

**Article info****Type of article:**

Original research paper

DOI:

<https://doi.org/10.58845/jstt.utt.2026.en.6.5.18-32>

***Corresponding author:**

Email address:

quan.lieuminh@hcmut.edu.vn

Received: 13/04/2026

Received in Revised Form:
29/07/2026

Accepted: 15/08/2026

An efficient hybrid framework for reliability-based size and shape optimization of trusses considering uncertainties

Anh Hoang Nguyen^{1,2}, Qui Xuan Lieu^{1,2}, Van Quan Tran³, Quan Minh Lieu^{1,2,*}

¹Faculty of Civil Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, Dien Hong Ward, Ho Chi Minh City 700000, Vietnam

²Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Xuan Ward, Ho Chi Minh City 700000, Vietnam

³AMS4 Research Group (Advanced Materials and Smart Structure for Sustainable System), University of Transport Technology, 54 Trieu Khuc Street, Thanh Liet Ward, Hanoi 100000, Vietnam

Abstract: Reliability-based design optimization (RBDO) of truss structures has become a powerful tool for achieving optimal designs under uncertainties in material properties, cross-sectional areas, and applied loadings. By integrating the sequential optimization and reliability assessment (SORA) into an efficient optimization tool based on the success-history-based adaptive differential evolution (SHADE) algorithm, this paper presents a novel RBDO methodology, which is the so-called SORA-SHADE, for the optimization of shape and size in truss structures under uncertainties of input data. The proposed paradigm is tested on well-known 2D and 3D trusses with and without the reliability analyses for validation. For reliability-based analysis cases, each cross-sectional area is considered as a random variable under uncertainties. The results show that the proposed approach is highly effective and reliable in finding optimal designs that meet high reliability requirements.

Keywords: Reliability-based design optimization, Uncertainties, Reliability analyses, Truss optimization, Success-history-based adaptive differential evolution (SHADE), Sequential optimization and reliability assessment (SORA).

1. Introduction

Truss structures hold fundamental significance within the discipline of civil engineering, serving diverse functions in roofs, transmission towers, and bridge supports. Given their critical importance, the structural optimization of trusses has attracted substantial academic interest. Many authors have applied metaheuristic algorithms to optimize truss structures with continuous design variables. Interested readers

can refer to [1–12] for more details.

Among these non-gradient-based techniques, Differential Evolution (DE) [13] stands out as a preeminent paradigm. Recognized for its exceptional ability to identify high-quality solutions with remarkable stability, DE has been extensively adopted across diverse engineering applications [14–16]. Nevertheless, owing to the inherent nature of meta-heuristic techniques, the standard DE algorithm still demands significant computational

resources to secure a globally optimal solution. Consequently, a vast array of modified and hybrid DE variants has been developed to mitigate this issue, as documented in several studies [17–24]. In particular, another advanced metaheuristic approach, known as Success-History based Adaptive DE (SHADE) [25], presents a novel adaptive DE algorithm that improves upon JADE [8] by employing a distinct, history-based parameter adaptation mechanism. Specifically, the average values of successful control parameters, CR and F, from each generation are recorded in historical memories, M_{CR} and M_F . By maintaining a diverse parameter set, SHADE dynamically guides the adaptation process, ensuring a more robust and resilient search. Due to these superior traits, SHADE has seen widespread implementation throughout the engineering sector.

Conventionally, engineering relies on deterministic design optimization (DDO) to identify the best solution by operating at the strict boundaries of problem constraints [26]. However, real-world engineering involves numerous unpredictable variables, including material characteristics, dimensions, boundary conditions, and applied loads. As a result, a deterministic optimal design often carries a high risk of failure when these constraints are challenged [27]. Thus, reliability-based design optimization (RBDO) methodologies have been developed to overcome this challenge [28–32]. By effectively balancing safety and cost efficiency, RBDO yields designs that remain both secure and economical despite inherent system uncertainties [33–35].

Because they rely heavily on gradients, analytical methods typically offer better efficiency than simulation-based ones for specific RBDO cases, for instance, linear systems involving normally distributed random variables. Prominent examples are the moment matching method [36] and Most Probable Point (MPP) methods [37, 38], with the reliability index approach (RIA) and performance measurement approach (PMA) being

the most prominent MPP techniques. RIA simplifies RBDO into a basic DDO problem by using FORM and SORM to convert reliability constraints into reliability index constraints [39, 40]. PMA, proven by Lee et al. to be a more efficient and stable alternative to RIA [41], lowers computational demands by solving an inverse MPP optimization to turn reliability constraints into deterministic ones [42, 43]. Additionally, the SORA method introduced by Du et al. further improves efficiency by splitting each iteration into two steps: optimizing modified constraints and finding inverse MPPs [33, 44]. This strategy effectively decouples the traditional two-level RBDO into consecutive single-level tasks, drastically reducing calculation time.

In this article, a novel RBDO approach is introduced by integrating SORA and SHADE, namely SORA-SHADE, to optimize the size and shape of trusses. The proposed method takes advantage of the robust and resilient search ability of SHADE to find the best solution no matter where the calculation starts, while SORA reduces the computational cost significantly in finding MPPs thanks to the decoupling strategy. Well-known 2D and 3D trusses are applied to examine the effectiveness of SORA-SHADE by comparing results with and without structural reliability analysis. Statistical findings are presented to provide baseline reference solutions for future comparative studies.

The rest of this article is structured as follows. Section 2 summarizes the theoretical background of the SHADE and SORA. Section 3 presents the workflow of the proposed method. Section 4 then illustrates two numerical examples to validate its accuracy and efficiency. Finally, Section 5 ends with conclusions drawn from numerical investigations.

2. Theoretical background

2.1. Success-History-based Adaptive Differential Evolution

The SHADE [25] is an adaptive variant of the classical DE approach that employs a historical

memory mechanism to automatically tune the control parameters F and CR throughout the optimization process. The overall procedure consists of six fundamental steps as follows

Initialization phase

Firstly, the algorithm maintains a population of NP individuals, where the i th individual ($i = 1, 2, \dots, NP$) denotes a vector of d design variables initialized as

$$\begin{aligned} x_{i,j}^{t=0} &= x_{\min,j} + \text{rand}_{i,j}[0,1] (x_{\max,j} - x_{\min,j}), \\ j &= 1, 2, \dots, d, \end{aligned} \quad (1)$$

where $\text{rand}_{i,j}[0,1]$ denotes a uniformly distributed random number within $[0,1]$; $x_{\min,j}$ and $x_{\max,j}$ stand for the lower and upper bounds of $x_{i,j}$, respectively.

In addition, SHADE maintains two historical memory arrays $\mathbf{M}_{CR}$ and $\mathbf{M}_F$, each of size H , initialized as

$$\begin{aligned} M_{CR,h} &= M_{F,h} = 0.5, \\ h &= 1, 2, \dots, H. \end{aligned} \quad (2)$$

Parameter generation

In every iteration, the control parameters CR_i and F_i for individual $\mathbf{x}_i$ are determined by initially choosing a random index $r_i \in [1, H]$, subsequently drawing

$$CR_i = \text{rand}_N(M_{CR,r_i}, 0.1), \quad (3)$$

$$F_i = \text{rand}_C(M_{F,r_i}, 0.1), \quad (4)$$

where $\text{rand}_N(\mu, \sigma^2)$ and $\text{rand}_C(\mu, s)$ denote values drawn from the normal and Cauchy distributions with mean μ and scale parameter 0.1 , respectively. Any CR_i value outside $[0,1]$ is clipped to the nearest bound; $F_i > 1$ is truncated to 1, while $F_i \leq 0$ triggers resampling until a valid value is obtained.

Mutation phase

A mutant vector $\mathbf{v}_i^t$ is created using the *current-to-pbest/1* strategy

$$\mathbf{v}_i^t = \mathbf{x}_i^t + F_i (\mathbf{x}_{\text{pbest}}^t - \mathbf{x}_i^t) - F_i (\mathbf{x}_{r_1}^t - \tilde{\mathbf{x}}_{r_2}^t) \quad (5)$$

where $\mathbf{x}_{\text{pbest}}^t$ is randomly chosen from the top

$[p_i \times N]$ individuals of the current population, with $p_i = \text{rand}[p_{\min}, 0.2]$ and $p_{\min} = 2/N$; $\mathbf{x}_{r_1}^t$ is randomly chosen from the current population; and $\mathbf{x}_{r_2}^t$ is randomly drawn from the union $\mathbf{P} \cup \mathbf{A}$, with all indices mutually distinct and different from i . Individuals replaced in the selection step are preserved in $\mathbf{A}$; whenever $|\mathbf{A}| > N$, randomly selected elements are removed to maintain $|\mathbf{A}| \leq N$

If any component $v_{i,j}^t$ violates its bounds, it is corrected as

$$v_{i,j}^t = \begin{cases} \frac{x_{\min,j} + x_{i,j}^t}{2} & \text{if } v_{i,j}^t < x_{\min,j}, \\ \frac{x_{\max,j} + x_{i,j}^t}{2} & \text{if } v_{i,j}^t > x_{\max,j}, \end{cases} \quad (6)$$

Crossover phase

The trial vector $\mathbf{u}_i^t = \{u_{i,1}^t, u_{i,2}^t, \dots, u_{i,d}^t\}$ is formed

by mixing $\mathbf{x}_i^t$ and $\mathbf{v}_i^t$ via binomial crossover

$$u_{i,j}^t = \begin{cases} v_{i,j}^t & \text{if } \text{rand}_{i,j}[0,1] \leq CR_i \text{ or } j = j_{\text{rand}}, \\ x_{i,j}^t & \text{otherwise,} \end{cases} \quad (7)$$

where j_{rand} represents a randomly selected index from $[1, d]$ that ensures that at least one dimension is always inherited from the mutant vector.

Selection phase

Each trial vector $\mathbf{u}_i^t$ is compared against its parent $\mathbf{x}_i^t$. The better individual survives to the next generation

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{u}_i^t & \text{if } f(\mathbf{u}_i^t) \leq f(\mathbf{x}_i^t), \\ \mathbf{x}_i^t & \text{otherwise.} \end{cases} \quad (8)$$

Whenever $f(\mathbf{u}_i^t) \leq f(\mathbf{x}_i^t)$, the successful CR_i and F_i values are stored in sets S_{CR} and S_F , respectively, together with the fitness improvement $\Delta f_i = f(\mathbf{u}_i^t) - f(\mathbf{x}_i^t)$.

Memory update

At the end of each generation, if $S_{CR} \neq \emptyset$ and $S_F \neq \emptyset$, the k th memory entries are updated as

$$M_{CR,k}^{t+1} = \text{mean}_{WA}(S_{CR}) = \sum_i w_i \times S_{CR,i}, \quad (9)$$

$$M_{F,k}^{t+1} = \text{mean}_{WL}(S_F) = \frac{\sum_i w_i \times S_{F,i}^2}{\sum_i w_i \times S_{F,i}}, \quad (10)$$

where the weight $w_i = \frac{\Delta f_i}{\sum_j \Delta f_j}$ reflects the relative

fitness improvement of each successful individual. The index k is then incremented; if $k > H$, it is reset to 1. If no improvement occurs in a generation, the memory remains unchanged.

The iterative process terminates when the convergence criterion $\delta = |f_{\text{mean}} / f_{\text{best}} - 1| \leq 10^{-6}$ is satisfied, or the maximum number of iterations $t_{\text{max}} = 500$ is reached [9].

2.2. Sequential Optimization and Reliability Assessment

A typical model of a reliability-based design optimization (RBDO) problem is given by

$$\begin{aligned} &\text{Minimize:} && f(\mathbf{d}, \mathbf{X}, \mathbf{P}), \\ &\text{Design Variables:} && \mathbf{DV} = \{\mathbf{d}, \boldsymbol{\mu}_x\}, \\ &\text{Subject to:} && \text{Prob}\{g_i(\mathbf{d}, \mathbf{X}, \mathbf{P}) \leq 0\} \geq R_i, \\ &&& i = 1, 2, \dots, m, \end{aligned} \quad (11)$$

where $f(\mathbf{d}, \mathbf{X}, \mathbf{P})$ denotes the objective function; $\mathbf{d}$ and $\mathbf{X}$ represent the vectors of deterministic and random design variables, respectively; $\mathbf{P}$ signifies the vector of random parameters; $g_i(\mathbf{d}, \mathbf{X}, \mathbf{P})$ designates the constraint functions; $R_i = \Phi(\beta_i)$ indicate the desired probabilities of constraint satisfaction; $\Phi(\bullet)$ denotes the cumulative distribution function (CDF) of the standard normal distribution; β_i denotes the target reliability index; $\boldsymbol{\mu}_x$ stands for the average vector of the random design variables $\mathbf{X}$.

The fundamental principle of the SORA lies in adjusting the failed probabilistic constraint boundaries to the viable region using reliability data gathered from the previous iteration. Subsequently, these probabilistic constraints are converted into equivalent deterministic

formulations. This allows the design to improve step-by-step while saving a massive amount of computing time. Further comprehensive details can be found in [33].

2.3. Constraint treatment

The penalty function approach is adopted to convert constrained problems into corresponding unconstrained ones. Accordingly, the constrained optimization problem formulated in Eq. (11) can be reformulated as follows

$$\text{Min: } f_{\text{penalty}} = (1 + \varepsilon_1 v)^{\varepsilon_2} f(\mathbf{d}, \mathbf{X}, \mathbf{P}), \quad (12)$$

in which

$$v = \sum_{i=1}^{i,s} \max\{0, g_{i,s}(\mathbf{d}, \mathbf{X}, \mathbf{P})\}, \quad (13)$$

where v is the whole number of violated constraints; $g_{i,s}(\mathbf{d}, \mathbf{X}, \mathbf{P})$ refers to the s th constraint within the i th constraint group. It is important to emphasize that the above penalty function is applied to the equivalent deterministic optimization problem after using the SORA approach to convert probabilistic constraints into deterministic ones.

In order to gain better optimal solutions, the dynamic approach is applied to select the remaining penalty parameters. Accordingly, ε_1 and ε_2 are assigned the values 1 and 1.5, respectively. In which, ε_2 steadily increases by 0.05 after each iteration until it reaches 3.

3. The SORA-SHADE framework

This section introduces the proposed hybrid approach termed SORA-SHADE, a novel hybrid method combining the SORA and SHADE algorithms. The step-by-step process of this method is shown in the flowchart in Fig. 1.

Since there is no MPP information in the first iteration, the starting points $\mathbf{x}_{\text{MPP}}$ and $\mathbf{p}_{\text{MPP}}$ are simply set to the average values of the random variables and parameters. Consequently, the initial shift is set to zero $\mathbf{s}_0 = \mathbf{0}$, and the SHADE algorithm is employed to tackle the deterministic optimization problem without considering probabilistic

constraints. Following this step, as specific constraints become passive, the resulting optimal point $\mu_x^{(1)} = \{\mu_{x_1}^{(1)}, \mu_{x_2}^{(1)}, \dots, \mu_{x_n}^{(1)}\}$ lies exactly on the boundaries of these deterministic constraint limits. Subsequently, the inverse MPP search method is utilized to identify the MPPs that meet the required

reliability. Finally, the shifting vector s_i for the i th constraint is determined as

$$s_i = (s_1, s_2, \dots, s_n) = \mu_x^{(1)} - x_{MPP}^i, \quad i = 1, 2, \dots, m. \quad (14)$$

The updated constraint boundary for the next deterministic optimization phase is defined as

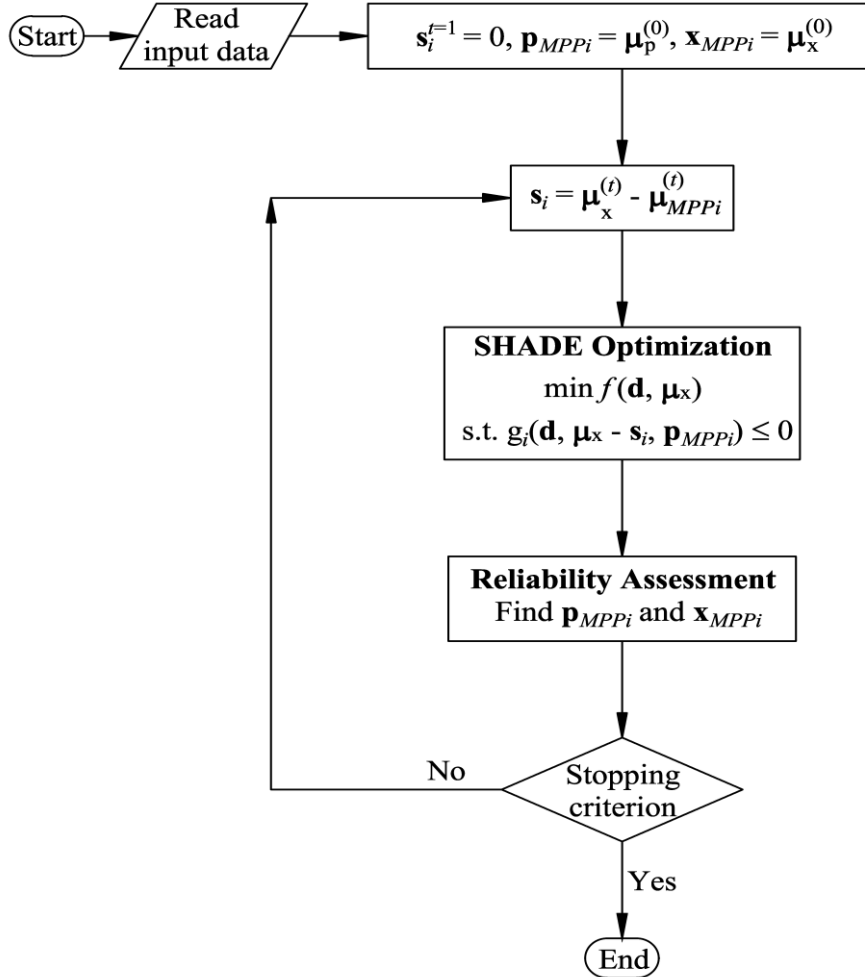


Fig. 1. The SORA-SHADE flowchart

$$g_i(\mu_x - s_i) \leq 0. \quad (15)$$

This process requires shifting the design variable location μ_x in the next cycle to meet the probabilistic constraints. As a result, the feasible design space becomes progressively smaller, but the design becomes safer and more reliable. These iterations terminate when the stopping criteria are met.

4. Numerical examples

In this part, three well-known trusses related to size and shape optimization are investigated

considering uncertainties for cross-sectional area variables A_i , i.e., $P \sim N(A_i, (0.05A_i)^2)$. The natural frequency constraints are applied to probabilistic ones. In addition, the population size NP and memory size H are set to 20. These parameters were carefully selected using a trial-and-error method. However, they were not reported in this work for a concise presentation. Besides, 10 independent runs were conducted for each case examined. Note that the above parameters are very carefully examined, but they are not reported

for a succinct presentation of a technical paper. Subsequently, the statistical results, such as the optimal weight and its corresponding number of objective function evaluations (nFEAs), the worst and the mean weight, standard deviation, the total computational time (CPUtime), and reliability index, are documented. In order to demonstrate the efficiency of the SORA-SHADE, a comparison of the solutions obtained by the proposed method with other optimization algorithms such as DE and PSO is implemented for all benchmark problems. Note that both DE and PSO are integrated with SORA to provide the results for comparative

purposes in this study, i.e., SORA-DE and SORA-PSO. The linear two-node truss member [45] is utilized for FE analysis. Lastly, Table 1 provides material properties and the restricted range of area variables for truss design, while the probability constraints on natural frequencies with the applied reliability indices are summarized in Table 2. All simulations were implemented in Python 3.10 on a laptop computer with an Intel Core™ i7-1360P CPU @ 2.20GHz, 16.0 GB RAM of memory, and Windows 11 Professional with a 64-bit operating system.

4.1. 10-bar planar truss

Table 1. Material properties design for truss structures

Example	Young's modulus E (Pa)	Material density ρ (kg/m ³)	Bounds of cross-sectional areas (m ²)
10-bar truss	6.98×10^{10}	2770	$0.645 \times 10^{-4} \leq A$
37-bar truss	2.1×10^{11}	7800	$1 \times 10^{-4} \leq A \leq 10 \times 10^{-4}$
52-bar truss	2.1×10^{11}	7800	$1 \times 10^{-4} \leq A \leq 10 \times 10^{-4}$

Table 2. Parameters for truss optimization subjected to multiple probabilistic frequency constraints

Example	Frequency constraints (Hz)	Constraint reliability indices β_i^t
10-bar truss	$f_1 \geq 7, f_2 \geq 15, f_3 \geq 20$	$\beta_1^t = \beta_2^t = \beta_3^t = 3$
37-bar truss	$f_1 \geq 20, f_2 \geq 40, f_3 \geq 60$	$\beta_1^t = \beta_2^t = \beta_3^t = 3$
52-bar truss	$f_1 \leq 15.9155, f_2 \geq 28.6497$	$\beta_1^t = \beta_2^t = 3$

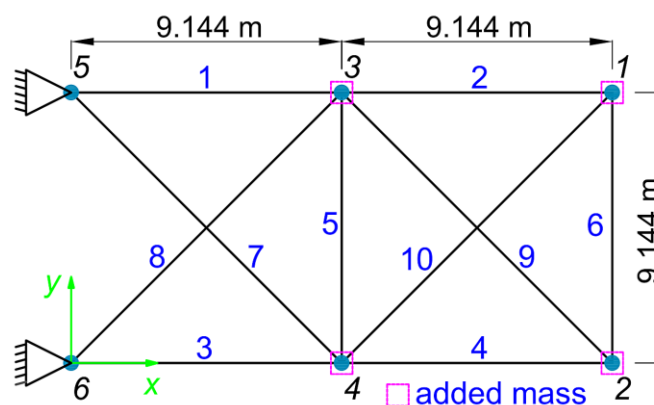


Fig. 2. The 10-bar planar truss structure

The 10-bar planar truss is illustrated in Fig. 2. Note that a lumped mass of 454 kg is attached at nodes 1, 2, 3, and 4.

Table 3 reports the optimal solutions obtained by different approaches. The outcomes

indicate that SORA-SHADE outperforms the SORA-DE and SORA-PSO in terms of computational cost, including both CPU time (s) and nFEAs. In addition, SORA-SHADE also yields better optimal weights. Additionally, probabilistic

constraints are satisfied because the reliability indices gained meet the desired values (β_1, β_2 , and $\beta_3 \geq 3$). Note that the second probabilistic constraint with the reliability index of $\beta_2 = \text{Inf}$ is inactive. The infinite reliability index of the second probabilistic constraint is attributed to the structural configuration. At the optimal solution, the second natural frequency $f_2 = 16.6511$ Hz is significantly higher than the lower bound of 15 Hz. The considerable difference between the obtained f_2 and the allowable value indicates that the constraint is satisfied for all realizations of the random cross-sectional areas, regardless of their uncertainties. As a result, the failure probability

associated with this constraint is effectively zero throughout the entire optimization process, resulting in $\beta_2 = \text{Inf}$ and rendering the constraint inactive. It can be seen that the optimal truss weight obtained by reliability-based analysis is higher than in the reliability-free analysis case. The truss weight can increase significantly when the structure is required to meet higher levels of the reliability index. Thus, conventional deterministic optimization may lead to inadequate and unsafe structures in practice. Eventually, the weight convergence histories of SORA-SHADE, SORA-DE, and SORA-PSO are sketched in Fig. 3.

4.2. 37-bar planar truss

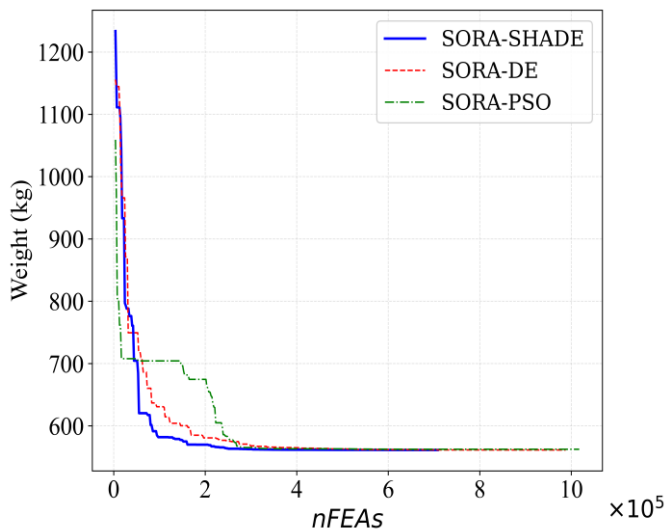


Fig. 3. The weight convergence history of the 10-bar truss obtained using SORA-SHADE, SORA-DE, and SORA-PSO

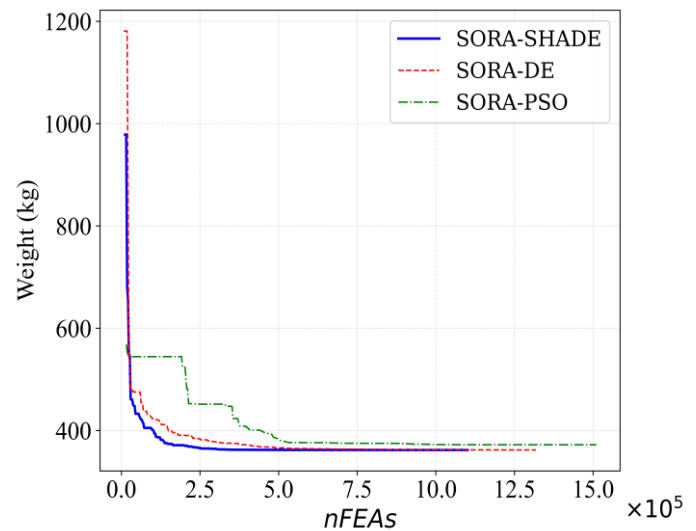


Fig. 5. The weight convergence history of the 37-bar truss obtained using SORA-SHADE, SORA-DE, and SORA-PSO

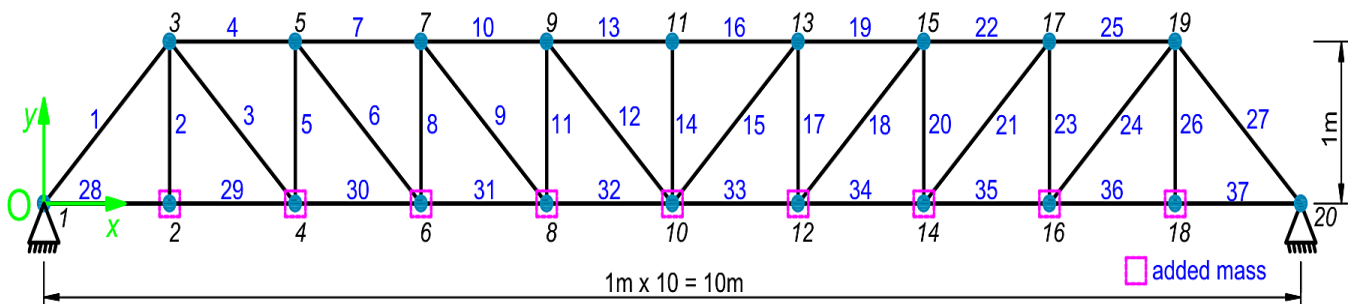


Fig. 4. The 37-bar planar truss structure

Fig. 4 shows the 37-bar planar truss's base structure. An additional 10 kg lumped mass is attached to all free nodes on the bottom chord. The design variables for this problem consist of the top

chord's nodal coordinates and the sizes of the truss members.

Table 4 shows a comparison of optimal results attained by the present approach with

different algorithms. It can be seen that the optimal weight, CPU time, and number of FE analyses obtained by SORA-SHADE are better than those of SORA-DE and SORA-PSO. As expected, the reliability-based solutions lead to slightly heavier structures than deterministic ones because additional safety regions are required to satisfy reliability constraints under uncertainties in cross-sectional area variables. In addition, probabilistic constraints are all fulfilled with the obtained reliability indices β_1, β_2 , and $\beta_3 \geq 3$. Lastly, Fig. 5 sketches the weight convergence histories.

4.3. 52-bar dome truss

The 52-bar dome truss is sketched in Fig. 6. All free nodes are allowed to displace within a ± 2

m range, and a 50 kg nonstructural mass is attached to every movable node.

Table 5 compares the obtained optimal solutions. The results demonstrate that all probabilistic constraints are completely fulfilled and achieve the required reliability indices without any violations. The larger the obtained reliability index, the higher the satisfaction of the probabilistic constraint. Similar to 10-bar, the first probabilistic constraint with the reliability index of $\beta_1 = \text{Inf}$ is inactive. Additionally, the results gotten from SORA-SHADE are better than other approaches. Eventually, Fig. 7 plots the convergence history of the optimal weight.

5. Conclusions

Table 3. Optimal solutions of the 10-bar planar truss using different methods

A_i (cm ²)	PSO [7]	HS [5]	SHADE	SORA-PSO	SORA-DE	SORA-SHADE
A_1	37.7120	34.2820	35.4555	37.0672	36.5421	38.1921
A_2	9.9590	15.6530	14.7931	16.2406	16.3554	15.8982
A_3	40.2650	37.6410	34.9959	39.4291	37.6829	36.4267
A_4	16.7880	16.0580	14.6649	17.5704	15.4155	16.0579
A_5	11.5760	1.0690	0.6450	0.6454	0.6453	0.6451
A_6	3.9550	4.7400	4.5549	4.7883	4.9238	4.9088
A_7	25.3080	22.5050	23.8485	23.6153	26.8170	25.5994
A_8	21.6130	24.6030	23.5494	26.3135	24.7281	25.8137
A_9	11.5760	12.8670	12.3245	12.1552	12.5624	12.9352
A_{10}	11.1860	12.0990	12.3699	13.0830	13.6883	13.0424
Best weight (kg)	537.9800	534.9900	524.4684	562.4102	561.2529	561.2270
Worst weight (kg)	-	-	531.6037	581.1835	565.7996	565.8022
Average weight (kg)	540.8900	537.6800	528.3789	571.9508	562.2258	563.8934
Standard deviation	6.840	2.490	3.2001	7.4676	2.0003	2.2392
CPUtime (s)	-	-	5.0251	466.8291	467.7003	452.9324
nFEAs	-	20000	9280	1017432	987737	708229
f_1 (Hz)	-	-	7.0001	7.2034	7.2044	7.2042
f_2 (Hz)	-	-	16.2004	16.7053	16.6220	16.6511
f_3 (Hz)	-	-	20.0000	20.6964	20.7166	20.7194
β_1	-	-	-	3.0062	3.0042	3.0043
β_2	-	-	-	Inf	Inf	Inf
β_3	-	-	-	3.0060	4.2403	3.9093

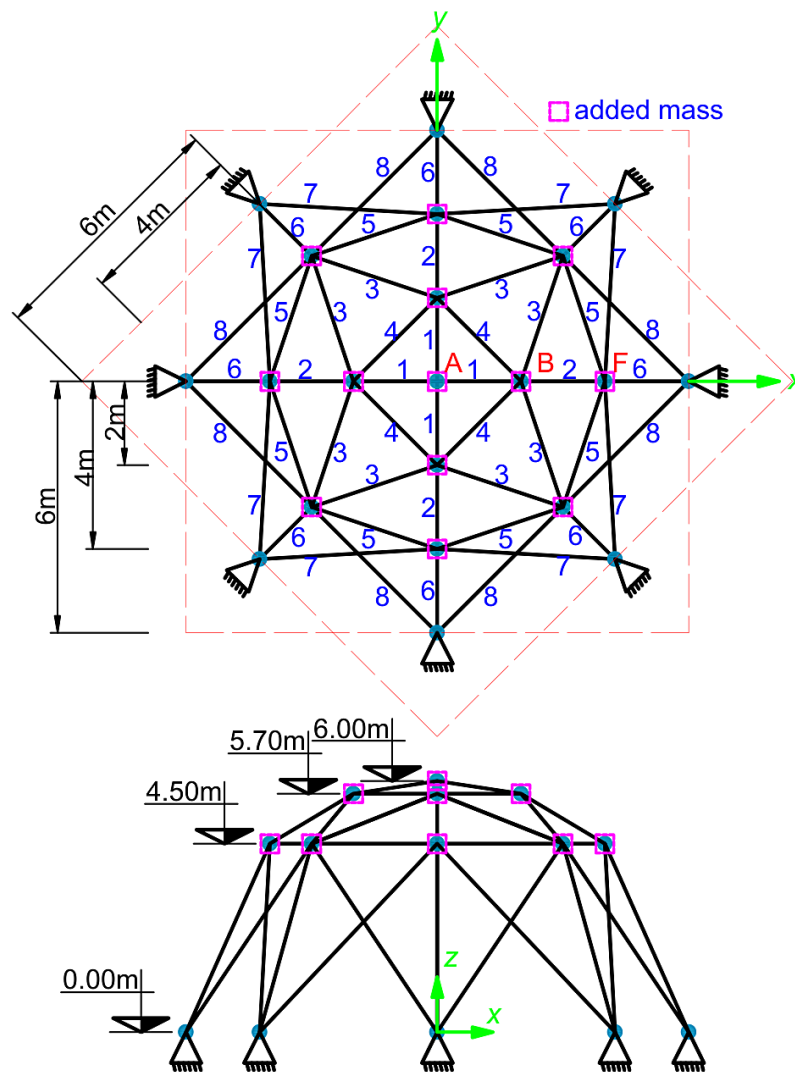


Fig. 6. The 52-bar dome truss structure

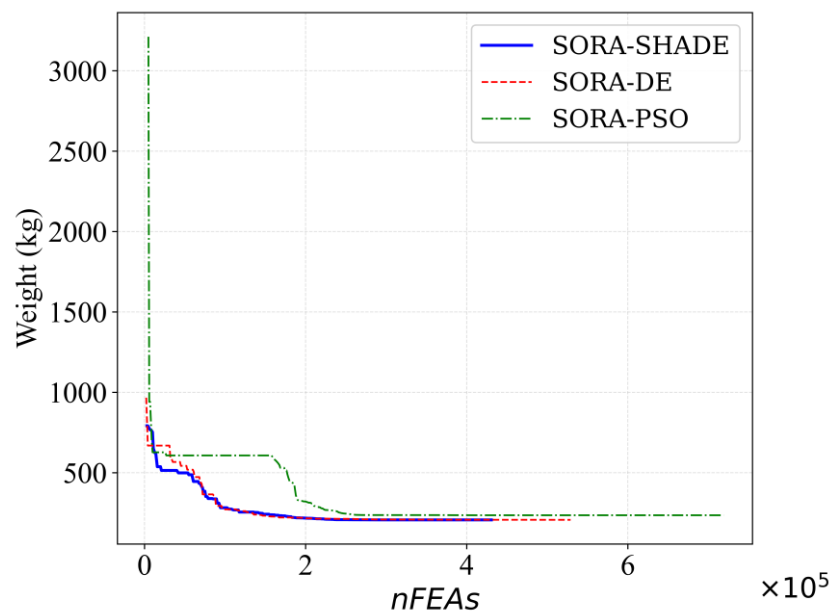


Fig. 7. The weight convergence history of the 52-bar truss obtained using SORA-SHADE, SORA-DE, and SORA-PSO

Table 4. Optimal solutions of the 37-bar planar truss using different methods

Design variables	PSO [7]	HS [5]	SHADE	SORA-PSO	SORA-DE	SORA-SHADE
Y_j (m), A_i (cm ²)						
$Y_3 - Y_{19}$	0.9637	0.8415	0.9041	0.7925	0.9109	0.9383
$Y_5 - Y_{17}$	1.3978	1.2409	1.2936	1.2058	1.2871	1.3237
$Y_7 - Y_{15}$	1.5929	1.4464	1.4660	1.4268	1.4746	1.5028
$Y_9 - Y_{13}$	1.8812	1.5334	1.5942	1.6790	1.6009	1.6355
Y_{11}	2.0856	1.5971	1.6693	2.0761	1.6756	1.7198
$A_1 - A_{27}$	2.6797	3.2031	2.8591	3.5572	3.2878	3.0039
$A_2 - A_{26}$	1.1568	1.1107	1.0403	1.0313	1.0198	1.0007
$A_3 - A_{24}$	2.3476	1.1871	1.0001	1.0021	1.0170	1.0004
$A_4 - A_{25}$	1.7182	3.3281	2.5852	3.9999	2.8060	2.7001
$A_5 - A_{23}$	1.2751	1.4057	1.1969	1.1519	1.2058	1.2887
$A_6 - A_{21}$	1.4819	1.0883	1.2038	1.2857	1.2399	1.3564
$A_7 - A_{22}$	4.6850	2.1881	2.6506	4.9570	2.8234	2.7324
$A_8 - A_{20}$	1.1246	1.2223	1.3452	1.6811	1.4538	1.3618
$A_9 - A_{18}$	2.1214	1.7033	1.4638	1.4366	1.5689	1.5801
$A_{10} - A_{19}$	3.8600	3.1885	2.6578	2.2082	2.4805	2.7286
$A_{11} - A_{17}$	2.9817	1.0100	1.2504	1.3023	1.3361	1.2797
$A_{12} - A_{15}$	1.2021	1.4074	1.3178	1.8638	1.4020	1.5088
$A_{13} - A_{16}$	1.2563	2.8499	2.4415	4.0779	2.8082	2.4587
A_{14}	3.3276	1.0269	1.000	1.679	1.003	1.0001
Best weight (kg)	377.2000	361.5000	359.0014	372.1862	362.0400	361.9512
Worst weight (kg)	-	-	359.7315	382.8351	362.3927	362.9521
Average weight (kg)	381.2000	362.0400	359.2829	378.1850	362.1777	362.3425
Standard deviation	4.2600	0.5200	0.2202	5.4510	0.1886	0.3940
CPUtime (s)	-	-	23.7444	2658.5539	2683.9338	2645.3558
nFEAs	-	20.000	6140	1510619	1317955	1101007
f_1 (Hz)	20.0001	20.0037	20.0001	20.5875	20.6034	20.5963
f_2 (Hz)	40.0003	40.0050	40.0000	41.2020	41.2369	41.1832
f_3 (Hz)	60.0001	60.0082	60.0000	61.9940	61.6438	61.6374
β_1	-	-	-	3.1021	3.0106	3.0022
β_2	-	-	-	3.1092	3.1018	3.0085
β_3	-	-	-	3.4582	3.0110	3.0028

This paper presents an efficient computational method that couples the SHADE with SORA methods to create the SORA-SHADE framework for reliability-based design optimization

of truss structures considering uncertainties of input data. The main idea of the SORA-SHADE method is to separate the optimization loop from the reliability assessment loop. Thus, this method

greatly improves calculation efficiency. By using information about the MPPs from the previous design point, the method moves the boundaries of any violated probabilistic constraints into the safe area. As a result, the optimal solution improves step by step through repeated iterations. The

proposed approach is verified through truss structure case studies. The obtained solutions indicate that the SORA-SHADE method achieves high effectiveness in resolving RBDO problems and could easily be adapted for many other engineering fields.

Table 5. Optimal solutions of the 52-bar dome truss using different methods

Design variables	PSO [7]	HS [5]	SHADE	SORA-PSO	SORA-DE	SORA-SHADE
Y_j (m), A_i (cm ²)						
Z_A	5.5344	4.7374	5.9820	4.3374	5.9590	5.9603
X_B	2.0885	1.5643	2.2676	1.2870	2.2336	2.2803
Z_B	3.9283	3.7413	3.7357	3.7000	3.7520	3.7234
X_F	4.0255	3.4882	3.9757	3.4893	3.9704	3.9676
Z_F	2.4575	2.6274	2.5000	2.5047	2.5003	2.5000
A_1	0.3696	1.0085	1.0003	1.0001	1.0014	1.0017
A_2	4.1912	1.4999	1.1063	2.1206	1.2566	1.2028
A_3	1.5123	1.3948	1.1985	2.0203	1.3418	1.2783
A_4	1.5620	1.3462	1.4449	1.2779	1.4327	1.5275
A_5	1.9154	1.6776	1.3969	2.0095	1.5244	1.5308
A_6	1.1315	1.3704	1.0003	1.0004	1.0006	1.0000
A_7	1.8233	1.4137	1.5927	2.3835	1.7253	1.7250
A_8	1.0904	1.9378	1.3817	1.2725	1.4949	1.5005
Best weight (kg)	228.3810	214.9400	193.1505	235.2759	206.6252	206.3244
Worst weight (kg)	-	-	212.2424	356.5831	215.3000	215.2584
Average weight (kg)	234.3000	229.8800	196.8589	294.1864	209.8964	208.3804
Standard deviation	5.2200	12.4400	6.4821	56.3886	4.7141	3.8606
CPUtime (s)	-	-	37.8426	3154.7721	2152.6226	2004.1647
nFEAs	-	20000	6820	718914	529015	430656
f_1 (Hz)	12.7510	12.2222	11.5085	15.2822	11.7990	11.9617
f_2 (Hz)	28.6490	28.6577	28.6479	29.5957	29.5636	29.5646
β_1	-	-	-	3.0029	Inf	Inf
β_2	-	-	-	3.0299	3.0065	5.1435

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM Vietnam for supporting this study.

Declaration of Generative AI

During the preparation of this work, the authors do not use Generative AI and AI-assisted technologies tools in order to generate the content

of the manuscript.

References

- [1] C.A.C. Coello, M. Rudnick, A.D. Christiansen. (1994). Using genetic algorithms for optimal design of trusses. *Proceedings Sixth International Conference on Tools with Artificial Intelligence. TAI 94, IEEE, pp. 88–94.* <https://doi.org/10.1109/TAI.1994.346509>
- [2] K.S. Lee, Z.W. Geem. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & Structures*, 82(9–10), 781–798. <https://doi.org/10.1016/j.compstruc.2004.01.002>
- [3] A. Kaveh, S. Talatahari. (2010). Optimal design of skeletal structures via the charged system search algorithm. *Structural and Multidisciplinary Optimization*, 41, 893–911. <https://doi.org/10.1007/s00158-009-0462-5>
- [4] A. Kaveh, M. Kalateh-Ahani, M.S. Masoudi. (2011). The CMA evolution strategy based size optimization of truss structures. *International Journal of Optimization in Civil Engineering (IJOCE)*, 1(2), 233–256.
- [5] L.F.F. Miguel, L.F.F. Miguel. (2012). Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. *Expert Systems with Applications*, 39(10), 9458–9467. <https://doi.org/10.1016/j.eswa.2012.02.113>
- [6] S.K. Azad, S.K. Azad, A.J. Kulkarni. (2012). Structural optimization using a mutation-based genetic algorithm. *International Journal of Optimization in Civil Engineering (IJOCE)*, 2(1), 81–101.
- [7] H.M. Gomes. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems with Applications*, 38(1), 957–968. <https://doi.org/10.1016/j.eswa.2010.07.086>
- [8] H. Akbulut. (2026). A modified starfish optimization algorithm (M-SFOA) for global optimization problems and its application to heart disease risk prediction. *Expert Systems with Applications*, 307, 131088. <https://doi.org/10.1016/j.eswa.2026.131088>
- [9] Q.X. Lieu. (2022). A novel topology framework for simultaneous topology, size and shape optimization of trusses under static, free vibration and transient behavior. *Engineering with Computers*, 38(6), 1–25. <https://doi.org/10.1007/s00366-022-01599-5>
- [10] Y. Xia, J. Liu, H. Qi. (2024). A population-based DNN-augmented optimization method for designing truss structures. *Swarm and Evolutionary Computation*, 89, 101613. <https://doi.org/10.1016/j.swevo.2024.101613>
- [11] Q.X. Lieu, D.T.T. Do, J. Lee. (2018). An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Computers & Structures*, 195, 99–112. <https://doi.org/10.1016/j.compstruc.2017.06.016>
- [12] S. Nguyen-Van, K.T. Nguyen, V.H. Luong, S. Lee, Q.X. Lieu. (2021). A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints. *Expert Systems with Applications*, 184, 115534. <https://doi.org/10.1016/j.eswa.2021.115534>
- [13] R. Storn, K. Price. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- [14] P. Civicioglu. (2012). Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. *Computers & Geosciences*, 46, 229–247. <https://doi.org/10.1016/j.cageo.2011.12.011>
- [15] S. Nguyen-Van, K.-W. Gwak. (2020). A Novel Determination of Boundaries of Cable Forces for Cable-Driven Parallel Robots with

- Frequency Constraint by Using Differential Evolution Algorithm. *Advances in Engineering Research and Application, Springer, Cham*, pp. 35–46. https://doi.org/10.1007/978-3-030-37497-6_4
- [16] S. Seok, A. Wang, M.Y. Chuah, D.J. Hyun, J. Lee, D.M. Otten, J.H. Lang, S. Kim. (2015). Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot. *IEEE/ASME Transactions on Mechatronics*, 20(3), 1117–1129. <https://doi.org/10.1109/TMECH.2014.2339013>
- [17] S. Sayah, A. Hamouda. (2013). A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Applied Soft Computing*, 13(4), 1608–1619. <https://doi.org/10.1016/j.asoc.2012.12.014>
- [18] A. Trivedi, D. Srinivasan, S. Biswas, T. Reindl. (2015). Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. *Swarm and Evolutionary Computation*, 23, 50–64. <https://doi.org/10.1016/j.swevo.2015.04.001>
- [19] S. Nguyen-Van, T.T.N. Nguyen, N. Nguyen-Dinh, Q.X. Lieu. (2021). Truss Optimization Under Frequency Constraints by Using a Combined Differential Evolution and Jaya Algorithm. *Advances in Engineering Research and Application, Springer, Cham*, pp. 861–873. https://doi.org/10.1007/978-3-030-64719-3_95
- [20] H. Zhang, D. Yue, X. Xie, S. Hu, S. Weng. (2015). Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch. *Applied Soft Computing*, 34, 312–323. <https://doi.org/10.1016/j.asoc.2015.05.012>
- [21] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, S.-Y. Chen. (2015). A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing*, 148, 75–82. <https://doi.org/10.1016/j.neucom.2012.08.075>
- [22] S.S. Reddy, J.Y. Park, C.M. Jung. (2016). Optimal operation of microgrid using hybrid differential evolution and harmony search algorithm. *Frontiers in Energy*, 10(3), 355–362. <https://doi.org/10.1007/s11708-016-0414-x>
- [23] Y. Lu, J. Zhou, H. Qin, Y. Li, Y. Zhang. (2010). An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects. *Expert Systems with Applications*, 37(7), 4842–4849. <https://doi.org/10.1016/j.eswa.2009.12.031>
- [24] G.-G. Wang, A.H. Gandomi, A.H. Alavi, G.-S. Hao. (2014). Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Computing and Applications*, 25(2), 297–308. <https://doi.org/10.1007/s00521-013-1485-9>
- [25] R. Tanabe, A. Fukunaga. (2013). Success-history based parameter adaptation for differential evolution. *2013 IEEE Congress on Evolutionary Computation, IEEE*, pp. 71–78. <https://doi.org/10.1109/CEC.2013.6557555>
- [26] M. Rashki, M. Miri, M.A. Moghaddam. (2014). A simulation-based method for reliability based design optimization problems with highly nonlinear constraints. *Automation in Construction*, 47, 24–36. <https://doi.org/10.1016/j.autcon.2014.07.004>
- [27] N.S. Hamzehkolaei, M. Miri, M. Rashki. (2016). An enhanced simulation-based design method coupled with meta-heuristic search algorithm for accurate reliability-based design optimization. *Engineering with Computers*, 32(3), 477–495. <https://doi.org/10.1007/s00366-015-0427-9>
- [28] M. Habashneh, M.M. Rad. (2023). An Investigation of the Recent Developments in Reliability-based Structural Topology Optimization. *Periodica Polytechnica Civil Engineering*, 67(3), 765–774. <https://doi.org/10.3311/PPci.22107>
- [29] R. Yadav, R. Ganguli. (2022). Reliability based and robust design optimization of truss and composite plate using particle swarm

- optimization. *Mechanics of Advanced Materials and Structures*, 29(13), 1892–1902. <https://doi.org/10.1080/15376494.2020.1843743>
- [30] N. Mashru, G.G. Tejani, P. Patel. (2025). Reliability-based multi-objective optimization of trusses with greylag goose algorithm. *Evolutionary Intelligence*, 18, 25. <https://doi.org/10.1007/s12065-024-01011-9>
- [31] L.A.R.d. Silva, A.J. Torii, A.T. Beck. (2024). System-reliability-based sizing and shape optimization of trusses considering millions of failure sequences. *Structural Safety*, 108, 102448. <https://doi.org/10.1016/j.strusafe.2024.102448>
- [32] H. Muayad, C. Raffaele, D. Marco, F. Hamed, M.M. Rad. (2025). Thermo-mechanical reliability-based topology optimization for imperfect elasto-plastic materials. *International Journal of Mechanics and Materials in Design*, 21, 1755–1776. <https://doi.org/10.1007/s10999-025-09799-9>
- [33] X. Du, W. Chen. (2004). Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design. *Journal of Mechanical Design*, 126(2), 225–233. <https://doi.org/10.1115/1.1649968>
- [34] J. Zhao, A.H.C. Chan, C. Roberts, K.B. Madelin. (2007). Reliability evaluation and optimisation of imperfect inspections for a component with multi-defects. *Reliability Engineering & System Safety*, 92(1), 65–73. <https://doi.org/10.1016/j.ress.2005.11.003>
- [35] S. Shan, G.G. Wang. (2008). Reliable design space and complete single-loop reliability-based design optimization. *Reliability Engineering & System Safety*, 93(8), 1218–1230. <https://doi.org/10.1016/j.ress.2007.07.006>
- [36] X. Du, W. Chen. (2001). A Most Probable Point-Based Method for Efficient Uncertainty Analysis. *Journal of Design and Manufacturing Automation*, 4(1), 47–66. <https://doi.org/10.1080/15320370108500218>
- [37] I. Lee, K.K. Choi, L. Du, D. Gorsich. (2008). Inverse analysis method using MPP-based dimension reduction for reliability-based design optimization of nonlinear and multi-dimensional systems. *Computer Methods in Applied Mechanics and Engineering*, 198(1), 14–27. <https://doi.org/10.1016/j.cma.2008.03.004>
- [38] A.M. Hasofer, N.C. Lind. (1974). Exact and Invariant Second-Moment Code Format. *Journal of the Engineering Mechanics Division*, 100(1), 111–121. <https://doi.org/10.1061/JMCEA3.0001848>
- [39] J. Tu, K.K. Choi, Y.H. Park. (1999). A New Study on Reliability-Based Design Optimization. *Journal of Mechanical Design*, 121(4), 557–564. <https://doi.org/10.1115/1.2829499>
- [40] R.V. Grandhi, L. Wang. (1998). Reliability-based structural optimization using improved two-point adaptive nonlinear approximations. *Finite Elements in Analysis and Design*, 29(1), 35–48. [https://doi.org/10.1016/S0168874X\(98\)00007-9](https://doi.org/10.1016/S0168874X(98)00007-9)
- [41] B.D. Youn, K.K. Choi. (2004). An Investigation of Nonlinearity of Reliability-Based Design Optimization Approaches. *Journal of Mechanical Design*, 126(3), 403–411. <https://doi.org/10.1115/1.1701880>
- [42] L. Graham-Brady. (2010). Structural design optimization considering uncertainties. *Structure and Infrastructure Engineering*, 6(5), 661–661. <https://doi.org/10.1080/15732470902792892>
- [43] B.D. Youn, K.K. Choi, R.-J. Yang, L. Gu. (2004). Reliability-based design optimization for crashworthiness of vehicle side impact. *Structural and Multidisciplinary Optimization*, 26(3–4), 272–283. <https://doi.org/10.1007/s00158-003-0345-0>
- [44] X. Li, C. Gong, L. Gu, Z. Jing, H. Fang, R. Gao. (2019). A reliability-based optimization method using sequential surrogate model and

Monte Carlo simulation. *Structural and Multidisciplinary Optimization*, 59(2), 439–460. <https://doi.org/10.1007/s00158-018-2075-3>.

[45] J.N. Reddy. (2005). An introduction to the finite element method. *McGraw-Hill, New York*.