

# MOTEO: A Novel Multi-Objective Thermal Exchange Optimization Algorithm for Optimal Design of Truss Structures

Nima Khodadadi (✉ [nimakhan.khodadadi@gmail.com](mailto:nimakhan.khodadadi@gmail.com))

Iran University of Science and Technology <https://orcid.org/0000-0002-8348-6530>

Siamak Talatahari

University of Tabriz

Armin Dadras Eslamlou

Iran University of Science and Technology

---

## Research Article

**Keywords:** Thermal Exchange Optimization (TEO), Multi- Objective Thermal Exchange Optimization (MOTEO) algorithm, Truss Structures, mathematical problems

**Posted Date:** July 21st, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-507206/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# MOTEO: A Novel Multi-Objective Thermal Exchange Optimization Algorithm for Optimal Design of Truss Structures

Nima Khodadadi<sup>1,\*</sup>, Siamak Talatahari<sup>2</sup>, Armin Dadras Eslamlou<sup>1</sup>

<sup>1</sup>Iran University of Science and Technology, Tehran, Iran,

Correspondence, Email: [nimakhan.khodadadi@gmail.com](mailto:nimakhan.khodadadi@gmail.com) \*

<sup>2</sup>Department of Civil Engineering, University of Tabriz, Tabriz, Iran

## Abstract

In the present paper, a physics-inspired metaheuristic algorithm is presented to solve multi-objective optimization problems. The algorithm is developed based on the concept of Newtonian cooling law that recently has been employed by the Thermal Exchange Optimization (TEO) algorithm to efficiently solve single-objective optimization problems. The performance of the multi-objective version of TEO (MOTEO) is examined through bi- and tri-objective mathematical problems as well as bi-objective structural design examples. According to the comparisons between the MOTEO and several well-known algorithms, the proposed algorithm can provide quality Pareto fronts with appropriate accuracy, uniformity and coverage for multi-objective problems.

**Keywords:** Thermal Exchange Optimization (TEO); Multi- Objective Thermal Exchange Optimization (MOTEO) algorithm; Truss Structures; mathematical problems

## 1. Introduction

In the design of structures, engineers often encounter some conflicting objectives that negatively influence each other. For instance, on the one hand, they want to increase the strength of the structure to enhance safety, and on the other hand, they may want to decrease the weight of the structure for economic purposes. In mathematics, the optimization of such problems that involve two or more objectives is called multi-objective optimization problems (MOPs) [1].

Generally, there are no easy, pat answers to MOPs. Their answer is usually represented through a set of optimal solutions with a tradeoff between the optimized competitive objectives so that none of these objectives can be improved without degrading some of the other objectives. Satisfying all constraints, Pareto front provides the designer with a set of solutions. One of which can be chosen by the designer as the final design. The application of metaheuristic optimization algorithms recently became a popular research field to resolve the multi-objective design of structures[2].

Metaheuristics are mainly inspired by natural mechanisms, such as evolutionary processes, physical rules, and animals' behavior [3-8]. These approximate optimization techniques are usually able to provide acceptably good solutions for complex problems in a reasonably short time. These methods, particularly population-based metaheuristics, are efficient tools that can be applied to solve MOPs [1]. In the following, a brief overview of two popular metaheuristics, namely Genetic Algorithm and Particle Swarm Optimization, is provided.

Genetic algorithm (GA) mimics the process of natural selection using the mutation, crossover and selection operators [9]. This algorithm originally was developed for single-objective problems. NSGA II is a popular multi-objective version of GA, in which a nondominated sorting mechanism is employed

to label the Pareto sets starting from the first nondominated front [10]. Afterward, a crowding distance measure is attributed to each solution using a crowded-comparison operator that guides the selection process. The algorithm prefers to survive the solutions with lower domination ranks according to the elitism notion, and it selects the solution located in less-crowded regions to preserve the diversity of solutions. Particle Swarm Optimization (PSO), as another popular algorithm, has been developed according to the swarm behavior of animals such as birds and fish [4]. In this algorithm, the particles (solutions) tend to follow the best personal experience as well as the globally best solution experienced by all particles until each iteration. In the multi-objective PSO (MOPSO) [11], a repository with a certain capacity is created in which a set of nondominated solutions are collected to be used in future steps. Furthermore, the objective function search space is divided into equal hypercubes to identify less populated places. This feature not only helps algorithm keep diversity in solutions and spread them along with the entire Pareto, but also provides designers with a wide variety of solutions rather than concentrating on specific regions.

Since structural optimization usually requires considerable computational resources to be executed, researchers constantly seek new efficient algorithms to decrease the computational costs and increase the accuracy of the solutions. Moreover, according to the no free lunch (NFL) theorem, there is no general-purpose search algorithm to outperform other methods in all problems; hence there is a strong need to develop specific modern algorithms for such specific types of problems [12]. For example, recently, Kaveh and Mahdavi [13] introduced MOCBO algorithm to minimize both displacement and weight of truss structures. Their results show that the presented algorithm can be used as an efficient technique for solving MOPs. In MOCBO, the solutions are simulated as colliding bodies, where the laws of the collision between objects are utilized for updating their variables. Luh and Chueh [14] proposed the constrained Multi-Objective Immune Algorithm (CMOIA) that considers the invading mechanism of a body against foreign substances as an analogy for the optimization process. In their research study, the cytokines concept was utilized for handling the design constraints, and the design of truss structures with displacement and weight objectives were examined. Their results indicate the effective performance of CMOIA compared to some other optimization methods. Tejani et al. [15] developed a multi-objective version of the Modified Adaptive Symbiotic Organisms Search (MOMASOS) algorithm which was developed based on the biological growth and survival process in organisms. They applied MOMASOS in structural optimization problems with mass and deflection objective functions and observed that its adaptive version provides superior results compared to those of other algorithms. In addition to the development of the new algorithms, the application of the existing algorithms to various MOPs also engrossed the interest of many researchers. For example, Nan et al. [16] recently optimized the coordinate of the node space truss structures using a multi-objective evolutionary algorithm (MOEA). Any design with suitable boundary conditions and topology can be selected out of their final Pareto set providing designs with both optimum structural weight and minimum displacement. Other similar researches on the development and application of multi-objective optimization can be found in Refs. [17-22].

In the present paper, a multi-objective version of the newly-established physics-based algorithm, so-called TEO [23,24], is introduced, and it is applied to several structural examples with weight and displacement objective functions. Various mathematical benchmark functions are also examined to show the accuracy and quality of its Pareto front compared to true Pareto. The results are compared with those of some well-known algorithms.

The remainder of the paper has been arranged as: In Section 2, the basic concepts of multi-objective optimization are presented; in Section 3, the Thermal Exchange Optimization and its multi-objective version (MOTEO) is presented; in Section 4, the results of mathematical and structural examples are discussed; and the concluding remarks are finally provided in Section 5.

## 2. Basic concepts in multi-objective optimization

This section encompasses some necessary definitions for multi-objective optimization theory that will be used in the next sections. In these definitions, the optimization problems are assumed to be minimization problems.

### Definition 1. The multi-objective optimization problem (MOP)

A MOP is defined as follows

$$MOP = \begin{cases} \min F(x) = \{f_1(x), f_2(x), \dots, f_n(x)\} \\ s. c. x \in S \end{cases} \quad (1)$$

where,  $F(x)$  denotes a set of at least two cost functions ( $f_i(x)$ ), which should be minimized;  $S$  is the space of feasible solutions  $x = (x_1, x_2, \dots, x_k)$ , which are usually constrained between maximum and minimum bounds corresponding to each variable. MOPs often possess more than one solution, none of which can dominate the others. The dominance concept is defined in the next paragraph.

### Definition 2. Pareto dominance

Let's consider two solutions  $u$  and  $v$  with cost functions values  $u = (u_1, u_2, \dots, u_n)$  and  $v = (v_1, v_2, \dots, v_n)$  that fulfill the constraints of a MOP. Providing that MOP is a minimization problem, the solution  $u$  is said to dominate  $v$  ( $u < v$ ), if and only if no component of  $v$  is smaller than the corresponding cost component of  $u$ , and at least one component of  $u$  must be smaller than that of  $v$ . This definition can be mathematically expressed as Eq. (2).

$$u < v \Leftrightarrow \forall i \in \{1, 2, \dots, n\}: u_i \leq v_i \wedge \exists i \in \{1, 2, \dots, n\}: u_i < v_i \quad (2)$$

For example, in Fig. 1 any of the solutions shown in blue cannot dominate red ones. According to this definition for Pareto dominance, the Pareto optimal set is defined as follows.

### Definition 3. Pareto optimal set

For a given MOP ( $F, S$ ), the Pareto optimal set ( $P$ ) is defined as Eq. (3). This set contains all feasible solutions, none of which can be dominated by any other feasible solution.

$$P = \{x \in S | \nexists x' \in S, F(x') < F(x)\} \quad (3)$$

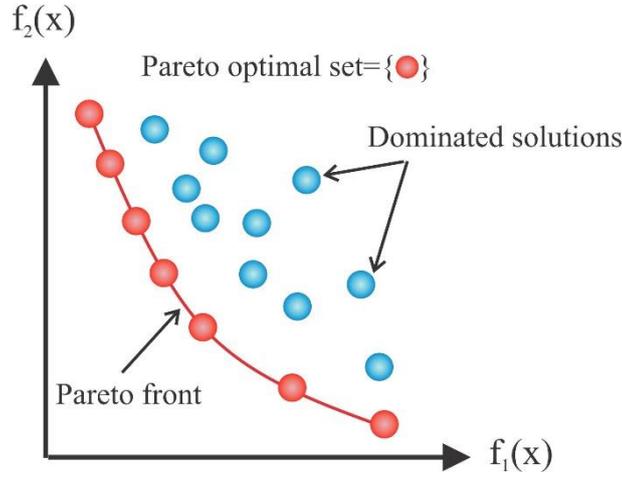
In Fig. 1, solutions drawn in red, for example, are the Pareto optimal set of solutions.

### Definition 4. Pareto front

A Pareto front ( $PF$ ) illustrates the Pareto optimal set in the objective space. According to the definitions mentioned above, it can be expressed as Eq. (4)

$$PF = \{F(x), x \in P\} \quad (4)$$

The red line shown in Fig. 1, illustrates the Pareto front curve, for example.



**Fig. 1.** Pareto optimal solutions

### 3. Thermal Exchange Optimization

Physics-inspired algorithms employ physical phenomena to update search agents inside the search space and thereby solving optimization problems. For example, Gravitational Search Algorithm (GSA) uses Newtonian gravity law, motion laws, and mass interactions to move the search agents (masses) [25]; and Big Bang and Big Crunch (BB-BC) algorithm mimics a theory with the same name to distribute the search particles randomly throughout the search space in the Big Bang phase and shrinks them to a single point as the center of mass [26]. Recently, Kaveh and Dardas [24] proposed a new metaheuristic optimization algorithm, so-called Thermal Exchange Optimization (TEO), which was inspired by Newtonian cooling law. In this algorithm, the search agents are divided into two searching groups, including environment and cooling objects whose temperature represents the optimization variables. In the following, firstly the steps of the original TEO algorithm for a single-objective problem are stated, and the necessary changes for devising a multi-objective version are then explained.

#### Step 1. Initialization of temperatures

First, the search agents—temperatures of objects—are randomly initialized throughout the search space. These solutions can be produced by Eq. (5).

$$\mathbf{T}_i^0 = \mathbf{T}_{min} + \mathbf{rand}_i \cdot (\mathbf{T}_{max} - \mathbf{T}_{min}); i = 1, 2, \dots, n \quad (5)$$

where  $n$  is the number of search objects;  $\mathbf{T}_i^0$  denotes the initial solution of the  $i$ th object;  $\mathbf{T}_{max}$  and  $\mathbf{T}_{min}$  are the vector of upper and lower bounds of solutions, respectively; and  $\mathbf{rand}_i$  is a vector of random numbers, independently generated for  $i$ th object, with a uniform distribution between 0 and 1.

#### Step 2. Evaluation of objects

After assigning temperatures to the objects, they are evaluated by cost functions. The objects are sorted according to their cost function values (costs) in descending order, and the first  $nPop$  number of objects—equal to the assumed number of objects—are preserved.

#### Step 3. Saving the best objects

Some of the historically best solutions are saved in Thermal Memory (TM). Therefore, in each iteration, if the new better solution(s) with a lower cost than the current TM is found, TM is updated and the new solutions with lower costs are saved. This step helps the algorithm keep the position of promising regions in the search space and boost the exploitation (intensification).

#### Step 4. Creating groups

The sorted objects are paired, such that the first half is considered as the environment for objects in the second half and vice versa. For example, as shown in Fig. 2,  $T_1$  is paired with  $T_{\frac{n}{2}+1}$ .

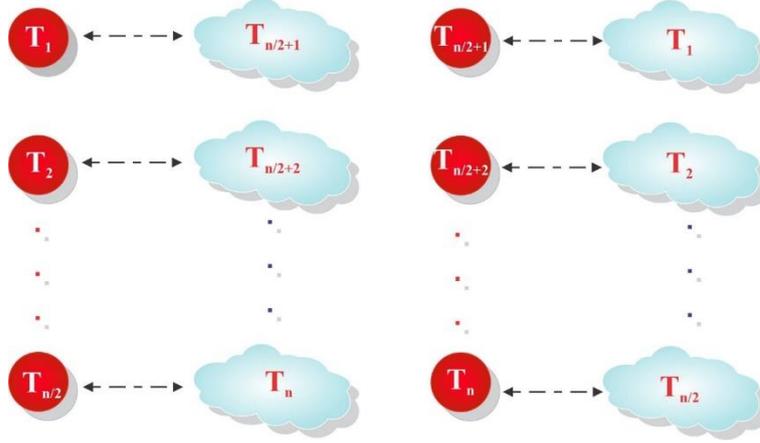


Fig. 2. Pairing cooling objects and environmental solutions

#### Steps 5 and 6. Calculating $\beta$ and $t$ parameters

Newton's law of cooling states that the heat loss rate of a lumped object is directly proportional to the difference in the temperatures between the body and its surroundings, as formulated in Eq. (6).

$$\frac{T(t) - T_{env}}{T_0 - T_{env}} = e^{-\beta t} \quad (6)$$

where  $T(t)$  is the new temperature in time  $t$  after the thermal exchange between an object with a temperature  $T_0$  and the environment with temperature  $T_{env}$ , and  $\beta$  is a constant relating to several parameters such as the object's specific heat capacity and density. As can be seen, when  $\beta$  has a higher value, the object tends to change less amount of temperature. In TEO, inspired by this phenomenon,  $\beta$  is defined such that the less the cost, the less the changes in the solution. Therefore,  $\beta$  is defined as follows

$$\beta = \frac{F_i}{F_{max}} \quad (7)$$

where,  $F_i$  and  $F_{max}$  respectively are the cost of the current object and the worst object with the highest cost among the population. Also, the time parameter  $t$  is related to the iterations and is defined as Eq. (8)

$$t_{iter} = \frac{iter}{MaxIter} \quad (8)$$

where  $iter$  denotes current iteration number and  $MaxIter$  stands for the maximum permitted iterations.

#### Step 7. Escaping local optima (i)

In order to escape from local minimums, the metaheuristics should regulate the solutions employing some stochastic mechanisms. For instance, GA uses mutation to avoid getting trapped in such solutions and to explore various chromosomes. In TEO, the current step is one of the twin mechanisms that are dedicated to this task. Here, the environmental solutions are randomly modified before temperature updating using Eq. (9).

$$\mathbf{T}_i^{env} = (1 - (c_1 + c_2 \times (1 - t)) \times \mathbf{rand}) \times \mathbf{T}_i'^{env} \quad (9)$$

where,  $\mathbf{T}_i'^{env}$  and  $\mathbf{T}_i^{env}$  are the temperatures of the object before and after the modification;  $c_1$  and  $c_2$  are internal controlling parameters; and  $\mathbf{rand}$  is a random vector containing random numbers between 0 and 1. The equation has been devised so that it can reduce the randomness as the algorithms reach the last iterations and thereby reducing the exploration and increasing the exploitation.

### Step 8. Temperature updating

According to Eq. (6), the new temperature of cooling objects can be calculated as follows.

$$\mathbf{T}_i^{old} = \mathbf{T}_i^{env} + (\mathbf{T}_i^{env} - \mathbf{T}_i^{old}) \times \exp(-\beta t) \quad (10)$$

where,  $\mathbf{T}_i^{new}$  and  $\mathbf{T}_i^{old}$  are the new updated and previous temperature of the  $i$ th object, and the  $\beta$  and  $t$  parameters were defined in Steps 5 and 6.

### Step 7. Escaping local optima (ii)

This step is the second mechanism designed to avoid trapping in local minima so that TEO can explore the global optimum solution. Herein, with the probability of  $pro$  one of the dimensions is randomly selected and its value is changed to a random number between its allowable upper and lower bounds.

### Step 8. Termination conditions

The maximum number of iterations is controlled in this step, and if the condition is met, the algorithms cease the search process and report the best solution found so far; otherwise, it resumes and returns to Step 2.

## 3.1. Multi-objective thermal exchange optimization

The steps of the multi-objective Thermal Exchange Optimization (MOTEO) version are provided in this section. Since MOTEO has structural similarities with the single-objective version, for sake of brevity just different steps are mentioned. The differences lie in the way that the objects are sorted and the way that  $\beta$  parameter is measured.

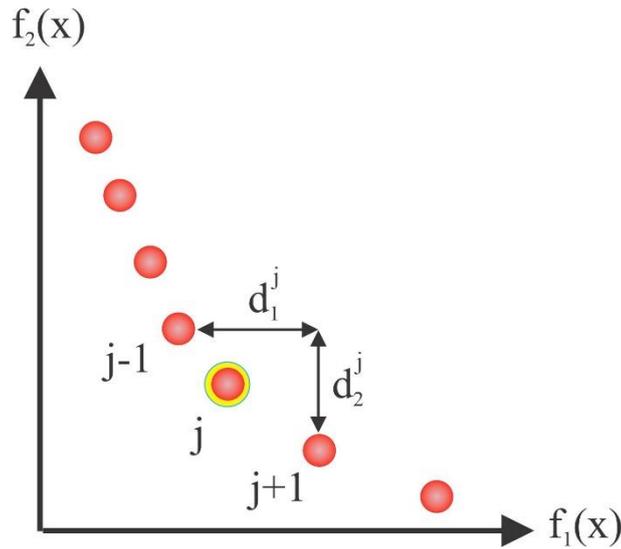
### 3.1.1 Selection mechanism

In multi-objective problems, dependent on the number of objectives functions, several cost values are attached to each object. Therefore, an idea regarding their ranking is the sorting based on their Pareto fronts. In this regard, a non-dominated sorting algorithm is executed to divide the solutions into several fronts [10]. A non-dominated sorting algorithm can be designed as follows: an index (equal to one) is assigned to each object, and the cost values are compared with those of other objects, as stated in Section 2. Any time that the object is dominated by another object, its index value is increased by one. After performing this step for all objects, the objects with the same index values are considered as a separate group. It is apparent that the objects' index values have an inverse relationship with their quality.

After performing the non-dominated sorting algorithm, the rank of each group against other groups is determined. However, their internal ranking has not been specified yet. In order to compare the objects inside a group, their crowding distance, as a secondary criterion, is calculated and compared with other groups. Crowding distance is calculated by Eq. (11) [10]

$$CD^j = \sum_{i=1}^M \frac{d_i^j}{|f_i^{max} - f_i^{min}|} \quad (11)$$

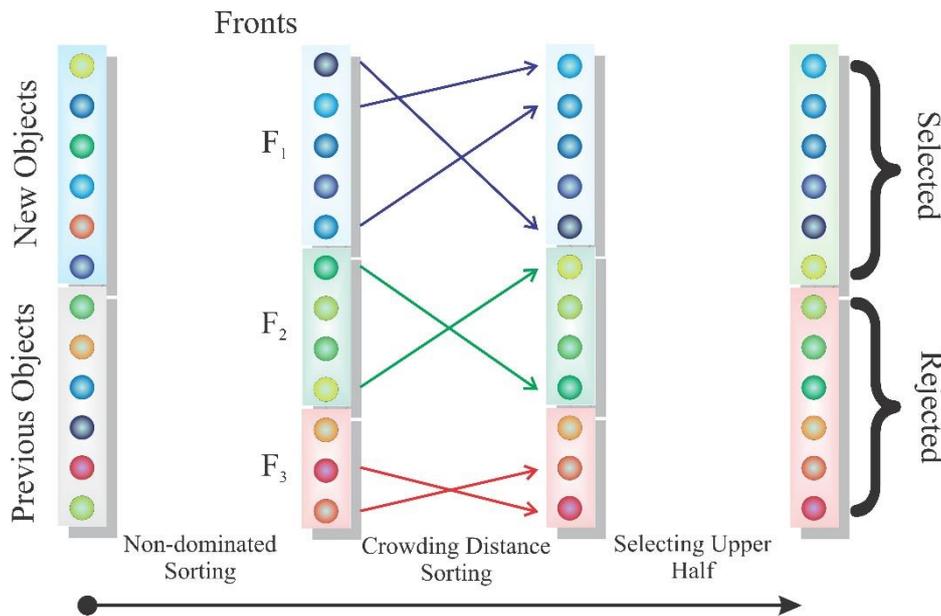
where,  $CD^j$  is the crowding distance of the  $j$ th object;  $d_i^j$  denotes the distance of two neighboring solutions of the  $j$ th object considering the  $i$ th cost function. This parameter for a bi-objective problem is illustrated in Fig. 3.  $f_i^{min}$  and  $f_i^{max}$  are respectively the cost of the best and worst objects considering the  $j$ th cost function. It is noteworthy that the crowding distance of the best and worst objects is assumed as infinity.



**Fig. 3.** An illustration of distance parameter

To preserve diversity, the crowding distance of the object located inside each group are compared, and those having higher distance attain superior ranks. To put it simply, if a solution  $T_i$  belongs to a better front than  $T_j$ , the former receives a better rank than the latter. If the front ranks are the same but  $CD^i > CD^j$ ,  $T_i$  receives a relatively better rank because of higher crowding distance.

The selection mechanism utilized in MOTEO is pictorially illustrated in Fig. 4. As seen, in each iteration the number of populations will remain constant.



**Fig. 4.** Selection mechanism of MOTEO

### 3.1.2 $\beta$ parameter

According to the ranking system explained above,  $\beta$  is redefined as follows:

$$\beta = \frac{r_i}{nPop} \quad (12)$$

where,  $r_i$  is the final rank of the solution  $T_i$  according to its Pareto front and crowding distance, and  $nPop$  stands for the number of populations.

The rest steps of the MOTEO are the same as TEO, and MOTEO can be implemented only by making the mentioned adjustments, as demonstrated in Fig. 5.

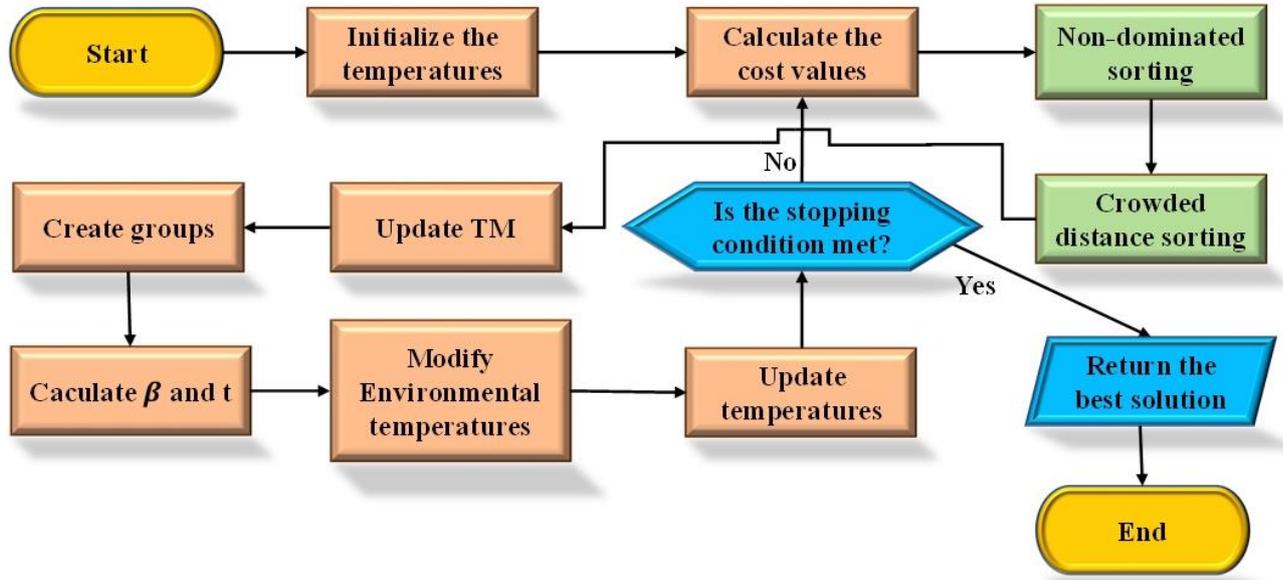


Fig. 5. Flowchart of MOTEO algorithm

## 4. Numerical Examples

Eight mathematical test functions and two well-known truss structures will analyze the efficiency of the proposed method. The structural problems and mathematical functions are employed to verify the ability of the multi-objective optimizers to handle problems with various characteristics such as non-convexity and non-linearity. The algorithm has been coded in MATLAB 2020b and the structures are analyzed with the direct stiffness method. The present work is performed with the following features on the computer: CPU 2.3 GHZ (an Intel Core i9 computer platform), Ram 16 GB 2400 MHz DDR4 on a computer with Macintosh (macOS BigSur).

### 4.1 Performance Metrics

We used four metrics in the following way to assess the results of the algorithms.[27,28]:

**Generational distance (GD):** This index calculates the distance between the true Pareto ( $PF_{true}$ ) and the obtained Pareto front ( $PF_{known}$ ). This distance is calculated by Eq. (13).

$$GD = \left( \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} dis_i^2 \right)^{\frac{1}{2}} \quad (13)$$

where  $n_{pf}$  is the number of non-dominated solutions in  $PF_{known}$  and  $dis_i$  is the Euclidean distance between the solution  $i$  in  $PF_{known}$  and its nearest solution in  $PF_{true}$ . It should be noted that a smaller value of  $GD$  implies a more desirable convergence of the algorithm.

**Spacing (S):** This measure indicates how evenly the obtained solutions are distributed along the  $PF_{known}$ . This measure is formulated as follows:

$$S = \left( \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} (d_i - \bar{d})^2 \right)^{\frac{1}{2}} \quad \text{Where} \quad \bar{d} = \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} d_i \quad (14)$$

where  $n_{pf}$  is the number of non-dominated solutions in  $PF_{known}$  and  $d_i$  is the Euclidean distance between the solution  $i$  in  $PF_{known}$  and the nearest solution in  $PF_{true}$ . The smaller value of  $GD$  shows better convergence.

**Maximum Spread (MS):** This metric evaluates how ‘‘well’’ the  $PF_{true}$  is covered by the  $PF_{known}$  through hyper-cubes formed by the extreme cost values observed in the  $PF_{true}$  and  $PF_{known}$ . It is defined as:

$$MS = \left[ \frac{1}{m} \sum_{i=1}^m \left[ \frac{\min(f_i^{max}, F_i^{max}) - \max(f_i^{min}, F_i^{min})}{F_i^{max} - F_i^{min}} \right]^2 \right]^{\frac{1}{2}} \quad (15)$$

where  $m$  is the number of objectives,  $f_i^{max}$  and  $f_i^{min}$  are the maximum and minimum of the  $i$ th objective in  $PF_{known}$ , respectively; and  $F_i^{max}$  and  $F_i^{min}$  are the maximum and minimum of the  $i$ th objective in  $PF_{true}$ , respectively. A larger value of  $MS$  indicates a better spread of solution as well.

**Inverted Generational Distance (IGD):** It is a metric for assessing the quality of approximations to the Pareto front obtained by multi-objective optimization algorithms. This measure is defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (16)$$

here,  $n$  is the number of elements in  $PF_{true}$ , and  $d_i$  is the Euclidean distance between the solution  $i$  in  $PF_{known}$  and its nearest solution in  $PF_{true}$ . The Zero value shows that all of the generated elements are in the true front of Pareto. [29]

## 4.2 Mathematical Test Function

In this section, the results of the proposed multi-objective optimization algorithm have been analyzed with eight well-known benchmark problems.[27,28]. These problems are introduced in the following and their formulations are provided in Table 1.

**Bi-objective functions:** Five examples, namely ZDT1, ZDT2, ZDT3, ZDT4, and ZDT5 are chosen as bi-objectives convex, disconnected convex, non-convex, and disconnected multi-objective optimization problems.

**Tri-objective functions:** Three examples, namely, DTLZ1, DTLZ2, and DTLZ3 are chosen as linear separable multimodal, concave scalable multimodal, and unimodal multi-objective optimization problems.

**Table 1**  
Multimodal benchmark functions with fixed-dimension

Function		d	Range
ZDT1	$F_1 = x_1, \quad F_2 = g \left( 1 - \sqrt{F_1/g} \right), \quad g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30	[0,1]
ZDT2	$F_1 = x_1, \quad F_2 = g \left( 1 - (F_1/g)^2 \right), \quad g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30	[0,1]
ZDT3	$F_1 = x_1, \quad F_2 = g \left( 1 - \sqrt{F_1/g} - F_1/g \sin(10\pi F_1) \right),$ $g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30	[0,1]
ZDT4	$F_1 = x_1, \quad F_2 = g \left( 1 - \sqrt{F_1/g} \right),$ $g = 1 + 10(d-1) + \sum_{i=2}^d (x_i^2 - 10\cos(4\pi x_i))$	10	$x_1 \in [0,1]$ $x_i \in [-5,5], i = 2, \dots, d$
ZDT6	$F_1 = 1 - \exp(-4x_1) \sin^6(6\pi x_1), \quad F_2 = g \left( 1 - (F_1/g)^2 \right),$ $g = 1 + 9 \left( \frac{\sum_{i=2}^d x_i}{d-1} \right)^{0.25}$	10	[0,1]
DTLZ1	$F_1 = \frac{1}{2} x_1 x_2 (1 + g), F_2 = \frac{1}{2} x_1 (1 - x_2) (1 + g),$ $F_3 = \frac{1}{2} (1 - x_1) (1 + g), g$ $= 100 \left( 10 + \sum_{i=3}^d (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$	12	[0,1]

$$\begin{aligned}
& F_1 = (1 + g) \cos\left(x_1 \left(\frac{\pi}{2}\right)\right) \cos\left(x_2 \left(\frac{\pi}{2}\right)\right), F_2 \\
\text{DTLZ2} \quad & = (1 + g) \cos\left(x_1 \left(\frac{\pi}{2}\right)\right) \sin\left(x_2 \left(\frac{\pi}{2}\right)\right), F_3 \quad 12 \quad [0,1] \\
& = (1 + g) \sin\left(x_1 \left(\frac{\pi}{2}\right)\right), g = \sum_{i=3}^d (x_i - 0.5)^2 \\
& F_1 = (1 + g) \cos\left(x_1^\pi \left(\frac{\pi}{2}\right)\right) \cos\left(x_2^\pi \left(\frac{\pi}{2}\right)\right), F_2 \\
\text{DTLZ4} \quad & = (1 + g) \cos\left(x_1^\pi \left(\frac{\pi}{2}\right)\right) \sin\left(x_2^\pi \left(\frac{\pi}{2}\right)\right), F_3 \quad 12 \quad [0,1] \\
& = (1 + g) \sin\left(x_1^\pi \left(\frac{\pi}{2}\right)\right), g = \sum_{i=3}^d (x_i - 0.5)^2
\end{aligned}$$

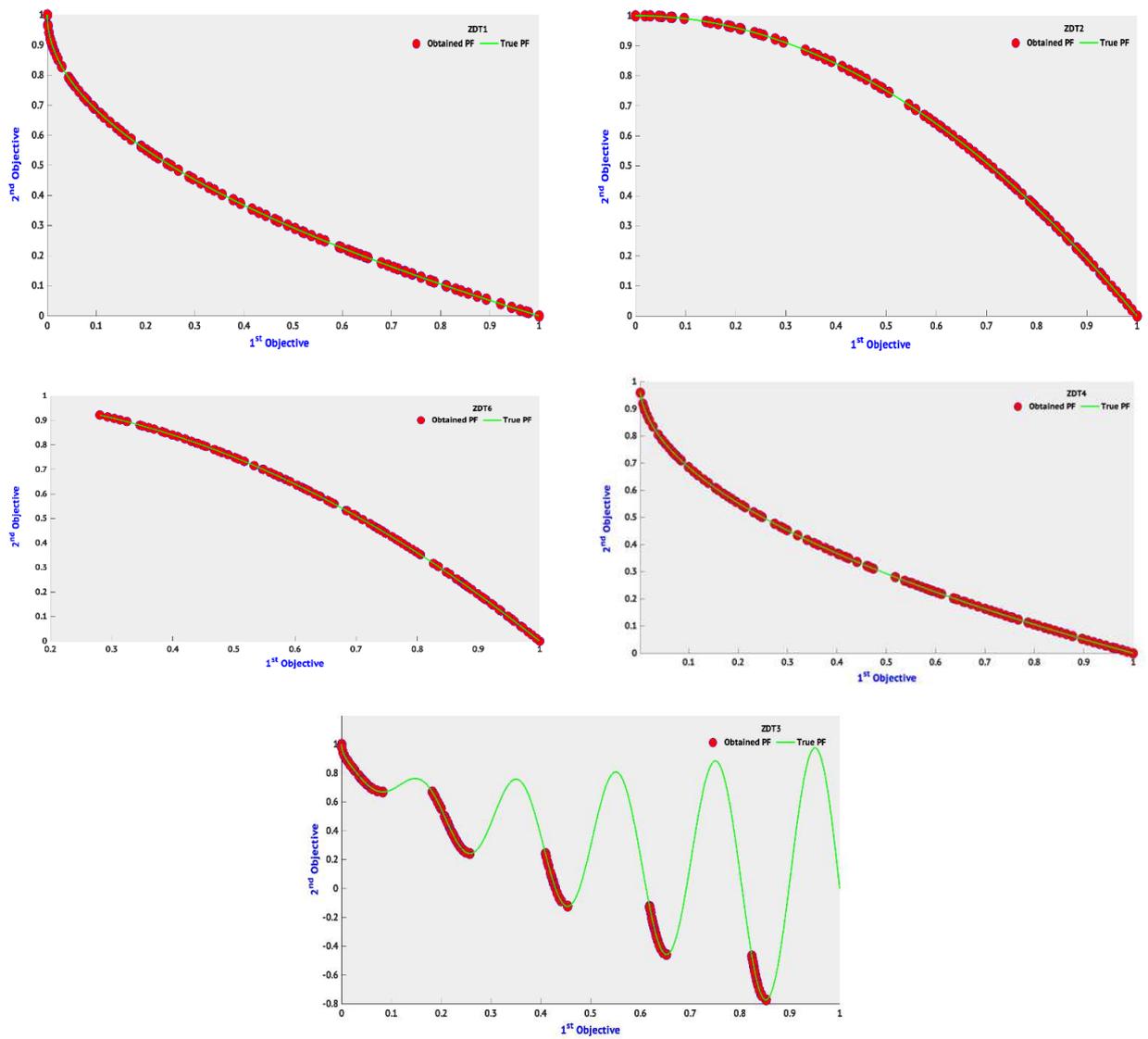
The results of the presented method are compared with the three famous multi-objective optimization algorithms—Strength Pareto Evolutionary Algorithm two (SPEA2) [28], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10], and Multi-Objective Particle Swarm Optimization (MOPSO) [11] algorithms. The internal parameters considered for these algorithms are shown in Table 2. In this study, the number of function evaluations for all algorithms is set to 300000 and the results are reported based on 30 independent runs.

The mean and standard deviation of GD, S, MS, and IGD metrics for bi-objective test functions along with tri-objective problems are presented in Tables 3-10. It should be mentioned that the boldfaced values indicate the best results obtained among the compared algorithms. As seen, MOTEO has the best performance in terms of GD, S, MS and IGD metrics. Therefore, it can be said that, compared to other algorithms, MOTEO has the shortest distance from true Pareto fronts, provides a uniform distribution of solutions, and covers a wider range of cost values. In terms of GD, S, and IGD, the NSGA-II is only second to MOTEO. NSGA II. The obtained best Pareto fronts of MOTEO for five bi-objective and three tri-objective problems are presented in Figs. 6 and 7, respectively. The figures prove that MOTEO is able to find a Pareto optimal set near the global Pareto front and to preserve solution diversities across space, as well. All in all, it can be inferred that most results of MOTEO are superior to those of other algorithms.

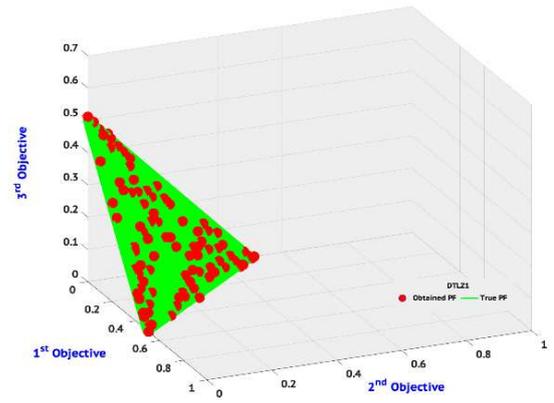
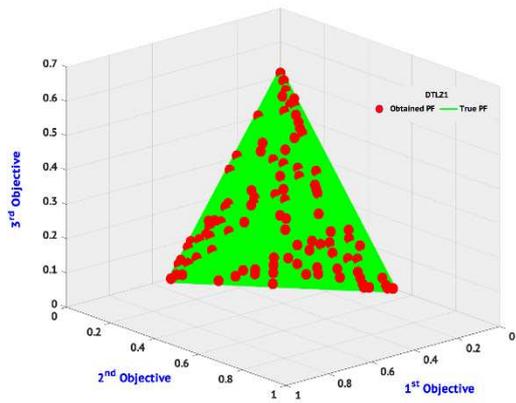
**Table 2**

Parameter setting of algorithms

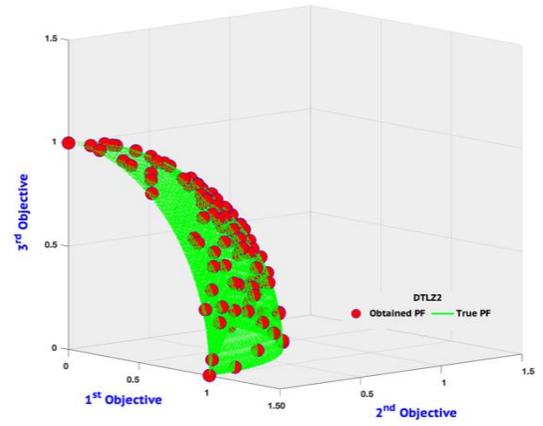
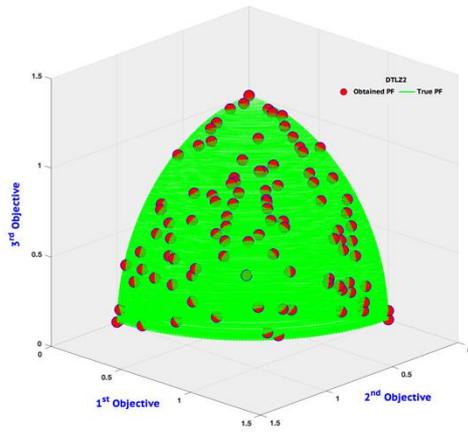
Parameters	Algorithm			
	MOPSO	NSGA-II	SPEA2	MOTEO
Crossover Probability ( $P_c$ )	-	0.9	0.9	-
Mutation Probability ( $P_w$ , or pro)	0.5	1/d	1/d	1/d
Population Size ( $N_{pop}$ )	100	100	100	100
Archive Size ( $N_{rep}$ , or TM)	100	100	100	100
Number of Adaptive Grid ( $N_{grid}$ )	30	-	-	-
Personal Learning Coefficient ( $C_1$ )	2	-	-	1
Global Learning Coefficient ( $C_2$ )	2	-	-	2
Inertia weight ( $w$ )	0.4	-	-	-



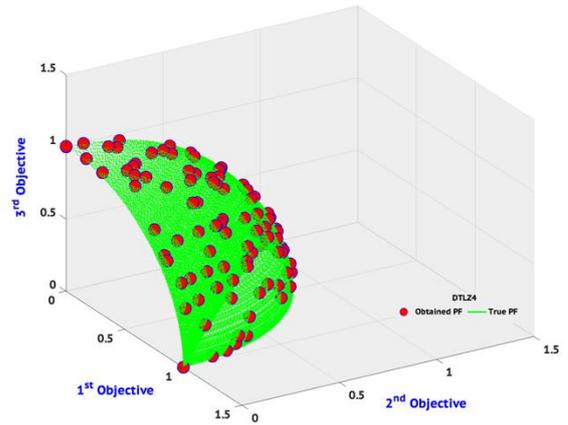
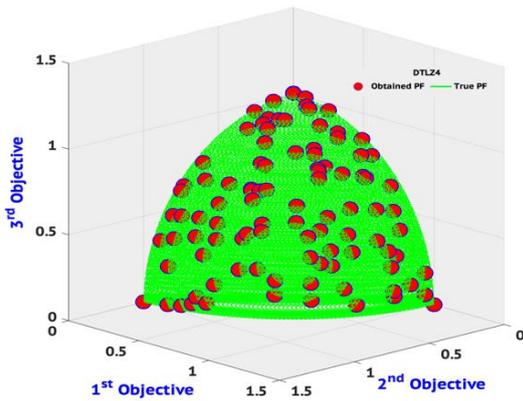
**Fig. 6.** True and resulted Pareto front by MOTEO on bi-objective test functions



(a)



(c)



(e)

**Fig. 7.** True and resulted Pareto front by MOTEQ on tri-objective test functions: (a) DTLZ1, (b) DTLZ2, (c) DTLZ3.

**Table 3**

The mean (and standard deviation) results of benchmark functions GD performance metric

Functions	Methods			
	<i>MOPSO</i>	<i>NSGA-II</i>	<i>SPEA2</i>	<i>MOTEO</i>
<b>ZDT1</b>	0.003603 (0.000478826)	0.0000862 ( <b>1.7992E-05</b> )	0.2523318 (0.0246142)	<b>0.0000794</b> (1.9398E-05)
<b>ZDT2</b>	0.0025378 (0.00098416)	<b>0.0000466</b> (2.1909E-06)	0.424548 (0.01670509)	0.000048 ( <b>1.41425E-06</b> )
<b>ZDT3</b>	0.0049528 (0.00177014)	0.0003064 (3.5267E-05)	0.229655 (0.02308383)	<b>0.0003032</b> ( <b>1.9305E-05</b> )
<b>ZDT4</b>	0.094439 (0.209639537)	0.0013652 (0.00038914)	68.9247954 (1.896318)	<b>0.0000962</b> ( <b>4.9857E-05</b> )
<b>ZDT6</b>	0.1506312 (0.33539696)	0.0000356 (2.3022E-06)	2.6209262 (0.0434244)	<b>0.0000338</b> ( <b>2.2804E-06</b> )
<b>DTLZ1</b>	137.015245 (59.061218)	0.2663104 (0.34871408)	226.106343 (24.68157)	<b>0.081109</b> ( <b>0.02861018</b> )
<b>DTLZ2</b>	0.250506 (0.032588467)	0.0018304 ( <b>0.00060063</b> )	0.2694456 (0.0299048)	<b>0.0117166</b> (0.0189848)
<b>DTLZ4</b>	0.2889046 (0.04331457)	<b>0.0018784</b> ( <b>0.00087037</b> )	0.290684 (0.02814046)	0.0044392 (0.0032198)

**Table 4**

The mean (and standard deviation) results of benchmark functions S performance metric

Functions	Methods			
	<i>MOPSO</i>	<i>NSGA-II</i>	<i>SPEA2</i>	<i>MOTEO</i>
<b>ZDT1</b>	0.0015786 (0.00017166)	0.00007 (2.018776E-05)	0.198854 (0.023103046)	<b>0.0000642</b> ( <b>1.9563E-05</b> )
<b>ZDT2</b>	0.0009258 (0.00028773)	<b>0.0000266</b> (2.3022E-06)	0.3322836 (0.01689598)	0.0000272 ( <b>8.3666E-07</b> )
<b>ZDT3</b>	0.003939 (0.001538756)	0.00021 (2.502765E-05)	0.2025156 (0.01734266)	<b>0.0001992</b> ( <b>1.3498E-05</b> )
<b>ZDT4</b>	0.0304846 (0.06694879)	0.0003332 (9.9384E-05)	46.0872712 (20.917399)	<b>0.000081</b> ( <b>5.40696E-05</b> )
<b>ZDT6</b>	0.0971702 (0.21661905)	0.0000196 (1.6733E-06)	2.1349914 (0.04490765)	<b>0.0000192</b> ( <b>1.4832E-06</b> )
<b>DTLZ1</b>	32.5875322 (22.990261)	0.0483346 (0.05513448)	166.487063 (17.720155)	<b>0.0183194</b> ( <b>0.00701707</b> )
<b>DTLZ2</b>	0.1309528 (0.02041799)	<b>0.0015968</b> ( <b>0.00063377</b> )	0.2082058 (0.02088225)	0.0114262 (0.01883207)
<b>DTLZ4</b>	0.1508502 (0.01591233)	<b>0.0018</b> ( <b>0.00080594587</b> )	0.2256692 (0.02275843)	0.0039606 (0.00339327)

**Table 5**

The mean (and standard deviation) results of benchmark functions MS performance metric

Functions	Methods			
	<i>MOPSO</i>	<i>NSGA-II</i>	<i>SPEA2</i>	<i>MOTEO</i>
<b>ZDT1</b>	0.998793 (0.000393155)	<b>1 (0)</b>	0.9675234 (0.01647403)	<b>1 (0)</b>
<b>ZDT2</b>	0.9986266 (0.00077024)	<b>1 (0)</b>	0.9184778 (0.03034199)	<b>1 (0)</b>
<b>ZDT3</b>	<b>0.980348</b> (0.017968845)	0.9288054 (0.00012914)	0.9470436 (0.02745689)	0.9289402 ( <b>0.0001069</b> )
<b>ZDT4</b>	0.9629876 (0.08271082)	0.9992984 (0.00020888)	<b>2.3572554</b> (0.40442515)	1 (0)
<b>ZDT6</b>	0.998267 (0.003427475)	<b>1 (0)</b>	0.9743385 (0.01979964)	<b>1 (0)</b>
<b>DTLZ1</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>
<b>DTLZ2</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>
<b>DTLZ4</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>	<b>1 (0)</b>

**Table 6**

The mean (and standard deviation) results of benchmark functions IGD performance metric

Functions	Methods			
	<i>MOPSO</i>	<i>NSGA-II</i>	<i>SPEA2</i>	<i>MOTEO</i>
<b>ZDT1</b>	0.0003604 (4.8014E-05)	0.0000084 (1.5166E-06)	0.0252332 (0.0024613)	<b>0.000008</b> (2.1213E-06)
<b>ZDT2</b>	0.0002538 (9.8538E-05)	<b>0.0000048</b> (4.4721E-07)	0.0424546 (0.0016705)	0.000005 (0)
<b>ZDT3</b>	0.0004954 (0.00017735)	0.0000308 (3.3466E-06)	0.0229656 (0.0023086)	<b>0.0000302</b> ( <b>1.9235E-06</b> )

<b>ZDT4</b>	0.009444 (0.020963737)	0.0001366 (3.9125E-05)	6.8924796 (0.1896319)	<b>0.0000096 (4.8278E-06)</b>
<b>ZDT6</b>	0.0150632 (0.03353982)	0.0000038 ( <b>4.4721E-07</b> )	0.2620924 (0.0043424)	<b>0.0000034 (5.4772E-07)</b>
<b>DTLZ1</b>	13.7015244 (5.9061217)	0.0266312 (0.03487119)	18.6106342 (9.277712)	<b>0.0081108 (0.00286131)</b>
<b>DTLZ2</b>	0.0250506 (0.00325876)	<b>0.000183 (5.99928E-05)</b>	0.0269446 (0.0029906)	0.0011716 (0.00189824)
<b>DTLZ4</b>	0.0288904 (0.00433123)	<b>0.0002196 (8.8022E-05)</b>	0.0290684 (0.0028139)	0.0004438 (0.00032185)

### 4.3 Truss Structure Examples

In this section, two truss structures with conflicting objective functions are tested by NSGA-II, MOPSO, SPEA2, and MOTEO to evaluate the performance of the proposed algorithm in solving real-world problems. All these examples are run 20 times by each algorithm. The number of iteration and all parameter settings are the same as Section 4.2. The minimizations of weight and displacement are considered as the objectives in these structural multi-objective problems as follows:

$$\begin{aligned}
 & \text{Find} && \{X\} = [x_1, x_2, \dots, x_{ng}] \\
 & \text{Minimize} && [W(\{x\}), U(\{x\})] \\
 & && \text{where } W(\{x\}) = \sum_{i=1}^{nm} \gamma_i \cdot A_i \cdot L_i(x) \\
 & \text{Subjected to} && \left\{ \begin{array}{l} g_j(\{X\}) \leq 0 \quad j = 1, 2, 3, \dots, ns \\ A_{\min} \leq A_i \leq A_{\max} \\ x_{\min} \leq x_i \leq x_{\max} \end{array} \right. \quad (17)
 \end{aligned}$$

where  $\{X\}$  is the vector of design variables,  $ng$  is the number of design variables, the structure weight is described as  $W(\{x\})$ , the displacement is described as  $U(\{x\})$ , and the number of constraints is denoted as  $ns$ . Here, the  $i$ th member's material density is shown by  $\gamma_i$ ,  $L_i$  is its length,  $A_i$  is the member's cross-sectional area, and  $g_j(\{X\})$  denotes design constraints. Also,  $x_{\min}$  and  $x_{\max}$  are the lower and upper bounds of the design variable vector, respectively. As the penalty function is simpler and easier to carry out, the limitations expressed in the following are handled:

$$f_{penalty}(X) = (1 + \varepsilon_1 \cdot \nu)^{\varepsilon_2} \quad \nu = \sum_{i=1}^n \max[0, \nu_i] \quad (18)$$

where  $\nu$  is the sum of the violation ratios of constraints and constants  $\varepsilon_1$  and  $\varepsilon_2$  are selected to enhance the exploration and exploitation rate of the search. Here,  $\varepsilon_1$  is set to 1, and  $\varepsilon_2$  is set to 1.5 at the first iteration of the search process and is gradually increased to 3 [30].

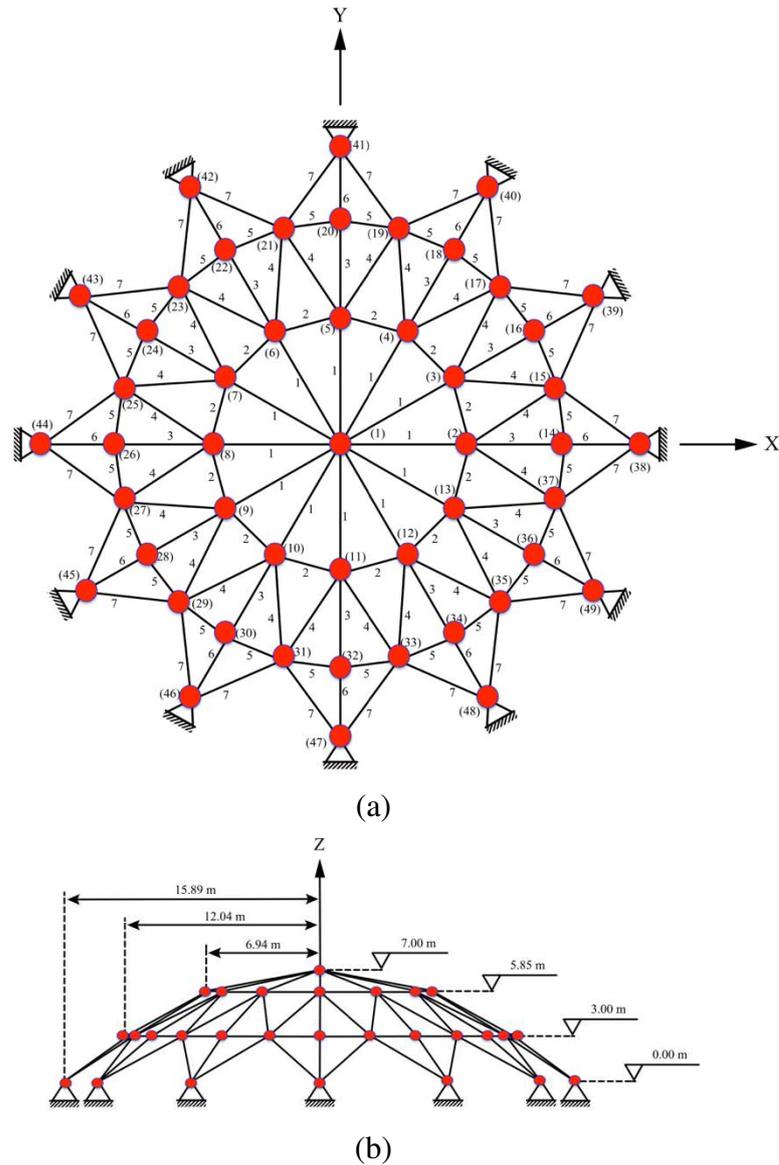
The allowable compressive and tensile stress constraints according to the AISC [31] is considered as:

$$\left\{ \begin{array}{l} \sigma_i = 0.6F_y \\ \sigma_i = \begin{cases} \left[ \left(1 - \frac{\lambda_i^2}{2C_c^2}\right) F_y \right] / \left[ \frac{5}{3} + \frac{8\lambda_i}{3C_c} - \frac{\lambda_i^3}{8C_c^3} \right] & \text{for } \lambda_i > C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \leq C_c \end{cases} \end{array} \right. \quad \begin{array}{l} \sigma_i \geq 0 \\ \sigma_i < 0 \end{array} \quad (19)$$

where the modulus of elasticity is described by  $E$ , the slenderness ratio is shown by  $\lambda_i$ ,  $C_c$  is defined as  $\sqrt{2\pi^2 E / F_y}$ , and the yield stress of steel is  $F_y$ .

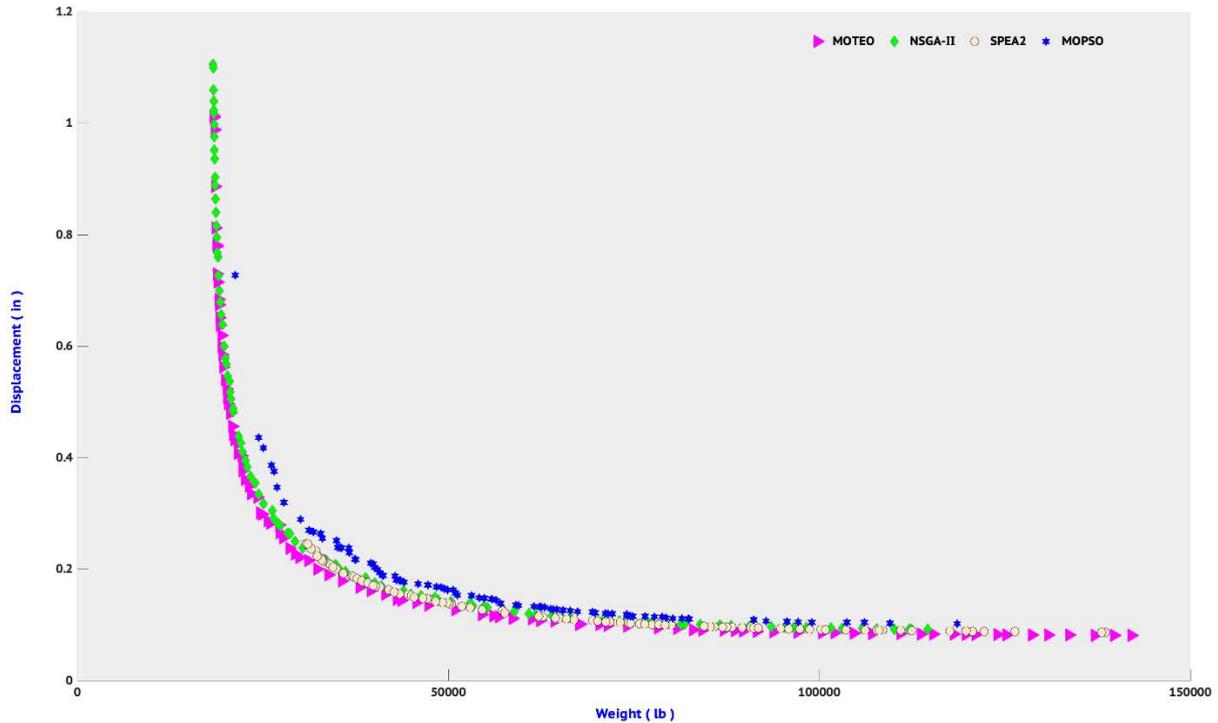
#### 4.3.1 A 120-bar Dome Truss

The optimization of 120-bar dome truss has been studied by different researchers to minimize the weight as a single objective [32,33]. In this paper, the total structural weight as well as the maximum displacement in all directions on all the nodes are considered as two optimization objectives. The density of the material and the modulus of elasticity are respectively  $0.288 \text{ lb/in}^3$  ( $7971.810 \text{ kg/m}^3$ ) and  $30.450 \text{ ksi}$  ( $2.1 \times 10^5 \text{ MPa}$ ), and the yield stress is taken as  $58 \text{ ksi}$  ( $400 \text{ MPa}$ ). This structure is drawn in Fig. 8. The dome is subjected to the vertical loading through all free nodes as following:  $-13.49 \text{ kips}$  ( $-60 \text{ kN}$ ) at node 1,  $-6.744 \text{ kips}$  ( $-30 \text{ kN}$ ) at nodes 2 to 14, and  $-2.248 \text{ kips}$  ( $-10 \text{ kN}$ ) at the remaining nodes. The truss members are categorized into seven independent size variables, and the cross-sectional areas are between  $0.775 \text{ in}^2$  ( $5 \text{ cm}^2$ ) and  $20 \text{ in}^2$  ( $129.032 \text{ cm}^2$ ).



**Fig. 8.** Scheme of the 120-bar dome truss structure: (a) top view, (b) side view.

As the true front of Pareto is not correctly identified, here the performance measurement used in the previous section cannot be applied. However, the best two extreme-point values in Table 7 are compared in order to provide a quantitative assessment of the performance of the optimizers. The extent of the obtained non-dominated front should be maximized, i.e., for each objective, a wide range of values should be covered by the non-dominated solutions [28]. As provided in the table, MOTEO is able to find solutions with lower displacement and weight in the extreme points. The best Pareto fronts of NSGA-II, SPEA2, MOPSO, and MOTEO obtained after 20 runs are plotted and compared in Fig. 9. It can be seen that the obtained set of solutions for MOTEO is distributed uniformly. Additionally, the obtained Pareto front by MOTEO generally dominated those obtained by MOPSO, SPEA2 and NSGA-II. The majority of the Pareto front can also be covered in a suitable convergence.



**Fig. 9.** Pareto front obtained by MOPSO, SPEA2, NSGA-II and. MOTEO algorithms on 120-bar dome truss structure

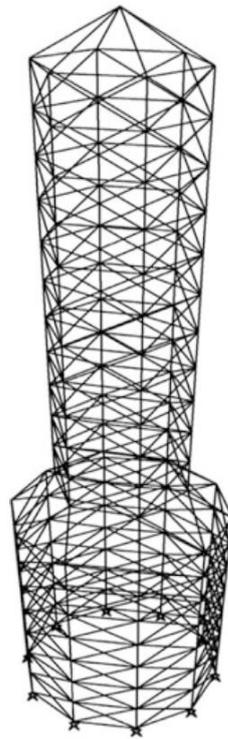
**Table 7**

Comparison of the results for 120-bar truss dome

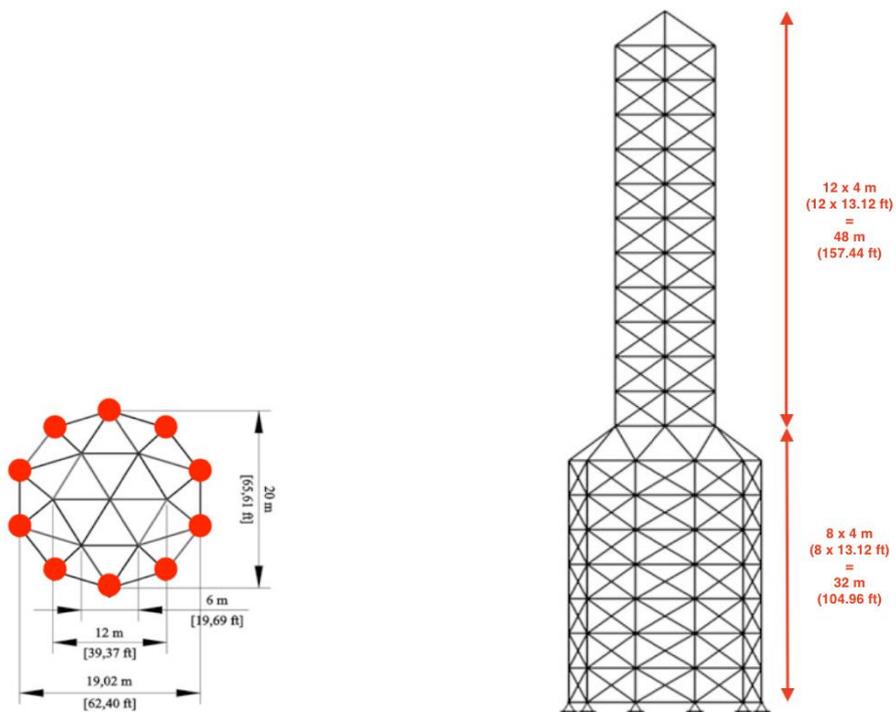
Methods	Extreme point one [ <i>in, lb</i> ]	Extreme point two [ <i>in, lb</i> ]
MOPSO	[0.086990, 116586.36]	[0.71648, 19272.85]
NSGA-II	[0.085341, 122577.07]	[1.15543, 18362.06]
SPEA2	[0.084981, 140077.12]	[0.24864, 26102.90]
MOTEO	[0.081152, 145107.69]	[1.01325, 18356.51]

### 4.3.2 A 582-bar Tower Truss

The single- and multi-objective optimization of the 582-bar tower truss structure, shown in Fig 10, has been studied in the literature [34]. Here, this structure with continuous sizing variables as a multi-objective optimization problem is investigated. Two objectives, namely the total used steel volume and the maximum displacement in all degrees of freedom on all nodes are optimized simultaneously. The members are symmetrically divided into 32 groups. The truss is subjected to the lateral loads in both x- and y-directions and a vertical load in the z-direction at all free nodes as following: 1.12 kips (5 kN), 1.12 kips (5 kN) and  $-6.744$  kips ( $-30$  kN), respectively. The lower and upper bounds of cross-sectional areas are taken as  $3.1$  in<sup>2</sup> (20 cm<sup>2</sup>) and 155 in<sup>2</sup> (1000 cm<sup>2</sup>). The yield stress of steel is 36 ksi (253.1 MPa) with the elasticity modulus of 29000 ksi (203893.6 MPa). According to ASD-AISC, the slenderness ratio for tension and compression members are respectively limited to 300 and 200 [31].



(a)

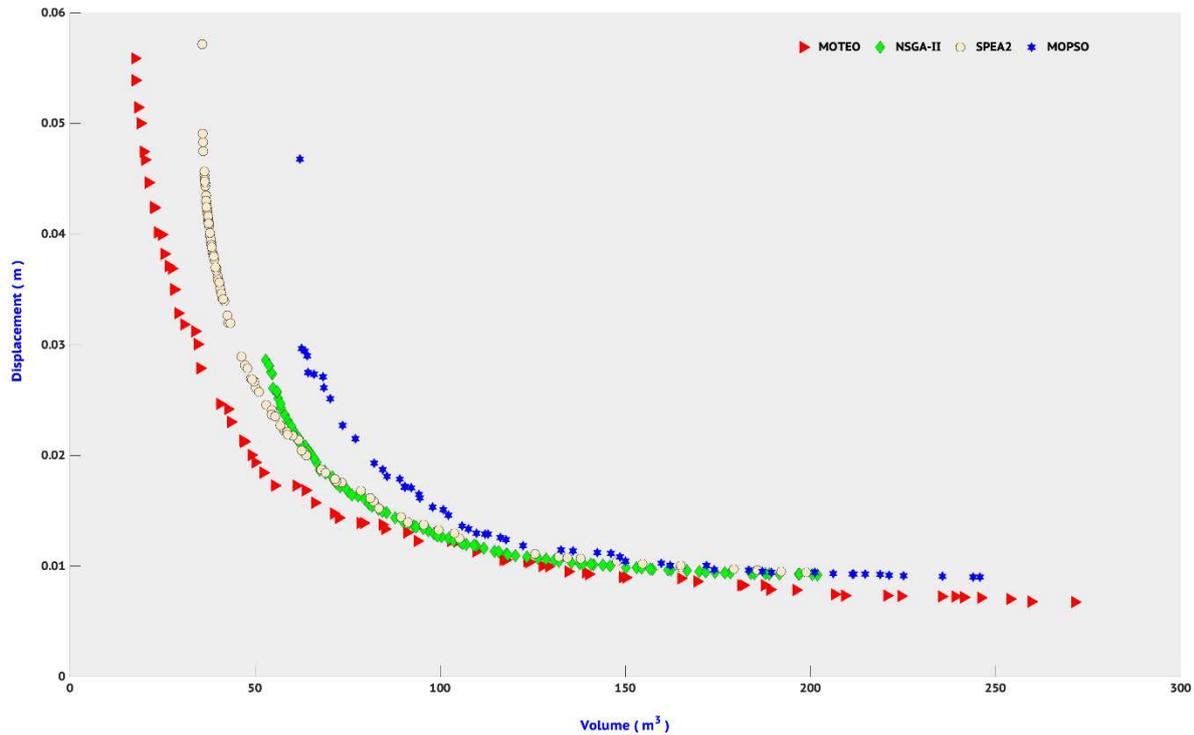


(b)

**Fig. 10.** Scheme of the 582-bar tower truss structure: (a) perspective view (b) dimensions.

Fig. 11 depicts the Pareto fronts obtained by NSGA-II, MOPSO, SPEA2, and MOTEO algorithms. As seen, the Pareto front obtained by MOTEO generally dominates the rest of the algorithms. Like the preceding example, the true Pareto front is not known for this example, and the performance metrics

are not applicable. Table 8 presents the two best extreme point values obtained by the algorithms proving that MOTEO covered a larger portion of the Pareto front. As can be seen, the minimum of both displacement and volume is obtained by MOTEO. These results show the proper convergence and sparsity efficiency of the MOTEO in optimizing problems with multiple conflicting functions.



**Fig. 11.** Pareto fronts obtained by MOPSO, SPEA2, NSGA-II and. MOTEO algorithms on 582-bar tower truss structure

**Table 8**  
Comparison of the results for 582-bar tower truss

Methods	Extreme point one [ $m, m^3$ ]	Extreme point two [ $m, m^3$ ]
MOPSO	[0.008910, 245.7983]	[0.046250, 62.1554]
NSGA-II	[0.009129, 201.9052]	[0.028608, 52.9431]
SPEA2	[0.009425, 198.9654]	[0.057544, 35.7754]
MOTEO	[0.008149, 271.3608]	[0.0554983, 17.5433]

## 5. Conclusion

In this paper, a multi-objective version of the recently-developed Thermal Exchange Optimization is proposed. The differences between the introduced Multi-objective Thermal Exchange Optimization (MOTEO) and the basic single-objective version mainly rely upon the definition of  $\beta$  parameter and ranking scheme. The rank of solutions is determined using a non-dominated sorting mechanism and crowded distance between obtained solutions, as illustrated in Fig. 4. According to the new ranking system,  $\beta$  parameter is calculated for each solution, and the position of solutions is updated accordingly.

In order to investigate the proposed MOTEO, firstly eight bi- and tri-objective mathematical problems have been selected and MOTEO's performances are compared to three well-established algorithms, in terms of four accuracy and Pareto quality metrics. The results reveal the efficiency and competitive performance of the MOTEO considering most of the measures and criteria. Also, to examine MOTEO in the optimization of real-world engineering problems, two structural examples with higher dimensions have been taken into account. It was observed that MOTEO is capable of solving multi-objective structural problems effectively and demonstrates a competitive and satisfactory performance compared to other well-known algorithms.

## **Compliance with ethical standards**

### **Conflict of interest**

The authors declare that there is no conflict of interest regarding the publication of this paper. The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

### **Ethical approval**

This article does not contain any studies with human participants or animals performed by any of the authors.

### **Funding details**

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

### **Informed consent**

This article has not been published previously and it is not under consideration for publication elsewhere. This publication is approved by the author and if accepted, it will not be published elsewhere in the same form, in English or in any other language, without the written consent of the copyright-holder.

### **Authors' contribution**

Nima, Siamak and Armin discussed and constructed the measures, found their applications and wrote the paper together.

## References

- [1] Talbi, E.-G.: Metaheuristics: from design to implementation, vol. 74. John Wiley & Sons, 2009.
- [2] Kaveh, A., Eslamlou, A.D.: Metaheuristic Optimization Algorithms in Civil Engineering: New Applications. Springer, 2020.
- [3] A. Kaveh, N. Khodadadi, B.F. Azar, S. Talatahari, Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections, Engineering with Computers. 2020, 1–21.
- [4] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, 1994, 1942-1948.
- [5] Salcedo-Sanz, S.: Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. Physics Reports 2016, 655, 1-70.
- [6] Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Computational Intelligence Magazine 2016, 1(4), 28-39.
- [7] Kaveh, A., Talatahari, S., Khodadadi, N.: Stochastic paint optimizer: theory and application in civil engineering. Engineering with Computers, 2020.
- [8] Kaveh, A., Dadras Eslamlou, A., Khodadadi, N.: Dynamic Water strider algorithm for optimal design of skeletal structures. Periodica Polytechnica Civil Engineering, 2020,64(3), 904-916.

- [9] Holland, J.H.: Genetic Algorithms. *Scientific American* 1992, 267(1), 66-73.
- [10] 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.
- [11] Coello, C.A.C., Lechuga, M.S.: MOPSO: a proposal for multiple objective particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2002, 1052, 1051-1056.
- [12] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1997, 1(1), 67-82.
- [13] Kaveh, A., Mahdavi, V.R.: Multi-objective colliding bodies optimization algorithm for design of trusses. *Journal of Computational Design and Engineering* 2019, 6(1), 49-59.
- [14] Luh, G.-C., Chueh, C.-H.: Multi-objective optimal design of truss structure with immune algorithm. *Computers & Structures* 2004, 82(11), 829-844.
- [15] Tejani, G.G., Pholdee, N., Bureerat, S., Prayogo, D., Gandomi, A.H.: Structural optimization using multi-objective modified adaptive symbiotic organisms search. *Expert Systems with Applications* 2019, 125, 425-441.
- [16] Nan, B., Bai, Y., Wu, Y.: Multi-Objective Optimization of Spatially Truss Structures Based on Node Movement. *Applied Sciences* 2020, 10(6), 1964.
- [17] Kumar, S., Tejani, G.G., Pholdee, N., Bureerat, S.: Multi-objective modified heat transfer search for truss optimization. *Engineering with Computers* 2020.
- [18] 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.
- [19] Kaveh, A., Ilchi Ghazaan, M.: A new VPS-based algorithm for multi-objective optimization problems. *Engineering with Computers* 2020, 36(3), 1029-1040.
- [20] Yahya, M., Saka, M.P.: Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights. *Automation in Construction* 2014, 38, 14-29.
- [21] Bakhshinezhad, S., Mohebbi, M.: Multi-objective optimal design of semi-active fluid viscous dampers for nonlinear structures using NSGA-II. *Structures* 2020, 24, 678-689.
- [22] Etedali, S., Hasankhoie, K., Sohrabi, M.R.: Optimal design of pure-friction isolators with and without restoring device: A multi-objective cuckoo search-based approach for seismic-excited structures. *Structures* 2020, 25, 708-719.
- [23] Kaveh, A., Dadras, A., Bakhshpoori, T.: Improved thermal exchange optimization algorithm for optimal design of skeletal structures. *Smart Structures and Systems* 2018, 21(3), 263-278.
- [24] Kaveh, A., Dadras, A.: A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software* 2017, 110, 69-84.
- [25] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A Gravitational Search Algorithm. *Information Sciences* 2009, 179(13), 2232-2248.
- [26] Erol, O.K., Eksin, I.: A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software* 2006, 37(2), 106-111.

- [27] Deb, K.: An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 2000, 186(2), 311-338.
- [28] Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 2000, 8(2), 173-195.
- [29] Sierra, M.R., Coello Coello, C.A.: Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and  $\epsilon$ -Dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2005, 505-519.
- [30] Kaveh, A., Talatahari, S., Khodadadi, N.: Hybrid Invasive Weed Optimization-Shuffled Frog-Leaping Algorithm for Optimal Design of Truss Structures. *Iranian Journal of Science and Technology, Transactions of Civil Engineering* 2020, 44(2), 405-420.
- [31] AISC, A.: *Manual of steel construction—allowable stress design*. American Institute of Steel Construction (AISC), Chicago 1989.
32. Kaveh, A., Laknejadi, K.: A new multi-swarm multi-objective optimization method for structural design. *Advances in Engineering Software* 2013, 58, 54-69.
- [33] Soh, C.K., Yang, J.: Fuzzy Controlled Genetic Algorithm Search for Shape Optimization. *Journal of Computing in Civil Engineering* 1996, 10(2), 143-150.
- [34] Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P.: Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computers & Structures* 2009, 87(5), 284-302.