity can be observed by the wave-like patterns of the contour plots. The target reliability values of these two cases are obtained from direct MCSs with a sample size equals 10^7 , which are 0.997752 and 0.997304 respectively. In order to compare the three algorithms on the same number of total levels, the accuracy threshold is set to 0.001, the LSSR is set to 256, the RI precision is set to 0.005, and the corresponding number of total levels is 8. Figs. 5 and Table 1 compare the samplings and the reliability estimation results of the three proposed algorithms, in which each algorithm is further compared with direct MCSs using the identical sample size (shown in Table 1). The

reliability estimation of the reliability index precision supervised fast searching algorithm uses the Chi-square approximation in Eq. (8). It can be seen that: 1, both the reliability accuracy supervised searching algorithm and the limit-state surface resolution supervised searching algorithm have comparable accuracy, which have the identical reliability estimation result if there is no shape changes on the LSS; 2, the reliability index precision supervised fast searching algorithm has both the lowest computational cost and the lowest accuracy, whose reliability estimation is conservative; 3, the limit-state surface resolution supervised searching algorithm has the ability to probe sharp changes on the LSS, which is therefore most accurate for the case of $g_2(x_1, x_2)$; 4, all these three algorithm can deal with LSSs with high nonlinearity, this is because algorithms employ a binary search strategy on each dimension. At this point, it is noted that the RI using the point strictly closest to the origin would yield an even worse result than that obtain by the reliability index precision supervised fast searching algorithm when sharp changes exist. For instance, the RI represented by the point strictly closest to the origin for the case of $g_2(x_1, x_2)$ equals 2.1095, and the corresponding reliability approximation is 0.8919, the error is 0.1054, which is 11.54 times larger of that of the reliability index precision supervised fast searching algorithm.

4. Engineering implementations of the proposed algorithms

In this section, the proposed algorithms are applied to the reliability evaluation of a 3-span 5-story elastic frame (shown in Figs. 6). The simulation is performed on CALFEM [47], which is a MATLAB based finite element simulation program. Altogether two cases are studied using the identical frame structure subjected to 5 horizontal loads, yet the loading configurations are different for different cases. For the beams of the frame structural, the elastics modulus is identically 2.1738×10^7 kN/m², the cross-section's area and the moment of inertia are identically 5.0606×10^{-1} m² and 2.1375×10^{-2} m⁴; for the columns, the elastics modulus is identically 2.3796×10^7 kN/m², the cross-section's area and the moment of inertia are identically 3.7303×10^{-1} m² and 2.3279×10^{-2} m⁴. The loading configurations of these two cases are demonstrated in Figs. 6. For Case 1, the 3rd to 5th stories are subjected an identical random load P_1 , 1st and 2nd stories are subjected to two random loads P_2 and P_3 ; for Case 2, each story is subjected to a different load, i.e., P_1 to P_5 . Each P_i is independent and obeys a lognormal distribution, whose means and standard deviations are listed in Table 2. Consequently, the input dimensionalities for Case 1 and Case 2 are respectively 3 and 5. The PFs for these two cases are identical, i.e.:

$$g(\mathbf{x}) = 0.06 - \Delta(\mathbf{x}) \quad (12)$$

where $\Delta(\mathbf{x})$ is the horizontal displacement of the top floor's right corner.

The target reliability values for these two cases are 0.9999982 and 0.9999966, which are obtained from direct MCSs using a total of 10^7 samples. For Case 1, the accuracy threshold is set to 0.00001, the LSSR is set to 16, the RI precision is set to 0.05, which are in accordance with 5 total levels; for Case 2, the accuracy threshold is set to 0.00005, the LSSR is set to 8, the RI precision is set to 0.1, and the total number of levels is 4. Table 3 shows the reliability estimation results via the three proposed algorithms, and Figs. 7 show the corresponding sampling results for Case 1. The RIs computed by the reliability index precision supervised fast searching algorithm are 5.2014 and 8.7763 for Case 1 and Case 2 respectively. Table 3 demonstrates that all these three algorithms are accuracy and efficient. In addition, it can be observed that: 1, the computational efficiency of the reliability accuracy supervised searching algorithm is amazing that a sample size on the order of 10^2 can reach an accuracy on the order of 10^{-7} ; 2, both the reliability accuracy supervised searching algorithm and the limit-state surface resolution supervised searching algorithm have sufficient accuracy, which is achieved by adopting the interpolation process; 3, if shape change on LSS is absent, the limit-state surface resolution supervised searching algorithm has nearly the same accuracy with the computational efficiency of the reliability accuracy supervised searching algorithm, but its computational cost is significantly higher due to the enhanced search for a closed LSS; 4, the accuracy of the reliability index precision supervised fast searching algorithm is the lowest, but its computational efficiency is the highest for Case 1; 5, the reliability accuracy supervised searching algorithm outplays the reliability index precision supervised fast searching algorithm in terms of computational efficiency in Case 2, which implies the superiority of the reliability accuracy supervised searching algorithm among these three algorithms.

5. Conclusion

The research proposes three algorithms for structural reliability estimation based on k -D tree and BFS. Details of the realizations, numerical validations and engineering examples of the proposed algorithms are provided. Furthermore, the following conclusions are drawn:

- 1, Using a k -D tree and BFS for redundant sampling reduction can significantly increase the computational efficiency of structural reliability estimation;
- 2, the interpolation process can dramatically increase the accuracy of reliability estimation;
- 3, using k -D tree and BFS for reliability estimation can guarantee a stable computational cost;

- 4, the binary search strategy can deal with PFs with high non-linearity;
- 5, the reliability accuracy can be supervised by the total reliability portion of LSGs;
- 6, computing the reliability through a closed LSS can increase the accuracy by accounting for the influence of sharp changes on LSSs;
- 7, RI can be fast obtained through a spatial binary search with a culling process.

6. Appendix

6.1 Pseudo code for the reliability accuracy supervised searching algorithm

```

Reliability=0.0
CurrentGrids=null
ChildGrids=null
Accuracy=infinity
Transfer the inputs into standard Gaussian variables and modify the performance function
ChildGridFormation(input space)
Level=1
CurrentGrids=ChildGrids
while Accuracy>AccuracyRequirement
    Accuracy=0.0
    for each CurrentGrid in CurrentGrids
        ChildGridFormation(CurrentGrid)
    end
    Level=Level+1
    CurrentGrids= ChildGrids
end
for each ChildGrid in ChildGrids
    Reliability=Reliability+ChildGrid's interpolated reliability portion
end
function ChildGridFormation(CurrentGrid)
    for each Center in AllCenters
        if Center has not been constructed
            TempNode.Coordinates= Center's coordinates
            TempNode.GValue=G(Center's coordinates)
        end
    end
    end
    for each combination of vertices and centers forming a valid ChildGrid
        if ChildGrid's type is SG, Type-2 or Type-3 LSG
            Reliability=Reliability+CurrentGrid's reliability portion
        else if ChildGrid's type is Type-1 LSG
            Accuracy=Accuracy+CurrentGrid's reliability portion
            append ChildGrid to ChildGrids
        end
    end
end

```

```

    end
end

```

6.2 Pseudo code for the limit-state surface resolution supervised searching algorithm

```

Reliability=0.0
LimiteStateSuraface=null
CurrentGrids=null
ChildGrids=null
MaxLevel=log2(Resolution)+1
Transfer the inputs into standard Gaussian variables and modify the performance function
ChildGridFormation(input space)
Level=1
CurrentGrids=ChildGrids
while Level<MaxLevel
    Resolution=0.0
    for each CurrentGrid in CurrentGrids
        ChildGridFormation(CurrentGrid)
    end
    Level=Level+1
    CurrentGrids= ChildGrids
end
LimiteStateSuraface=interpolating ChildGrids
TempGirds=all grids on the last level
FindSafeGrids(any SG in TempGrids)
Reliability= CountReliability(TempGirds)
function ChildGridFormation(CurrentGrid)
    for each Center in AllCenters
        if Center has not been constructed
            TempNode.Coordinates= Center's coordinates
            TempNode.GValue=G(Center's coordinates)
        end
    end
    end
    for each combination of vertices and centers forming a valid ChildGrid
        if ChildGrid's type is Type-1 or Type-2 LSG
            Resolution=Resolution+CurrentGrid's reliability portion
            append ChildGrid to ChildGrids
        end
    end
    end
    while (ChildGrids cannot form a closed hyper-surface)
        for each TrailGrid adjacent to the discontinuities
            if TrailGrid's type is type 1 or type-2 LSG
                append TrailGrid to ChildGrids
            end
        end
    end
end
end

```

```

end
function FindSafeGrids(seed)
    for each TempGrid in seed's neighbor grid
        if TempGrid's type is empty
            TempGrid's type = SG
            FindSafeGrids(TempGrid)
        end
    end
end
end
function CountReliability(TempGrids)
    for each TempGrid in TempGrids
        if TempGrid's type is SG or type-2 LSG
            Reliability = Reliability + TempGrid's reliability portion
        else if TempGrid's type is type-1 LSG
            Reliability = Reliability + TempGrid's interpolated reliability portion
        end
    end
end
end

```

6.3 Pseudo code for the reliability index precision supervised fast searching algorithm

```

Reliability=0.0
ReliabilityIndex=infinity
MaxReliabilityIndex=infinity
IndexGrid=null
CurrentGrids=null
ChildGrids=null
MaxLevel=round(log2(1/Precision))+1
Transfer inputs into standard Gaussian variables
ChildGridFormation(input space)
Level=1
CurrentGrids=ChildGrids
while Level<MaxLevel
    for each CurrentGrid in CurrentGrids
        ChildGridFormation(CurrentGrid)
    end
    Level=Level+1
    IndexGrid = the grid in CurrentGrids whose max distance to the origin is the minimum
    MaxReliabilityIndex = IndexGrid's max distance to the origin
    for each CurrentGrid in CurrentGrids
        if CurrentGrid's min distance to the origin > MaxReliabilityIndex
            delete CurrentGrid from CurrentGrids
        end
    end
    CurrentGrids= ChildGrids
end
end

```

```

Finalize ReliabilityIndex by interpolating ChildGrids
Compute the reliability from ReliabilityIndex
function ChildGridFormation(CurrentGrid)
    for each Center in AllCenters
        if Center has not been constructed
            TempNode.Coordinates= Center's coordinates
            TempNode.GValue=G(Center's coordinates)
        end
    end
    for each combination of vertices and centers forming a valid ChildGrid
        if ChildGrid's type is type 1 or type-2 LSG
            append ChildGrid to ChildGrids
        end
    end
end
end

```

Acknowledgement

The authors wish to acknowledge the financial supports provided by the National Natural Science Foundation of China (No. 52178301) and the Science Research Foundation of Wuhan Institute of Technology (K2021030).

Reference

- [1] R.E. Melchers, A.T. Beck, Structural reliability analysis and prediction, John Wiley & sons 2018.
- [2] Z.P. Bazant, J.-L. Le, Probabilistic mechanics of quasibrittle structures: strength, lifetime, and size effect, Cambridge University Press 2017.
- [3] Z.P. Bazant, S.-D. Pang, Mechanics-based statistics of failure risk of quasibrittle structures and size effect on safety factors, Proceedings of the National Academy of Sciences 103(25) (2006) 9434-9439.
- [4] F.W. Zok, On weakest link theory and Weibull statistics, Journal of the American Ceramic Society 100(4) (2017) 1265-1268.
- [5] Z. Xu, R. Ballarini, J.-L. Le, A renewal weakest-link model of strength distribution of polycrystalline silicon MEMS structures, Journal of Applied Mechanics 86(8) (2019) 081005.
- [6] S.-D. Pang, Z.P. Bazant, J.-L. Le, Statistics of strength of ceramics: finite weakest-link model and necessity of zero threshold, International journal of fracture 154(1-2) (2008) 131-145.
- [7] W. Luo, Z.P. Bazant, Fishnet model for failure probability tail of nacre-like imbricated lamellar materials, Proceedings of the National Academy of Sciences 114(49) (2017) 12900-12905.
- [8] Z. Xu, J.-L. Le, A first passage based model for probabilistic fracture of polycrystalline silicon MEMS structures, Journal of the Mechanics and Physics of Solids 99 (2017) 225-241.
- [9] Z. Xu, J.-L. Le, On power-law tail distribution of strength statistics of brittle and quasibrittle structures, Engineering Fracture Mechanics 197 (2018) 80-91.
- [10] O. Ditlevsen, H.O. Madsen, Structural reliability methods, Wiley New York 1996.
- [11] H. Zhang, Interval importance sampling method for finite element-based structural reliability assessment under parameter uncertainties, Structural Safety 38 (2012) 1-10.
- [12] H. Dai, H. Zhang, W. Wang, G. Xue, Structural reliability assessment by local approximation of

- limit state functions using adaptive Markov chain simulation and support vector regression, *Computer - Aided Civil and Infrastructure Engineering* 27(9) (2012) 676-686.
- [13] D. Straub, I. Papaioannou, Bayesian updating with structural reliability methods, *Journal of Engineering Mechanics* 141(3) (2015) 04014134.
- [14] W.-L. Loh, On Latin hypercube sampling, *The annals of statistics* 24(5) (1996) 2058-2080.
- [15] V.L. Parsons, Stratified sampling, *Wiley StatsRef: Statistics Reference Online* (2014) 1-11.
- [16] M.D. Shields, J. Zhang, The generalization of Latin hypercube sampling, *Reliability Engineering & System Safety* 148 (2016) 96-108.
- [17] R.E. Caflisch, Monte carlo and quasi-monte carlo methods, *Acta numerica* 7 (1998) 1-49.
- [18] Y.-G. Zhao, T. Ono, Moment methods for structural reliability, *Structural safety* 23(1) (2001) 47-75.
- [19] A. Haldar, S. Mahadevan, First-order and second-order reliability methods, *Probabilistic structural mechanics handbook*, Springer 1995, pp. 27-52.
- [20] Y.-G. Zhao, Z.-H. Lu, Fourth-moment standardization for structural reliability assessment, *Journal of Structural Engineering* 133(7) (2007) 916-924.
- [21] H.P. Gavin, S.C. Yau, High-order limit state functions in the response surface method for structural reliability analysis, *Structural safety* 30(2) (2008) 162-179.
- [22] C. Wang, H. Zhang, Q. Li, Moment-based evaluation of structural reliability, *Reliability Engineering & System Safety* 181 (2019) 38-45.
- [23] A. Der Kiureghian, T. Dakessian, Multiple design points in first and second-order reliability, *Structural Safety* 20(1) (1998) 37-49.
- [24] Y.-G. Zhao, T. Ono, Third-moment standardization for structural reliability analysis, *Journal of Structural Engineering* 126(6) (2000) 724-732.
- [25] S.S. Afshari, F. Enayatollahi, X. Xu, X. Liang, Machine learning-based methods in structural reliability analysis: A review, *Reliability Engineering & System Safety* 219 (2022) 108223.
- [26] J.E. Hurtado, *Structural reliability: statistical learning perspectives*, Springer Science & Business Media 2004.
- [27] H. Linxiong, L. Huacong, P. Kai, X. Hongliang, A novel kriging based active learning method for structural reliability analysis, *Journal of Mechanical Science & Technology* 34(4) (2020).
- [28] H.-s. Li, Z.-z. Lü, Z.-f. Yue, Support vector machine for structural reliability analysis, *Applied Mathematics and Mechanics* 27(10) (2006) 1295-1303.
- [29] W.J. de Santana Gomes, Structural reliability analysis using adaptive artificial neural networks, *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg* 5(4) (2019).
- [30] M.M. Zuniga, A. Murangira, T. Perdrizet, Structural reliability assessment through surrogate based importance sampling with dimension reduction, *Reliability Engineering & System Safety* 207 (2021) 107289.
- [31] Z. Xiang, Y. Bao, Z. Tang, H. Li, Deep reinforcement learning-based sampling method for structural reliability assessment, *Reliability Engineering & System Safety* 199 (2020) 106901.
- [32] K. Cheng, Z. Lu, Structural reliability analysis based on ensemble learning of surrogate models, *Structural Safety* 83 (2020) 101905.
- [33] S. Marelli, B. Sudret, An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis, *Structural Safety* 75 (2018) 67-74.
- [34] P. Huang, H.-Z. Huang, T. Huang, A novel algorithm for structural reliability analysis based on finite step length and Armijo line search, *Applied Sciences* 9(12) (2019) 2546.
- [35] M.A. Shayanfar, M.A. Barkhordari, M. Barkhori, M. Barkhori, An adaptive directional importance

- sampling method for structural reliability analysis, *Structural Safety* 70 (2018) 14-20.
- [36] J.-X. Gong, P. Yi, A robust iterative algorithm for structural reliability analysis, *Structural and Multidisciplinary Optimization* 43(4) (2011) 519-527.
- [37] S. Katsuki, D.M. Frangopol, Hyperspace division method for structural reliability, *Journal of Engineering Mechanics* 120(11) (1994) 2405-2427.
- [38] C. Zhong, M. Wang, C. Dang, W. Ke, Structural reliability assessment by salp swarm algorithm-based FORM, *Quality and Reliability Engineering International* 36(4) (2020) 1224-1244.
- [39] C. Elegbede, Structural reliability assessment based on particles swarm optimization, *Structural safety* 27(2) (2005) 171-186.
- [40] Z. Jing, J. Chen, X. Li, RBF-GA: An adaptive radial basis function metamodeling with genetic algorithm for structural reliability analysis, *Reliability Engineering & System Safety* 189 (2019) 42-57.
- [41] J. Cheng, Hybrid genetic algorithms for structural reliability analysis, *Computers & Structures* 85(19-20) (2007) 1524-1533.
- [42] L. Deng, M. Ghosn, S. Shao, Development of a shredding genetic algorithm for structural reliability, *Structural safety* 27(2) (2005) 113-131.
- [43] M.A. Shayanfar, M.A. Barkhordari, M.A. Roudak, A new effective approach for computation of reliability index in nonlinear problems of reliability analysis, *Communications in Nonlinear Science and Numerical Simulation* 60 (2018) 184-202.
- [44] J.-x. Gong, P. Yi, N. Zhao, Non-gradient-based algorithm for structural reliability analysis, *Journal of Engineering Mechanics* 140(6) (2014) 04014029.
- [45] Q. Xiao, Evaluating correlation coefficient for Nataf transformation, *Probabilistic Engineering Mechanics* 37 (2014) 1-6.
- [46] A.M. Hasofer, N.C. Lind, Exact and invariant second-moment code format, *Journal of the Engineering Mechanics division* 100(1) (1974) 111-121.
- [47] M. Ristinmaa, G. Sandberg, K.-G. Olsson, CALFEM as a tool for teaching university mechanics, *International journal of innovation in science and mathematics education* 5(1) (2000).

Figures

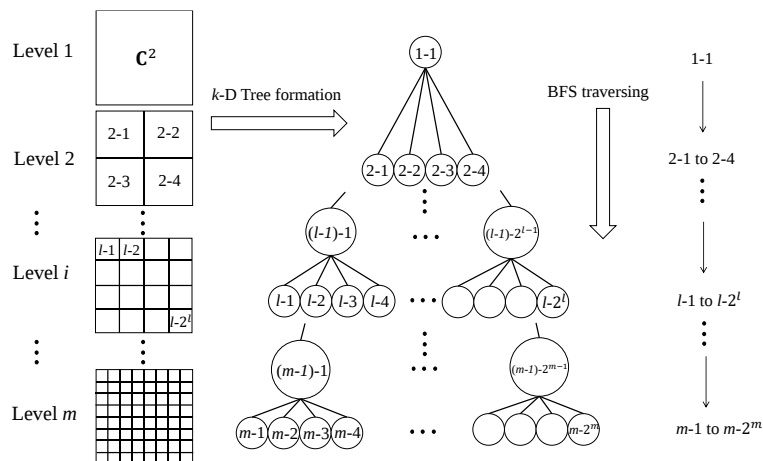


Figure 1. Illustration on k-D Tree construction and BFS traversing.

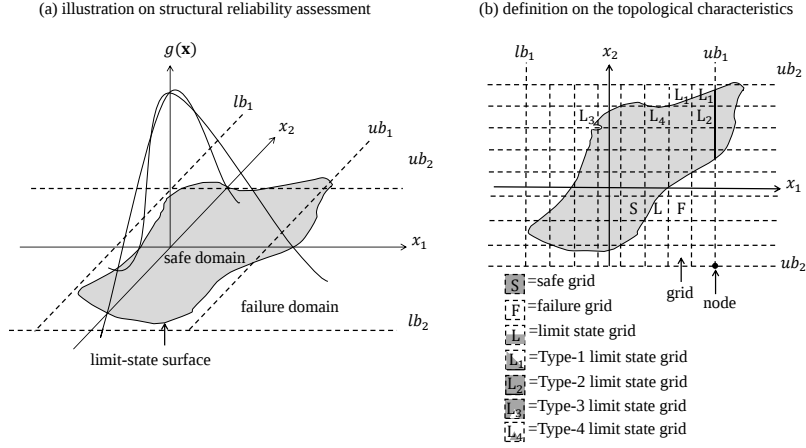
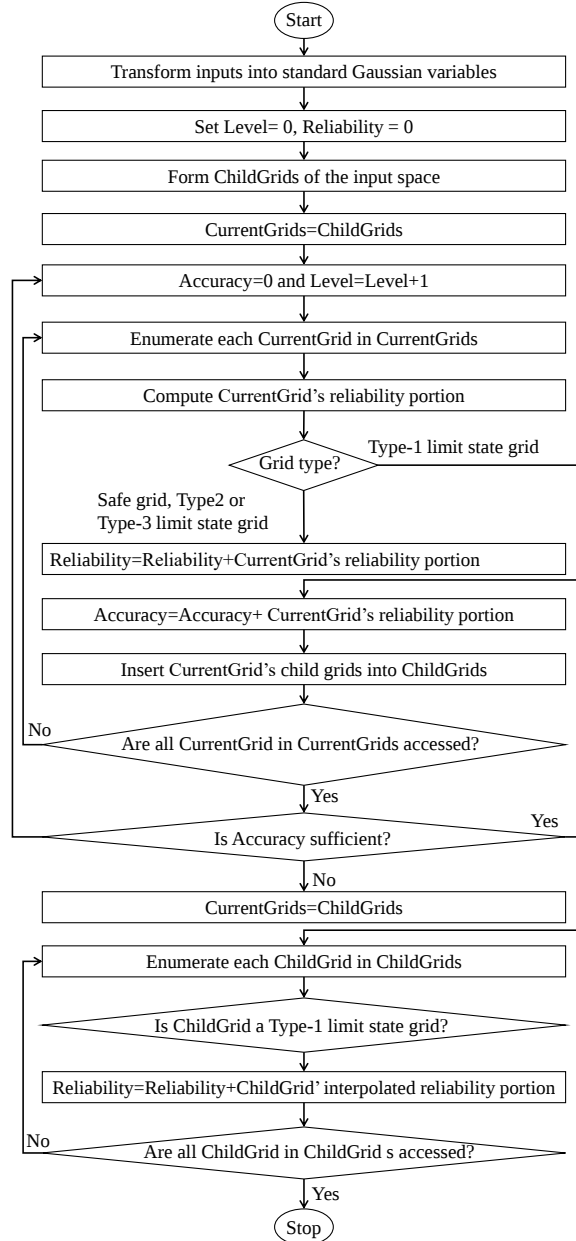
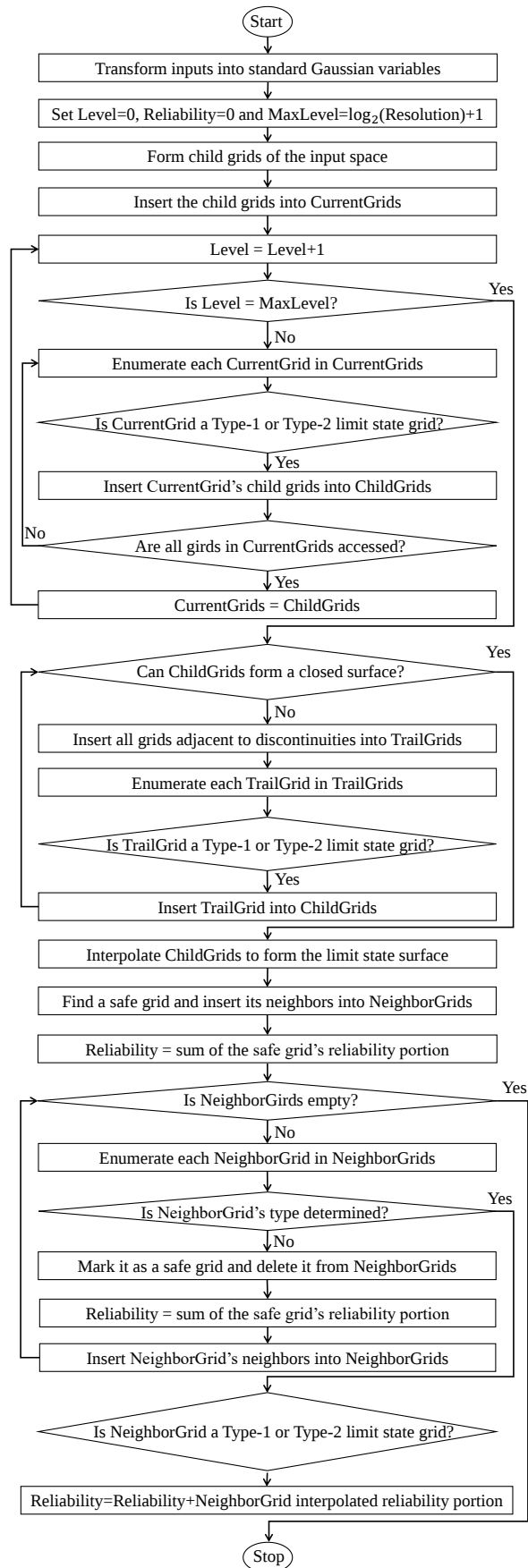


Figure 2. Illustration on the problem statement: (a) configuration of structural reliability estimation, (b) definition on the topological characteristics.

(a) the reliability accuracy supervised searching algorithm



(b) the limit-state surface resolution supervised searching algorithm



(c) the reliability index precision supervised fast searching algorithm

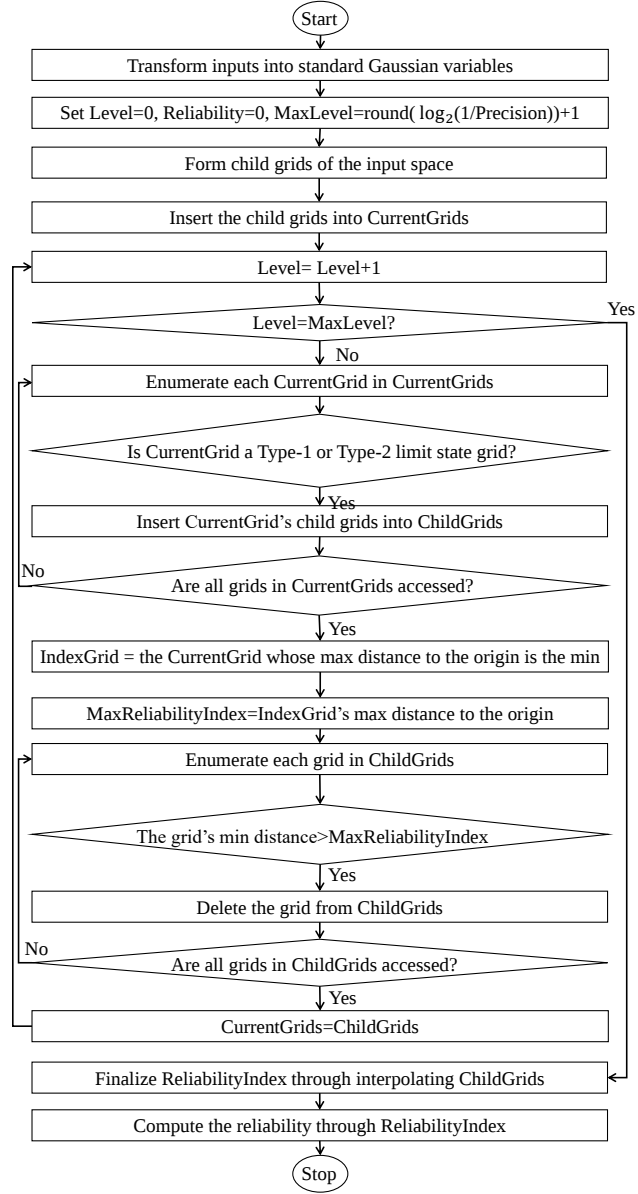


Figure 3. Flowchart of the proposed algorithms: (a) the reliability accuracy supervised searching algorithm, (b) the limit-state surface resolution supervised searching algorithm, (c) the reliability index precision supervised fast searching algorithm.

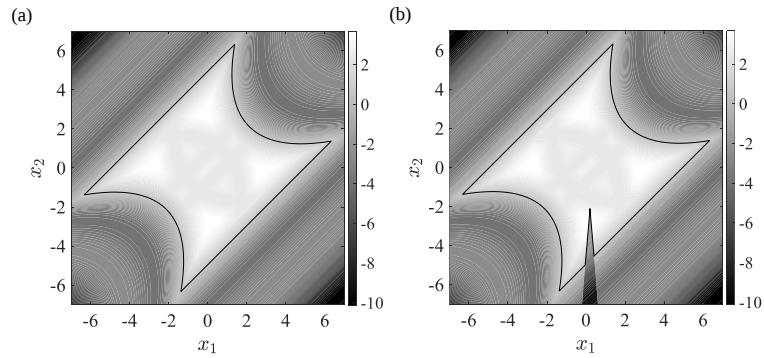


Figure 4. The exemplary performance function and the corresponding limit-state surface: (a) g_1 , (b) g_2 .

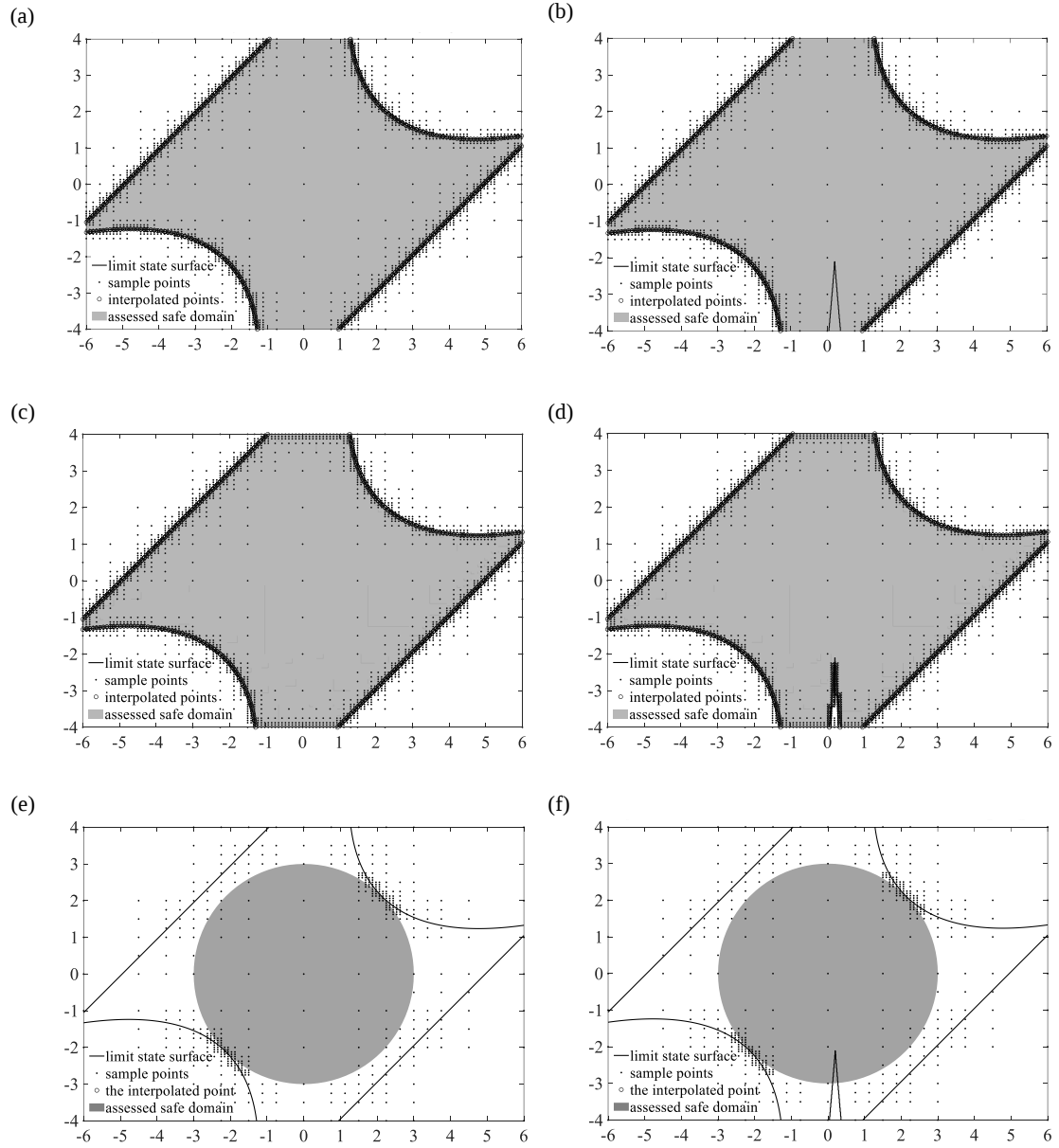


Figure 5. Samplings of the structural reliability assessments: (a) the reliability accuracy supervised searching algorithm using g_1 , (b) the reliability accuracy supervised searching algorithm using g_2 , (c) the limit-state surface resolution supervised searching algorithm using g_1 , (d) the limit-state surface resolution supervised searching algorithm using g_2 , (e) the reliability index precision supervised fast searching algorithm using g_1 , (f) the reliability index precision supervised fast searching algorithm using g_2 .

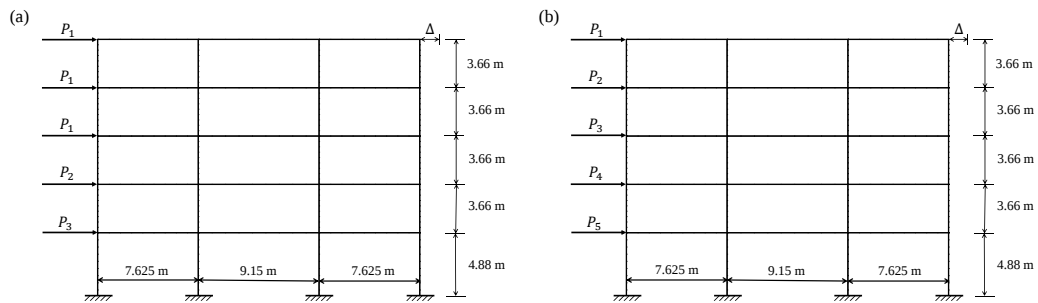


Figure 6. Configurations of the 3-span 5-story frames: (a) Case 1, (b) Case 2.

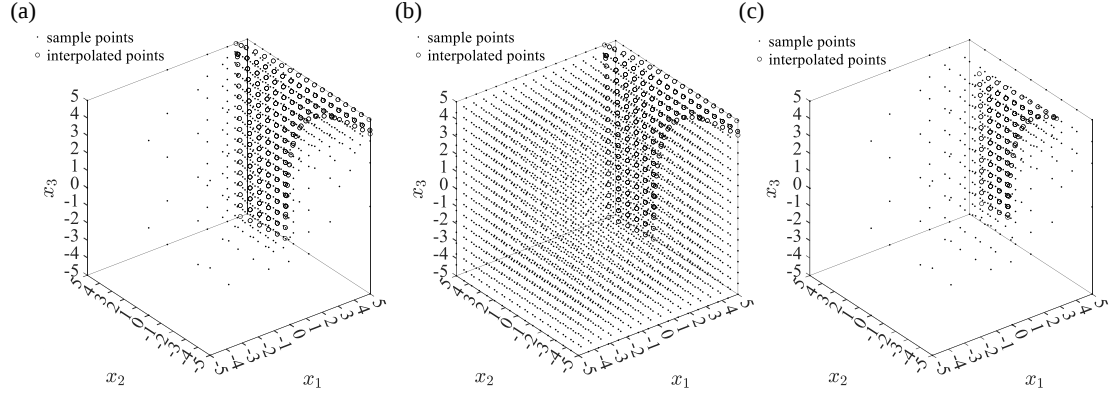


Figure 7. Samplings of structural reliability assessments of Case 1: (a) the reliability accuracy supervised searching algorithm, (b) the reliability accuracy supervised searching algorithm, (c) the limit-state surface resolution supervised searching algorithm.

Tables

Table 1. Reliability estimation result and comparison of the exemplary cases.

Algorithm	PF	Sample size	Error	Mean error using MCS
the reliability accuracy supervised searching algorithm	g_1	1813	0.267×10^{-4}	9.01×10^{-4}
the limit-state surface resolution supervised searching algorithm	g_1	1957	0.267×10^{-4}	8.55×10^{-4}
the reliability index precision supervised fast searching algorithm	g_1	459	88.5×10^{-4}	17.3×10^{-4}
the reliability accuracy supervised searching algorithm	g_2	1813	4.26×10^{-4}	9.49×10^{-4}
the limit-state surface resolution supervised searching algorithm	g_2	2047	1.54×10^{-4}	9.37×10^{-4}
the reliability index precision supervised fast searching algorithm	g_2	459	84.0×10^{-4}	19.2×10^{-4}

Table 2. Description of the random inputs.

Case number	Variable	Distribution	Mean	Standard deviation
1	P_1 , kN	lognormal	133.454	40.04
1	P_2 , kN	lognormal	88.97	35.59
1	P_3 , kN	lognormal	71.175	28.47
2	P_1 , kN	lognormal	133.454	40.04
2	P_2 , kN	lognormal	133.454	40.04
2	P_3 , kN	lognormal	133.454	40.04
2	P_4 , kN	lognormal	88.97	35.59
2	P_5 , kN	lognormal	71.175	28.47

Table 3. Reliability estimation result and comparison of the engineering example.

Algorithm	Case number	Sample size	Error
the reliability accuracy supervised searching algorithm	1	612	4.75×10^{-8}
the limit-state surface resolution supervised searching	1	3681	4.75×10^{-8}
the reliability index precision supervised fast searching	1	422	3.93×10^{-6}
the reliability accuracy supervised searching algorithm	2	746	5.33×10^{-7}
the limit-state surface resolution supervised searching	2	58837	5.33×10^{-7}
the reliability index precision supervised fast searching	2	2525	3.40×10^{-6}