Multi-objective SHADE with manta ray foraging optimizer for structural design problems

Changting Zhong[a], Gang Li[a,*], Zeng Meng[b], Haijiang Li[c,**], Wanxin He[a]

[a] Department of Engineering Mechanics, State Key Laboratory of Structural Analyses for Industrial Equipment, Dalian University of Technology, Dalian 116024, China

[b] School of Civil Engineering, Hefei University of Technology, Hefei 230009, China

[c] BIM for Smart Engineering Centre, Cardiff School of Engineering, Cardiff University, Queen's Buildings, Cardiff CF24 3AA, Cardiff, CF24 3AA, Whales, UK

Emails: Changting Zhong, zhongct@dlut.edu.cn;

Gang Li (corresponding author), ligang@dlut.edu.cn; Zeng Meng, mengz@hfut.edu.cn;

Haijiang Li (joint corresponding author), lih@cardiff.ac.uk; Wanxin He, hewanxin@dlut.edu.cn;

Abstract: This paper presents a hybrid multi-objective success history-based parameter adaptive differential evolution (SHADE) with manta ray foraging optimizer (MRFO) for structural design problems, called MO-SHADE-MRFO. In the proposed algorithm, the updating rules of SHADE, a variant of differential evolution with great performance, are combined with the operators from MRFO, a recent swarm-based metaheuristic algorithm inspired from the manta ray with cyclone, chain and somersault foraging behaviors, which can balance the exploration and exploitation of the algorithm for structural design problems. Furthermore, MO-SHADE-MRFO utilizes the external archive to save and update the obtained Pareto fronts during the optimization process. The proposed algorithm is verified by multi-objective truss optimization problems with two objectives of minimizing the structural weight and the compliance, including 10-bar, 25-bar, 37-bar, 120-bar, 200-bar and 942-bar truss problems. Moreover, 9 different multi-objective metaheuristic algorithms are implemented to compare with the proposed algorithm, where three metrics are used to measure the performance of the algorithms, including hypervolume (HV), inverted generational distance (IGD), and spacing-to-extent (STE). According to the experimental results, MO-SHADE-MRFO can provide the best statistical values of HV, IGD and STE in most cases, ranking the first among the compared algorithms. Besides, the proposed algorithm also gives well-distributed Pareto solutions for the tested problems, indicating the effectiveness of the hybrid mechanism of SHADE and MRFO.

Keywords: Structural design; Multi-objective problem; Metaheuristics; Success history-based parameter adaptive differential evolution; Manta ray foraging optimizer

Abbreviations:

Name	Description
MOPs	Multi-objective optimization problems
PFs	Pareto fronts
NFL	No free lunch
PSO	Particle swarm optimization
GA	Genetic algorithm
DE	Differential evolution
SHADE	Success history-based parameter adaptive differential evolution
SHAMODE	Success history-based parameter adaptive multi-objective differential evolution
MRFO	Manta ray foraging optimization
MOPSO	Multi-objective particle swarm optimization
NSGA-II	Non-dominated sorting genetic algorithm
MOEA/D	Multi-objective evolutionary algorithm based on decomposition
MOGOA	Multi-objective grasshopper optimization algorithm
MOMVO	Multi-objective multi-verse optimization
MOWCA	Multi-objective water cycle algorithm
MOSSA	Multi-objective salp swarm algorithm
UPSEMOA	Unrestricted population size evolutionary multi-objective optimization algorithm
MO-SHADE-MRFO	Multi-objective hybrid SHADE and MRFO
HV	Hypervolume
IGD	Inverted generational distance
STE	Spacing-to-extent
SD	Standard deviation
FR	Friedman test rank

1 Introduction

The optimal truss design is a major area of interest in the field of structural design [1]. According to the number of objective functions, truss design problems can be divided into single-objective and multi-objective optimization problems. The objectives generally include mass, displacement, compliance, natural frequencies, and so on. The design variables can be classified as shapes, geometry sizes and topology. The topology variables determine the initial layout of a truss structure, while sizing and shape variables define the element cross-sectional areas and nodal positions, respectively. The constraints of truss structure usually include stress, displacement, bifurcation buckling, and natural frequencies, which often result in non-convex feasible regions and make designers struggle to search for the optimal solution in truss design. The optimal truss design is a very challenging task usually solved by the cumbersome trial-and-error in the practical engineering design, thus we need to develop more advanced artificial intelligence techniques [2][3], and the application of the optimization technique is a better choice.

In general, two classes of optimizers can be utilized for truss optimization problems, namely, gradient-based algorithms [4] and metaheuristic algorithms [5]. The gradient-based algorithms are efficient in the optimization process but less effective in many cases of truss design problems due to complicated coding and derivative dependence, and especially the difficulties in handling multi-modal or discrete optimization problems. On the other hand, the metaheuristic algorithms are derivative-free with high convergence capacity for highly nonlinear optimization problems. Many reports show that metaheuristic algorithms can be employed to solve almost any type of optimization problems for the advantages of simplicity, flexibility, and derivative independence, and has increasing popularity in the optimal truss design [6][7].

The basic concept of metaheuristic algorithms is to mimic the stochastic behavior of natural systems, which can balance the exploration and exploitation to search the design space effectively. Over the past decades, a variety of metaheuristic algorithms have been developed and successfully applied in different optimization problems, including particle swarm optimization (PSO) [8], genetic algorithm (GA) [9], differential evolution (DE) [10], bat algorithm [11], grey wolf optimizer [12], salp swarm algorithm [13], Harris hawks optimization [14], equilibrium optimizer [15][16], marine predator algorithm [17], beluga whale optimization [18], and so on. However, due to the inherent randomness in the metaheuristic algorithms, the performance of metaheuristic algorithms may be unstable for different optimization problems. Besides, some of them may get stuck in the local optimum due to the poor exploration capacity in solving complex optimization problems. This motivates the development of the self-adaptive metaheuristic algorithm, which can automatically adjust the algorithmic parameters to treat specific optimization problems. Pholdee and Bureerat [19] investigated several top self-adaptive metaheuristic algorithms, including the winners of annual completion at the congress on evolutionary computation (CEC competitions), to solve truss optimization problems. The results demonstrate that the top

metaheuristics can provide good performance in truss optimization problems, and novel powerful metaheuristic algorithms could be noteworthy for truss optimization problems.

The single-objective optimal truss design accounts for the major research of truss optimization, with the common objective function of minimizing the structural weight/mass with the constraints of stress, displacements, or frequencies. A great many of metaheuristics have been investigated on such a single-objective design problem. Kaveh and Talatahari [20] introduced the imperialist competitive algorithm to find the optimal design of skeletal structures and illustrated the performance of the algorithm using two space trusses and two frame structures. Kaveh and Mahdavi [21] developed the colliding bodies optimization algorithm to solve optimization problems of the benchmark truss structures. Mortazavi et al. [22] developed an integrated PSO with fly-back mechanism and weight particle concept for sizing and layout of truss structures, which is competitive with several state-of-the-art metaheuristic algorithms. Farshchin et al. [23] proposed the multi-class teaching-learning-based optimization to solve shape and sizing truss optimization problems, under the constraints of multiple natural frequencies. Degertekin et al. [24] employed the parameter-free Jaya algorithm to minimize the weight of truss structures under natural frequency constraints. Moreover, other modifications of metaheuristic in solving single-objective truss design problems have been developed, including hybrid optimality criterion and GA [25], chaotic firefly algorithms [26], Newton metaheuristic algorithm [27], discrete advanced Jaya algorithm [28], game theory-based Jaya algorithm [29], peloton dynamics optimization algorithm [30], enhanced forensic-based investigation [31], etc. Nevertheless, when it comes to consider multi-objective optimization problems, these algorithms may not provide good performance because of the high requirements of balance in exploration and exploitation.

Compared to the abundant single-objective cases, there has been much less research work in multi-objective truss optimization, due to the characteristics of conflicting objectives, nonlinear constraints, multi-modality, and non-convex feasible regions [32]. However, multi-objective optimization is the more common situation than single-objective optimization in the real-world applications, with advantages of decision-making to handle the multiple objectives. In the multi-objective truss optimization, the selection of the algorithm is critical to influence the obtained solutions. According to the general classification of multi-objective metaheuristic algorithms, they can be divided into three parts: Pareto-based algorithm, indicator-based algorithm, and decomposition-based algorithms, including several well-known algorithms: non-dominated sorting genetic algorithm II (NSGA-II) [33], multiobjective PSO (MOPSO) [34], multiobjective evolutionary algorithm based on decomposition (MOEA/D) [35]. Moreover, several recently developed multi-objective metaheuristic algorithms have also attracted much attention, such as multi-objective grasshopper optimization algorithm (MOGOA) [36], multi-objective multi-verse optimization (MOMVO) [37], multi-objective water cycle algorithm (MOWCA) [38], multi-objective salp swarm algorithm (MOSSA) [39], unrestricted population size

evolutionary multi-objective optimization algorithm (UPSEMOA) [40]. Although, these algorithms can provide good performance in solving multi-objective optimization problems, some of them may stuck in truss optimization problems due to the imbalance between the exploration and exploitation in the updating rules. Researchers developed several metaheuristic algorithms for truss optimization, such as success history-based adaptive multi-objective differential evolution (SHAMODE) and SHAMODE with whale optimization (SHAMODE-WOA) [41][42], outperforming the compared algorithms in terms of measure indices. However, results also showed that no algorithm can outperform compared algorithms for all truss optimization problems, following the No Free Lunch (NFL) theory [43], which motivates researchers to develop more novel and high-quality algorithms for multi-objective truss optimization problems.

This paper presents a hybrid multi-objective metaheuristic algorithm for truss optimization problems, called MO-SHADE-MRFO, combined with the updating rules from the success history-based parameter adaptive differential evolution (SHADE) and manta ray foraging optimizer (MRFO). SHADE is a variant of differential evolution with high-performance [44], achieving the 3rd rank in the CEC 2014 competition. MRFO is a novel swarm-based metaheuristic algorithm, inspired from the chain, cyclone and somersault behaviors of the manta rays [45]. SHADE and MRFO are very popular in the field of optimization, which can provide good performance for different optimization problems [46][47][48]. The MO-SHADE-MRFO utilizes the external archive to save and update the Pareto solutions, and hybridizes the updating rules from SHADE and MRFO to balance the exploration and exploitation, which can enhance the convergence behavior for truss optimization. To verify the proposed algorithm, 6 different truss structures with two objectives are implemented, with the comparison of 9 different multi-objective metaheuristic algorithms, showing the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows: Section 2 reviews the related works on the multi-objective metaheuristic algorithms for truss optimization. Section 3 introduces some definitions of the multi-objective optimization. Then, Section 4 presents the basic theory of MO-SHADE-MRFO, including the updating rules of SHADE and MRFO. Section 5 introduces the mathematical formulation of multi-objective truss problem, and Section 6 shows the comparison results of six truss problems. Finally, the conclusions are listed in Section 8.

2 Related works

This section provides the developments of multi-objective metaheuristic algorithms for truss optimization problems. The research works of Pareto-based algorithms are the main issue in this field. Coello et al. [49] developed the genetic algorithm for multi-objective truss optimization. In [50], a modified PSO was enhanced by the selection and preservation of diversity, called FC-MOPSO, to obtain the Pareto solutions of multi-objective truss optimization problems. A modified symbiotic organism

search (MOSOS) was invented for truss optimization problems, which was verified by 5 different truss design problems [51]. Kaveh and Mahdavi [52] proposed a multi-objective colliding bodies optimization algorithm for structural design problems, which was verified by benchmark functions, 120-bar truss dome, and 582-bar tower truss, where Pareto fronts obtained by MOCBO were better than those of NSGA-II and MOPSO. Kumar et al. [53] investigated the performance of multi-objective passing vehicle search (MOPVS) in structural design problems, illustrating that the algorithm can achieve good Pareto fronts. Tejani et al. [54] developed a multi-objective heat transfer search algorithm in solving structural design problems, verified by 5 truss optimization problems. As follows, a modified heat transfer search algorithm was also developed for structural design problems [55]. Premkumar et al. [56] suggested a multi-objective gradient-based optimizer for structural design problems, which had good performance in solving truss optimization problems, including 10-bar, 25-bar, 60-bar, 37-bar, 72-bar, 120-bar, 200-bar, and 942-bar problems. Other developments in this research field include novel MOPVS [57], multiobjective forensic-based investigation algorithm [58], multiobjective thermal exchange optimization algorithm [59], decomposition-based multi-objective heat transfer search [60], etc. Although most of the above algorithms can provide good Pareto solutions, some of them may not obtain stable values in terms of different measure indices (hypervolume, inverted generational distance, etc.).

Except for the above algorithms, several works about the variants of differential evolution for multi-objective truss optimization problems were also investigated. Vargas et al. [61] investigated the generalized differential evolution (GDE) and its variants in solving multi-objective truss optimization problems, with two objectives of the total weight and the structural displacement, showing good Pareto fronts in the tested truss structures. Carvalho et al. [62] investigated different differential evolution algorithms in solving truss optimization problems with 3 objectives, which proved the feasibility in truss optimization with many-objective. Lemonge et al. [63] developed the third evolution step of GDE (GDE3) with the adaptive penalty method for truss optimization problems with complex constraints of frequencies. Anosri et al. [64] proposed SHAMODE with interval scheme (iSHAMODE) to solve truss optimization problems with two objectives of the total weight and reliability index, which had better statistical values of hypervolume than the compared algorithms. In [42], a comprehensive review of multi-objective metaheuristic algorithms in solving truss optimization problem was discussed, illustrating that SHAMODE and SHAMODE-WOA were outstanding among 14 different multi-objective metaheuristic algorithms. However, SHADE is highly focused on the intensification, while the diversification is more important in the multi-objective, and the exploration capacity of SHADE for multi-objective can be further enhanced by hybridizing another powerful metaheuristic algorithm. Thus, this work presents the MO-SHADE-MRFO by combining SHADE and MRFO to balance the exploration and exploitation, which can enhance the convergence behavior for multi-objective truss optimization problems.

3 Brief review on multi-objective optimization

The basic concepts and definitions of multi-objective optimization [65] are reviewed briefly in this section. In general, multi-objective optimization problems have multiple conflicting objectives to be optimized under the constraints, which can be formulated as:

$$\begin{aligned} &\text{Find} && \mathbf{x}=\{x_1, x_2, \ldots, x_m\} \\ &\text{Min} && F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_o(\mathbf{x})] \\ &\text{s.t.} && g_i(\mathbf{x}) \geq 0, \quad i = 1,2,\ldots,p \\ & && h_j(\mathbf{x}) = 0, \quad j = 1,2,\ldots,n \\ & && lb_k \leq x_k \leq ub_k, \quad k = 1,2,\ldots,m \end{aligned} \tag{1}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ denotes the design variable vector, $F(\mathbf{x})$ represents the o-dimensional vector of objective functions, g_i is the i-th inequality constraint, h_j denotes the j-th equality constraint, p and n represent the number of inequality and equality constraints, respectively, m is the number of design variable, lb and ub are the lower and upper bounds of the design variables, and o denotes the number of objective functions. In multi-objective optimization problems, multiple objectives are required to be computed simultaneously, while a good solution for one objective function may be bad for another objective function. Therefore, it is difficult to find a solution which satisfies all objective functions. Usually, a set of feasible solutions are existed in the multi-objective optimization problems. To compare feasible solutions, the *Pareto* theorem is introduced to characterize the dominance relations of different solutions [65].

Pareto-optimality: The solution $\mathbf{x} \in \boldsymbol{X}$ is denoted the Pareto optimum if and only if:

$$\nexists\, \mathbf{y} \in \boldsymbol{X} \,|\, F(\mathbf{y})\ \mathrm{p}\ F(\mathbf{x}) \tag{2}$$

Pareto-dominance: Supposing two vectors: $\mathbf{x}=\{x_1, x_2, \ldots, x_m\}$ and $\mathbf{y}=\{y_1, y_2, \ldots, y_m\}$, the vector $\mathbf{y}$ dominates the vector $\mathbf{x}$ (denoted as $\mathbf{y}\ \mathrm{p}\ \mathbf{x}$), if and only if:

$$\forall i \in \{1,2,...,j\} : f_i(\mathbf{y}) \leq f_i(\mathbf{x}) \wedge \exists i \in \{1,2,...,j\} : f_i(\mathbf{y}) < f_i(\mathbf{x}) \tag{3}$$

Pareto set: The Pareto set is defined as a set in the search space including all Pareto optimal vectors, expressed as follows:

$$P_s = \{\mathbf{x}, \mathbf{y} \in \boldsymbol{X} \,|\, \exists F(\mathbf{y})\ \mathrm{f}\ F(\mathbf{x})\} \tag{4}$$

Pareto front: The Pareto front represents the collection of Pareto optimal solutions in the Pareto set, defined as:

$$P_f = \{F(\mathbf{x}) | \mathbf{x} \in P_s\} \tag{5}$$

The Pareto set is the key concept for solutions of multi-objective optimization problems, which is defined as the relationship of design space and objective space, shown in Fig. 1. The feasible regions in

the design space are mapped into the regions in the objective space, while the infeasible regions are out of feasible solutions determined by the constraints of optimization problems. Two points *A* and *B* in the design space are mapped into the objective space, indicating that *A* dominates *B*. The target of multi-objective optimization algorithms is to find well-distributed feasible solutions, which are approximately close to Pareto optimal front.

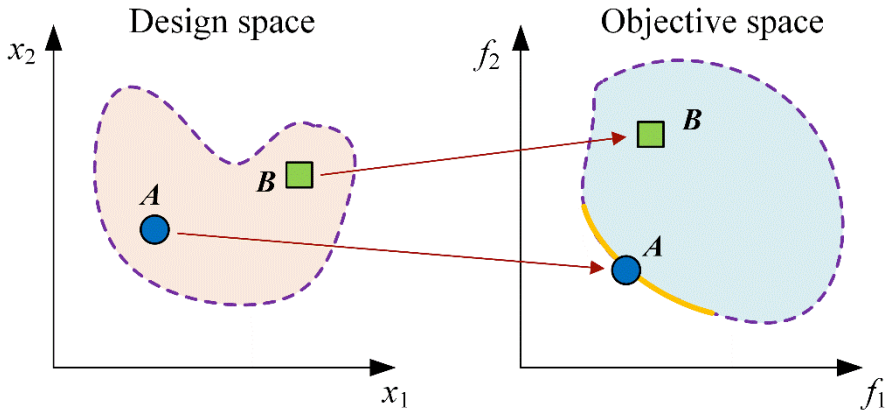


Fig. 1 Design space and objective space

4 The proposed MO-SHADE-MRFO algorithm

In this section, two metaheuristic algorithms, success-history based parameter adaptive differential evolution (SHADE) and manta ray foraging optimization (MRFO), are briefly introduced which are the basic components of the proposed algorithm. SHADE is a modified version of differential evolution by introducing the success history-based parameter adaptation technique, and MRFO is inspired by the foraging behavior of manta ray with good convergence in solving benchmark and real-world optimization problems. Finally, the procedure of MO-SHADE-MRFO is illustrated.

4.1 Success history-based parameter adaptive differential evolution (SHADE)

SHADE is a variant of differential evolution, which is a powerful metaheuristic algorithm and ranked the 3rd in the competition of CEC 2014 [44]. The original differential evolution highly depends on the settings of the algorithmic parameters (scaling factor *F*, crossover rate *CR*, population size *N*, chosen mutation/crossover strategies). To control the algorithmic parameters automatically, SHADE introduces the success-history based parameter adaptation, which can enhance the robustness of differential evolution. Therefore, SHADE is applicable to various optimization problems without trial and error. If the control parameters are adopted into appropriate values in different phase of algorithm, the convergence rate can be enhanced. The basic phases of SHADE include initialization, mutation, crossover, selection, and historical memory updating.

In the initialization of SHADE, the population and historical memories are generated randomly

within the feasible domain, and the external archive is initialized as an empty set. The historical memories of scaling factor $M_{CR,q}$ and crossover rate $M_{F,q}$ are initially set as 0.5:

$$M_{CR,q} = M_{F,q} = 0.5, \quad q = 1,2,...,H \tag{6}$$

where H denotes the size of historical memories M_F and M_{CR}.

In the mutation phase of SHADE, the current-to-pbest/1 mutation strategy is employed to update the mutation vector, which utilizes the information of the best solution and other good solutions, considering the external archive A in SHADE to maintain the diversity of the population, where A stores the inferior parents recently replaced by offspring. The updating mechanism of current-to-pbest/1 mutation is expressed as:

$$\boldsymbol{v}_i = \boldsymbol{x}_i + F_i\left(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i\right) + F_i\left(\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}\right) \tag{7}$$

where $\boldsymbol{v}_i$ represents the mutant vector of the individual $\boldsymbol{x}_i$, $\boldsymbol{x}_{pbest}$ is randomly selected from the top $N*p$ individuals in the current generation, N is the population size, p is generated in [2/N, 0.2] based on the uniform distribution. $\boldsymbol{x}_{r1}$ is chosen randomly from the population, and $\boldsymbol{x}_{r2}$ is chosen from the union population (population N and external archive A). In Eq. (7), F_i can be written as:

$$F_i = randc_i\left(u_F, 0.1\right) \tag{8}$$

where u_F denotes a randomly chosen value from the historical memory of the scaling factor M_F, $randc_i(u_F, 0.1)$ represents that the value is generated by the Cauchy distribution with the mean of u_F and the variance of 0.1. Therefore, the mutation operator is controlled by the parameter $F_i \in [0,1]$.

The crossover operator of SHADE is similar to the basic DE. The trial vector is implemented from parent's vector $\boldsymbol{x}_i$ and mutation vector $\boldsymbol{v}_i$:

$$u_{i,k} = \begin{cases} v_{i,k}, & if \quad rand \le CR_i \quad or \quad k = k_{rand} \\ x_{i,k}, & otherwise \end{cases} \tag{9}$$

where $rand$ is randomly selected within (0,1), k_{rand} is an integer selected from [1, D], D represents the number of design variable, k = 1, 2, …, N. CR_i is the updated crossover rate:

$$CR_i = randn_i\left(u_{CR}, 0.1\right) \tag{10}$$

where u_{CR} is randomly selected from the historical memory of crossover rate M_{CR}, $randn_i(u_{CR}, 0.1)$ denotes that the crossover rate CR_i is generated by the normal distributions with the mean of u_{CR} and the variance of 0.1. If CR_i is out of [0,1], it is set as the boundary value.

In the selection process of SHADE, the previous and current fitness values of individuals $f(\boldsymbol{u}_i(t))$ and $f(\boldsymbol{x}_i(t))$ are compared, and the new individual is generated according to the better solution:

$$\boldsymbol{x}_i(t+1)=\begin{cases}\boldsymbol{u}_i(t), & if \quad f(\boldsymbol{u}_i(t))\le f(\boldsymbol{x}_i(t))\\ \boldsymbol{x}_i(t), & otherwise\end{cases} \tag{11}$$

The historical memory mechanism is employed in SHADE to adapt the mutation factor F_i and crossover factor CR_i. Historical memories M_F and M_{CR} are initialized in Eq. (6) and updated in each iteration. If the values of $M_{F,q}$ and $M_{CR,q}$ are out of the boundary [0,1], they are set to the boundary values. Thus, the historical memories are calculated as:

$$M_{F,q}=\begin{cases}mean_{WL}(S_F), & if \quad S_F\neq\varnothing\\ M_{F,q}, & otherwise\end{cases} \tag{12}$$

$$M_{CR,q}=\begin{cases}mean_{WL}(S_{CR}), & if \quad S_{CR}\neq\varnothing\\ M_{CR,q}, & otherwise\end{cases} \tag{13}$$

where $\boldsymbol{S}_F$ and $\boldsymbol{S}_{CR}$ are the successful recorded trial vectors for the mutation factor F_i and crossover rate C_{Ri}, respectively. $mean_{WL}(S)$ represents the weighted Lehmer mean function:

$$mean_{WL}(S)=\frac{\sum_{k=1}^{|S|}w_k\cdot S^2}{\sum_{k=1}^{|S|}w_k\cdot S} \tag{14}$$

where the weight vector parameter w_k is given by:

$$w_k=\frac{\left|f(\boldsymbol{u}_k)-f(\boldsymbol{x}_k)\right|}{\sum_{k=1}^{|S_{CR}|}\left|f(\boldsymbol{u}_k)-f(\boldsymbol{x}_k)\right|} \tag{15}$$

where $f(\boldsymbol{u}_k)$ and $f(\boldsymbol{x}_k)$ are the objective values of the trial vector and parent vector of the k-th individuals, respectively. More details of SHADE can be found in [44].

4.2 Manta ray foraging optimization (MRFO)

The manta ray foraging optimization (MRFO) was proposed by Zhao et al. [45] to mimic the behaviors of the manta rays, which is one of the largest known marine creatures. MRFO is a population-based optimization algorithm, and has high performance in solving benchmark and real-world optimization problems according to the experimental results [45]. MRFO includes three basic behaviors: chain foraging, cyclone foraging, and somersault foraging.

In the chain foraging, a foraging chain is formulated in manta rays to find the plankton position. In each iteration, each search agent is updated according to both the best solution and the solution in front of it. The d-th dimension position of the search agent Z_i^d is updated using the following equations:

$$Z_i^d(t+1)=\begin{cases}Z_i^d(t)+r\left(Z_{best}^d(t)-Z_i^d(t)\right)+\alpha\left(Z_{best}^d(t)-Z_i^d(t)\right), & i=1\\ Z_i^d(t)+r\left(Z_{i-1}^d(t)-Z_i^d(t)\right)+\alpha\left(Z_{best}^d(t)-Z_i^d(t)\right), & i=2,\mathrm{K},N\end{cases} \tag{16}$$

where t is the current iteration, r refers to a random number within [0,1], N indicates the total number of

agents, and α represents a weighting coefficient which can be updated by:

$$\alpha = 2 \times r \times \sqrt{\left|\log(r)\right|} \tag{17}$$

In the cyclone foraging, the manta rays formulate a long chain that moves towards the food by a spiral. The search agent not only follows the individual in front of it, but also moves towards the position of food. The updating mechanism of the search agent is defined as:

$$Z_i^d(t+1) = \begin{cases} Z_{best}^d(t) + r\left(Z_{best}^d(t) - Z_i^d(t)\right) + \beta\left(Z_{best}^d(t) - Z_i^d(t)\right), & i = 1 \\ Z_{best}^d(t) + r\left(Z_{i-1}^d(t) - Z_i^d(t)\right) + \beta\left(Z_{best}^d(t) - Z_i^d(t)\right), & i = 2,\text{K},N \end{cases} \tag{18}$$

where β denotes the weighting factor, calculated as:

$$\beta = 2\exp\left(r_1\left(\frac{M_{iter} - t + 1}{M_{iter}}\right)\right) \times \sin\left(2\pi r_1\right) \tag{19}$$

where r_1 is a random number in (0, 1). Besides, to enhance the exploration capacity, some manta rays may not move to the food position:

$$Z_i^d(t+1) = \begin{cases} Z_{rand}^d(t) + r\left(Z_{rand}^d(t) - Z_i^d(t)\right) + \beta\left(Z_{rand}^d(t) - Z_i^d(t)\right), & i = 1 \\ Z_{rand}^d(t) + r\left(Z_{i-1}^d(t) - Z_i^d(t)\right) + \beta\left(Z_{rand}^d(t) - Z_i^d(t)\right), & i = 2,\text{K},N \end{cases} \tag{20}$$

where $Z_{rand}^d(t)$ is the position randomly generated in the search space:

$$Z_{rand}(t) = l_b + r\left(u_b - l_b\right) \tag{21}$$

where l_b and u_b are the lower and upper bounds of design variables.

The somersault foraging is the last phase of the MRFO algorithm. In this phase, the food position is regarded as a pivot, and each manta ray tends to swim and move around the food position. Thus, the updated position of a manta ray in the somersault foraging behavior is usually near to the best position, which can be illustrated as follows:

$$Z_i^d(t+1) = Z_i^d(t) + S \times \left(r_2 Z_{best}^d(t) - r_3 Z_i^d(t)\right), \quad i = 1, 2, ..., N \tag{22}$$

where S represents the somersault factor, to determine the somersault range of manta rays and S=2. r_2 and r_3 are random numbers in (0,1). More details on MRFO can be found in [45].

4.3 The procedure of MO-SHADE-MRFO

SHADE is a very effective metaheuristic algorithm, especially for solving single-objective optimization problems. However, there has been little research on SHADE for solving multi-objective optimization problems. Panagant et al. [41] proposed multi-objective version of SHADE for solving multi-objective truss optimization problems. According to their results, the intensification part is highly emphasized in SHADE, but the diversification is more important in multi-objective optimization.

Therefore, it may suffer from the premature or oscillation of some measurement indicators (inverted generational distance, spacing to extent) in solving multi-objective truss optimization problems.

In this work, a hybrid meta-heuristic algorithm called MO-SHADE-MRFO is presented for multi-objective truss optimization problems, with the combination of SHADE and MRFO. In MO-SHADE-MRFO, the external archive is utilized to save and update the obtained Pareto solutions during the optimization iterations, where the updating rules of SHADE and MRFO are hybridized to balance the exploration and exploitation. The modification is made at the reproduction process, where each mutant vector has a chance to be further updated with the cyclone or chain behaviors of MRFO. The main procedure of MO-SHADE-MRFO for multi-objective optimization problem is provided as follows:

Step 1: Define the optimization problems including objectives, constraints, design variables, and set the parameters of the algorithm, including population size, maximum iterative number, Pareto archive size, and so on.

Step 2: Generate the initial population randomly among the boundaries of design variables and calculate the fitness values of all objectives, then use Pareto theory to save the Pareto solutions into the archive in the initialization step.

Step 3: Enter the main loop of the algorithm, set the historical memories M_F and M_{CR} as 0.5 initially.

Step 4: Update the position of each particle using the mutation operator from SHADE by Eqs. (7-8).

Step 5: Implement the updating rules of MRFO after the mutation phase of SHADE, and update the position of each search agent by chain foraging of Eq. (16), cyclone foraging of Eqs. (18) and (20), and somersault foraging of Eq. (22).

Step 6: Update the position of each search agent using the crossover operator from SHADE by Eq. (9), after the updating rules of MRFO.

Step 7: Check the constraints of the updated positions and calculate the fitness values.

Step 8: Compare the obtained solution with the former solution by the Pareto theory, and then update the Pareto solutions in the archive. If the archive number achieves to the maximum, remove the position with the most crowded segment and add the obtained Pareto solution into the external archive. This step of the algorithm is similar to other Pareto-based algorithms [36][39].

Step 9: Implement the adaptive strategies from SHADE, and then update the historical memories by Eqs. (12-13).

Step 10: If the stop criterion (maximum iterative number) is satisfied, store the archive and get out of the main loop. Otherwise, go to step 3 to update the Pareto solutions.

Step 11: Finally, output the final Pareto solutions of multi-objective optimization problems.

To fully illustrate the procedure of MO-SHADE-MRFO, the flowchart and the pseudo-code are provided in Fig. 2 and Algorithm 1, respectively.

Algorithm 1: pseudo-code of MO-SHADE-MRFO

Define the objective functions, and constraints of multi-objective truss optimization problems
Set parameters, such as population size, maximum iteration, and maximum Pareto archive size
Initialize population P_1 and select non-dominated solutions from P_1 to be initial Pareto front
For t = 1:M_{iter} do // Main loop
    // reproduction process
    Calculate the scaling factor F_i by Eq. (8);
    Evaluate the mutation vector v_i in the mutation phase of SHADE by Eq. (7)
    For i = 1 to N do // Start MRFO updating mechanism
        If *rand* < 0.5 // Cyclone foraging
            Calculate β by Eq. (19)
            If $t/M_{iter} > rand$
                Update the position of individual using Eq. (18)
            Else
                Update the position of individual using Eq. (20)
            End If
        Else // Chain foraging
            Calculate α by Eq. (17)
            Update the position of individual using Eq. (16)
        End If
        Update the position of individual using Eq. (22) // Somersault foraging
    End For // End MRFO updating mechanism
    Calculate the crossover rate CR_i by Eq. (10)
    Calculate the crossover operator using Eq. (9)
    x_{G+1} = best NP solutions with highest non-dominated levels from $x_G \cup u_G$
    Sind = set of indices of u_G that survived and are included in x_G
    $Pareto_{G+1}$ = non-dominated solutions from $Pareto_G \cup u_G$
    $A_{G+1} = A_G \cup x_{Sind,G}$ // adaptive strategies
    Calculate the cell of historical memories $M_{F,q}$ and $M_{CR,q}$ by Eq. (12) and Eq. (13), respectively.
    Record the Pareto solutions
End For

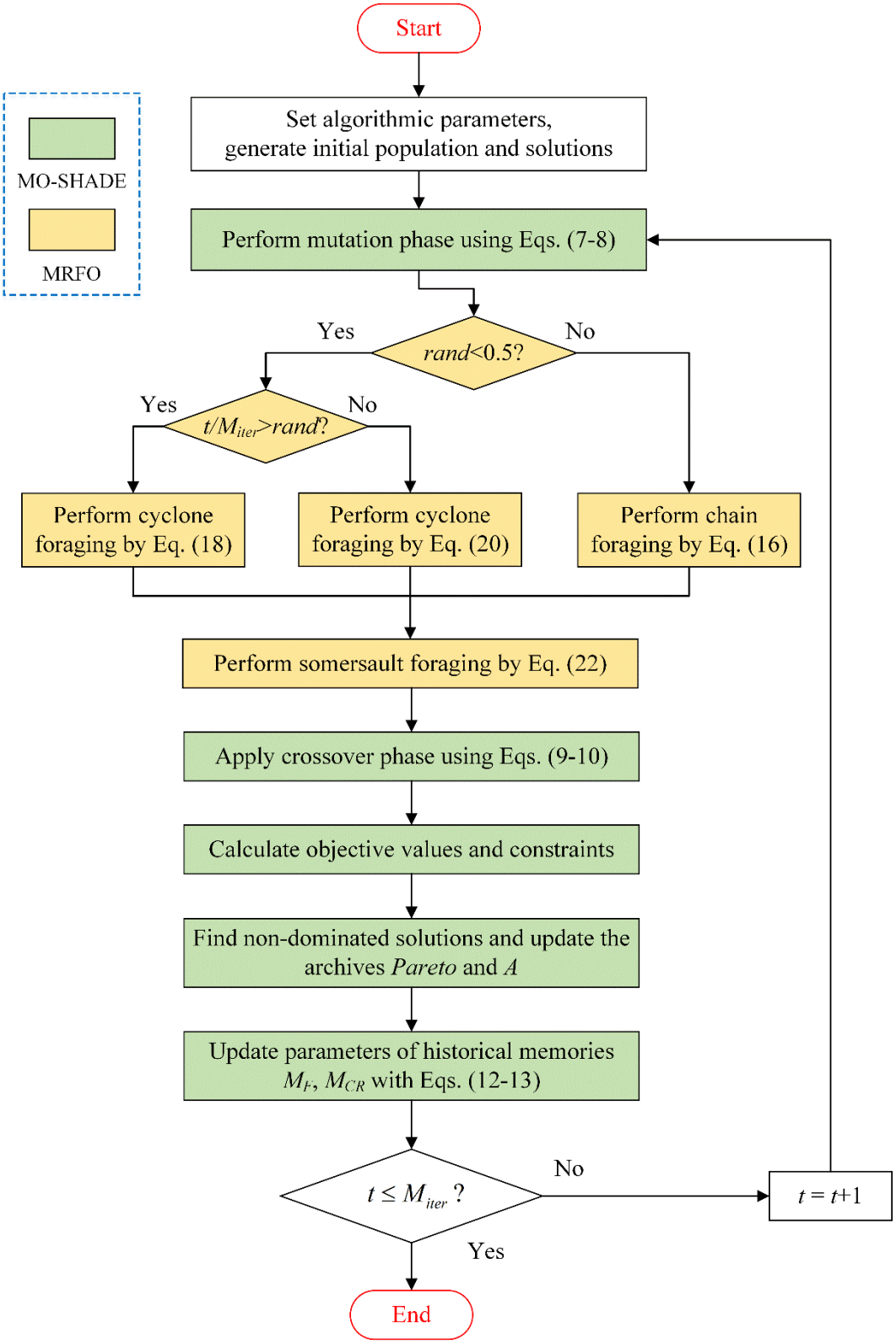


Fig. 2. Flowchart of MO-SHADE-MRFO

5 Mathematical formulation of multi-objective truss optimization

The multi-objective truss optimization design is a challenging task due to the conflicting objectives, complicated constraints, and discrete design variables of cross-sectional areas. This paper focuses on the multi-objective truss optimization problem, to minimize the total weight and compliance under allowable stress constraints. However, there are conflicts between the different objectives, which are impossible to both achieve optimal value at the same time. In addition, the nonlinear stress constraints also limit the search space of design variables, leading to the more intricate optimal design problems. Different from the single-objective problem, the multi-objective truss optimization problem aims to obtain the Pareto solution set rather than a unique optimal solution.

The basic formulation of a multi-objective truss optimization can be presented as:

$$\begin{aligned} &\text{Find} && A=\{A_1, A_2, \ldots, A_m\} \\ &\text{Min} && f_1(A)=\sum_{i=1}^{m} A_i \rho_i L_i\,, \quad f_2(A)=compliance=\boldsymbol{u}^T\boldsymbol{F} \\ &\text{s.t.} && |\sigma_i|-\sigma_i^{\max}\le 0\,, \quad A_i^{\min}\le A_i\le A_i^{\max} \end{aligned} \tag{23}$$

where A_i is the design variable of the cross-sectional area for i-th element, m is the number of design variables, f_1 and f_2 are the structural mass and compliance, respectively, ρ_i and L_i are the mass density and length of the i-th element, respectively, σ_i and $\sigma_i^{\max}$ are the tress and the allowable value of the i-th element, and $A_i^{\min}$ and $A_i^{\max}$ are the lower and upper bounds of cross-sectional areas of design variables. The compliance is computed by the vector product of displacement $\boldsymbol{u}$ and force $\boldsymbol{F}$.

6 Results and discussions

To verify the MO-SHADE-MRFO algorithm, 6 multi-objective truss optimization problems are implemented, consisting of consisting of 10-bar, 25-bar, 37-bar, 120-bar, 200-bar, and 942-bar truss structures, depicted in Figs. 3-8. The related data of truss problems are listed in Table 1, and the detailed nodes, elements and loads are given as follows. The details of these truss structures can also be founded in [42].

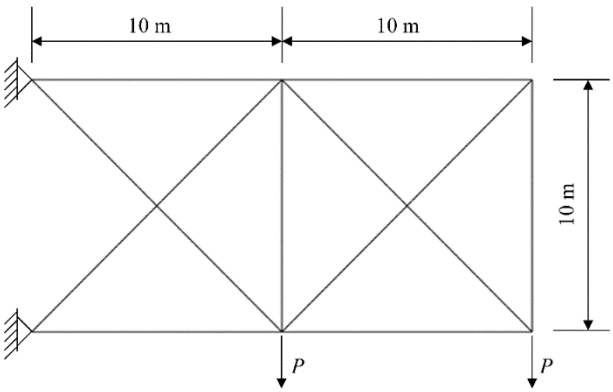


Fig. 3. 10-bar truss

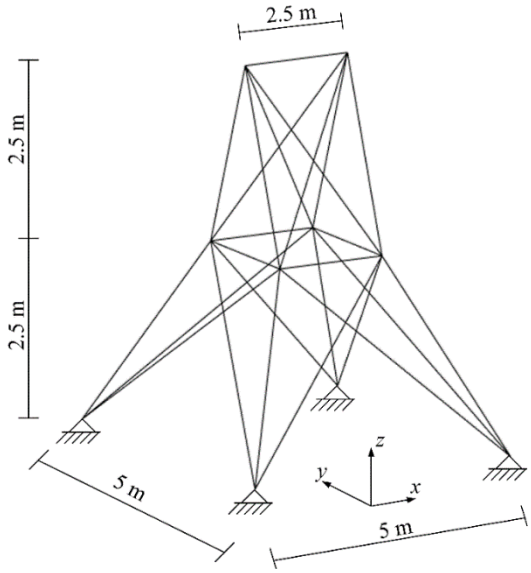


Fig. 4. 25-bar truss

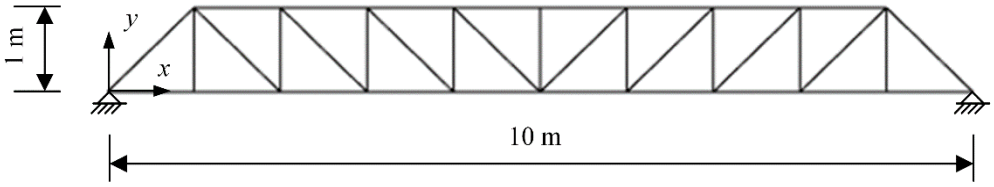


Fig. 5. 37-bar truss

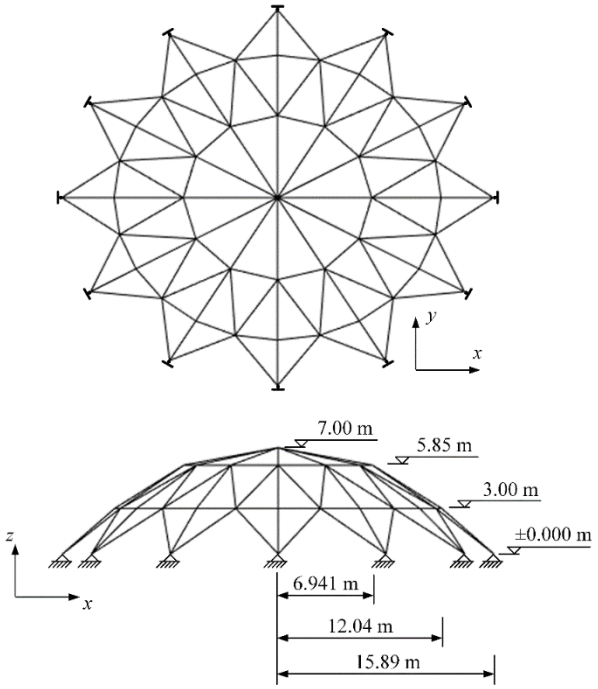


Fig. 6. 120-bar truss

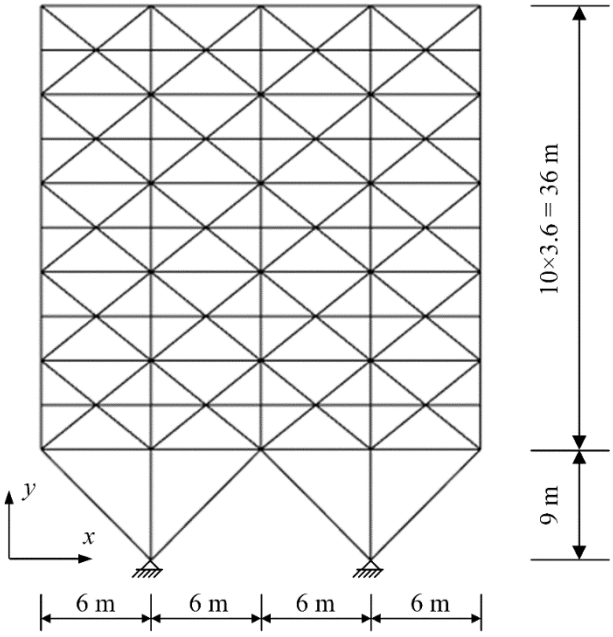


Fig. 7. 200-bar truss

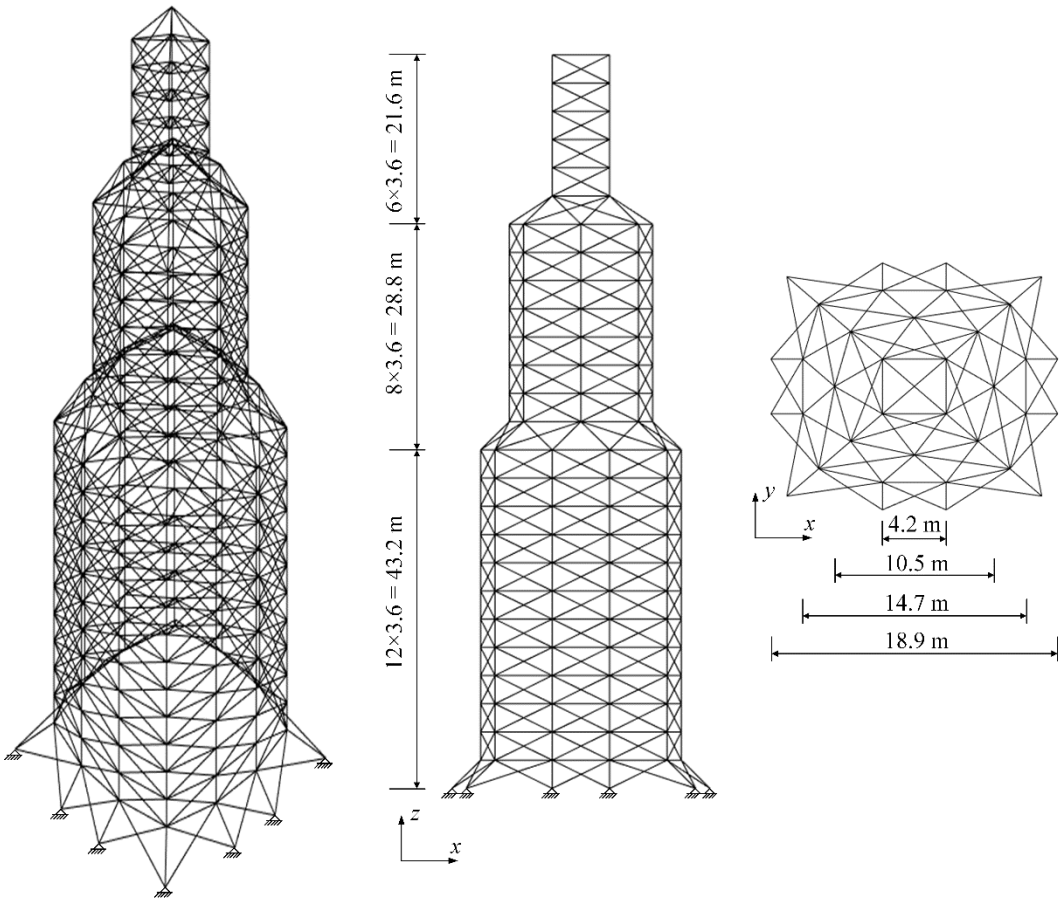


Fig. 8. 942-bar truss

Table 1 Design consideration of the truss problems

Problem	Design variable	Maximum stress σ^{max} (MPa)	Material density P (kg/m^3)	Elastic modulus E (GPa)	Loading (kN)
10-bar	A_i, (i=1,2,…,10)	400	7850	200	Nodes 2, 4 (F_y=-1000)
25-bar	A_i, (i=1,2,…,8)	400	7850	200	Node 1 (F_x=-100, F_y=F_z=-1000)
					Node 2 (F_y=-1000, F_z=-1000)
					Node 3 (F_x=50) Node 4 (F_x=60)
37-bar	A_i, (i=1,2,…,15)	400	7850	200	Nodes 2-10 (F_y=-100)
120-bar	A_i, (i=1,2,…,7)	400	7850	200	Nodes 13-36 (F_z=-500)
					Nodes 37-48 (F_z=-1500)
					Node 49 (F_z=-5000)
200-bar	A_i, (i=1,2,…,29)	400	7850	200	Supplementary materials
942-bar	A_i, (i=1,2,…,59)	400	7850	200	Supplementary materials

Note: More details of truss problems are provided in the Supplementary materials

6.1 Experimental setups

In the experiments, 9 different multi-objective metaheuristic algorithms are implemented to compare the MO-SHADE-MRFO, and the brief description of compared algorithms is provided as follows:

Multi-objective particle swarm optimization (MOPSO) [34]: Particle swarm optimization (PSO) is one of the most well-known metaheuristic algorithms, inspired by the preying of fishes or birds. MOPSO is a modified version of PSO to solve multi-objective optimization problems, with the adaptive grid mechanism to save the external archive and the mutation strategy to maintain the diversity of population.

Non-dominated sorting genetic algorithm II (NSGA-II) [33]: NSGA-II is a very famous multi-objective algorithm. The non-dominated sorting is used to reduce computational complexity, while the crowding distance operator can maintain the diversity of generation. The elitist strategy is considered in NSGA-II to ensure the best individual not losing during optimization and expand the search space.

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [35]: It is another popular algorithm in multi-objective field, which can transfer a multi-objective optimization problem into a set of scalar subproblem based on uniformly distributed aggregation weight vectors, providing a general framework in multi-objective problems. The decomposition strategy is embedded in multi-objective metaheuristic algorithms to maintain the diversity of population.

Multi-objective grasshopper optimization algorithm (MOGOA) [36]: The original version of grasshopper optimization algorithm mimics the behavior of grasshoppers in nature to find the best solution in optimization problems. In the MOGOA, the archive and leader selections are employed to estimate the Pareto optimal front of a multi-objective problem.

Multi-objective multi-verse optimization (MOMVO) [37]: The multi-verse optimization is proposed with the inspiration of multi-verse in physics. MOMVO is developed based on the same concept of the multi-verse optimization for solving multi-objective optimization problems, combined with archive and selector in updating mechanism.

Multi-objective water cycle algorithm (MOWCA) [38]: The water cycle algorithm mimics the water cycle process in nature. MOWCA is the multi-objective version of water cycle algorithm while the sea and rivers are considered as the non-dominated solution. The crowding-distance strategy and archive non-dominated solution are embedded to avoid local stagnation and maintain diversity of population.

Multi-objective salp swarm algorithm (MOSSA) [39]: The behavior of salp chain in sea is simulated in the salp swarm algorithm, and the salps in chain are divided as the leader and followers, which can be used to solve single-objective optimization problems. The updating mechanism of MOSSA is inherited the salp swarm algorithm, and the leader is chosen as the archive of non-dominated solutions.

Unrestricted population size evolutionary multi-objective optimization algorithms (UPSEMOA)

[40]: The basic concept of UPESMOA is to create a greater size of non-dominated solutions which can enhance the diversity, and the population size is not defined by user but is dependent on the minimum number of points required by generation of mechanism.

Success history-based parameter adaptive multi-objective differential evolution (SHAMODE) [41]: SHAMODE is the multi-objective version of SHADE, where the updating rule of SHADE is provided in Section 4.

For the proposed and compared algorithms, the population size and maximum iterative number are set as 100 and 100, respectively. Each algorithm is run 30 times to test the robustness for truss optimization problems, and the Friedman ranking test [66] is implemented to measure the performance of the algorithm, which is a non-parametric testing method to determine the difference of obtained samples. The algorithmic parameters of all algorithms are listed in Table 2.

Table 2 Algorithmic parameters of multi-objective metaheuristic algorithms

Algorithm	Parameters	Values
# all algorithms	Population size, Maximum iterations, Independent runs	100, 100, 30
MOPSO	Cognitive and social constant	c_1=1.5, c_2=1.5
	Inertia weight linearly decreased at interval	[0.9 0.4]
NSGA-II	Mutation probability	0.1
MOEA/D	Number of neighboring weight vector	6
	Mutation rate	0.1
MOGOA	C_{max}, C_{min}	1, 0.00004
MOMVO	Max and min of WEP	[1 0.2]
MOWCA	Number of river summation	4
MOSSA	Leading position update probability	0.5
UPESMOA	Crossover probability	0.7
	Scaling factor	0.8
	Probability of choosing element from offspring	0.5
SHAMODE	Archive ratio	1.4
MO-SHADE-MRFO	Archive ratio	1.4

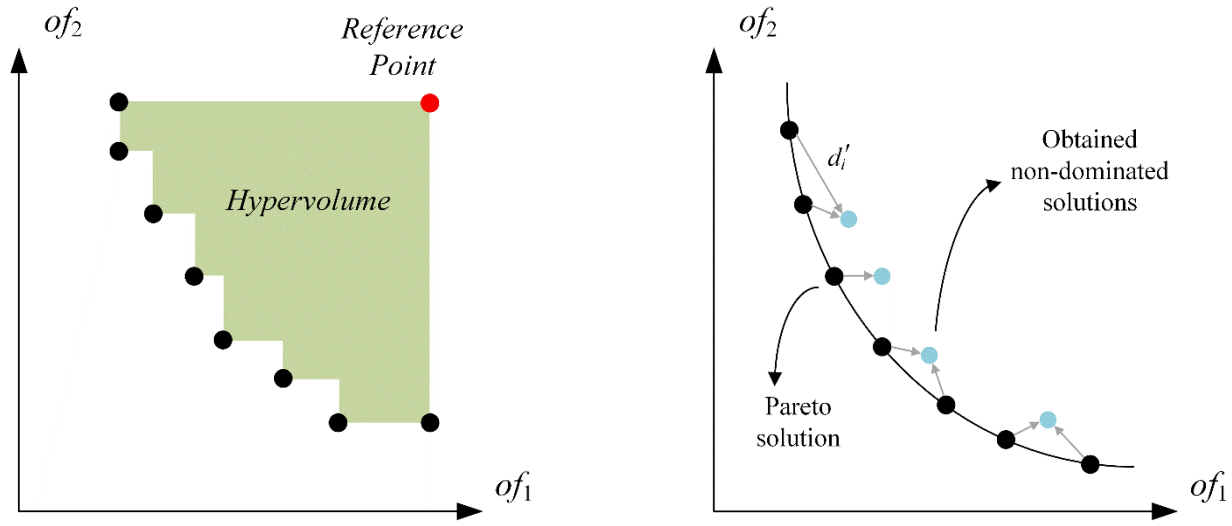
In the multi-objective truss optimization problems, three different metrics are used to evaluate the performance of metaheuristic algorithms, including hyper volume (HV) to measure spread of Pareto front, spacing-to-extent (STE) to measure spacing and extent of a front, and inverted generational distance (IGD) to measure the distances between the Pareto front and the reference front. The metrics are evaluated by:

$$HV = \left(\mathrm{U}_{i=1}^{|P|} V_i \right) \tag{24}$$

$$Spacing = \frac{1}{|P|-1} \sum_{i=1}^{|P|} \left(d_i - \bar{d} \right)^2, \quad Extent = \sum_{i=1}^{M} \left| f_i^{\max} - f_i^{\min} \right|, \; STE = Spacing/Extent \tag{25}$$

$$IGD = \frac{\sqrt{\sum_{i=1}^{|P|} (d_i')^2}}{|P|} \tag{26}$$

where P denote the number of solutions in the Pareto front, d_i is the Euclidian distance between the i-th solution of objective functions and its nearest neighbor true Pareto optimal solution, d'_i denotes the Euclidian distance between the i-th true Pareto optimal solution and the nearest true Pareto optimal solution. $\bar{d}$ denotes the mean value of all d_i, $f_i^{\max}$ and $f_i^{\max}$ are maximum and minimum values of objective function of the front, and V_i represents the hypervolume generated from two vertices between the i-th non-dominated solution and referent point, as illustrated in Fig. 9(a). The schematic view of IGD is also displayed in Fig. 9(b). It should be noted that the larger HV value is, the better performance of the algorithm will be, meaning that the HV combined by the solutions and reference point is large. Besides, the smaller IGD and STE values are, the better performance of the algorithm will be, denoting that the algorithm has good convergence and distribution.



(a) Hypervolume (HV) (b) IGD

Fig. 9. Schematic views of metrics

Table 3 Statistical hypervolume (HV) results for each truss problem

Algorithm		MOPSO	NSGA-II	MOEA/D	MOGOA	MOMVO	MOWCA	MOSSA	UPSEMOA	SHAMODE	Proposed
10-	Best	2.09E+09	1.38E+09	2.05E+09	2.04E+09	2.29E+09	3.51E+08	2.05E+09	2.03E+09	2.37E+09	**2.38E+09**
bar	Mean	2.25E+09	2.18E+09	2.22E+09	2.17E+09	2.32E+09	1.75E+09	2.16E+09	2.21E+09	2.38E+09	**2.38E+09**
	Worst	2.33E+09	2.35E+09	2.34E+09	2.29E+09	2.35E+09	2.28E+09	2.26E+09	2.30E+09	2.38E+09	**2.39E+09**
	SD	5.71E+07	2.31E+08	6.88E+07	6.74E+07	1.57E+07	4.14E+08	6.25E+07	7.26E+07	3.08E+06	**1.64E+06**
	FR	5.40	6.03	6.37	7.40	3.27	9.23	7.80	6.50	1.90	**1.10**
25-	Best	5.49E+08	2.85E+08	5.44E+08	5.03E+08	5.58E+08	3.31E+08	5.24E+08	2.67E+08	**5.69E+08**	5.68E+08
bar	Mean	5.65E+08	4.78E+08	5.61E+08	5.24E+08	5.60E+08	4.60E+08	5.44E+08	5.22E+08	**5.69E+08**	5.69E+08
	Worst	5.67E+08	5.39E+08	5.70E+08	5.47E+08	5.62E+08	5.59E+08	5.53E+08	5.61E+08	**5.70E+08**	5.69E+08
	SD	3.25E+06	5.61E+07	6.63E+06	1.26E+07	1.12E+06	5.86E+07	6.86E+06	5.09E+07	3.27E+05	**2.43E+05**
	FR	3.23	9.13	4.13	7.70	4.67	9.23	6.30	7.47	**1.20**	1.93
37-	Best	1.33E+08	1.31E+08	1.31E+08	1.18E+08	1.41E+08	1.13E+08	1.16E+08	6.75E+07	**1.47E+08**	**1.47E+08**
bar	Mean	1.37E+08	1.39E+08	1.39E+08	1.28E+08	1.43E+08	1.33E+08	1.27E+08	1.29E+08	**1.47E+08**	**1.47E+08**
	Worst	1.43E+08	1.43E+08	1.44E+08	1.36E+08	1.44E+08	1.40E+08	1.33E+08	1.40E+08	**1.48E+08**	**1.48E+08**
	SD	2.63E+06	3.29E+06	3.14E+06	4.79E+06	8.41E+05	5.95E+06	4.14E+06	1.61E+07	1.98E+05	**1.13E+05**
	FR	5.93	4.93	5.07	8.73	3.27	7.30	9.20	7.57	**1.47**	1.53
120-	Best	7.66E+10	4.37E+10	7.00E+10	7.44E+10	8.10E+10	6.60E+09	7.12E+10	6.41E+10	8.44E+10	**8.46E+10**
bar	Mean	7.94E+10	7.53E+10	7.83E+10	7.83E+10	8.27E+10	5.66E+10	7.83E+10	7.48E+10	8.47E+10	**8.48E+10**
	Worst	8.22E+10	8.22E+10	8.27E+10	8.10E+10	8.35E+10	8.16E+10	8.15E+10	7.90E+10	8.49E+10	**8.49E+10**
	SD	1.57E+09	7.09E+09	3.24E+09	1.50E+09	5.64E+08	2.14E+10	2.55E+09	3.71E+09	1.52E+08	**6.85E+07**
	FR	5.33	7.30	6.10	6.30	3.07	9.07	6.23	8.60	1.57	**1.43**
200-	Best	2.23E+10	2.51E+10	2.56E+10	1.91E+10	2.58E+10	1.98E+10	2.13E+10	2.40E+10	2.70E+10	**2.75E+10**
bar	Mean	2.43E+10	2.61E+10	2.62E+10	2.26E+10	2.63E+10	2.47E+10	2.31E+10	2.61E+10	2.75E+10	**2.76E+10**
	Worst	2.68E+10	2.69E+10	2.68E+10	2.45E+10	2.67E+10	2.60E+10	2.46E+10	2.71E+10	2.77E+10	**2.77E+10**
	SD	1.11E+09	5.28E+08	3.16E+08	1.20E+09	1.80E+08	1.26E+09	7.51E+08	7.99E+08	1.78E+08	**3.64E+07**
	FR	7.67	4.70	4.87	9.47	4.33	7.27	9.13	4.57	1.70	**1.30**
942-	Best	1.07E+11	1.17E+11	1.21E+11	8.73E+10	1.15E+11	1.07E+11	9.51E+10	1.14E+11	1.27E+11	**1.29E+11**
bar	Mean	1.18E+11	1.20E+11	1.23E+11	9.94E+10	1.20E+11	1.15E+11	1.03E+11	1.26E+11	1.29E+11	**1.30E+11**
	Worst	1.23E+11	1.25E+11	1.26E+11	1.07E+11	1.22E+11	1.21E+11	1.12E+11	1.30E+11	1.31E+11	**1.31E+11**
	SD	4.43E+09	2.28E+09	1.20E+09	3.74E+09	1.71E+09	3.08E+09	3.23E+09	3.14E+09	9.20E+08	**4.07E+08**
	FR	6.23	5.60	4.17	9.83	6.13	7.57	9.13	3.27	1.93	**1.13**

Note: SD denotes the standard deviation, FR represents the Friedman rank.

Table 4 Statistical inverted generational distance (IGD) for each truss problem

Algorithm		MOPSO	NSGA-II	MOEA/D	MOGOA	MOMVO	MOWCA	MOSSA	UPSEMOA	SHAMODE	Proposed
10-	Best	1.06E+03	4.37E+02	1.39E+03	5.12E+02	2.46E+02	1.76E+03	1.02E+03	7.61E+02	1.37E+02	**7.92E+01**
bar	Mean	2.12E+03	1.05E+03	2.81E+03	2.04E+03	4.52E+02	4.94E+03	2.82E+03	1.39E+03	2.88E+02	**1.35E+02**
	Worst	3.36E+03	1.93E+03	5.10E+03	3.75E+03	8.61E+02	7.68E+03	4.22E+03	2.48E+03	4.89E+02	**2.13E+02**
	SD	5.98E+02	4.13E+02	8.39E+02	9.18E+02	1.61E+02	1.80E+03	7.85E+02	4.01E+02	8.26E+01	**3.91E+01**
	FR	6.77	4.53	8.03	6.77	2.97	9.50	8.07	5.20	2.17	**1.00**
25-	Best	4.85E+01	1.17E+02	4.29E+01	1.86E+02	7.53E+01	1.63E+02	1.01E+02	1.67E+02	2.71E+01	**2.07E+01**
bar	Mean	1.28E+02	4.19E+02	4.66E+02	6.18E+02	1.14E+02	1.50E+03	3.38E+02	3.19E+02	4.04E+01	**2.60E+01**
	Worst	2.88E+02	8.02E+02	1.02E+03	1.42E+03	1.65E+02	2.15E+03	9.75E+02	9.10E+02	8.53E+01	**5.36E+01**
	SD	5.93E+01	1.66E+02	2.52E+02	3.55E+02	2.42E+01	5.01E+02	1.86E+02	1.33E+02	1.14E+01	**5.62E+00**
	FR	3.77	7.23	7.27	7.77	3.63	9.53	6.27	6.50	2.00	**1.03**
37-	Best	2.63E+02	1.66E+02	1.57E+02	1.45E+02	6.29E+01	2.94E+02	2.83E+02	1.76E+02	4.10E+01	**1.42E+01**
bar	Mean	4.01E+02	2.99E+02	4.92E+02	3.61E+02	1.27E+02	5.54E+02	5.11E+02	3.27E+02	7.79E+01	**2.95E+01**
	Worst	5.09E+02	4.94E+02	7.81E+02	5.39E+02	3.29E+02	9.54E+02	8.99E+02	4.98E+02	1.25E+02	**5.95E+01**
	SD	6.50E+01	8.49E+01	1.18E+02	9.26E+01	5.90E+01	1.67E+02	1.49E+02	8.41E+01	2.59E+01	**1.02E+01**
	FR	6.83	5.17	8.33	6.27	2.93	8.50	8.13	5.70	2.10	**1.03**
120-	Best	8.67E+03	2.96E+03	1.17E+04	7.75E+03	1.45E+03	6.96E+03	8.11E+03	4.97E+03	7.45E+02	**5.05E+02**
bar	Mean	1.69E+04	7.54E+03	1.96E+04	1.45E+04	2.36E+03	3.51E+04	1.63E+04	1.22E+04	2.50E+03	**1.02E+03**
	Worst	2.35E+04	1.81E+04	3.04E+04	2.50E+04	4.54E+03	5.79E+04	3.18E+04	2.09E+04	5.43E+03	**2.31E+03**
	SD	3.57E+03	3.33E+03	5.10E+03	4.54E+03	7.63E+02	1.34E+04	5.49E+03	3.86E+03	1.17E+03	**4.26E+02**
	FR	7.43	4.37	8.23	6.83	2.47	9.50	6.90	5.70	2.40	**1.17**
200-	Best	3.22E+03	2.24E+03	4.07E+03	3.89E+03	2.08E+03	3.05E+03	4.36E+03	2.05E+03	7.36E+02	**1.70E+02**
bar	Mean	5.62E+03	4.25E+03	5.74E+03	6.61E+03	3.76E+03	5.83E+03	6.20E+03	3.70E+03	2.13E+03	**4.27E+02**
	Worst	8.26E+03	6.40E+03	7.13E+03	9.00E+03	4.97E+03	9.33E+03	8.02E+03	5.73E+03	3.78E+03	**8.15E+02**
	SD	1.26E+03	1.03E+03	7.65E+02	1.24E+03	7.21E+02	1.46E+03	1.06E+03	9.14E+02	7.03E+02	**1.78E+02**
	FR	7.00	4.73	7.60	8.53	4.10	7.27	8.30	4.27	2.20	**1.00**
942-	Best	5.57E+03	3.75E+03	9.73E+03	1.36E+04	7.87E+03	6.97E+03	1.14E+04	1.57E+03	2.90E+03	**5.04E+02**
bar	Mean	9.79E+03	8.28E+03	1.48E+04	1.67E+04	1.20E+04	1.10E+04	1.47E+04	3.08E+03	4.57E+03	**7.69E+02**
	Worst	1.32E+04	1.27E+04	2.04E+04	2.02E+04	1.78E+04	1.51E+04	1.97E+04	5.95E+03	6.74E+03	**1.39E+03**
	SD	1.97E+03	2.10E+03	2.79E+03	1.58E+03	2.17E+03	2.14E+03	1.88E+03	7.97E+02	9.85E+02	**1.84E+02**
	FR	5.50	4.57	8.27	9.47	6.93	5.97	8.30	2.07	2.93	**1.00**

Table 5 Statistical spacing-to-extent (STE) for each truss problem

Algorithm		MOPSO	NSGA-II	MOEA/D	MOGOA	MOMVO	MOWCA	MOSSA	UPSEMOA	SHAMODE	Proposed
10-	Best	6.18E-03	5.08E-03	3.52E-03	**1.11E-03**	5.25E-03	9.13E-03	3.50E-03	3.35E-03	4.02E-03	4.25E-03
bar	Mean	1.37E-02	1.38E-02	7.82E-03	7.63E-03	8.36E-03	3.54E-02	8.31E-03	1.25E-02	5.95E-03	**5.56E-03**
	Worst	2.59E-02	5.66E-02	2.59E-02	1.73E-02	1.19E-02	9.34E-02	1.38E-02	3.12E-02	8.50E-03	**7.12E-03**
	SD	4.97E-03	1.11E-02	4.61E-03	3.68E-03	1.57E-03	2.12E-02	2.61E-03	6.52E-03	1.28E-03	**7.33E-04**
	FR	7.73	6.73	3.97	4.70	5.47	9.60	4.97	6.77	2.77	**2.30**
25-	Best	4.94E-03	5.43E-03	3.00E-03	**2.05E-05**	6.12E-03	6.54E-03	3.31E-03	7.71E-04	3.94E-03	3.22E-03
bar	Mean	9.72E-03	1.81E-02	4.94E-03	5.84E-03	8.96E-03	2.68E-02	9.00E-03	1.47E-02	5.12E-03	**4.57E-03**
	Worst	1.64E-02	6.81E-02	1.13E-02	1.14E-02	1.24E-02	9.25E-02	1.63E-02	4.51E-02	7.36E-03	**6.30E-03**
	SD	2.66E-03	1.21E-02	1.66E-03	3.37E-03	1.49E-03	1.97E-02	3.33E-03	8.73E-03	7.18E-04	**6.27E-04**
	FR	6.43	8.53	2.53	3.77	6.20	9.03	5.73	7.60	3.00	**2.17**
37-	Best	7.12E-03	5.14E-03	3.17E-03	**4.79E-04**	6.07E-03	5.14E-03	3.01E-03	9.92E-04	3.55E-03	3.99E-03
bar	Mean	1.21E-02	1.28E-02	6.03E-03	7.33E-03	8.21E-03	1.80E-02	8.97E-03	1.38E-02	5.98E-03	**5.28E-03**
	Worst	2.36E-02	2.41E-02	1.75E-02	1.14E-02	1.29E-02	4.32E-02	1.50E-02	6.70E-02	1.05E-02	**6.82E-03**
	SD	3.91E-03	4.72E-03	3.53E-03	2.30E-03	1.64E-03	8.16E-03	2.62E-03	1.22E-02	1.78E-03	**8.28E-04**
	FR	7.50	7.37	3.07	4.50	5.33	8.87	5.87	6.67	3.20	**2.63**
120-	Best	7.32E-03	2.88E-03	4.04E-03	**7.16E-04**	6.02E-03	7.99E-03	3.72E-03	2.66E-03	4.12E-03	4.60E-03
bar	Mean	1.20E-02	1.42E-02	7.82E-03	6.23E-03	8.67E-03	4.08E-02	9.98E-03	1.69E-02	5.88E-03	**5.76E-03**
	Worst	2.38E-02	2.53E-02	2.43E-02	1.29E-02	1.12E-02	8.89E-02	3.36E-02	3.58E-02	1.10E-02	**8.36E-03**
	SD	3.85E-03	5.87E-03	4.74E-03	3.59E-03	1.25E-03	2.60E-02	6.02E-03	8.09E-03	1.39E-03	**9.60E-04**
	FR	7.13	7.27	3.63	3.37	5.27	9.37	5.50	7.87	2.87	**2.73**
200-	Best	4.50E-03	4.05E-03	3.01E-03	**5.66E-04**	4.62E-03	5.33E-03	2.61E-03	2.16E-03	3.04E-03	3.28E-03
bar	Mean	9.16E-03	1.44E-02	6.70E-03	6.36E-03	6.94E-03	1.24E-02	8.24E-03	8.87E-03	7.79E-03	**4.88E-03**
	Worst	1.66E-02	4.71E-02	2.99E-02	1.48E-02	1.10E-02	2.74E-02	2.28E-02	4.88E-02	2.17E-02	**8.21E-03**
	SD	3.25E-03	9.08E-03	4.97E-03	3.27E-03	1.58E-03	5.07E-03	3.90E-03	8.49E-03	4.05E-03	**1.25E-03**
	FR	6.63	8.57	3.47	4.27	5.20	8.03	5.93	5.03	5.33	**2.53**
942-	Best	4.00E-03	4.21E-03	4.24E-03	**1.65E-03**	4.16E-03	6.29E-03	2.92E-03	2.67E-03	3.62E-03	2.86E-03
bar	Mean	6.34E-03	9.89E-03	9.82E-03	6.96E-03	9.85E-03	1.16E-02	9.21E-03	5.74E-03	4.82E-03	**3.87E-03**
	Worst	9.34E-03	3.94E-02	3.56E-02	2.69E-02	4.17E-02	2.58E-02	2.92E-02	3.08E-02	6.81E-03	**5.49E-03**
	SD	1.36E-03	7.32E-03	5.99E-03	5.29E-03	7.38E-03	4.09E-03	5.16E-03	4.92E-03	8.86E-04	**5.08E-04**
	FR	5.33	6.93	7.00	4.83	6.87	8.63	6.63	3.47	3.50	**1.80**

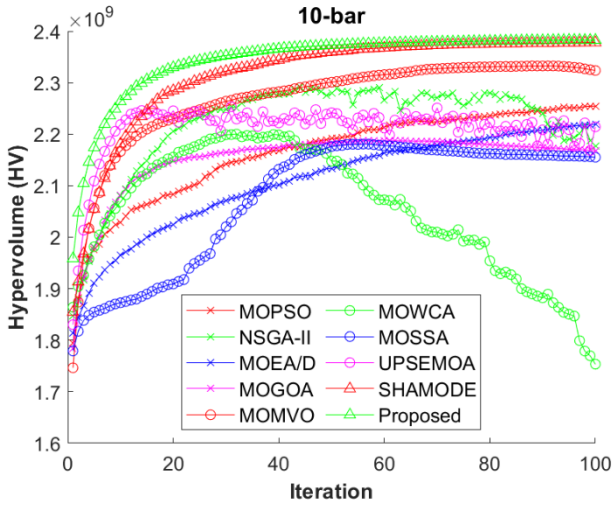
6.2 10-bar truss

This subsection considers the 10-bar truss example as the first benchmark (Fig. 3), which was evaluated by the proposed MO-SHADE-MRFO and 9 compared multi-objective metaheuristic algorithms. The details of loads, elements and dimensions are summarized in Table 1. The statistical results of HV, IGD and STE in the 10-bar truss problem solved by all metaheuristic algorithms are reported in Tables 3-5, while the best results of metrics are highlighted in bold.

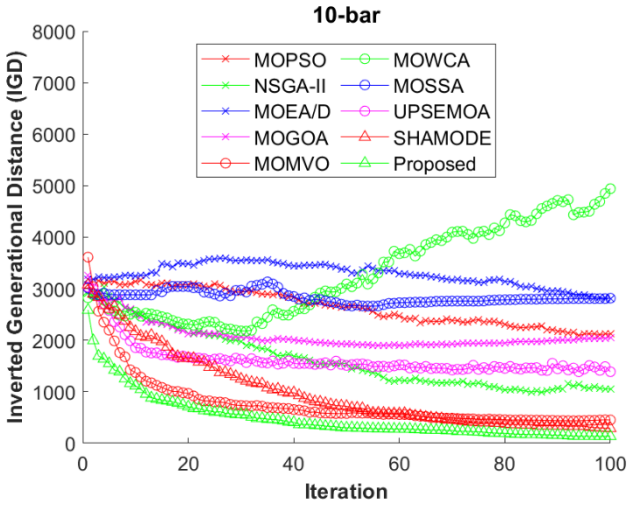
According to the results of HV in 10-bar truss from Table 3, the proposed algorithm obtains the first rank among 10 algorithms, with the best Friedman rank value (FR=1.10). It achieves the best statistical values of HV, while the mean HV value is 2.38E+09. The second and third algorithms are MOSHADE (FR=1.90) and MOMVO (FR=3.27), respectively. In comparison, three classical algorithms (MOPSO, NSGA-II and MOEA/D) provide unstable HV results in the 10-bar truss problem. Table 4 summarizes the statistical IGD values by each algorithm in the 10-bar truss, while the best, mean, worst and SD values of IGD by the proposed algorithm are also better than the results from comparing algorithms. The proposed algorithm ranks first with the FR value 1.0, while the second algorithm is SHAMODE with the FR value 2.17, the third algorithm is MOMVO with the FR value 2.97. For STE values from Table 5, the mean, worst and SD values of STE from the proposed algorithm are also better than other algorithms. Obviously, the proposed algorithm ranks first among 10 algorithms, with the FR value 2.30, followed by SHAMODE and MOEA/D.

To illustrate the convergence of the proposed algorithm, the iterative curves of metrics in 10-bar truss are displayed in Fig. 10(a)-(c). It can be seen that the proposed algorithm and SHAMODE have good consistency and convergence rate in HV and IGD curves, but the initial convergence speed of the proposed algorithm is better than SHAMODE. Moreover, unstable results or oscillation of three metrics among search history occur in some compared algorithms, such as MOPSO, NSGA-II, MOWCA, MOSSA and UPSEMOA. In addition, the obtained Pareto fronts for 10-bar truss are displayed in Fig. 10(d). The proposed algorithm can provide smooth and well-distributed Pareto solutions, while the Pareto solutions of NSGA-II, MSSA, MOWCA and UPSEMOA are scattered and non-continuous.

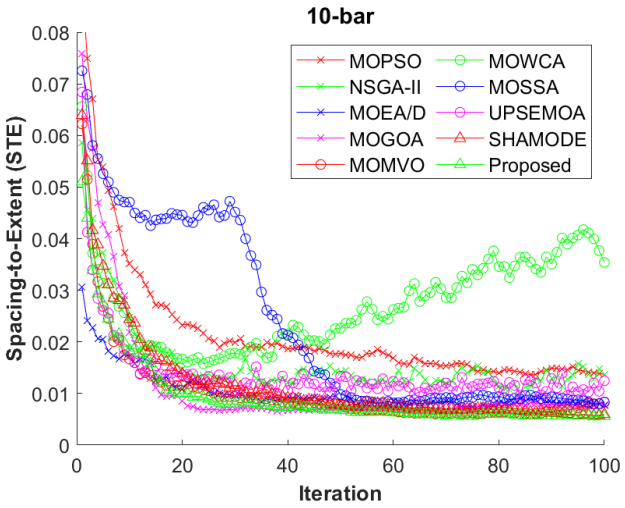
It can be concluded that MO-SHADE-MRFO is better than the compared algorithms in this problem with high convergence and coverage. The good convergence of MO-SHADE-MRFO is achieved by the operators from SHADE and MRFO. In the mutation and crossover phases, the success-based history parameter adaptation in SHADE can ensure convergence in the objective space, while the operators (cyclone, chain, somersault) in MRFO embedded into the SHADE can enhance the diversity and convergence in the objective space. Moreover, the good coverage of the proposed algorithm can be attributed to its archive maintenance. Therefore, the proposed algorithm can perform well on the 10-bar truss problem.



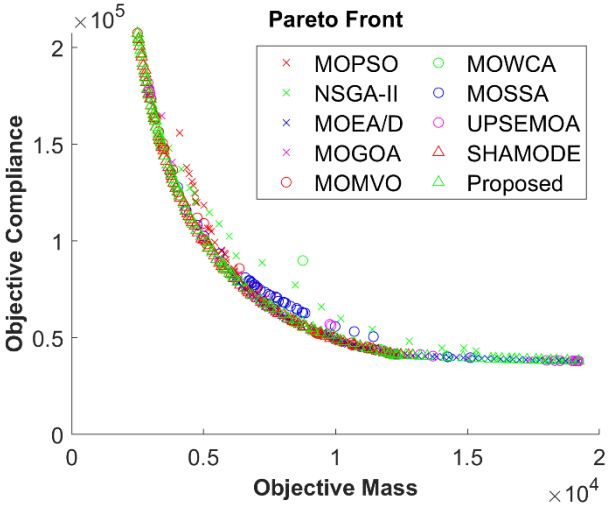
(a) HV iterative curves



(b) IGD iterative curves



(c) STE iterative curves



(d) Pareto front

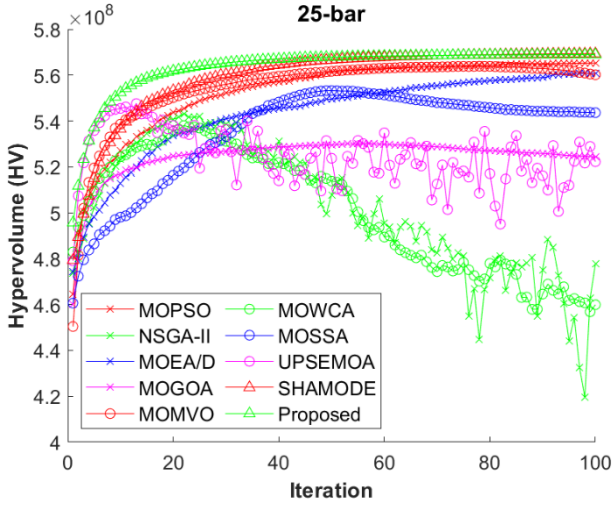
Fig. 10. Iterative curves and Pareto front of 10-bar truss problem

6.3 25-bar truss

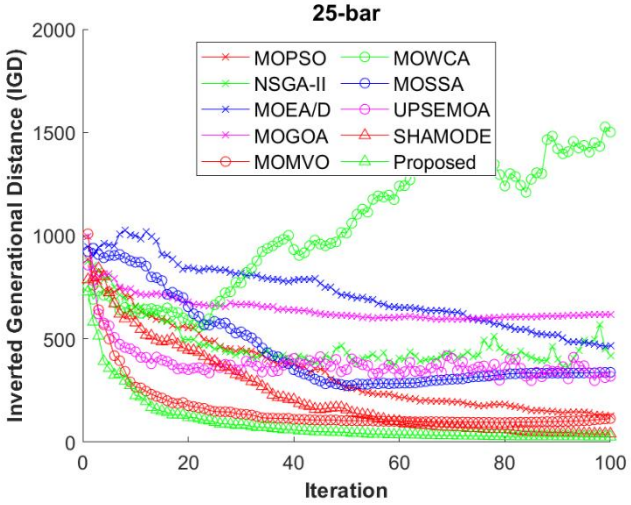
A 25-bar truss is the second problem depicted in Fig. 4. The density and elastic modulus in this problem are 7850 kg/m^3 and 200 GPa, respectively, and the allowable tensile and compressive stresses are 400 MPa. The cross-sectional areas of 25 elements are grouped as 8 design variables. The details of boundaries, design variables, and properties are summarized in Table 1.

The statistical results of three metrics (HV, IGD, STE) in the 25-bar truss problem obtained by all multi-objective metaheuristic algorithms are summarized in Tables 3-5. The proposed algorithm ranks the first in IGD (FR=1.03) and STE (FR=2.17), and the second in HV (FR=1.93), outperforming the compared algorithms. For comparison, SHAMODE ranks the first in HV values, the second in IGD values, and the third in STE values; NSGA-II and WOWCA are the last two algorithms in three metrics. The statistical results of three metrics demonstrate the superior performance of the proposed algorithm.

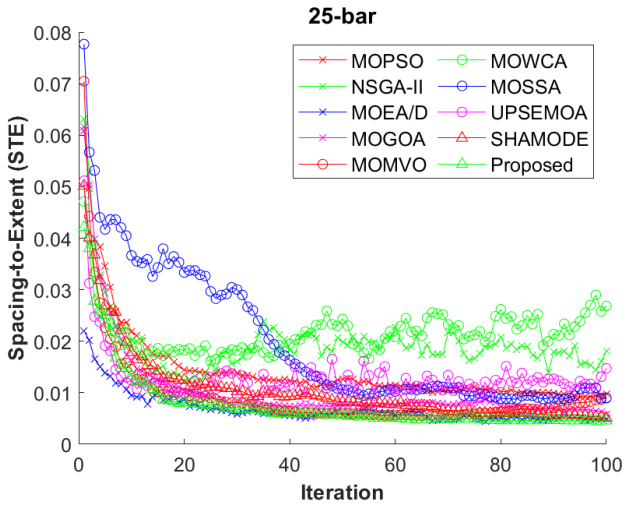
Figs. 11(a-c) display the convergence curves of three metrics (HV, IGD, STE) of each algorithm in the 25-bar truss problem, respectively. UPSEMOA, MOWCA and NSGA-II generate unstable convergence curves in the HV metric during the optimization process, as shown in Fig. 11(a), while MOGOA and MOSSA provide premature convergence. MOPSO, MOMVO, SHAMODE and the proposed algorithm can achieve stable convergence curve, while the proposed algorithm has the best convergence rate in the HV metric. Based on the convergence curves of IGD shown in Fig. 11(b), UPSEMOA, NSGA-II, MOEA/D, MOSSA and MOWCA have oscillation curves during optimization, while the proposed algorithm has the stable and fast convergence rate in IGD metric. Fig. 11(c) indicates the proposed algorithm provides good performance of STE convergence, with satisfactory competitiveness among the compared algorithms. The Pareto front of 25-bar truss problem calculated by each algorithm is also illustrated in Fig. 11(d). MSSA, NSGA-II and MOMVO provide discontinuous Pareto fronts, and the best Pareto fronts obtained by the proposed algorithm are well-distributed.



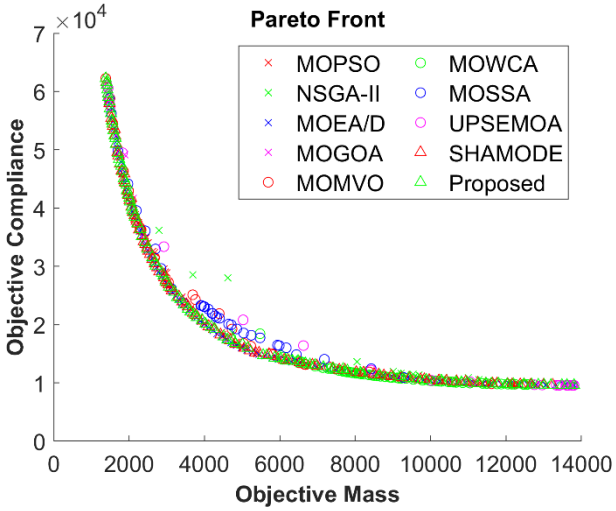
(a) HV iterative curves



(b) IGD iterative curves



(c) STE iterative curves



(d) Pareto front

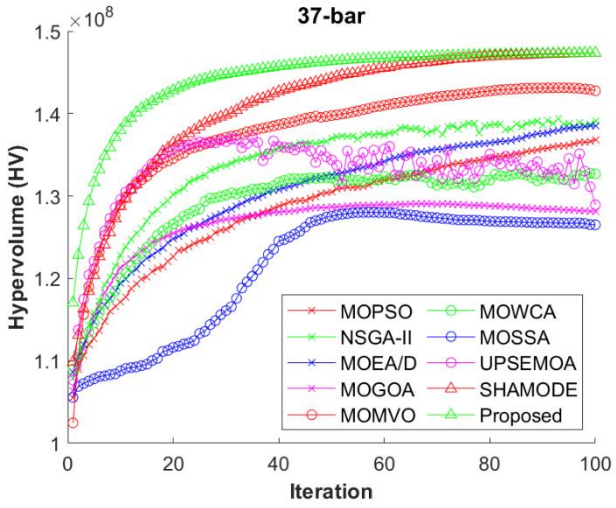
Fig. 11. Iterative curves and Pareto front of 25-bar truss problem

6.4 37-bar truss

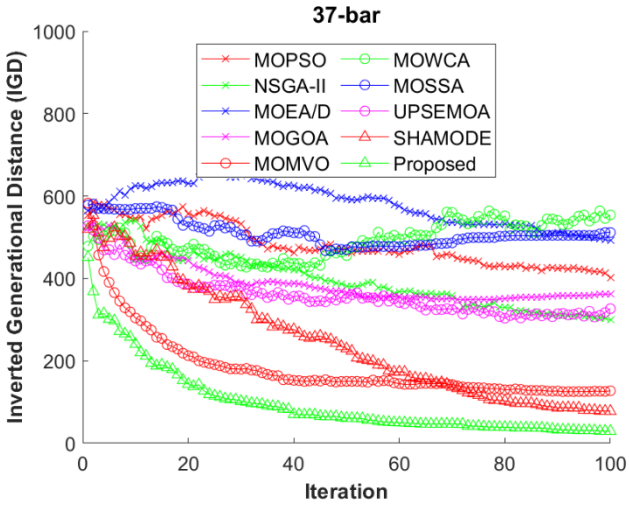
The third benchmark problem is 37-bar truss problem, as shown in Fig. 5. The details of boundaries, design variables, and properties are summarized in Table 1. The cross-sectional areas of 37 bars are grouped as 15 discrete design variables. The statistical results are summarized in Tables 3-5.

The results of the HV metric in Table 3 illustrate that both SHAMODE and the proposed algorithm achieve good convergence in HV values. The mean HV value by the proposed algorithm is 1.47E+08, which is equal to SHAMODE and better than other compared algorithms. MOPSO, MOMVO, NSGA-II and MOEA/D are inferior to the proposed algorithm, and MOSSA ranks the last in the HV metric. In the IGD results from Table 4, the proposed algorithm achieves the first rank with the FR value 1.03, which is far better than other algorithms. SHAMODE and MOMVO are the follower algorithms with FR values 2.10 and 2.93, respectively. Table 5 also illustrates that the proposed algorithm performs well on the STE metric, especially in the mean, worst and SD values of STE, which achieves the first rank with the FR value 2.63, followed by MOEA/D and SHAMODE. In all, the proposed algorithm has very competitive performance in the HV, IGD and STE metrics in the 37-bar truss problem.

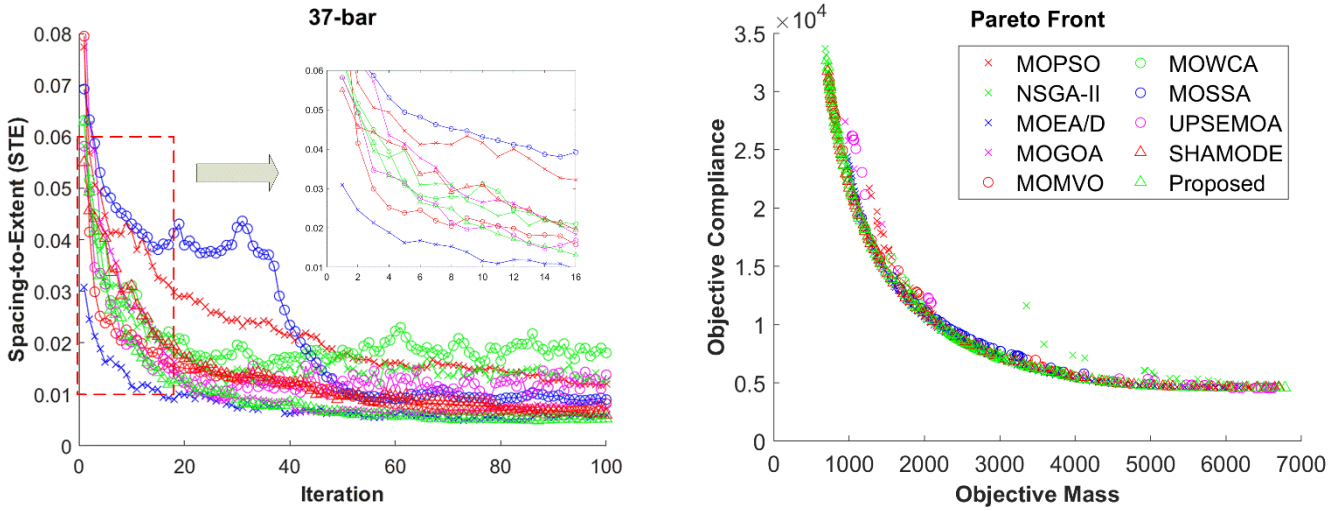
According to the HV and IGD curves from Fig. 12(a) and (b), the proposed algorithm has good convergence performance in the convergence rate in the HV and IGD metrics, while SHAMODE and MOMVO followed as the second and third at the convergence ability. UPSEMOA, MOWCA, NSGA-II, MOSSA and MOGOA have premature convergence or oscillation curves. Fig. 12(c) shows the convergence curve of STE. Some compared algorithms cannot provide good convergence, but the proposed algorithm can achieve good convergence ability. Fig. 12(d) displays the Pareto fronts obtained by each algorithm, and the proposed algorithm can perform a smooth and well-distributed Pareto solutions, which is better than the other compared algorithms. The results indicate that the MO-SHADE-MRFO helps in preventing the local optima trap due to the good balance between global diversification and local intensification.



(a) HV iterative curves



(b) IGD iterative curves



(c) STE iterative curves

(d) Pareto front

Fig. 12. Iterative curves and Pareto front of 37-bar truss problem

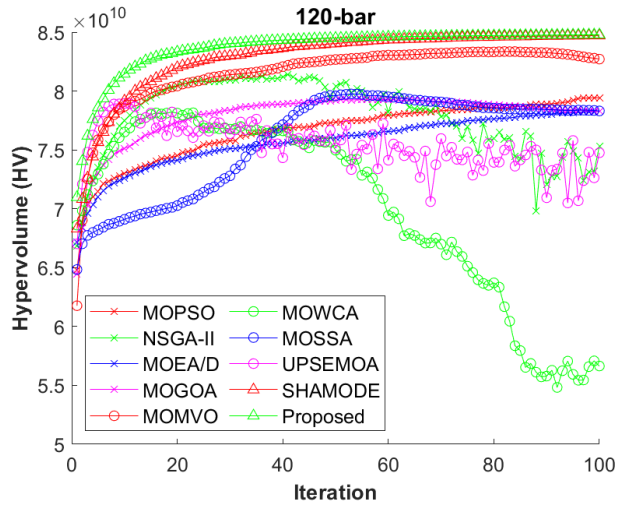
6.5 120-bar truss

A 120-bar truss problem is considered as the fourth benchmark problem, which is depicted in Fig. 6. The details of the 120-bar truss problem are listed in Table 1. The cross-sectional areas of 120 elements are grouped as 7 discrete design variables. The statistical results of three metrics (HV, IGD, STE) by 10 algorithms are summarized in Tables 3-5.

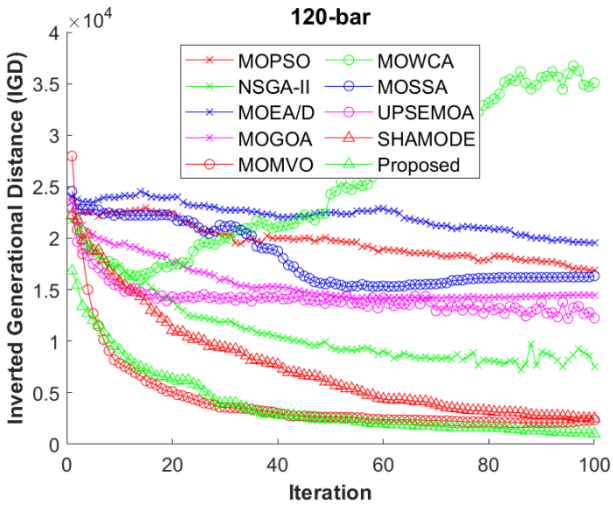
From the statistical results of HV in Table 3 for the 120-bar truss, the proposed algorithm ranks the first in the HV values with the FR value 1.43, while the second and third algorithms are SHAMODE (FR=1.57) and MOMVO (FR=3.07), respectively. The classical algorithms including MOPSO (FR=5.33), NSGA-II (FR=7.30) and MOEA/D (FR=6.10) rank the median among 10 algorithms, while MOWCA ranks the last in the HV values (FR=9.07). The IGD values of the 120-bar truss problem are also displayed in Table 4, illustrating that the proposed algorithm has superior performance and ranks the first (FR=1.17) among 10 algorithms. Based on the STE values in Table 5, the mean, worst and SD values of STE by the proposed algorithm are better than the results from other algorithms. The proposed algorithm also achieves the first rank with the FR value 2.73, followed by SHAMODE (FR=2.87) and MOGOA (FR=3.37). Thus, the proposed algorithm has superior performance in the convergence and coverage abilities for the 120-bar truss problem.

The convergence curves of HV shown in Fig. 13(a) illustrate that MOWCA, NSGA-II, UPSEMOA and MOSSA may have the oscillating curves. Meanwhile, the premature occurs in algorithms MOEA/D, MOPSO, MOGOA, and MOMVO. SHAMODE and the proposed algorithm outperforms other algorithms, and the proposed algorithm has faster convergence rate than SHAMODE. In Figs. 13(b-c), the proposed algorithm provides very competitive convergence capacities in the IGD and STE metrics, which is superior to most algorithms. The results of Pareto fronts illustrated in Fig. 13(d) show that the proposed

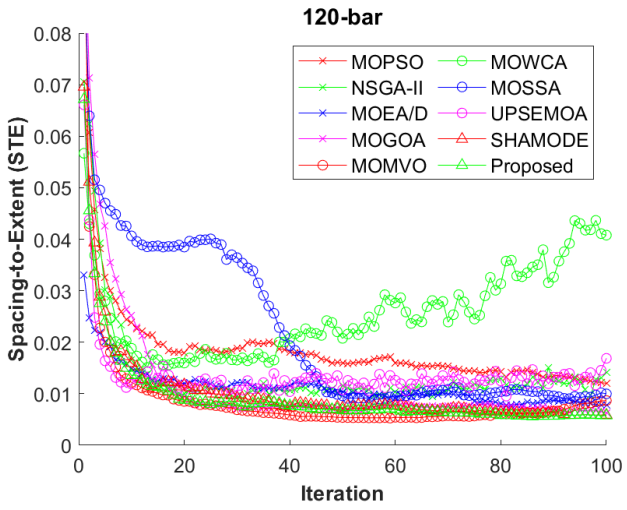
algorithm can provide spread, consistent and smooth Pareto solutions.



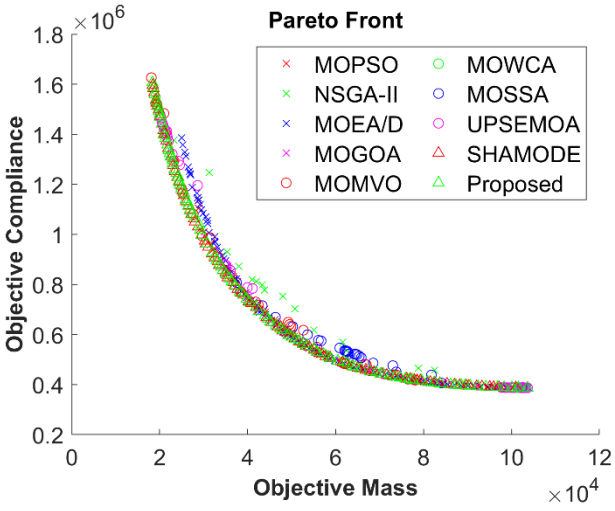
(a) HV iterative curves



(b) IGD iterative curves



(c) STE iterative curves



(d) Pareto front

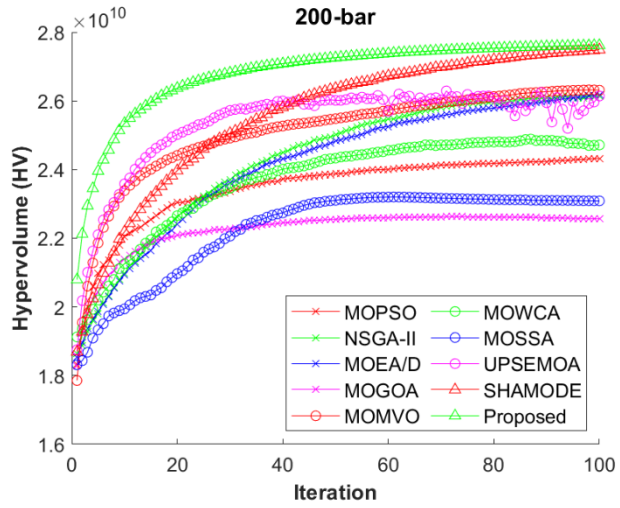
Fig. 13. Iterative curves and Pareto front of 120-bar truss problem

6.6 200-bar truss

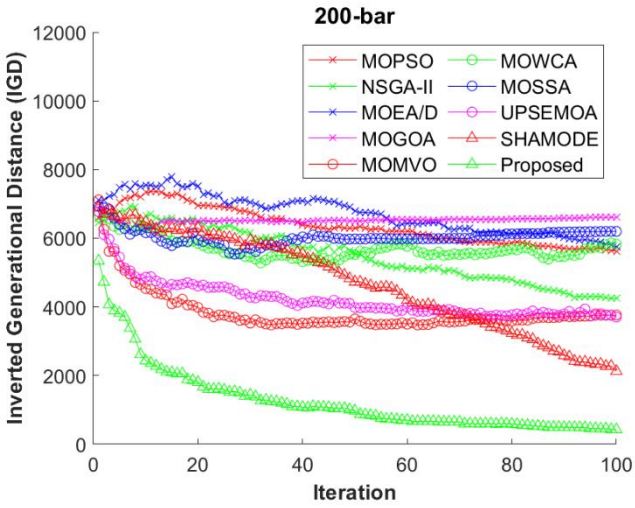
This subsection comprises the 200-bar truss example (Fig. 7), to test the performance of the proposed algorithm. Table 1 gives the details of loads, elements, and dimensions of the 200-bar truss. The 200-bar truss, which has 200 elements (meaning 200 constraints in the truss optimization problem) and 29 design variables.

The statistical results of HV, IGD and STE of 200-bar truss problem are summarized in Tables 3-5, demonstrating that the proposed algorithm achieves the best values of HV (FR=1.30), IGD (FR=1.00) and STE (FR=2.53) metrics among 10 algorithms. Fig. 14 displays the iterative curves of HVs, IGDs, STEs and the obtained Pareto fronts of 10 algorithms for this problem. The performance of the proposed algorithm is far better than the compared algorithms, especially in HV and IGD curves. In addition, the

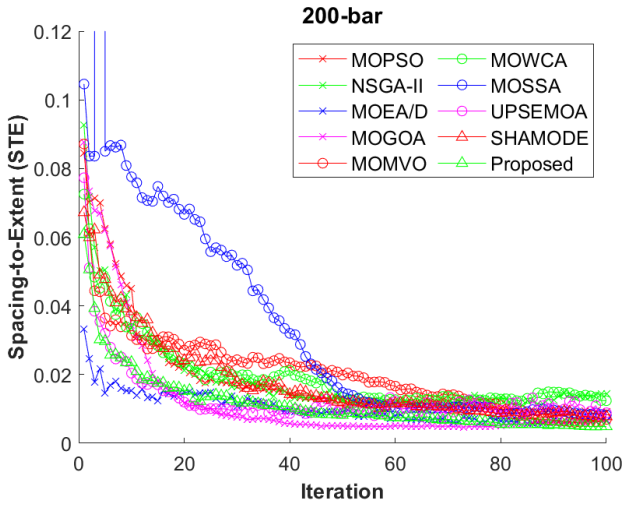
proposed algorithm can provide a well-spread and smooth Pareto front in this problem.



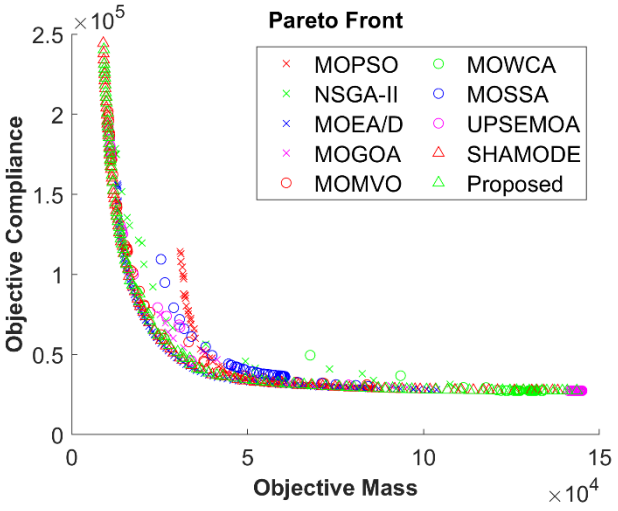
(a) HV iterative curves



(b) IGD iterative curves



(c) STE iterative curves



(d) Pareto front

Fig. 14. Iterative curves and Pareto front of 200-bar truss problem

6.7 942-bar truss

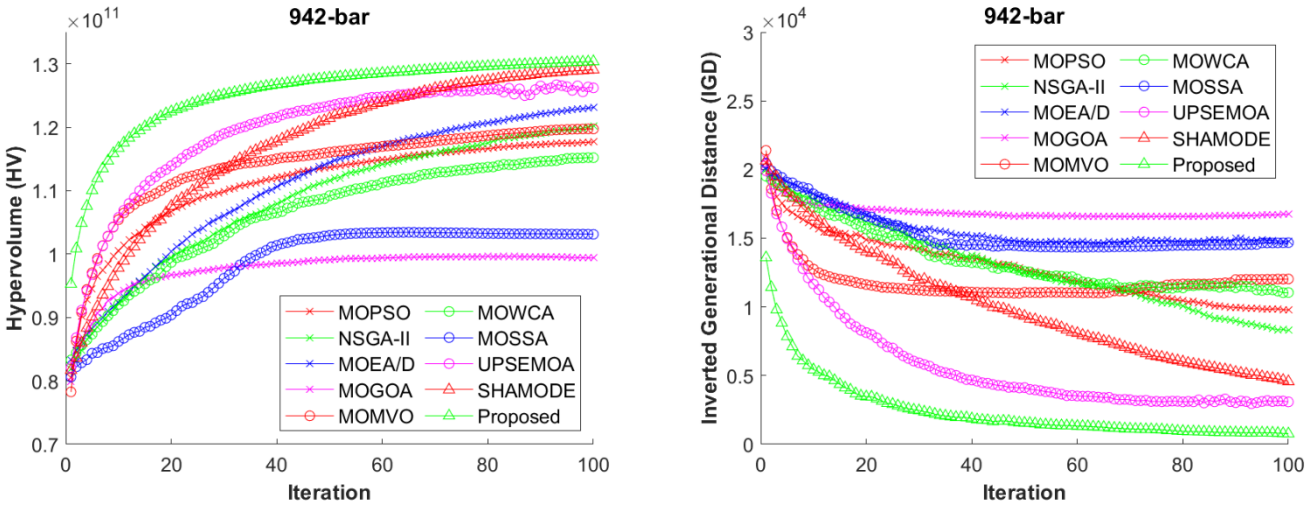
The 3D tower truss structure is the last benchmark problem in this study, depicted in Fig. 8, with 942 elements and 942 constraints, which greatly limits the search space of the design variables. The total 942 elements are grouped to 59 parts based on the structural symmetry, so this problem is typically a high-dimensional optimization problem. All properties of loads, elements and dimensions are listed in Table 1.

The statistical results of three metrics (HV, IGD, STE) of the 942-bar truss are summarized in Tables 3-5. According to the HV results from Table 3, the proposed algorithm achieves good performance in the HV metric, with the mean HV value 1.30E+11 and the FR value 1.13, which is better than the compared algorithms. The SHAMODE and UPSEMOA achieve the second and the third algorithm, with the FR

value 1.93 and 3.27, respectively. Table 4 summarizes the statistical results of IGD, illustrating that the proposed algorithm ranks the first among 10 algorithms, with the FR value 1.00, while the second and third algorithms are UPSEMOA (FR=2.07) and SHAMODE (FR=2.93), respectively. The results of STE values in Table 5 illustrate that the proposed algorithm ranks the first (FR=1.80) among the 10 algorithms, while SHAMODE (FR=3.47) and UPSEMOA (FR=3.50) rank 2nd and 3rd, respectively. To sum up, the proposed algorithm has superior performance in the statistical results of three metrics for this problem.

The convergence capacities of three metrics are also analyzed to evaluate the proposed algorithm. Fig. 15(a) illustrates the HV iterative curves of the proposed algorithm and other compared algorithms. It is obvious that the proposed algorithm has the fastest convergence rate, better than other compared algorithms. Fig. 15(b) illustrates the IGD iterative curves, demonstrating the good convergence ability of the proposed algorithm. In all, the proposed algorithm has good performance in solving complex multi-objective truss optimization problem.

According to the convergence behaviors of algorithms in different multi-objective truss optimization problems, some compared algorithms such as NSGA-II, MOEA/D, MOPSO, UPSEMOA and MOWCA give unstable convergence behaviors of HV and IGD metrics in several tested problems, illustrating their insufficient capacity of exploration and exploitation for multi-objective truss optimization problems. However, the proposed algorithm can provide the table and fast convergence behavior of HV and IGD metrics in all tested optimization problems, by using the updating rules of MO-SHADE-MRFO to ensure the exploration capacity, which can obtain and update high-quality Pareto solutions. It should be noted that although the convergence behaviors of SHAMODE and MO-SHADE-MRFO are similar, the obtained HV and IGD values of MO-SHADE-MRFO are better than those of SHAMODE, which benefits from the hybridization of SHADE and MRFO in the updating mechanism.



(a) HV iterative curves

(b) IGD iterative curves

Fig. 15. Iterative curves of 942-bar truss problem

7 Conclusions

A hybrid multi-objective metaheuristic algorithm called MO-SHADE-MRFO is developed to solve structural design problems, utilizing two powerful metaheuristic algorithms, the success-history based parameter adaptive differential evolution (SHADE) and manta ray foraging optimizer (MRFO). MO-SHADE-MRFO is a Pareto-based algorithm, using the external archive to save and update the obtained Pareto fronts, which is convenient to handle the conflict objectives in the multi-objective optimization. Besides, the updating rules of cyclone, chain and somersault foraging behaviors from MRFO are combined with the SHADE to balance the exploration and exploitation in the MO-SHADE-MRFO, which can improve the convergence capacity for multi-objective structural design problems.

In the experiments, 6 truss optimization problems with 10-bar, 25-bar, 37-bar, 120-bar, 200-bar and 942-bar are implemented to verify the proposed algorithm, considering two objectives of minimizing the weight and compliance, under the constraints of elemental stress. The proposed algorithm is also compared with 9 different multi-objective metaheuristic algorithms, consisting of MOPSO, NGSA-II, MOEA/D, MOGOA, MOMVO, MOWCA, MOSSA, UPSEMOA and SHAMODE, where the performances is measured by the metrics of HV, IGD and STE. The experimental results indicate that MO-SHADE-MRFO achieves the best rank of three metrics and outperforms the compared algorithms. The convergence behaviors of HV, IGD and STE from MO-SHADE-MRFO are also better than those of the compared algorithms, and the well-distributed Pareto fronts are also obtained by MO-SHADE-MRFO. It can be concluded that MO-SHADE-MRFO is very competitive in solving multi-objective truss optimization problems.

In our future works, high-dimensional optimization problems with many-objectives are interested to handle by the proposed algorithm. Besides, more hybridization of different operators can be extended into the proposed algorithm.

Acknowledgement

The supports of the National Key Research and Development Program (Grant No. 2019YFA0706803) and the National Natural Science Foundation of China (Grant Nos. 11872142 and 11821202) are greatly appreciated. We sincerely thank Dr. Natee Panagant for opening the source code of multi-objective truss optimization on the website.

References

[1] Rashani A R, Camp C V, Rostamian M, Azizi K, Gandomi A H. Population-based optimization in structural engineering: a review. Artificial Intelligence Review, 2021, https://doi.org/10.1007/s10462-021-10036-w

[2] Ahmadi M, Taghavirashidizadeh A, Javaheri D, Masoumain A, Ghoushchi SJ, Pourasad Y. DQRE-SCnet: a novel hybrid approach for selecting users in Federated learning with deep-Q-reinforcement learning based on spectral clustering. Journal of King Saud University – Computer and Information Sciences, 2022, 34:7445−7458. https://doi.org/10.1016/j.jksuci.2021.08.019

[3] Fathimathul R P P, Orban R, Vadivel K S, Subramanian M, Muthusamy S, Elminaam D S A, Nabil A, Abulaigh L, Ahmadi M, Ali M A S. A novel method for the classification of butterfly species using pre-trained CNN models. Electronics, 2022, 11(13):2016. https://doi.org/10.3390/electronics11132016

[4] Meng Z, Li G, Wang B P, Hao P. A hybrid chaos control approach of the performance measure functions for reliability-based design optimization. Computers and Structures, 2015, 146:32−43. https://doi.org/10.1016/j.compstruc.2014.08.011

[5] Meng Z, Li G, Wang X, Sait S M, Yıldız A R. A comparative study of metaheuristic algorithms for reliability-based design optimization problems. Archives of Computational Methods in Engineering, 2021, 28:1853−1869. https://doi.org/10.1007/s11831-020-09443-z

[6] Falcone R, Lima C, Martinelli E. Soft computing techniques in structural and earthquake engineering: a literature review. Engineering Structures, 2020, 207:110269. https://doi.org/10.1016/j.engstruct.2020.110269

[7] Renkavieski C, Parpinelli R S. Meta-heuristic algorithms to truss optimization: Literature mapping and application. Expert Systems With Applications, 2021, 182:115197. https://doi.org/10.1016/j.eswa.2021.115197

[8] Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks (ICNN), Perth, Australia, 1995. P. 1942−1948.

[9] Holland J H. Adaptation in natural and artificial systems. Michigan: Ann Arbor, University of Michigan Press; 1975

[10] Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 1997, 11: 341−359. https://doi.org/10.1023/A:1008202821328

[11] Yang X S, Gandomi A H. Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 2012, 29(5): 464−483. https://doi.org/10.1108/02644401211235834

[12] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer. Advances in Engineering Software, 2014, 69: 46−61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[13] Mirjalili S, Gandomi A H, Mirjalili S Z, Saremi S, Faris H, Mirjalili S M. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 2017, 114: 163−191. http://dx.doi.org/10.1016/j.advengsoft.2017.07.002

[14] Heidari A A, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. Future Generation Computer Systems, 2019, 97:849−872. https://doi.org/10.1016/j.future.2019.02.028

[15] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a novel optimization algorithm. Knowledge-Based Systems, 2020, 191: 105190. https://doi.org/10.1016/j.knosys.2019.105190

[16] Zhong C, Li G. Comprehensive learning Harris hawks-equilibrium optimization with terminal replacement mechanism for constrained optimization problems. Expert Systems With Applications, 2022, 192: 116432. https://doi.org/10.1016/j.eswa.2021.116432

[17] Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi A H. Marine predators algorithm: a nature-inspired metaheuristic. Expert Systems With Applications, 2020, 152:113377. https://doi.org/10.1016/j.eswa.2020.113377

[18] Zhong C, Li G, Meng Z. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm.

Knowledge-Based Systems, 2022, 251:109215. https://doi.org/10.1016/j.knosys.2022.109215

[19] Pholdee N, Bureerat S. A comparative study of eighteen self-adaptive metaheuristic algorithm for truss sizing optimization. KSCE Journal of Civil Engineering, 2018, 22(8):2982−2993. https://doi.org/10.1007/s12205-017-0095-y

[20] Kaveh A, Talatahari S. Optimum design of skeletal structures using imperialist competitive algorithm. Computers and Structures, 2010, 88:1220−1229. https://doi.org/10.1016/j.compstruc.2010.06.011

[21] Kaveh A, Mahdavi V R. Colliding bodies optimization method for optimum discrete design of truss structures. Computers and Structures, 2014, 139:43−53. https://doi.org/10.1016/j.compstruc.2014.04.006

[22] Mortazavi A, Toğan V. Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm. Applied Soft Computing, 2017, 51:239−252.

[23] Farshchin M, Camp C V, Maniat M. Multi-class teaching-learning-based optimization for truss design with frequency constraints. Engineering Structures, 2016, 106:355−369. https://doi.org/10.1016/j.engstruct.2015.10.039

[24] Degertekin S O, Yalcin Bayar G, Lamberti L. Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints. Computers and Structures, 2021, 245:106461. https://doi.org/10.1016/j.compstruc.2020.106461

[25] Zuo W, Bai J, Li B. A hybrid OC-GA approach for fast and global truss optimization with frequency constraints. Applied Soft Computing, 2014, 14:528−535. https://doi.org/10.1016/j.asoc.2013.09.002

[26] Kaveh A, Mahdipour Moghanni R, Javadi S M. Optimum design of large steel skeletal structures using chaotic firefly optimization algorithm based on the Gaussian map. Structural and Multidisciplinary Optimization, 2019:879−894. https://doi.org/10.1007/s00158-019-02263-1

[27] Gholizadeh S, Danesh M, Gheyratmand C. A new Newton metaheuristic algorithm for discrete-performance-based design optimization of steel moment frames. Computers and Structures, 2020, 234:106250. https://doi.org/10.1016/j.compstruc.2020.106250

[28] Degertekin S O, Lamberti L, Ugur I B. Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm. Applied Soft Computing, 2019, 79:363−390. https://doi.org/10.1016/j.asoc.2019.03.058

[29] Mahjoubi S, Bao Y. Game theory-based metaheuristics for structural design optimization. Computer-Aided Civil and Infrastructure Engineering, 2021, 36:1337−1353. https://doi.org/10.1111/mice.12661

[30] Poitras G J, Cormier G, Nabolle A S. Peloton dynamics optimization: algorithm for discrete structural optimization. ASCE Journal of Structural Engineering, 2021, 147(10):04021164. https://doi.org/10.1061/(ASCE)ST.1943-541X.0003113

[31] Kaveh A, Hamedani K B, Kamalinejad M. An enhanced forensic-based investigation algorithm and its application to optimal design of frequency-constrained dome structures. Computers and Structures, 2021, 256:106643. https://doi.org/10.1016/j.compstruc.2021.106643

[32] Li G, Hu H. Risk design optimization using many-objective evolutionary algorithm with application to performance-based wind engineering of tall buildings. Structural Safety, 2014, 48: 1−14. https://doi.org/10.1016/j.strusafe.2014.01.002

[33] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182−197. https://doi.org/10.1109/4235.996017

[34] Coello Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation, 2004, 8(3):256−279. https://doi.org/10.1109/TEVC.2004.826067

[35] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 2007, 11(6):712−731. https://doi.org/10.1109/TEVC.2007.892759

[36] Mirjalili S Z, Mirjalili S, Saremi S, Faris H, Aljarah I. Grasshopper optimization algorithm for multi-objective

optimization problems. Applied Intelligence, 2018, 48:805−820. https://doi.org/10.1007/s10489-017-1019-8

[37] Mirjalili S, Jangir P, Mirjalili S Z, Saremi S, Trivedi I N. Optimization of problems with multiple objectives using the multi-verse optimization problem. Knowledge-Based Systems, 2017, 134:50−71. https://doi.org/10.1016/j.knosys.2017.07.018

[38] Sadollah A, Eskandar H, Kim J H. Water cycle algorithm for solving constrained multi-objective optimization problems. Applied Soft Computing, 2015, 27:179−298. https://doi.org/10.1016/j.asoc.2014.10.042

[39] Mirjalili S, Gandomi A H, Mirjalili S Z, Saremi S, Faris H, Mirjalili S M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 2017, 114:163−191. https://doi.org/10.1016/j.advengsoft.2017.07.002

[40] Aittokoski T, Miettinen K. Efficient evolutionary approach to approximate the Pareto-optimal set in multiobjective optimization, UPS-EMOA. Optimization Methods and Software, 2010, 25(6):841−858. https://doi.org/10.1080/10556780903548265

[41] Panagant N, Bureerat S, Tai K. A novel self-adaptive hybrid multi-objective meta-heuristic for reliability design of trusses with simultaneous topology, shape and sizing optimization design variables. Structural and Multidisciplinary Optimization, 2019, 60:1937−1955. https://doi.org/10.1007/s00158-019-02302-x

[42] Panagant N, Pholdee N, Bureerat S, Yıldız A R, Mirjalili S. A comparative study of recent multi-objective metaheuristics for solving constrained truss optimization problems. Archives of Computational Methods in Engineering, 2021, 28:4031−4047. https://doi.org/10.1007/s11831-021-09531-8

[43] Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1):67−82. https://doi.org/10.1109/4235.585893

[44] Tanabe R, Fukunaga A. Success-history based parameter adaptation for differential evolution. In: Proceedings of IEEE Congress on Evolutionary Computation, Cancún, México, 2013. p. 1−8. https://doi.org/10.1109/CEC.2013.6557555

[45] Zhao W, Zhang Z, Wang L. Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Engineering Applications of Artificial Intelligence, 2020, 87:103300. https://doi.org/10.1016/j.engappai.2019.103300

[46] Chakraborty S, Sharma S, Saha A K, Chakraborty S. SHADE-WOA: A metaheuristic algorithm for global optimization. Applied Soft Computing, 2021, 113:107866. https://doi.org/10.1016/j.asoc.2021.107866

[47] Kahraman H T, Akbel M, Duman S. Optimization of optimal power flow problem using multi-objective manta ray foraging optimizer. Applied Soft Computing, 2022, 116:108334. https://doi.org/10.1016/j.asoc.2021.108334

[48] Got A, Zouache D, Moussaoui A. MOMRFO: Multi-objective manta ray foraging optimizer for handling engineering design problems. Knowledge-Based Systems, 2022, 237:107880. https://doi.org/10.1016/j.knosys.2021.107880

[49] Coello C A, Christiansen A D. Multiobjective optimization of trusses using genetic algorithms. Computers and Structures, 2000, 75:647−660. https://doi.org/10.1016/S0045-7949(99)00110-8

[50] Mokarram V, Banan M R. A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables. Structural and Multidisciplinary Optimization, 2018, 57:509−533. https://doi.org/10.1007/s00158-017-1764-7

[51] Tejani G G, Pholdee N, Bureerat S, Prayogo D. Multiobjective adaptive symbiotic organisms search for truss optimization problems. Knowledge-Based Systems, 2018, 161:398−414. https://doi.org/10.1016/j.knosys.2018.08.005

[52] Kaveh A, Mahdavi V R. Multi-objective colliding bodies optimization algorithm for design of trusses. Journal of Computational Design and Engineering, 2019, 6:49−59. https://doi.org/10.1016/j.jcde.2018.04.001

[53] Kumar S, Tejani G G, Pholdee N, Bureerat S. Multi-objective passing vehicle search algorithm for structure optimization. Expert Systems With Applications, 2021, 169:114511. https://doi.org/10.1016/j.eswa.2020.114511

[54] Tejani G G, Kumar S, Gandomi A H. Multi-objective heat transfer search algorithm for truss optimization. Engineering with Computers, 2021, 37:641−662. https://doi.org/10.1007/s00366-019-00846-6

[55] Kumar S, Tejani G G, Pholdee N, Bureerat S. Multi-objective modified heat transfer search for truss optimization. Engineering with Computers, 2021, 37:3439−3454. https://doi.org/10.1007/s00366-020-01010-1

[56] Premkumar M, Jangir, Sowmya R. MOGBO: a new multiobjective gradient-based optimizer for real-world structural optimization problems. Knowledge-Based Systems, 2021, 218:106856. https://doi.org/10.1016/j.knosys.2021.106856

[57] Kaveh A, Ghazaan M I. A new VPS-based algorithm for multi-objective optimization problems. Engineering with Computers, 2020, 36:1029−1040. https://doi.org/10.1007/s00366-019-00747-8

[58] Chou J S, Truong D N. Multiobjective forensic-based investigation algorithm for solving structural design problems, Automation in Construction, 2022, 134:104084. https://doi.org/10.1016/j.autcon.2021.104084

[59] Kumar S, Jangir P, Tejani G G, Premkumar M. MOTEO: a novel physics-based multiobjective thermal exchange optimization algorithm to design truss structures. Knowledge-Based Systems, 2022, 242:108422. https://doi.org/10.1016/j.knosys.2022.108422

[60] Kumar S, Jangir P, Tejani G G, Premkumar M. A decomposition based multi-objective heat transfer search algorithm for structure optimization. Knowledge-Based Systems, 2022, 253:109591. https://doi.org/10.1016/j.knosys.2022.109591

[61] Vargas D E C, Lemonge A C C, Barbosa H J C, Bernardino H S. Solving multi-objective structural optimization problems using GDE3 and NSGA-II with reference points. Engineering Structures, 2021, 239:112187. https://doi.org/10.1016/j.engstruct.2021.112187

[62] Carvalho J P G, Carvalho É C R, Vargas D E C, Hallak P H, Lima B S L P, Lemonge A C C. Multi-objective optimum design of truss structures using differential evolution algorithms. Computers and Structures, 2021, 252:106544. https://doi.org/10.1016/j.compstruc.2021.106544

[63] Lemonge A C C, Carvalho J P G, Hallak P H, Vargas D E C. Multi-objective truss structural optimization considering natural frequencies of vibration and global stability. Expert Systems With Applications, 2021, 165:113777. https://doi.org/10.1016/j.eswa.2020.113777

[64] Anosri S, Panagant N, Burrerat S, Pholdee N. Success history based adaptive multi-objective differential evolution variants with an interval scheme for solving simultaneously topology, shape and sizing truss reliability optimisation. Knowledge-Based Systems, 2022, 253:109533. https://doi.org/10.1016/j.knosys.2022.109533

[65] Ngatchou P, Zarei A, El-Sharkawi M. Pareto multi objective optimization. In: Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, Arlington, VA, USA, 2005. p. 84−91. https://doi.org/10.1109/ISAP.2005.1599245

[66] Derrac J, García, S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 2011, 1(1):3−18. https://doi.org/10.1016/j.swevo.2011.02.002