Expert Systems With Applications 205 (2022) 117640

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient decoupling-assisted evolutionary/metaheuristic framework for expensive reliability-based design optimization problems

Zeng Meng [a,b,1,*], Ali Rıza Yıldız [c], Seyedali Mirjalili [d,e]

[a] School of Civil Engineering, Hefei University of Technology, Hefei 230009, PR China
[b] Department of Engineering Mechanics, State Key Laboratory of Structural Analyses for Industrial Equipment, Dalian University of Technology, Dalian 116024, PR China
[c] Department of Automotive Engineering, Uludağ University, Görükle, Bursa, Turkey
[d] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Australia
[e] Yonsei Frontier Lab, Yonsei University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Reliability-based design optimization
Fractional-order equilibrium optimizer algorithm
Metaheuristic
Particle's memory saving strategy
Evolutionary algorithm
Optimization
Algorithm

ABSTRACT

Reliability-based design optimization (RBDO) algorithm is to minimize the objective under the probabilistic factors. While gradient-based and classical evolutionary RBDO algorithms provide promising performance on simple optimization problems, they are likely to perform poorly on challenging problems, including the multimodal functions, discrete design spaces, non-differential problems, etc. This paper proposes a unified framework to improve the performance of existing RBDO algorithms for complex RBDO problems. Our framework is based on three new strategies: generalized decoupling evolutionary and metaheuristic RBDO framework, particle's memory saving strategy, and adaptive fractional-order equilibrium optimizer algorithm. The proposed algorithm is characterized by a decoupling strategy to enable the parallel operation of the inner reliability computation and outer deterministic optimization, a particle's memory saving strategy to provide effective guidance from the previous iteration, and the adaptive fractional-order equilibrium optimizer algorithm to enhance the search efficiency and global convergence capacity. To evaluate the performance of the proposed algorithm, a wide range of experiments are conducted on different types of use cases. The experimental results demonstrate that our algorithm provides superior performance over other comparative algorithms.

1. Introduction

Evolutionary and metaheuristic algorithms have exhibited a remarkable capacity in addressing a wide range of optimization problems in engineering systems (Faramarzi et al., 2020a; Guirguis, 2020). Particularly, owing to the collaborative search in the design space through a population of solutions instead of a single solution, evolutionary and metaheuristic algorithms are suitable for solving multimodal and non-differential problems (Bai et al., 2021; Ho-Huu et al., 2018). However, because of the presence of intrinsic uncertainties in all practical engineering applications, the disposal of these variations and randomness should be carried out (Jung, Cho, & Lee, 2019; Liu & Paulino, 2019; Liu et al., 2020; Wang et al., 2021). To achieve this goal, the reliability-based design optimization (RBDO) methodology is utilized, which handles the uncertainty parameters well (Meng et al., 2022; Yi et al., 2020; Youn et al., 2003). Owing to the efficiency and simplicity, the first-order reliability method (FORM) is very popular and widely employed in RBDO (Yang, Zhang, & Han, 2020; Zhu et al., 2020). Previous studies show that the existing RBDO algorithms can be classified into four categories: double loop algorithms, single-loop algorithms, decoupling algorithms, and hybrid algorithms (Aoues & Chateauneuf, 2010; Tu, Choi, & Park, 1999; Yang, Zhang, Cheng, & Han, 2021). However, these RBDO algorithms are established based on gradient optimization algorithms, so they lack capacity to handle the multimodal function, discrete design variable, and non-differential problems (Jensen et al., 2020; Schuëller & Jensen, 2008; Valdebenito & Schuëller, 2010).

Therefore, the evolutionary and metaheuristic algorithms, including genetic algorithm (GA) (Vasconcelos et al., 2001), particle swarm optimization (PSO) (Poli et al., 2007), differential evolution (DA) (Das &

* Corresponding author at: School of Civil Engineering, Hefei University of Technology, Hefei 230009, PR China.
E-mail addresses: mengz@hfut.edu.cn (Z. Meng), aliriza@uludag.edu.tr (A. Rıza Yıldız), ali.mirjalili@gmail.com (S. Mirjalili).
[1] ORCID ID: https://orcid.org/0000-0002-5648-5866.

https://doi.org/10.1016/j.eswa.2022.117640
Received 30 September 2021; Received in revised form 24 March 2022; Accepted 27 May 2022
Available online 30 May 2022
0957-4174/© 2022 Elsevier Ltd. All rights reserved.

Suganthan, 2010), etc., become a promising way to address the above difficulties (Brest et al., 2006; Casciati, 2014; Dan, 2009). According to Deb et al. (2009), the success proportion of classical gradient double loop algorithms only 49% through 2500 runs for simple test problems with two local optima, but the success proportion of GA algorithm can up to a hundred percent. Similar work has been done by Yang and Hsieh (2011), which is a successful application for discrete variables and non-smooth function. The difference is that they combine the merits of auto-tuning and boundary-approaching with PSO. With the global search demand, common optimization algorithms, such as sequential approximate programming (Yi et al., 2008), sequential quadratic programming (Jiang et al., 2019), convex linearization algorithm (Cho & Lee, 2011), and so on, usually cannot find the global optimum for problems with multiple local optima (Wang et al., 2019). Another reason is that these gradient algorithms always find the local optimum along the descent direction of objective function. Therefore, the gradient algorithms, which heavily rely on the initial search point, are unsuitable for disposing of the multimodal function and discrete design space problems (Jung, Cho, & Lee, 2020; Liu et al., 2018). More recently, Meng et al. (2021) conducted a state-of-the-art investigation of evolutionary and metaheuristic-based RBDO algorithms, such as PSO (Piotrowski et al., 2020), artificial bee colony (Karaboga & Basturk, 2007), grey wolf optimizer (GWO) (Mirjalili et al., 2014), water cycle algorithm (Eskandar et al., 2012), etc., and the results indicated the evolutionary and metaheuristic-based RBDO algorithms were superior to gradient-based RBDO algorithms in terms of global optimization capability.

On the other hand, the RBDO model consists of a double loop optimization model, where the inner reliability/optimization computation is of critical importance (Guo et al., 2009). Therefore, the evolutionary and metaheuristic-based RBDO algorithms are also used to tackle the shortcoming of the global optimization capabilities of the reliability computation problem. For instance, Elegbede (2005) combined the FORM and PSO, which aims to search the global design point effectively. Zhao et al. (2015) employed the chaos theory to improve the global search capacity, where the reliability problems with multiple design points are successfully solved. For challenging composite laminate structures with multiple failure criteria, das Neves Carneiro, G., & Conceição António, C. (2019) used the GA algorithm for global optimization. In addition, the DE (Pedroso, 2017; Zaeimi & Ghoddosian, 2018) and salp swarm algorithms (Zhong et al., 2020) were adopted to provide more opportunities for obtaining the global optimum.

The above-mentioned studies reflect the importance of evolutionary and metaheuristic algorithms for reliability analysis and RBDO. However, the existing evolutionary and metaheuristic-based RBDO algorithms remain to use the classical gradient algorithms in the inner loop, such as the Hasofer-Lind Rackwitz-Fiessler algorithm (Zhao et al., 2018), advanced mean value method (Keshtegar & Chakraborty, 2018), etc. In other words, these algorithms still not out of range of gradient algorithms, and the numerical difficulties of multimodal function, discrete design variable, and non-differential problem are not fundamentally conquered. Therefore, it is of great significance to put forward RBDO approaches using evolutionary and metaheuristic-based algorithms in both inner and outer loops. However, unlike the deterministic evolutionary and metaheuristic algorithms, the complete evolutionary and metaheuristic-based RBDO algorithms require performing the reliability analysis for each individual, and thus the computational cost should be unaffordable. For instance, if a simple example with three constraints is implemented using 2000 function calls in both inner reliability analysis and outer optimization, the total number of function calls is up to 1.2×10^7! Especially for high dimensional problems, the computational cost must be unacceptable owing to the "curse of dimensionality".

This study aims to solve the above-mentioned problems by proposing a new decoupling metaheuristic RBDO algorithm, where the reliability computation and optimization are executed separately and paralleled. Furthermore, a new metaheuristic algorithm, adaptive fractional-order equilibrium optimizer (AFOEO), is adopted to accelerate the convergence speed. The outline of this work is organized as follows: the background theories involved in the proposed RBDO algorithm are reviewed in section 2. Section 3 displays the proposed evolutionary and metaheuristic-based RBDO algorithms. Section 4 conducts experimental tests and discussion. Section 5 summarizes the concluding remarks.

2. Background

2.1. RBDO algorithms

In this study, the RBDO algorithm is implemented under an uncertain environment (Liu et al., 2019). Some studies (Li et al., 2021; Rosario et al., 2019) reveal that RBDO methodology usually achieves a more reliable design by comparing it with the deterministic optimization algorithm. Another merit is that deterministic optimization can be deemed as only one special solution with reliability index value equalling to 0. The RBDO formulation is defined as follows:

$$\begin{aligned} &\text{find} \quad \boldsymbol{d} \\ &\min \quad C(\boldsymbol{d}) \\ \text{s.t.} \quad &\text{Prob}(\mathrm{g_i}(\mathbf{d},\mathbf{x})\geqslant 0)\geqslant \mathrm{R_i^t} \quad \mathrm{i}=1,2,...,\mathrm{ng} \\ &\boldsymbol{d}^{\mathrm{L}}\leqslant \boldsymbol{d}\leqslant \boldsymbol{d}^{U} \end{aligned} \tag{1}$$

where $\mathbf{d}$ is the design variable vector with lower bound $\boldsymbol{d}^{\mathrm{L}}$ and upper bound $\boldsymbol{d}^{U}$. $\times$ is the random design variable vector. $\boldsymbol{\mu}_{\boldsymbol{x}}$ is the mean. C is the objective function. ng is the number of the performance function g.R_i^t is the allowable reliability value. $\text{Prob}(g_i(\boldsymbol{d},\boldsymbol{x})\geqslant 0)$ denotes the cumulative distribution function, which can be solved by using the following the complex multidimensional integral.

$$\text{Prob}(g_i(\boldsymbol{d},\boldsymbol{x})\geqslant 0)=\int \ldots \int_{g_i(\boldsymbol{d},\boldsymbol{x})\geqslant 0} \int f_X(\boldsymbol{x})d\boldsymbol{x} \tag{2}$$

where f_X denotes the joint probability density function. In an engineering system, it is impractical to tackle the above integral. Therefore, the FORM is usually employed in RBDO through two different ways: reliability index approach and performance measure approach (PMA).

2.2. Performance measure approach

PMA demonstrates promising local search capacity and good convergence, and thus it is commonly applied to engineering system (Tu, Choi, & Park, 2001). However, PMA is a local search algorithm (Torii et al., 2019; Youn et al., 2004). The expression of PMA is as follows:

$$\begin{aligned} &find \quad d \\ &\min \quad C(d) \\ s.t. \quad &g_i(d,u)\geqslant 0 \qquad i=1,...,ng \\ &d^L\leqslant d\leqslant d^U \end{aligned} \tag{3}$$

where g_i denotes the performance measure function that is usually performed in the standard normal space (U-space). $\boldsymbol{u}$ denotes the standard normal variable vector in U-space, which is computed by the random variable vector $\boldsymbol{x}$. The performance measure function g_i is obtained by searching the most probable target point (MPTP) $\boldsymbol{u}_{\mathrm{MPTP}}$.

$$\begin{aligned} &\text{find} \qquad \boldsymbol{u} \\ &\min \qquad g_i(\boldsymbol{u}) \\ \text{s.t.} \quad &\|\mathbf{u}\|=\beta^t \end{aligned} \tag{4}$$

where β^t denotes the target reliability index. To compute the MPTP efficiently, the advanced mean value (AMV) algorithm is extensively adopted by using the Karush–Kuhn–Tucker conditions.

$$\boldsymbol{u}^{k+1}=-\beta^t\frac{\nabla_{\boldsymbol{u}}g(\boldsymbol{d},\boldsymbol{u}^k)}{\left\|\nabla_{\boldsymbol{u}}g(\boldsymbol{d},\boldsymbol{u}^k)\right\|} \tag{5}$$

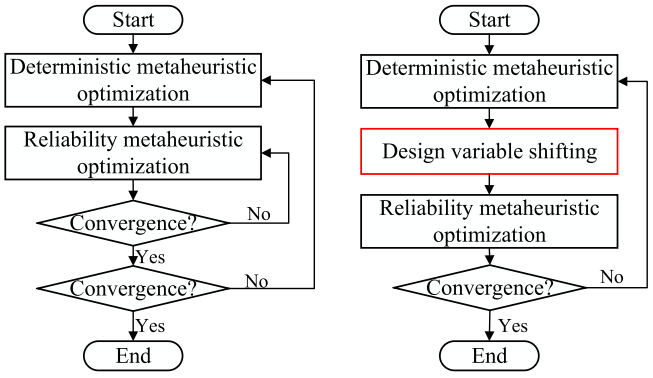

Fig. 1. Different optimization strategies (a) double loop metaheuristic RBDO method (b) DMEO.

where the subscript "k" is the iterative step number. $\nabla_u g$ is the gradients of performance function with respect to random variables $\boldsymbol{u}$. Considering the instability of AMV for the nonlinear problem, the improved algorithms, such as hybrid mean value algorithm, directional stability transformation algorithm, limited descent-based mean value method, etc. (Lee, Choi, & Gorsich, 2010; Meng et al., 2017; Youn et al., 2003), are put forward to promote stability. For these gradient algorithms, the computation of sensitivity is key, in which the multi-point linearization and approximate sensitivity method can be employed owing to the high efficiency (Liu et al. 2016a, 2016b).

The main purpose of the above MPTP algorithms is to search the optimum along the direction of the gradient descent (Rackwitz, 2001). Once the RBDO problems contain multiple MPTPs or unusable derivative, the classical gradient MPTP algorithms become unavailable. Thus, evolutionary and metaheuristic algorithms provide a powerful tool, and the universality and global search capacity are significantly promoted. Additional details on evolutionary and metaheuristic-based MPTP algorithms can be referred to in Refs. (Pedroso, 2017; Yaseen & Keshtegar, 2018).

2.3. Equilibrium optimizer algorithm

In this paper, equilibrium optimizer (EO), a recently developed metaheuristic algorithm, exhibits superiority over the well-known and other recent algorithms (GA, PSO, GWO, SSA, and gravitational search algorithm) in terms of exploration and exploitation (Faramarzi et al., 2020b; Zhong & Li, 2022). The basic idea of EO is to search for the best generic mass balance on a control volume. The equilibrium concentration is analogous to a particle's position of PSO. Its updating rule of equilibrium concentration is formulated as follow:

$$d_i^{iter+1} = d_{eq,i}^{iter} + \left(d_i^{iter} - d_{eq,i}^{iter}\right)F_i^t + \frac{G_i^{iter}}{\lambda_i V}\left(1 - F_i^{iter}\right) \tag{6}$$

where *iter* is the current iterative number. d_i^{iter} denotes the ith component of the concentration vector. $d_{eq,i}^{iter}$ is the candidate solution. V is the control volume. G^{iter} is the generation rate that is used to carry out the exploitation phase, which is determined by the generation probability. The parameter F is the exponential term, which can be formulated as below:

$$F_i^{iter} = \exp\left(-\lambda_i\left(\left(1 - \frac{iter}{Max\_iter}\right)^{a_2\frac{iter}{Max\_iter}} - \bar{t}_0\right)\right) \tag{7}$$

$$\bar{t}_0 = \frac{1}{\lambda_i}\ln\left(-a_1 sign(r_i - 0.5)\left[1 - e^{-\lambda_i \bar{t}}\right]\right) + \bar{t} \tag{8}$$

where λ_i and r_i are the random number that lies in the range of [0 1]. *Max_iter* is the allowable iterative number. a_1 and a_2 are two parameters, which are utilized to control the exploration and exploitation ability. Additional details of EO algorithm can be referred to in (Faramarzi et al., 2020b). For RBDO using metaheuristic algorithm, assume we use N population points and *Max_iter* iterations in both inner MPTP computation and outer deterministic optimization. The number of function evaluations for a RBDO problem with ng constraints should be.

$$(Max\_iter \times N)^2 \times ng \tag{9}$$

It should be mentioned that the number of function evaluations of other evolutionary and metaheuristic algorithms is identical to that of EO.

3. Proposed efficient generalized evolutionary and metaheuristic-based RBDO algorithm

In this section, a new generalized evolutionary and metaheuristic-based RBDO algorithm within both inner and outer optimization loops is proposed. Inspired by the sequential optimization and reliability assessment (SORA) method (Du & Chen, 2004), a new generalized decoupling metaheuristic RBDO framework is put forward, where the EO is adopted as the foundation. This new variant is called the decoupling metaheuristic equilibrium optimizer (DMEO). The main contribution of the DMEO is proposing and combining the three strategies, including the global reliability search strategy, particle's memory saving strategy, and adaptive fractional-order (FO) calculus strategy, within decoupling metaheuristic RBDO framework. The purpose of the first global reliability search strategy is to search the global MPTP, and it can guide the DMEO to search accurately. The second particle's memory saving strategy is to generate competitive individuals from the history computation information, thereby speeding up the optimization process. The third adaptive FO calculus methodology is adopted for global and local search. The hybrid utilization of these strategies can ensure the high efficiency of DMEO for global optimization. This algorithm can be very suitable for addressing the complex RBDO problems, such as discrete design space, high dimensional example, and non-differential problem.

3.1. Generalized decoupling metaheuristic RBDO framework

As mentioned before, the evolutionary and metaheuristic-based RBDO algorithms require executing the reliability analysis optimization repeatedly at each particle, which leads to unbearably expensive computation cost, as shown in Fig. 1(a). Inspired by SORA method (Du & Chen, 2004), the DMEO splits the reliability analysis from the optimization process of the outer loop, as shown in Fig. 1(b). Therefore, it avoids the inner reliability optimization but only shifts the design variables and probabilistic constraints, as shown in Fig. 2. As such, each particle only requires perform deterministic analysis other than operating reliability computation. In this way, the computation cost can be greatly improved. Unlike the SORA algorithm, the DMEO consists of the swarm, and each particle should be shifted in the optimization process. In general, the decoupling metaheuristic RBDO formulas are as follows:

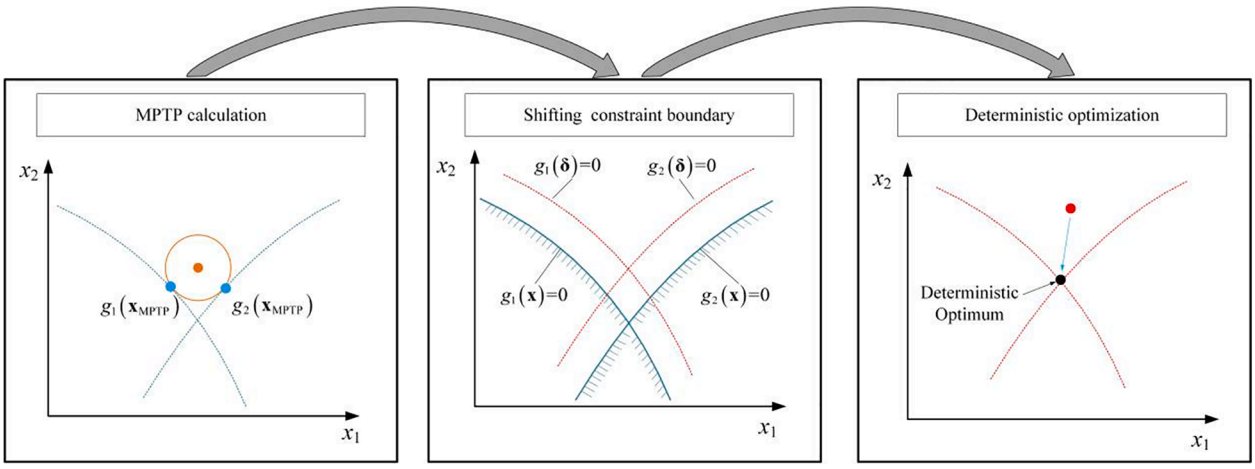

Fig. 2. Schematic diagram of DMEO.

$$\begin{aligned} &\text{find} && \boldsymbol{d}^k \\ &\min && C(\boldsymbol{d}^k) \\ \text{s.t.}\quad & g_i(\boldsymbol{\delta}_i^k, \boldsymbol{x}_i^{k-1}) \geqslant 0 && i = 1, \ldots, ng \\ & \boldsymbol{d}^L \leqslant \boldsymbol{d}^k \leqslant \boldsymbol{d}^U \end{aligned} \tag{10}$$

where the subscript "k" is the iterative number of DMEO. $\boldsymbol{\delta}^k$ denotes the new design variable, which is used by shifting the constraint boundary using shift-vector $\boldsymbol{s}$. In DMEO of Eq. (10), the new design variables for jth particle can be computed as follows:

$$\boldsymbol{\delta}_{i,j}^k = \boldsymbol{d}_{i,j}^k - \boldsymbol{s}_{i,j}^{k-1} \tag{11}$$

It should be noted that the shift vector for each particle is the same, which is only related to the ith constraint.

$$\boldsymbol{s}_{i,j}^{k-1} = \boldsymbol{d}_j^{k-1} - \boldsymbol{x}_{\mathrm{MPTP}j}^{k-1} \tag{12}$$

where the $\boldsymbol{x}_{\mathrm{MPTP}j}^{k-1}$ is MPTP that is transformed from the $\boldsymbol{u}_{\mathrm{MPTP}j}^{k-1}$. Since the reliability computation requires handling the complex equality constraint, we use the trigonometric function to transform $\mathbf{u}$ in Cartesian coordinate to $\boldsymbol{\theta}$ in polar coordinate (Chen et al., 2013). The coordinate transformation can be expressed as follows:

$$\begin{cases} u_1 = \beta^t \sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{n-2})\sin(\theta_{n-1}) \\ u_2 = \beta^t \sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{n-2})\cos(\theta_{n-1}) \\ u_3 = \beta^t \sin(\theta_1)\sin(\theta_2)\cdots\cos(\theta_{n-2}) \\ \vdots \\ u_{n-1} = \beta^t \sin(\theta_1)\cos(\theta_2) \\ u_n = \beta^t \cos(\theta_1) \end{cases} \tag{13}$$

Based on Eq. (13), the equality constraint $\|\boldsymbol{u}\| = \beta^t$ of Eq. (4) can be satisfied naturally. Then, the optimization model can be simplified as.

$$\begin{aligned} &\text{find} && \boldsymbol{\theta} \\ &\min && g(\boldsymbol{\theta}) \end{aligned} \tag{14}$$

where the $\boldsymbol{\theta}$ is the $(ND-1)$-dimensional random variables in θ-space, and each component lies in the range of [0, 2π]. Based on the previous transformation, the equality constraint of Eq. (4) vanishes, thus the EO can be used to solve the Eq. (14) conveniently. Under the proposed framework, the number of function evaluations of reliability analysis should be.

$$Max\_k \times Max\_iter \times N \times ng \tag{15}$$

where *Max_k* denotes the maximum iteration number of the decoupling framework. According to the relative works of the decoupling method, the value of *Max_k* is usually very small (far less than *Max_iter* × *N*). Therefore, the proposed method should be very efficient, and it only needs computing *Max_k* × *ng* times for reliability analysis. In this case, the computational cost is perfectly acceptable. First, Algorithm 1 is offered below to demonstrate the pseudocode of the framework of the DMEO from the general viewpoint.

Algorithm 1 Pseudocode of generalized decoupling metaheuristic RBDO framework

1: k=0; error=1.
% k is the iterative number of the proposed generalized decoupling metaheuristic RBDO framework. Error is the relative error of objective function value between two successive iterations.
2: Initialize the shifted vector $s_{ij}^k = \mathbf{0}$; $j = 1, \ldots, ng$.
3: **While** k<Max_k or error< Max_error
% Max_error is the maximum value of relative error.
4: Generate N population points [$\mathbf{d}_1$; $\mathbf{d}_2$; …,$\mathbf{d}_N$] by Latin hypercube sampling method (Vořechovský, 2015) and particle's memory saving strategy. Compute all points/particles in the population using the function evaluations of the fitness function (Rodrigues et al., 2016).
5: Compute the shifted design variables using the $\boldsymbol{\delta}_{ij}^{k+1} = \boldsymbol{d}_{ij}^{k+1} - \boldsymbol{s}_{ij}^k$; $j = 1, \ldots, ng$ for the ith constraint. Update the points using the AFOEO to obtain the best solution $\boldsymbol{d}_j^k$; $j = 1, \ldots, ng$.
6: **For** j=1:ng
7: Generate NM population points [$\boldsymbol{\theta}_1$; $\boldsymbol{\theta}_2$; …, $\boldsymbol{\theta}_{NM}$] by Latin hypercube sampling method and particle's memory saving strategy. Compute all points/particles in the population using the function evaluations of the fitness function.
8: Compute the best solution of MPTP $\boldsymbol{\theta}^k$ using the AFOEO. During the computing process of the fitness function, the random variables $\boldsymbol{\theta}$ is transformed to $\mathbf{x}$ by using the Eq. (13) and Rosenblatt transformation(Lee et al., 2008) in the order. Obtain the best solution of MPTP $\boldsymbol{x}_{\mathrm{MPTP}j}^k$ in the x-space.
9: **End For**
10: Compute the shift-vector $\boldsymbol{s}_{ij}^k = \boldsymbol{d}_j^k - \boldsymbol{x}_{\mathrm{MPTP}j}^k$; $j = 1, \ldots, ng$.
11: $error = \left|C(\boldsymbol{d}^k) - C(\boldsymbol{d}^{k-1})\right| \Big/ C(\boldsymbol{d}^k)$.
12: **End While**

3.2. Adaptive fractional-order equilibrium optimizer algorithm

3.2.1. Adaptive strategy for exponential parameter

In EO, the exponential term F_i^{iter} determines the convergence speed, and it also impacts the exploitation and exploration capacities. On the one hand, as the increase of iterative number, the search speed should be reduced to promote the exploitation ability. On the other hand, the search speed should be accelerated according to the fitness value. Thus, a new adaptive strategy is proposed to update the exponential term, which is stated as:

$$F_i^{iter} = \exp\left(-\lambda_i\left(\left(\frac{1}{iter}\right)^{\xi} - \bar{t}_0\right)\right) \tag{16}$$

where.

$$\xi = \begin{cases} \left|\dfrac{\text{best}(f^{iter}) - \text{mean}(f^{iter})}{\text{best}(f^{iter}) - \text{worst}(f^{iter})}\right| & if \quad \text{best}(f^{iter}) \neq \text{worst}(f^{iter}) \\ \left|\text{best}(f^{iter}) - \text{mean}(f^{iter})\right| & if \quad \text{best}(f^{iter}) = \text{worst}(f^{iter}) \end{cases} \tag{17}$$

where f^{iter} denotes the fitness value.

3.2.2. Opposition-based learning

Inspired from the social revolutions with sudden change in human society, opposition-based learning (OBL) strategy is developed by Tizhoosh (2005), which aims to promote the global convergence capacity. OBL takes advantages of existing solution direction and opposite search direction. Let $\boldsymbol{d}$ and $\boldsymbol{\theta}$ are candidate solutions. Considering $\boldsymbol{d} \in [\boldsymbol{d}^{\mathrm{L}}, \boldsymbol{d}^{\mathrm{U}}]$ and $\boldsymbol{\theta} \in [\boldsymbol{\theta}^{\mathrm{L}}, \boldsymbol{\theta}^{\mathrm{U}}]$, the opposite candidate solutions $\widetilde{\boldsymbol{d}}$ and $\widetilde{\boldsymbol{\theta}}$ are stated as:

$$\widetilde{\boldsymbol{d}} = \boldsymbol{d}^{\mathrm{L}} + \boldsymbol{d}^{\mathrm{U}} - \boldsymbol{d}\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^{\mathrm{L}} + \boldsymbol{\theta}^{\mathrm{U}} - \boldsymbol{\theta} \tag{18}$$

For RBDO problem, the utilization of OBL is summarized as below. Assume that $f(\boldsymbol{d}, \boldsymbol{\theta})$ is fitness function. During the iterative process, when $f(\boldsymbol{d}, \boldsymbol{\theta}) \geqslant f(\widetilde{\boldsymbol{d}}, \widetilde{\boldsymbol{\theta}})$, the opposite candidate solution is preserved. On the contrary, when $f(\boldsymbol{d}, \boldsymbol{\theta}) < f(\widetilde{\boldsymbol{d}}, \widetilde{\boldsymbol{\theta}})$, the original solution is kept. This condition is repeatedly checked until the satisfying the stopping criterion.

3.2.3. Fractional-order equilibrium optimizer algorithm

EO exhibits a promising global search capability but encounters the inefficient local search capability that may lead to a poor convergence rate. In order to tackle the limitation, the FO calculus (Wang et al., 2017) is embedded into the EO by taking the merits of the memory saving property of the previous solutions. The basic idea of FO is to perform the integral and derivative with arbitrary order. Comparing to the integer calculus, FO calculus is more flexible and accurate, and the works of Yousri et al. (2019) disclosed that FO calculus could enhance the local search ability. There are a series of definitions of FO, in which Grunwald–Letnikov is the classical one, and it is stated as follows:

$$D^{\alpha}(d(iter)) = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{n=0}^{\infty}(-1)^n\frac{\Gamma(\alpha+1)}{\Gamma(n+1)\Gamma(\alpha-n+1)}x(iter-nh) \tag{19}$$

where $D^{\alpha}(d(iter))$ is the Grunwald–Letnikov fractional derivative of α order. Γ is the gamma function. When the discrete system is considered, Eq. (19) becomes.

$$D^{\alpha}(d(iter)) = \frac{1}{T^{\alpha}}\sum_{n=0}^{r}\frac{(-1)^n\Gamma(\alpha+1)d(iter-nT)}{\Gamma(n+1)\Gamma(\alpha-n+1)} \tag{20}$$

where T denotes the sampling period, and r denotes the number from the previous event. Especially when $\alpha = 1$, the Eq. becomes.

$$D^{1}(d(iter)) = d(iter+1) - d(iter) \tag{21}$$

where $D^{1}(d(iter))$ is the difference between two continuous events. Based on the definition of EO of Eq. (6), the formulation can be rewritten as.

$$D^{1}\left(d_i^{iter+1}\right) = d_{eq,i}^{iter} + \left(F_i^{iter} - 1\right)d_i^{iter} - d_{eq,i}^{iter}F_i^{iter} + \frac{G_i^{iter}}{\lambda_i V}\left(1 - F_i^{iter}\right) \tag{22}$$

By substituting the fractional derivative α order to the fractional-order one, we can obtain the following expressions:

$$D^{\alpha}\left(d_i^{iter+1}\right) = d_{eq,i}^{iter} + \left(F_i^{iter} - 1\right)d_i^{iter} - d_{eq,i}^{iter}F_i^{iter} + \frac{G_i^{iter}}{\lambda_i V}\left(1 - F_i^{iter}\right) \tag{23}$$

Based on the Grunwald–Letnikov definition in Eq. (22), the Eq. (24) for $T = 1$ can be formulated as.

$$D^{\alpha}\left(d_i^{iter+1}\right) = d_i^{iter+1} + \sum_{n=0}^{r}\frac{(-1)^n\Gamma(\alpha+1)d_i^{iter+1-n}}{\Gamma(n+1)\Gamma(\alpha-n+1)} \tag{24}$$

Combing Eq. (22) and Eq. (24), the formulation of the EO using FO can be described by.

$$d_i^{iter+1} = -\sum_{n=0}^{r}\frac{(-1)^n\Gamma(\alpha+1)d_i^{iter+1-n}}{\Gamma(n+1)\Gamma(\alpha-n+1)} + d_{eq,i}^{iter} + \left(F_i^{iter} - 1\right)d_i^{iter} - d_{eq,i}^{iter}F_i^{t} + \frac{G_i^{iter}}{\lambda_i V}\left(1 - F_i^{iter}\right) \tag{25}$$

When the first four terms, i.e., $r = 4$, is considered, the update formulation can be expanded as.

$$\begin{aligned} d_i^{iter+1} = {} & \alpha d_i^{iter} - \frac{1}{2!}\alpha(\alpha-1)d_i^{iter-1} + \frac{1}{3!}\alpha(\alpha-1)(\alpha-2)d_i^{iter-2} \\ & - \frac{1}{4!}\alpha(\alpha-1)(\alpha-2)(\alpha-3)d_i^{iter-3} + d_{eq,i}^{iter} \\ & + \left(d_i^{iter} - d_{eq,i}^{iter} - 1\right)F_i^{iter} + \frac{G_i^{iter}}{\lambda_i V}\left(1 - F_i^{iter}\right) \end{aligned} \tag{26}$$

It is well known that the high order derivative relates to the global convergence property and the small value α leads to the local convergence ability. According to Yousri et al. (2020), the value α can be selected as 0.1 in this study. Besides, to make a tradeoff between exploitation and exploration, we use the particle's memory saving strategy to generate potential candidates. In general, we propose the AFOEO by combing the merits of adaptive strategy, OBL, and FO. Algorithm 2 provides the pseudocode of AFOEO in the proposed

Table 1
Results of reliability analysis for evolutionary and metaheuristic algorithms.

Algorithms	Parameter Setting	Algorithms	Parameter Setting
GA	$pc = 1.0, pm = 0.1$	DE	$CR = 0.8, F = 0.5$
PSO	$w_{\min} = 0.9, w_{\max} = 0.4,$ $c_1 = 2, c_2 = 2$	GWO	Only the common parameters (*Max_iter* and a)
ABC	Limit $= NP \times n$	EO	$a_1 = 2$, $a_2 = 1$

decoupling RBDO framework, which is adopted in both MPTP computation and deterministic optimization.

Algorithm 2 Pseudocode of AFOEO

1. *iter*=0.
2. Set the parameters $a_1 = 2$; $a_2 = 1$; $\alpha = 0.1$.
3. Initialize the particle's populations $\mathbf{d}_i$ (i=1,…,N).
4. **While** *iter*<*Max_iter*
5. **For** i=1: N
6. Compute the fitness of all particles $fit(d_i)$ (i=1,…,N).
7. **If** $fit\ (d_i) < fit\ (d_{eq,1})$
8. $d_{eq,1} = d_i$; $fit(d_{eq,1}) = fit(d_i)$.
9. **Elseif** $fit(d_i) < fit(d_{eq,1})$ & $fit(d_i) < fit(d_{eq,2})$
10. $d_{eq,2} = d_i$; $fit(d_{eq,2}) = fit(d_i)$.
11. **Elseif** $fit(d_i) < fit(d_{eq,1})$ & $fit(d_i) < fit(d_{eq,2})$ & $fit(d_i) < fit(d_{eq,3})$.
12. $d_{eq,3} = d_i$; $fit(d_{eq,3}) = fit(d_i)$.
13. **Elseif** $fit(d_i) < fit(d_{eq,1})$&$fit(d_i) < fit(d_{eq,2})$&$fit(d_i) < fit(d_{eq,3})$&$fit(d_i) < fit(d_{eq,4})$.
14. $d_{eq,4} = d_i$; $fit(d_{eq,4}) = fit(d_i)$.
15. **End If**

(*continued on next column*)

(*continued*)

Algorithm 2 Pseudocode of AFOEO

16. **End For**
$d_{ave} = (d_{eq,1} + d_{eq,2} + d_{eq,3} + d_{eq,4})/4$.
Construct the equilibrium vector
$d_{ave} = (d_{eq,1} + d_{eq,2} + d_{eq,3} + d_{eq,4})/4$.
17. **For** i=1: N
18. Randomly choose the candidate from the equilibrium vector.
19. Compute the values F_i^{iter} by Eqs. (16) and (17).
20. Compute the values of G_i^{iter} according to the history information.
21. Sort the particles according to the values of fitness.
22. Update the best particle by Eq. (26).
23. Perform the OBL by Eq. (18).
24. Update other particles by Eq. (6).
25. **End For**
26. *iter* =*iter*+1.
27. **End While**

3.3. Particle's memory saving strategy

Under the decoupling framework, the global optimum can be found. To further accelerate the convergence rate, the particle's memory saving strategy is employed. This method makes the utmost use of the iterative history information of MPTP, where the MPTP solution from the previous iterative step can be selected as an individual of the initial population at the current MPTP search in the polar coordinate. This is because the MPTP at the previous iterative step is located at the neighborhood of the accurate MPTP. Also, the optimum of design

Table 2
RBDO results for different cases.

Cases	Algorithms	Success Proportion	Best	Mean	Worst	SD	Times (s)	F-evaluations
TC1	PMA	66.3333%	−12.1741	Not convergence			1.1465	341
	SORA	58.3333%	−12.1741	Not convergence			0.4901	218
	SLA	65.3333%	−12.1741	Not convergence			0.8187	98
	CSLA	68.3333%	−12.1741	Not convergence			6.5166	172
	GA	0%	−11.9257	11.8889	50.3421	18.6009	3921.9817	5.48E7
	PSO	94.3333%	−13.1542	−10.3964	27.5574	8.1462	2395.0902	5.48E7
	ABC	1%	−12.1694	−10.2165	−0.0691	1.4963	3109.9044	5.44E7
	DE	4.3333%	−14.2328	−13.1996	−12.1746	0.3268	3085.5515	5.48E7
	GWO	35.6667%	−13.0255	−12.0839	−11.7527	0.2160	2430.9356	5.40E7
	EO	88.6667%	−13.2631	−12.2354	−12.1739	0.1849	3188.7092	5.40E7
	DMEO	**100%**	**−12.1741**	**−12.1741**	**−12.1740**	**1.1276E−8**	**3.2312**	**8.45E4**
TC2	PMA	0.6472%	6.0739	8.0674	11.8983	1.0076	7.0747	6.89E3
	SORA	8.1258%	8.8001	9.9704	0.4015	0.1442	0.1442	1.33E3
	SLA	0	6.2143	8.8094	11.5500	0.9316	0.3850	422
	CSLA	0	6.2772	8.1114	11.3560	1.0851	4.5381	872
	GA	6.5217%	6.0453	7.1103	9.8212	0.7761	3958.4109	5.48E7
	PSO	88.2353%	6.0391	6.0740	6.8350	0.1046	2379.0337	5.48E7
	ABC	100%	6.0391	6.0391	6.0391	3.2027E−14	3082.6194	5.44E7
	DE	99.6711%	6.0391	6.0400	6.3081	0.0154	3161.5440	5.48E7
	GWO	99.6667%	6.0391	6.0415	6.1617	0.0077	2428.8967	5.40E7
	EO	100%	6.0391	6.0391	6.0391	3.1138E−14	3231.9223	5.40E7
	DMEO	**100%**	**6.0391**	**6.0391**	**6.0391**	**3.1138E−14**	**3.9057**	**1.04E5**
TC3	GA	0%	0.0226	0.0289	0.0438	4.4746E−2	64974.8955	3.67E7
	PSO	86.6667%	0.0213	0.0215	0.0224	2.9772E−4	31799.7474	3.67E7
	ABC	100%	0.0214	0.0215	0.0216	6.2658E−5	32753.9536	3.60E7
	DE	100%	0.0213	0.0213	0.0214	2.9587E−5	39012.6300	3.67E7
	GWO	96.8750%	0.0213	0.0215	0.0218	1.6666E−4	31894.4027	3.60E7
	EO	94.1176%	0.0213	0.0215	0.0219	1.8555E−4	37392.8698	3.60E7
	DMEO	**100%**	**0.0213**	**0.0216**	**0.0216**	**7.6276E−5**	**17.5870**	**4.77E5**
TC4	GA	90%	3.6165	3.6769	3.8832	0.1157	209007.3040	1.43E9
	PSO	100%	3.6150	3.6150	3.6150	4.8648E−16	134099.7850	1.43E9
	ABC	100%	3.6150	3.6150	3.6150	1.9505E−8	138001.0356	1.41E9
	DE	100%	3.6150	3.6150	3.6150	9.4906E−16	153200.0140	1.43E9
	GWO	100%	3.6151	3.6159	3.6193	0.0017	135555.0273	1.41E9
	EO	100%	3.6150	3.6150	3.6150	9.4206E−16	152349.5122	1.41E9
	DMEO	**100%**	**3.6150**	**3.6150**	**3.6150**	**1.3550E−15**	**57.3729**	**1.18E6**

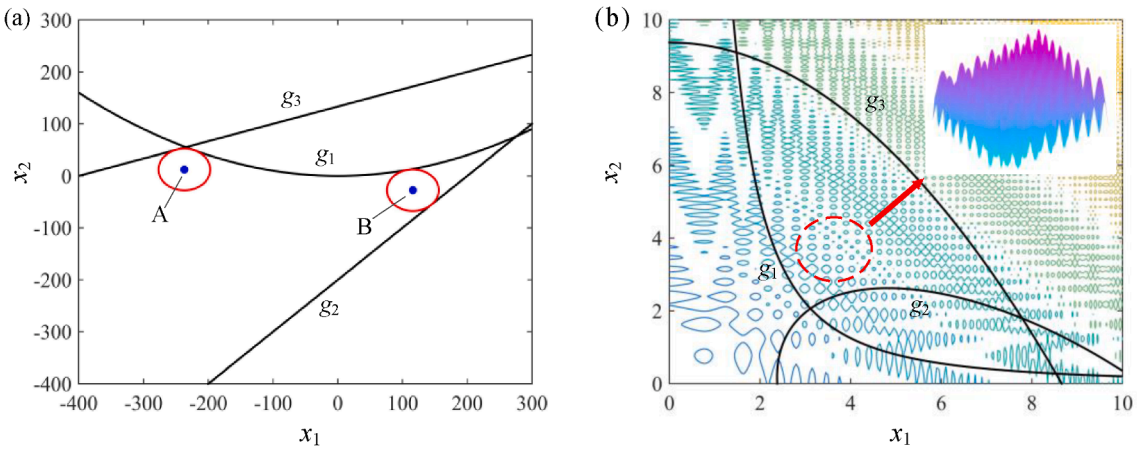

Fig. 3. Schematic diagram of two different cases (a) TC1: RBDO example with two local optima that is difficultly to be solved by the gradient algorithms (b) TC2: multimodal RBDO problem that should be solved by the evolutionary and metaheuristic-based RBDO algorithms with strongly global convergence capacity.

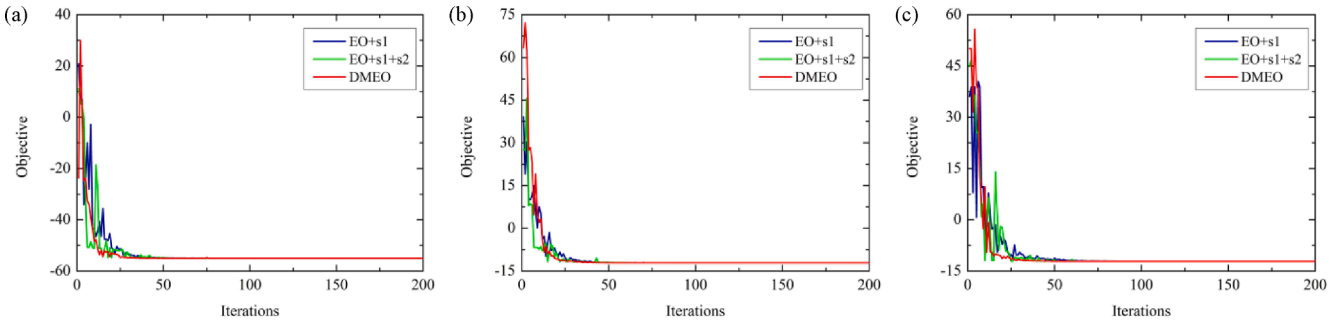

Fig. 4. The iterative histories for TC1 in different iterative stages (a) step 1 (b) step 2 (c) step 3.

Table 3
RBDO results for the high dimensional example.

Case	Algorithms	Best	Mean	Worst	SD	Times (s)	F-evaluations
TC5	PMA	−25.9607	−25.9549	−25.7783	0.0333	13.0860	2.55E4
	SORA	−25.9589	−25.9550	−25.9515	0.0024	23.3426	4.42E4
	DMEO	**−26.0867**	**−25.7578**	**−25.5176**	**0.1223**	**319.8525**	**5.24E6**
TC6	PMA	Not convergence					
	SORA	Not convergence					
	DMEO	**11.1427**	**11.4815**	**12.1835**	**0.2913**	**464.9262**	**5.13E5**

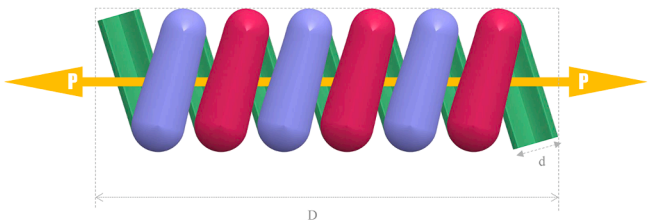

Fig. 5. Schematic figure of the tension/compression spring.

variables of the $(k-1)$th iterative step can be used as the initial population of the kth iterative step. To prove this, we denote $\boldsymbol{\theta}^k = g^{-1}(\boldsymbol{d}^k, C^k)$, and C^k is the accordingly performance function value. Then, we can expand the performance function $g^{-1}(\boldsymbol{d}^k, C^k)$ at $\boldsymbol{\theta}^{k-1} = g^{-1}(\boldsymbol{d}^{k-1}, C^{k-1})$ using Taylor expansion.

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \nabla g^{-1}(\boldsymbol{d}^{k-1}, C^{k-1})(\boldsymbol{d}^k - \boldsymbol{d}^{k-1}) + O(\left\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\right\|^2) \tag{27}$$

where ∇g^{-1} is the sensitivity. $O(\cdot)$ represents the high order small value with respect to the $\left\|\boldsymbol{d}^k - \boldsymbol{d}^{k-1}\right\|^2$. From Eq. (27), we can observe that the difference between $\boldsymbol{\theta}^k$ and $\boldsymbol{\theta}^{k-1}$ is proportional to the difference between $\boldsymbol{d}^k$ and $\boldsymbol{d}^{k-1}$. It should be noted that the optimum $\boldsymbol{d}^k$ is located at the neighborhood of optimum$\boldsymbol{d}^{k-1}$, and thus the MPTP at the $(k-1)$th iterative step is also located at the neighborhood of MPTP at the (k)th iterative step in θ-space. Thus, we can keep the optimum at the $(k-1)$th iterative step as a component of the initial population at the (k)th iterative step.

$$\boldsymbol{d}^k_{int1} = \boldsymbol{d}^{k-1}_{Nbest} \tag{28}$$

$$\boldsymbol{\theta}^k_{int1} = \boldsymbol{\theta}^{k-1}_{Nbest} \tag{29}$$

where $\boldsymbol{d}^k_{int1}$ and $\boldsymbol{\theta}^k_{int1}$ are the best solutions of design variables and MPTP at the kth iterative step. These mechanics promote exploitation capability by generating a potential candidate. The pseudocode of particle's

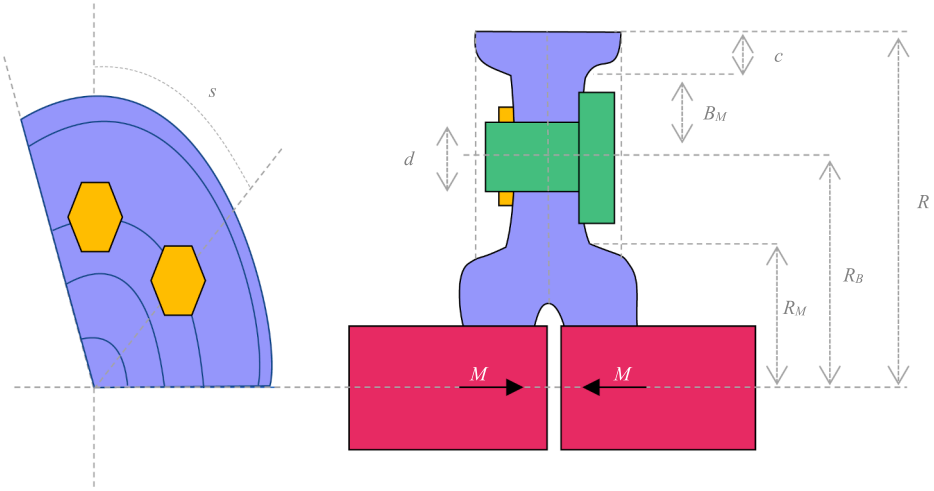

Fig. 6. Schematic figure of the bolted rim.

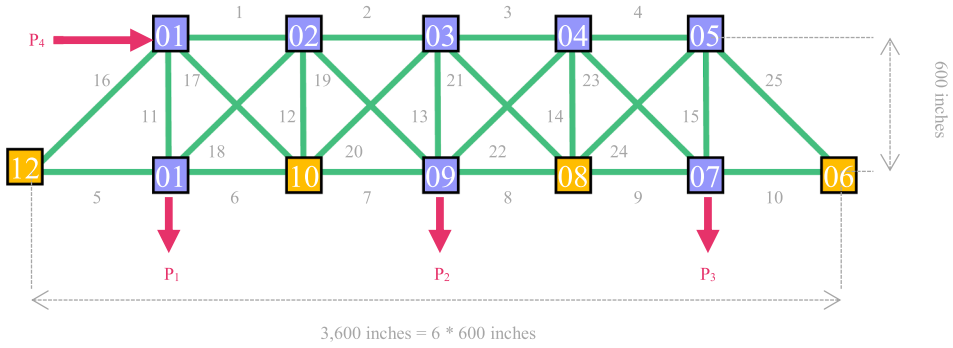

Fig. 7. Schematic figure of the truss.

memory saving strategy is in Algorithm 3.

Algorithm 3 Pseudocode of particle's memory saving strategy

iter=0; *k*=0.
% *iter* is the iterative number of AFOEO algorithm.
While *k*<Max_*k* or error< Max_error
IF *k*=0
Generate *N* population points [$\mathbf{d}_1$; $\mathbf{d}_2$; …,$\mathbf{d}_N$] by Latin hypercube sampling method.
Else
$\boldsymbol{d}_{intl}^{k} = \boldsymbol{d}_{Nbest}^{k-1}$. Generate another *N*-1 population points [$\mathbf{d}_2$; $\mathbf{d}_3$; …,$\mathbf{d}_N$] by Latin hypercube sampling method.
End If
Update the design variables to find the best population point $\boldsymbol{d}_{Nbest}^{k}$ by using the AFOEO.
For *j*=1:*ng*
IF *k*=0
Generate *N* population points [$\mathbf{d}_1$; $\mathbf{d}_2$; …,$\mathbf{d}_N$] by Latin hypercube sampling method.
Else
$\boldsymbol{\theta}_{intl}^{k} = \boldsymbol{\theta}_{Nbest}^{k-1}$. Generate another *N* population points [$\boldsymbol{\theta}_2$; $\boldsymbol{\theta}_3$; …, $\boldsymbol{\theta}_N$] by Latin hypercube sampling method.
End IF
Update the random variables to find the best population point by $\boldsymbol{\theta}_{Nbest}^{k}$ using the AFOEO.
Transform the best population point from $\boldsymbol{\theta}_{Nbest}^{k}$ to $\mathbf{x}^k$.
End For
$error = \left|C(\boldsymbol{d}^k) - C(\boldsymbol{d}^{k-1})\right| \Big/ C(\boldsymbol{d}^k)$.
End While

3.4. Parameter settings of the proposed DMEO

The parameter settings of the AFOEO are same as those of EO, which are presented in Table 1. In the local search, the fractional derivative parameter α is selected as 0.1 based on the research of Yousri et al. (2019). To ensure the global convergence capacity, the parameters of a_1 and a_2 are set to 2 and 1, respectively. The RBDO optimum can be deemed as convergence when the relative error ε of fitness value between two continuous iterations is less than 10^{-3}. The population size should be larger than the 10 × D dimension. The maximum iteration number is set to 200 for all sub-deterministic optimization problems. Because the search space of the MPTP computation algorithm is far smaller than that of the deterministic optimization, the maximum number of iterations of MPTP computation is half of that of the deterministic optimization.

3.5. Exploration and exploitation ability of EO

To sum up, the main mechanisms for performing exploitation and exploration in DMEO are as follows:

- Decoupling: decoupling, separate the deterministic optimization and reliability analysis to avoid computing MPTP for each particle. This characteristic directly promotes the computational efficiency.
- Coordinate transformation: by using polar coordinate, the complex constrained optimization model in transformed into simple unconstrained optimization model. This will avoid creating constraint violation solutions, contributing to exploitation and exploration.
- Particle's memory saving: as the progress of iteration, exploitation becomes more and more important. Therefore, in the next the sub-optimization problem, the best solution obtained from the previous sub-optimization is in the vicinity of the optimum of the current sub-

optimization, the particle's memory saving is conducive to local search, resulting in exploitation.

- Adaptive fractional-order calculus: this mechanism controls the speed of the non-elite particles moving toward the elite particle. During the iterative process, when the distance between the particles is large, the adaptive strategy, OBL, and FO calculus are aid to provide promising search direction and speed, which is also helpful for exploring unknown search spaces, i.e. improving the exploitation and exploration capacity.

3.6. Computational complexity

The computational complexity is crucial for measuring the computational time. The Big-O notation is used, which is employed to describe the worst-case performance of the DMEO (Faramarzi et al., 2020b). Unlike the deterministic optimization, the complexity of DMEO is simultaneously decided by the MPTP search and design variables update. Specifically, the particle number (N), the dimension of random variables (nr), the dimension of design variables (nd), the iterative numbers of deterministic optimization and MPTP search (Max_iter), the constraint number (ng), the maximum iteration number of the decoupling framework (Max_k), the function computation cost of MPTP search (cr), and the function computation cost of deterministic optimization (cd) should be considered.

$$O(\text{DMEO}) = O(\text{MPTP search}) + O(\text{deterministic optimization}) \tag{30}$$

Thus, the computational complexity becomes.

$$O(\text{DMEO}) = O(Max\_k \times Max\_iter \times N \times (ng \times nr + nd) + Max\_k \times Max\_iter \times N \times (cr + cd)) \tag{31}$$

From Eq. (31), it can be seen that complexity of DMEO is polynomial function. Therefore, DMEO is a very efficient algorithm.

4. Experimental study

4.1. RBDO algorithms

To validate the proposed method, several widely used benchmark test cases are used: TC1 (multimodal case with two local optima) (Deb et al., 2009), TC2 (multimodal case with three constraints) (Daum et al., 2007), TC3 (tension/compression spring design case with discrete design variable) (Mirjalili et al., 2014) and TC4 (bolted rim case with mixed variables) (Giraud-Moreau & Lafon, 2002). The details of these examples can be seen in the Appendix. The purpose of these tests is to analyze the superiority of the exploitation and exploration capacity in terms of the complex system. The numerical results are compared in terms of following metrics: success proportion, mean solution, best solution, worst solution, standard deviation, and CPU time. The success proportion is computed by the following formulations:

$$P_s = \sum_{i=1}^{n} I\left(\left\|\frac{\boldsymbol{x}_i^* - \boldsymbol{x}_i^{\text{best}}}{\boldsymbol{x}_i^*}\right\| \leqslant \varepsilon\right) \times 100\% \tag{32}$$

where I denotes the indicator function. $\boldsymbol{x}_i^{\text{best}}$ denotes the best solution of the ith computation. $\boldsymbol{x}_i^*$ denotes the reference value of best solution, which is computed by using the metaheuristic algorithms with 10,000 simulations (both inner and outer loops) and 30 runs. The value of ε is 0.01.

4.2. Comparative algorithms

Only a few evolutionary and metaheuristic algorithms have already been proposed for solving RBDO models without computing the gradient information. Therefore, this article considers five representative evolutionary and metaheuristic algorithms for solving RBDO problems in both inner and outer loops, i.e., GA, PSO, ABC, DE, GWO, and EO (Karaboga & Akay, 2009). Moreover, the classical gradient algorithm is also used for comparisons, such as PMA, SORA, SLA, and CSLA (Jeong & Park, 2017; Liang et al., 2007; Meng et al., 2018). To comprehensively investigate the effects of initial point for PMA, SORA, SLA, and CSLA, we randomly generate the initial point of design variables, and all these gradient-based RBDO algorithms are also repeatedly operated using different initial points. For gradient-based RBDO algorithms, the iteration is deemed as the converging if $\left\|\frac{d^k - d^{k-1}}{d^k}\right\| \leqslant 10^{-6}$ (outer loop) and $\left\|\frac{x^k - x^{k-1}}{x^k}\right\| \leqslant 10^{-6}$ (inner loop) are satisfied. For evolutionary and metaheuristic-based RBDO algorithms, the population size and the iterations are identical to those of the FOEO in both MPTP search and deterministic optimization, which aims to compare different algorithms in a fair way. Other parameters can be seen in Table 1.

4.3. Empirical results

For the first and second mathematical benchmark cases, all algorithms, including gradient, evolutionary and metaheuristic RBDO algorithms, are run 300 times to validate the global convergence capacity carefully. For other cases, all algorithms are run 30 times, and the mean and standard deviation of the results are recorded. All algorithms are executed using MATLAB on a personal computer with an Intel(R) Core (TM) i9-10940X CPU @ 3.30 GHz, 128.0 GB of RAM, 64-bit Windows 10 operating system, and the CPU time is adopted to demonstrate the efficiency. It should be emphasized that all convergence curves display the mean optimization results.

As can be seen from Table 2, DMEO performs significantly better than other methods on all-test problems (at least 1000 times faster than other evolutionary and metaheuristic-based RBDO algorithms in terms of CPU time). For the first and the second cases, it can be observed from Fig. 3 that they are multimodal problems. The first case has two local optima: A (−236.9867, 12.1741) and B (115.8741, −27.557380), where point A is the global optimum, as plotted in Fig. 3(a). From Table 2, the gradient-based RBDO approaches meet the convergence problem, and the success proportions of PMA, SORA, SLA, and CSLA are 66.3333%, 58.3333%, 65.3333%, and 68.3333%, respectively. For GA and ABC, since the convergence rate is slow, it also cannot converge to the point A. For DE, GWO, and EO, the best solution is less than − 12.1741; this is because the best solution violates the constraint. We can add the generation and population size to promote the performances, but it generates additional computational cost. PSO and EO exhibit superiority over other methods in terms of global convergence capacity. However, all these evolutionary and metaheuristic-based RBDO algorithms need to call the inner optimization for each individual of outer optimization, and the computational cost is very time-consuming. For DMEO, the outer optimization and inner optimization are decoupled and implemented in the order. The average maximum number of iterations of the decoupling framework Max_k is 3, which is far less than the product of population points and iteration ($Max\_iter \times N$). Therefore, the CPU time of DMEO is about 1000 times less than those of evolutionary and metaheuristic-

based RBDO algorithms. For TC2, the gradient algorithms, i.e. PMA, SORA, SLA, and CSLA, lack of ability to solve this problem. The evolutionary and metaheuristic-based RBDO algorithms demonstrate superiority, but the CPU times are too long. The proposed DMEO algorithm is the most efficient and effective method. For TC3 and TC4, similar phenomena can be found. In general, PSO, ABC, GWO, and EO show the competitiveness for global optimization. However, all these methods require too many computational costs, which seriously hinder the practical applications. The proposed DMEO method exhibits overwhelming superiority advantages in terms of efficiency and the convergence rate, which is at least 1000 times faster than evolutionary and metaheuristic-based RBDO algorithms in terms of CPU time.

Moreover, we discuss the proposed DMEO algorithm in detail based on experimental tests. In order to investigate the influence of different strategies, a comprehensive investigation of three variants of DMEO for TC1 is conducted as a representative, which includes nested double loop EO algorithms, the decoupling-based EO algorithms (EO + s1), the decoupling-based EO algorithms using particle's memory saving strategy (EO + s1 + s2), and the proposed DMEO method (EO + s1 + s2 + s3). The accordingly computational times are 3188.7092 s, 2.9948 s, 2.9933 s, 2.9921 s. It should be emphasized the double loop EO algorithms do not use a decoupling strategy, which can be not compared with other variants. From the experimental test, all three variants (EO + s1, EO + s1 + s2, and EO + s1 + s2 + s3) need to shift the constraint three times, i.e. $k = 3$, and the convergence curves are plotted in Fig. 4. It can be seen that different combinations of strategies interactively affect DMEO. The integration of above-mentioned strategies exhibits better impact in terms of the efficiency and global convergence capacity of DMEO than standalone strategy. Among them, EO + s1 + s2 + s3 is the most effective method, which means that the combination of all strategies promotes convergence the most. This phenomenon validates the reasonability of the mixed utilization of different strategies under the decoupling framework. Specifically, strategy s1 plays a more important role than strategies s2 and s3 to alleviate the computational cost, and this phenomenon is reasonable. Strategy s1 carries out the MPTP search and deterministic optimization parallelly, and thus the computational costs are only slightly slower than the deterministic optimization. Strategies s2 and s3 focus on promoting the convergence rate and global convergence capacity. In general, the integration of three strategies can make the proposed DMEO algorithm become very competitive from the aspect of efficiency, exploration and exploitation.

4.4. Validation on high-dimensional case studies

In this section, we test the performance of the proposed DMEO method on a high dimensional mathematic example (TC5) (Li et al., 2019) and a 25-bar truss design (TC6) (Ganzerli & Pantelides, 2000), the details are also listed in Appendix. Considering the nested double loop evolutionary and metaheuristic-based RBDO algorithms' unbearable computational cost, this section only uses PMA and SORA as the references, in which the accordingly maximum iterative number are 500 and 80. The initial iterative point of PMA and SORA are randomly generated to test the global convergence capacity. All algorithms, including PMA, SORA, and DMEO, are operated 30 times, and the best, mean, and worst values are listed in Table 3. Also, the standard deviations are compared.

For TC6, the performance function $g = 25 \quad \text{ksi} - \max\{\sigma_1, \sigma_2, \ldots, \sigma_{25}\}$ is a complex high dimensional non-difference problem that is computed by the finite element method. The maximum iterations and population size are also set to 500 and 80, respectively. All these methods are also carried out 30 times, and the statistical data are listed in Table 3.

The statistical results are summarized in Table 3, which can be found that the DMEO is significantly better than or comparably well to PMA and SORA. From the computational cost of the previous section, we can conclude that the double loop evolutionary and metaheuristic-based RBDO algorithms is about 1000 times slower than the proposed DMEO in terms of CPU time. In other words, it requires about 50 days a per run for TC6, which is unbearable. For TC5, since this complex RBDO problem contains multiple local optima, PMA and SORA are biased toward local search and different best solutions can be found with different initial points. DMEO achieves better global convergence. For TC6, PMA and SORA are incapable of solving this complex non-difference problem. This phenomenon demonstrates that the proposed DMEO not only achieves promising global search capacity, but also achieves high local search capacity. In summary, the proposed DMEO is highly effective for solving high-dimensional complex RBDO problems.

5. Conclusions

The target of this study was to introduce a general and efficient evolutionary and metaheuristic RBDO algorithm for the expensive RBDO problems. The optimization framework of the proposed DMEO algorithm adopts the basic concept of SORA. Comparing with the SORA, the proposed DMEO algorithm first extended the decoupling strategy into the evolutionary and metaheuristic computation field in order to guide the global search accurately. In addition, the particle's memory saving mechanism was developed to generate better local optima by making the merits of the history information. Finally, an adaptive fractional-order equilibrium optimizer strategy was developed to further promote the computational efficiency. DMEO can be considered as a substantial improvement for evolutionary and metaheuristic RBDO algorithms.

To verify the proposed DMEO, this algorithm was on several intensively utilized benchmark examples with different types: multimodal function, unimodal function, discrete design space, multiple constraints, and non-difference problem. The results showed that all three strategies could accelerate the convergence rate of the proposed DMEO. In addition, an overall performance comparison among DMEO, four popular gradient RBDO algorithms (including PMA, SORA, SLA, and CSLA), and six evolutionary and metaheuristic algorithms (including GA, PSO, ABC, DE, GWO, and EO) were implemented. The results showed that the DMEO achieves an extremely faster converge rate and higher convergence capacity. The superiority of the DMEO over another gradient, evolutionary, and metaheuristic algorithms demonstrates that the proposed universal decoupling evolutionary and metaheuristic DMEO framework is very effective.

CRediT authorship contribution statement

Zeng Meng: Conceptualization, Methodology, Software, Validation, Funding acquisition, Writing – original draft, Writing – review & editing. **Ali Rıza Yıldız:** Software, Writing – review & editing. **Seyedali Mirjalili:** Software, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The supports of the National Natural Science Foundation of China (Grant No. 11972143), the Fundamental Research Funds for the Central Universities of China (Grant No. JZ2020HGPA0112), and the Foundation of State Key Laboratory of Structural Analysis for Industrial Equipment from Dalian University of Technology (Grant No. GZ21101) are much appreciated.

Appendix

Mathematical example with two local optima

The first case is a mathematical example with two local optima, which is explored by the Deb et al. (2009). It contains two normal random variables $\times$ and y, while their standard deviations are$\sigma_x = \sigma_y = 10$.The RBDO model is expressed as follows:

$$\begin{aligned} &\min && -\mu_y \\ \text{s.t.}\quad &\text{Prob}[g_i(x,y)\geqslant 0]\leqslant \Phi(-\beta_i^t), && i=1,\quad 2,\quad 3 \\ &\text{where} && g_1(x,y) = x^2 - 1000y \\ & && g_2(x,y) = y - x + 200 \\ & && g_3(x,y) = x - 3y + 400 \\ & && -400\leqslant x,y\leqslant 300 \\ & && \beta_1^t = \beta_2^t = \beta_3^t = 4 \end{aligned} \tag{33}$$

Multimodal example

The second case is a multimodal example with many local optima, which can be seen in Daum et al. (2007). Both of two random variables follow the normal distribution, while their standard deviations are$\sigma_x = \sigma_y = 0.3$. The RBDO model is as follows:

$$\begin{aligned} &\min && \sin(3\mu_{x_1}^2) + \sin(3\mu_{x_2}^2) + \mu_{x_1} + \mu_{x_2} \\ \text{s.t.}\quad &\text{Prob}[g_i(\boldsymbol{x})\geqslant 0]\leqslant \Phi(-\beta^t), && i=1,\quad 2,\quad 3 \\ & && 0\leqslant d_1\leqslant 10, \qquad 0\leqslant d_2\leqslant 10 \\ &\text{where} && g_1(\boldsymbol{x}) = x_1^2 x_2/20 - 1 \\ & && g_2(\boldsymbol{x}) = (x_1 + x_2 - 5)^2/30 - (x_1 - x_2 - 12)^2/120 - 1 \\ & && g_3(\boldsymbol{x}) = 80/(x_1^2 + 8x_2 + 5) - 1 \\ & && \beta_1^t = \beta_2^t = \beta_3^t = 3 \end{aligned} \tag{34}$$

Tension/Compression spring

As shown in Fig. 5, a tension/compression spring subjected to the axially load is designed, and the volume is considered as the objective function. The design variables contain the coil diameter x_2, wire diameter x_1, and the spring coils number x_3, while the spring coils number is considered as the integer number. The RBDO model is formulated as follows:

$$\begin{aligned} &\min && (\mu_{x_1} + 2)\mu_{x_2}\mu_{x_3}^2 \\ \text{s.t.}\quad &\text{Prob}[g_i(\boldsymbol{x})\geqslant 0]\leqslant \Phi(-\beta^t), && i=1,\quad 2,\quad 3 \\ & && 0.05\leqslant \mu_{x_1}\leqslant 2, \qquad 0.25\leqslant \mu_{x_2}\leqslant 1.3, 2\leqslant \mu_{x_3}\leqslant 15 \\ &\text{where} && g_1(\boldsymbol{x}) = x_2^3 x_3/71,785x_1^4 - 1 \\ & && g_2(\boldsymbol{x}) = 1 - 1/5108x_1^2 + x_1x_2/12,566(x_1x_1^3 - x_1^4) - 4x_2^2 \\ & && g_3(\boldsymbol{x}) = 140.45x_1/x_2^2x_3 \\ & && g_4(\boldsymbol{x}) = 1 - x_2 - x_1/1.5 \\ & && \beta_1^t = \beta_2^t = \beta_3^t = 3 \end{aligned} \tag{35}$$

A bolted rim

As plotted in Fig. 6, a bolted rim is optimized with both discrete and continuous variables, which contains eleven probabilistic constraints. The bolts of diameter d and its number N are selected as discrete variables, and the radius R_B and torque M are continuous design variables. The RBDO model is as follows:

$$
\begin{aligned}
&\text{find} && \boldsymbol{d} = [R_B, M, d, N]^{\mathrm{T}} \\
&\min && f(x) = \beta_1\left(\frac{N}{N_m}\right) + \beta_2\left(\frac{R_B + \phi_4(d) + c}{R_m}\right) + \beta_3\left(\frac{M}{M_T}\right) \\
&\text{s.t.} && \mathrm{Prob}[g_i(\boldsymbol{x}) \geqslant 0] \leqslant \Phi(-\beta_i^t); \qquad i = 1 - 11 \\
& && g_1(\boldsymbol{x}) = \frac{\alpha M}{N R_B K(d)} - 1 \leqslant 0,\ g_2(\boldsymbol{x}) = 1 - \frac{2\pi R_B}{\phi_5(d) N} \leqslant 0 \\
& && g_3(\boldsymbol{x}) = 1 - \frac{R_B}{\phi_4(d)} + R_M \leqslant 0,\ g_4(\boldsymbol{x}) = N - N_{\max} \leqslant 0 \\
& && g_5(\boldsymbol{x}) = R_B - R_{\max} \leqslant 0,\ g_6(\boldsymbol{x}) = N_M - N \leqslant 0 \\
& && g_7(\boldsymbol{x}) = R_M - R_B \leqslant 0,\ g_8(\boldsymbol{x}) = M - M_{\max} \leqslant 0 \\
& && g_9(\boldsymbol{x}) = M_T - M \leqslant 0,\ g_{10}(\boldsymbol{x}) = d - 24 \leqslant 0 \\
& && g_{11}(\boldsymbol{x}) = 6 - d \leqslant 0 \\
& && \beta_1^t = \beta_2^t = \cdots = \beta_{11}^t = 3.0 \\
& && R_B \widetilde{N}(,), M \widetilde{N}(,) \\
& && K(d) = \frac{0.9 f_m R_e \pi (\phi_1(d))^2}{4\sqrt{1 + 3(0.16\phi_3(d) f_1 / \phi_1(d))^2}} \\
& && M_T = 40Nm,\ M_{\max} = 1000Nm \\
& && f_m = 0.15,\ f_1 = 0.15 \\
& && \alpha = 1.5,\ R_e = 627MPa \\
& && N_M = 8,\ N_{\max} = 100, \\
& && R_M = 50mm,\ R_{\max} = 1000mm \\
& && 6 \leqslant d \leqslant 24,\ 8 \leqslant N \leqslant 100, \\
& && 50 \leqslant R_B \leqslant 100,\ 40 \leqslant M \leqslant 100
\end{aligned}
\tag{36}
$$

A high dimensional mathematical example

This example contains 18 random variables, where the means are considered as the design variables. All random variables have the same standard deviation 0.1. The RBDO model is represented by.

$$
\begin{aligned}
&\text{find} \qquad \boldsymbol{d} = [d_1, d_2, \ldots, d_{18}]^{\mathrm{T}} \\
&\min \qquad f(x) = \sum_{i=14}^{18} d_i - \sum_{j=1}^{13} d_j \\
&\text{s.t.} \qquad \mathrm{Prob}[g_i(\boldsymbol{x}) \geqslant 0] \leqslant \Phi(-\beta_i^t); \qquad i = 1 - 12 \\
&g_1 = 3x_5^2 + x_7^2 + 2x_{13}^2 + x_1x_{10} + 2x_4x_6 + 3x_2x_9 + 4x_3x_8 + 5x_{11}x_{12} + 0.001\cdot(x_{14} + x_{15}) \\
&\qquad -x_{16}x_{17} - x_{18} - 210 \\
&g_2 = 2x_2^2 + 4x_3^2 + 5x_1x_{10} + 4x_4x_5 + 3x_6x_7 + 2x_9x_8 + x_{11}x_{12} + x_{13}x_{12} + 0.001\cdot(x_{14} + x_{15}) \\
&\qquad -x_{16} - 2x_{17}^2 - x_{18}^2 - 200 \\
&g_4 = 2x_2^2 + 1.5x_3^2 + 2x_4^2 + x_7^2 + x_1x_{13} + 2x_8x_{12} + 3x_6x_{11} + 2x_4x_5 + 3x_{10}x_3 + 3x_1x_9 \\
&\qquad +0.001\cdot(x_{14} + x_{15}) - x_{16}x_2 - x_{17} - 180 \\
&g_5 = 2x_1^2 + 2x_6^2 + 3x_7^2 + x_{11}^2 + x_{12}x_{13} + 2x_8x_{10} + 3x_3x_9 + 2x_4x_5 + x_1x_{13} + 0.001\cdot(x_{14} + x_{15}) \\
&\qquad -x_{17}x_{18} - x_{16} - 180 \\
&g_6 = 10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + x_{10} + 6x_{11} + 4x_{12} + 2x_{13} \\
&\qquad +0.001\cdot(x_{14} + x_{15}) - 2x_{16} - 3x_{17} - x_{18} - 200 \\
&g_7 = 10x_1 + 8x_2 + 6x_3 + 4x_4 + 2x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + x_{10} + x_{11} + 3x_{12} + 5x_{13} \\
&\qquad +0.001\cdot(x_{14} + x_{15}) - x_{16} - x_{17} - x_{18} - 160 \\
&g_8 = 4x_1 + 3x_2 + 2x_3 + x_4 + 10x_5 + 9x_6 + 8x_7 + 7x_8 + 6x_9 + 5x_{10} + 4x_{11} + 3x_{12} + 2x_{13} \\
&\qquad +0.001\cdot(x_{14} + x_{15}) - 5x_{16} - 2x_{17} - 3x_{18} - 180 \\
&g_9 = \sum_{i=1}^{13} ix_i + 0.001\cdot(x_{14} + x_{15}) - 3x_{16} - 2x_{17} - x_{18} - 300 \\
&g_{10} = \sum_{i=1}^{13} ix_{14-i} + 0.001\cdot(x_{14} + x_{15}) - x_{16} - 3x_{17} - x_{18} - 280 \\
&g_{11} = 0.001\cdot\sum_{i=1}^{13} x_i - 10\cdot x_{14}^2\cdot x_{15} + 200 \\
&g_{12} = 0.001\cdot\sum_{i=1}^{13} x_i - 4\cdot(x_{14} + x_{15} - 5)^2 - (x_{14} - x_{15} - 12)^2 + 120 \\
&\beta_1^t = \beta_2^t = \cdots = \beta_{12}^t = 3.0 \\
&2 \leqslant d_i \leqslant 4; \qquad i = 1 - 18
\end{aligned}
\tag{37}
$$

A 25-bar truss

The weight of a 25-bar truss is minimized, which has 24 degrees of freedom, as shown in Fig. 7. The elasticity modulus of these bars is selected as 29,000 ksi. The lengths of the bars are 600 in, and the heights are 600 in. All truss areas are selected as the design variables. There are four external forces: $P_1 = P_3 = 400$ kip, $P_2 = 500$ kip, and $P_4 = 300$ kip. All forces are considered as random variables, where the coefficients of variation are 0.1. The RBDO formulation is expressed as follows:

$$
\begin{aligned}
&\text{find} \qquad \mathbf{d} = [A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10},\ A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}, A_{17}, A_{18}, A_{19},\ A_{20}, A_{21}, A_{22}, A_{23}, A_{24}, A_{25}]^T \\
&\min \qquad V(\mathbf{d}) \\
&\text{s.t.} \qquad \mathrm{Prob}[g(\mathbf{d}, P_1, P_2, P_3, P) \geqslant 0] \leqslant \Phi(-\beta^t); \\
&\text{where} \quad g = 25 \quad \text{ksi} - \max\{\sigma_1, \sigma_2, \ldots, \sigma_{25}\} \\
&\qquad\qquad \beta^t = 3, \quad j = 1, \ldots, 25 \\
&\qquad\qquad 0.1 \leqslant A_i \leqslant 25, \quad i = 1, \ldots, 25
\end{aligned}
\tag{38}
$$

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2022.117640.

References

Aoues, Y., & Chateauneuf, A. (2010). Benchmark study of numerical methods for reliability-based design optimization. *Structural and Multidisciplinary Optimization, 41*, 277–294.

Bai, B., Zhang, J., Wu, X., Zhu, G. W., & Li, X. (2021). Reliability prediction-based improved dynamic weight particle swarm optimization and back propagation neural network in engineering systems. *Expert Systems with Applications, 177*, Article 114952.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*, 646–657.

Casciati, S. (2014). Differential evolution approach to reliability-oriented optimal design. *Probabilistic Engineering Mechanics, 36*, 72–80.

Chen, Z., Qiu, H., Gao, L., & Li, P. (2013). An optimal shifting vector approach for efficient probabilistic design. *Structural and Multidisciplinary Optimization, 47*, 905–920.

Cho, T. M., & Lee, B. C. (2011). Reliability-based design optimization using convex linearization and sequential optimization and reliability assessment method. *Structural Safety, 33*, 42–50.

Dan, S. (2009). Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation, 12*, 702–713.

das Neves Carneiro, G., & Conceição António, C.. (2019). Global optimal reliability index of implicit composite laminate structures by evolutionary algorithms. *Structural Safety, 79*, 54–65.

Das, S., & Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation, 15*, 4–31.

Daum, D. A., Kalyanmoy, D., & Branke, J. (2007). *Reliability-based optimization for multiple constraints with evolutionary algorithms* (pp. 911–918). New York: IEEE Congress on Evolutionary Computation.

Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A. K., & Padmanabhan, D. (2009). Reliability-Based Optimization Using Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation, 13*, 1054–1074.

Du, X., & Chen, W. (2004). Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design, 126*, 225–233.

Elegbede, C. (2005). Structural reliability assessment based on particles swarm optimization. *Structural Safety, 27*, 171–186.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures, 110–111*, 151–166.

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020a). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications, 152*, Article 113377.

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020b). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191*, Article 105190.

Ganzerli, S., & Pantelides, C. P. (2000). Optimum structural design via convex model superposition. *Computers & Structures, 74*, 639–647.

Giraud-Moreau, L., & Lafon, P. (2002). A comparison of evolutionary algorithms for mechanical design components. *Engineering Optimization, 34*, 307–322.

Guirguis, D., et al. (2020). Evolutionary black-box topology optimization: Challenges and promises. *IEEE Transactions on Evolutionary Computation, 24*, 613–633.

Guo, X., Bai, W., Zhang, W., & Gao, X. (2009). Confidence structural robust design and optimization under stiffness and load uncertainties. *Computer Methods in Applied Mechanics and Engineering, 198*, 3378–3399.

Ho-Huu, V., Duong-Gia, D., Vo-Duy, T., Le-Duc, T., & Nguyen-Thoi, T. (2018). An efficient combination of multi-objective evolutionary optimization and reliability analysis for reliability-based design optimization of truss structures. *Expert Systems with Applications, 102*, 262–272.

Jensen, H. A., Jerez, D. J., & Valdebenito, M. (2020). An adaptive scheme for reliability-based global design optimization: A Markov chain Monte Carlo approach. *Mechanical Systems and Signal Processing, 143*, Article 106836.

Jeong, S. B., & Park, G. J. (2017). Single loop single vector approach using the conjugate gradient in reliability based design optimization. *Structural and Multidisciplinary Optimization, 55*, 1329–1344.

Jiang, C., Qiu, H., Li, X., Chen, Z., Gao, L., & Li, P. (2019). Iterative reliable design space approach for efficient reliability-based design optimization. *Engineering with Computers, 36*, 151–169.

Jung, Y., Cho, H., & Lee, I. (2019). Reliability measure approach for confidence-based design optimization under insufficient input data. *Structural and Multidisciplinary Optimization, 60*, 1967–1982.

Jung, Y., Cho, H., & Lee, I. (2020). Intelligent initial point selection for MPP search in reliability-based design optimization. *Structural and Multidisciplinary Optimization, 62*, 1809–1820.

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*, 108–132.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471.

Keshtegar, B., & Chakraborty, S. (2018). An efficient-robust structural reliability method by adaptive finite-step length based on Armijo line search. *Reliability Engineering & System Safety, 172*, 195–206.

Lee, I., Choi, K. K., Du, L., & Gorsich, D. (2008). Dimension reduction method for reliability-based robust design optimization. *Computers & Structures, 86*, 1550–1562.

Lee, I., Choi, K. K., & Gorsich, D. (2010). Sensitivity analyses of FORM-based and DRM-based performance measure approach (PMA) for reliability-based design optimization (RBDO). *International Journal for Numerical Methods in Engineering, 82*, 26–46.

Li, F., Cai, X., Gao, L., & Shen, W. (2021). A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Transactions on Cybernetics, 51*, 1390–1402.

Li, M., Sadoughi, M., Hu, C., Hu, Z., Eshghi, A. T., & Lee, S. (2019). High-dimensional reliability-based design optimization involving highly nonlinear constraints and computationally expensive simulations. *Journal of Mechanical Design, 141*, Article 051402.

Liang, J., Mourelatos, Z. P., & Nikolaidis, E. (2007). A single-loop approach for system reliability-based design optimization. *Journal of Mechanical Design, 129*, 1215–1224.

Liu, K., Novelino, L. S., Gardoni, P., & Paulino, G. H. (2020). Big influence of small random imperfections in origami-based metamaterials. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 476*, 20200236.

Liu, K., & Paulino, G. H. (2019). Tensegrity topology optimization by force maximization on arbitrary ground structures. *Structural and Multidisciplinary Optimization, 59*, 2041–2062.

Liu, K., Paulino, G. H., & Gardoni, P. (2016a). Reliability-based topology optimization using a new method for sensitivity approximation-application to ground structures. *Structural and Multidisciplinary Optimization, 54*, 553–571.

Liu, K., Paulino, G. H., & Gardoni, P. (2016b). Segmental multi-point linearization for parameter sensitivity approximation in reliability analysis. *Structural Safety, 62*, 101–115.

Liu, X., Wang, X., Xie, J., & Li, B. (2019). Construction of probability box model based on maximum entropy principle and corresponding hybrid reliability analysis approach. *Structural and Multidisciplinary Optimization, 61*, 599–617.

Liu, X., Wu, Y., Wang, B., Yin, Q., & Zhao, J. (2018). An efficient RBDO process using adaptive initial point updating method based on sigmoid function. *Structural and Multidisciplinary Optimization, 58*, 2583–2599.

Meng, Z., Guo, L. B., & Wang, X. (2022). A general fidelity transformation framework for reliability-based design optimization with arbitrary precision. *Structural and Multidisciplinary Optimization, 65*, 14. https://doi.org/10.1007/s00158-021-03091-y

Meng, Z., Li, G., Wang, X., Sait, S. M., & Yıldız, A. R. (2021). A comparative study of metaheuristic algorithms for reliability-based design optimization problems. *Archives of Computational Methods in Engineering, 28*, 1853–1869.

Meng, Z., Li, G., Yang, D., & Zhan, L. (2017). A new directional stability transformation method of chaos control for first order reliability analysis. *Structural and Multidisciplinary Optimization, 55*, 601–612.

Meng, Z., Yang, D., Zhou, H., & Wang, B. P. (2018). Convergence control of single loop approach for reliability-based design optimization. *Structural and Multidisciplinary Optimization, 57*, 1079–1091.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Pedroso, D. M. (2017). FORM reliability analysis using a parallel evolutionary algorithm. *Structural Safety, 65*, 84–99.

Piotrowski, A. P., Napiorkowski, J. J., & Piotrowska, A. E. (2020). Population size in particle swarm optimization. *Swarm and Evolutionary Computation, 58*, Article 100718.

Poli, R., Kennedy, J., & Blackwell, T. (2007). *Particle swarm optimization. Swarm intelligence, 1*, 33–57.

Rackwitz, R. (2001). Reliability analysis—a review and some perspectives. *Structural Safety, 23*, 365–395.

Rodrigues, M. C., de Lima, B. S. L. P., & Guimarães, S. (2016). Balanced ranking method for constrained optimization problems using evolutionary algorithms. *Information Sciences, 327*, 71–90.

Rosario, Z., Fenrich, R. W., & Iaccarino, G. (2019). Cutting the double loop: Theory and algorithms for reliability-based design optimization with parametric uncertainty. *International Journal for Numerical Methods in Engineering, 118*, 718–740.

Schuëller, G. I., & Jensen, H. A. (2008). Computational methods in optimization considering uncertainties – An overview. *Computer Methods in Applied Mechanics and Engineering, 198*, 2–13.

Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* (pp. 695–701).

Torii, A. J., Lopez, R. H., Beck, A. T., & Miguel, L. F. F. (2019). A performance measure approach for risk optimization. *Structural and Multidisciplinary Optimization, 60*, 927–947.

Tu, J., Choi, K. K., & Park, Y. H. (1999). A new study on reliability-based design optimization. *Journal of Mechanical Design, 121*, 557–564.

Tu, J., Choi, K. K., & Park, Y. H. (2001). Design potential method for robust system parameter design. *AIAA journal, 39*, 667–677.

Valdebenito, M. A., & Schuëller, G. I. (2010). Reliability-based optimization considering design variables of discrete size. *Engineering Structures, 32*, 2919–2930.

Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C., & Saldanha, R. R. (2001). Improvements in genetic algorithms. *IEEE Transactions on Magnetics, 37*, 3414–3417.

Vořechovský, M. (2015). Hierarchical refinement of Latin hypercube samples. *Computer-Aided Civil and Infrastructure Engineering, 30*, 394–411.

Wang, Y., Hao, P., Guo, Z., Liu, D., & Gao, Q. (2019). Reliability-based design optimization of complex problems with multiple design points via narrowed search region. *Journal of Mechanical Design, 142*, 1–62.

Wang, J., Wen, Y., Gou, Y., Ye, Z., & Chen, H. (2017). Fractional-order gradient descent learning of BP neural networks with Caputo derivative. *Neural Networks, 89*, 19–30.

Wang, J., Zhou, Z. J., Hu, C. H., Tang, S. W., & Cao, Y. (2021). A new evidential reasoning rule with continuous probability distribution of reliability. *IEEE Transactions on Cybernetics*, 1–13.

Yang, I., & Hsieh, Y. H. (2011). Reliability-based design optimization with discrete design variables and non-smooth performance functions: AB-PSO algorithm. *Automation in Construction, 20*, 610–619.
Yang, M. D., Zhang, D. Q., Cheng, C., & Han, X. (2021). Reliability-based design optimization for RV reducer with experimental constraint. *Structural and Multidisciplinary Optimization, 63*, 2047–2064.
Yang, M., Zhang, D., & Han, X. (2020). New efficient and robust method for structural reliability analysis and its application in reliability-based design optimization. *Computer Methods in Applied Mechanics and Engineering, 366*, Article 113018.
Yaseen, Z. M., & Keshtegar, B. (2018). Limited descent-based mean value method for inverse reliability analysis. *Engineering with Computers, 35*, 1237–1249.
Yi, J., Bai, J., He, H., Zhou, W., & Yao, L. (2020). A multifactorial evolutionary algorithm for multitasking under interval uncertainties. *IEEE Transactions on Evolutionary Computation, 24*, 908–922.
Yi, P., Cheng, G., & Jiang, L. (2008). A sequential approximate programming strategy for performance-measure-based probabilistic structural design optimization. *Structural Safety, 30*, 91–109.
Youn, B. D., Choi, K. K., & Park, Y. H. (2003). Hybrid analysis method for reliability-based design optimization. *Journal of Mechanical Design, 125*, 221–232.
Youn, B. D., Choi, K. K., Yang, R. J., & Gu, L. (2004). Reliability-based design optimization for crashworthiness of vehicle side impact. *Structural and Multidisciplinary Optimization, 26*, 272–283.
Yousri, D., AbdelAty, A. M., Radwan, A. G., Elwakil, A. S., & Psychalinos, C. (2019). Comprehensive comparison based on meta-heuristic algorithms for approximation of the fractional-order Laplacian sα as a weighted sum of first-order high-pass filters. *Microelectronics Journal, 87*, 110–120.
Yousri, D., Elaziz, M. A., & Mirjalili, S. (2020). Fractional-order calculus-based flower pollination algorithm with local search for global optimization and image segmentation. *Knowledge-Based Systems, 197*, Article 105889.
Zaeimi, M., & Ghoddosian, A. (2018). Structural reliability assessment based on the improved constrained differential evolution algorithm. *Periodica Polytechnica Civil Engineering, 62*(2), 494–507.
Zhao, H., Ru, Z., Chang, X., & Li, S. (2015). Reliability analysis using chaotic particle swarm optimization. *Quality and Reliability Engineering International, 31*, 1537–1552.
Zhao, H., Zhu, C., & Ru, Z. (2018). Reliability analysis based on artificial bee colony (ABC) and its application in geotechnical engineering. *The Open Civil Engineering Journal, 12*, 96–107.
Zhong, C., & Li, G. (2022). Comprehensive learning Harris hawks-equilibrium optimization with terminal replacement mechanism for constrained optimization problems. *Expert Systems with Applications, 192*, Article 116432.
Zhong, C., Wang, M., Dang, C., & Ke, W. (2020). Structural reliability assessment by salp swarm algorithm–based FORM. *Quality and Reliability Engineering International, 36*, 1224–1244.
Zhu, S. P., Keshtegar, B., Chakraborty, S., & Trung, N. T. (2020). Novel probabilistic model for searching most probable point in structural reliability analysis. *Computer Methods in Applied Mechanics and Engineering, 366*, Article 113027.