

# Discrete Optimum Design of Planar Steel Curved Roof and Pitched Roof Portal Frames Using Metaheuristic Algorithms

Ali Kaveh<sup>1</sup>, Mohammad Iman Karimi Dastjerdi<sup>1</sup>, Ataollah Zaerreza<sup>1</sup>, Seyed Milad Hosseini<sup>1</sup>

<sup>1</sup> School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Iran

\* Corresponding author, e-mail: [alikaveh@iust.ac.ir](mailto:alikaveh@iust.ac.ir)

Received: 25 April 2021, Accepted: 25 June 2021, Published online: 14 July 2021

## Abstract

Portal frames are single-story frame buildings including columns and rafters, and their rafters can be either curved or pitched. These are used widely in the construction of industrial buildings, warehouses, gyms, fire stations, agricultural buildings, hangars, etc. The construction cost of these frames considerably depends on their weight. In the present research, the discrete optimum design of two types of portal frames including planar steel Curved Roof Frame (CRF) and Pitched Roof Frame (PRF) with tapered I-section members are presented. The optimal design aims to minimize the weight of these frame structures while satisfying some design constraints based on the requirements of ANSI/AISC 360-16 and ASCE 7-10. Four population-based metaheuristic optimization algorithms are applied to the optimal design of these frames. These algorithms consist of Teaching-Learning-Based Optimization (TLBO), Enhanced Colliding Bodies Optimization (ECBO), Shuffled Shepherd Optimization Algorithm (SSOA), and Water Strider Algorithm (WSA). Two main objectives are followed in this paper. The first one deals with comparing the optimized weight of the CRF and PRF structures with the same dimensions for height and span in two different span lengths (16.0 m and 32.0 m), and the second one is related to comparing the performance of the considered metaheuristics in the optimum design of these portal frames. The obtained results reveal that CRF is more economical than PRF in the fair comparison. Moreover, comparing the results acquired by SSOA with those of other considered metaheuristics reveals that SSOA has better performance for the optimal design of these portal frames.

## Keywords

curved roof frames, pitched roof frames, tapered members, discrete optimization, metaheuristic algorithms

## 1 Introduction

Portal frames are a type of structural frame in which its elements comprise columns and curved or pitched rafters. In these frames, the connections between columns and rafters are considered as moment-resisting. However, the connections between columns and base plates can be either pin-jointed or fixed-jointed, and the pin-based are more economical than the fixed-based [1]. Portal frames are widely used in the construction of industrial buildings, warehouses, gyms, fire stations, agricultural buildings, hangars, etc. These can be constructed in different forms. In this study, two types of them namely Pitched Roof Frame (PRF) and Curved Roof Frame (CRF) are investigated. Members of these frames can be either prismatic or non-prismatic (tapered). The prismatic members have a constant cross-section, while tapered members have a variable-cross section along their entire length. From the economic aspect, incorporating tapered members into the

portal frames leads to non-uniform distribution of bending moments and decreases the use of material over a wide range of spans [2].

Structural optimization, such as topology optimization [3–5], is one of the most studied research topics in engineering [6], and it has been attracted many research items. Since there is a limited number of existing resources in engineering design, designers attempt to find a solution fulfilling all requirements and have the lowest possible cost. Obtaining this optimum solution is a difficult task. It cannot be found with an exact method within a reasonable amount of time. Using approximate algorithms such as metaheuristic algorithms is the main alternative to find this solution [7]. Focusing on a relatively simple concept and easy implementation, not requiring the gradient information, and bypassing most local optima indicate why metaheuristic optimization algorithms are more effective

than other optimization methods and have become more prevalent in engineering applications in recent decades [8]. Algorithm of the Innovative Gunner (AIG) [9], Artificial Electric Field Algorithm (AEFA) [10], and Quantum Henry Gas Solubility (QHGS) [11] are examples of the recently developed metaheuristics. Metaheuristics as probabilistic solvers have been applied and developed for solving optimization problems in the field of civil engineering, such as structural optimization [12, 13], damage detection [14, 15], optimal design of retaining wall [16, 17], etc. Motivated by the potential application of metaheuristics, this study attempts to apply some population-based metaheuristics for the optimal design of the curved roof and pitched roof portal frames. In the following, some studies conducted to optimize the portal frames utilizing metaheuristics are reviewed.

Saka [18] optimized steel PRFs with haunched rafters using Genetic Algorithm (GA). Phan et al. [19, 20] proposed Real-Coded Niching GA (RC-NGA) for the design optimization of cold-form steel portal frame buildings. In 2015, they considered stressed-skin action and the semi-rigidity of the joints for the design optimization of a cold-formed steel portal frame using RC-NGA [21]. McKinstry et al. [22] optimized fabricated steel beams for long-span portal frames by GA. They found the optimum dimensions of the plates used for the columns, rafters, and haunches of these frames. In 2017, Kaveh et al. [23] proposed an efficient optimization method for the optimal design of steel PRFs with tapered members utilizing Colliding Bodies Optimization (CBO) and Enhanced Colliding Bodies Optimization (ECBO) algorithms. In 2019, Kaveh et al. [24] applied Enhanced Vibrating Particles System (EVPS) algorithm for the optimal design of steel CRFs with non-prismatic members.

In this paper, the discrete optimum design of two types of portal frames including planar steel Curved Roof Frame (CRF) and Pitched Roof Frame (PRF) with tapered I-section members are presented. The main objective of the present paper is to compare the optimized PRF and CRF with tapered I-section members. Furthermore, the performance of the considered metaheuristics in the optimum design of these portal frames is compared. The optimal design aims to minimize the weight of the frames while stratifying some design constraints based on the requirements of ANSI/AISC 360-16 and ASCE 7-10. For the fair comparison, the geometrical characteristics (height and span) of the frames are assumed to be the same, and they are compared in two different span lengths: 16.0 m

and 32.0 m. Optimal design of the frame structures is performed based on linking the existing commercial software SAP2000 and MATLAB through Open Application Programming Interface (OAPI) feature. Since the frame members are non-prismatic, design variables are considered to be dimensions of the cross-sections at the first and end of the frame members. Four different population-based metaheuristic algorithms are considered for the optimal design of the frames, and their performances are compared. These algorithms consist of Teaching-learning-based optimization (TLBO), Enhanced Colliding Bodies Optimization (ECBO), Shuffled Shepherd Optimization Algorithm (SSOA), and Water Strider Algorithm (WSA). The results indicate that CRF is better than PRF from the economic point of view in a fair comparison. Furthermore, comparing the results found by considered algorithms reveals that SSOA outperforms the other algorithms and is more efficient for the optimal design of these portal frames.

The rest of this paper is structured as follows: In Section 2, a description of utilized metaheuristic optimization algorithms is explained briefly. Section 3 presents the mathematical formulation for the discrete structural optimization of the portal frame problem and its design constraints. A brief description of structural loading is provided in Section 4. In Section 5, design examples and discussions on the obtained results are presented. The concluding remarks of this paper are finally driven in Section 6.

## 2 Metaheuristic algorithms

In this article, four population-based metaheuristic algorithms are employed for the optimal design of planar steel CRFs and PRFs. These algorithms are Teaching-Learning-Based optimization (TLBO), Enhanced Colliding Bodies Optimization (ECBO), Shuffled Shepherd Optimization Algorithm (SSOA), and Water Strider Algorithm (WSA). These optimization methods are described briefly in the following subsections.

### 2.1 Teaching-Learning-Based optimization (TLBO)

Rao et al. [25] introduced Teaching-learning-based optimization (TLBO) algorithm based on the learning process occurring in the school. Like most of the population-based optimizers developed in the literature, TLBO initializes with random solutions, each of which is named as student or Learner ( $L$ ). In each iteration of the TLBO, the best learner with the highest quality of the solution is named as the teacher. This algorithm comprises two sequential phases: the teacher phase and the learner phase. These

phases are iteratively performed in the iterative body of the algorithm to search for the optimum learner. TLBO has two parameters: number of learners ( $nL$ ) as population size and maximum number of function evaluations ( $MaxNFES$ ) as a stopping criterion. Since these two parameters are two prevalent parameters existing in any other population-based algorithm, TLBO can be called a parameter-less optimizer.

In the teacher phase, students are updated based on the knowledge transfer from a teacher. Thus, the class performance as a normal distribution of grades is improved by moving the average position of the students toward the best student (teacher). The teacher phase is mathematically formulated as below:

$$L_{new,i} = L_{old,i} + rand_{i,j} \cdot (T - F_i \times ML); \quad (1)$$

$$i = 1, 2, \dots, nL, \quad j = 1, 2, \dots, nd,$$

where  $L_{new,i}$  and  $L_{old,i}$  are respectively the new and old positions of the student;  $rand_{i,j}$  is a random number generated in the  $[0,1]$  interval;  $T$  represent the best learner who is considered as the teacher;  $F_i$  is a teaching factor that can be either 1 or 2;  $ML$  is the average position of the learners in the search space, and  $nd$  is the number of design variables. This phase indicates intensification or global search ability of the TLBO algorithm by moving the  $ML$  toward the teacher.

In the learner phase, students are cyclically updated based on the knowledge transfer from interaction with a randomly selected one. This phase can be mathematically stated as follows:

$$L_{new,i} = L_{old,i} + rand_{i,j} \cdot \begin{cases} L_i - L_{rs} & \text{if } f(L_i) < f(L_{rs}) \\ L_{rs} - L_i & \text{if } f(L_i) > f(L_{rs}) \end{cases}, \quad (2)$$

in which  $rand_{i,j}$  is a random number generated in the  $[0,1]$  interval;  $L_i$  is the  $i$ th student;  $L_{rs}$  denotes to a randomly selected student ( $rs \neq i$ );  $f(L_i)$  and  $f(L_{rs})$  are the objective function values of  $i$ th and randomly selected students, respectively. This phase shows the diversification or local search ability of the TLBO algorithm. Because each student attempts to find a better position by searching around its neighborhood and share information with a student selected randomly.

After each searching phase, the replacement strategy is carried out to keep the old learners or replace them with the newly generated ones. In this regard, the student with the smallest objective function value or better quality is preferred to the old student. For further clarity, the pseudo-code of the TLBO algorithm is given in Algorithm 1.

---

**Algorithm 1** Pseudo-code of the TLBO

---

```

Set the algorithm parameters:  $nL$  and  $MaxNFES$ 
Generate the initial learners randomly in the search space
Evaluate the initial learners
while  $NFES \leq MaxNFES$ 
    Determinate the teacher for the learners
    Calculate the mean position of the learners
    Generate the new learners using Eq. (1)
    Evaluate the new learners
    Apply the replacement strategy between the new and old learners
    Generate the new learners using Eq. (2)
    Evaluate the new learners
    Apply the replacement strategy between the new and old learners
end while
Report the best learner found by the TLBO algorithm
    
```

---

**2.2 Enhanced Colliding Bodies Optimization (ECBO)**

Colliding Bodies Optimization (CBO) is a simple and efficient population-based metaheuristic algorithm. This optimization method inspires by the one-dimensional collision between two bodies. Like TLBO, there are no algorithm-specific parameters for CBO so that it can be called parameter-less metaheuristic. Although CBO has these advantages, it suffers from some shortcomings. In order to alleviate these handicaps, Kaveh and Illchi Ghazan [26] developed an enhanced variant of the algorithm, namely ECBO. In their proposed algorithm, a memory that saves the number of the best solutions obtained so far and a mechanism that changes some components of colliding bodies (CBs) are used to improve the algorithm's performance. The memory can increase the convergence speed of ECBO compared to standard CBO, and the mechanism makes a chance for CBs to escape from local optima and prevents unwanted premature convergence. Like other population-based metaheuristics, ECBO starts from a set of candidate solutions, each of which is called as a Colliding Body (CB). These CBs are randomly generated within the search space. Thereafter, the objective function values of the CBs are evaluated. For each CB, a specified mass is defined according to the following equation:

$$m_i = \frac{1 / f(CB_i)}{\sum_{i=1}^{nCB} 1 / f(CB_i)}; \quad i = 1, 2, \dots, nCB, \quad (3)$$

where  $f(CB_i)$  represents the objective function value of the  $i$ th CB, and  $nCB$  is the number of colliding bodies. To save a number of the best solutions obtained so far, a Colliding Memory (CM) is used in the iterative body of the ECBO algorithm. For this purpose, the vector of solutions saved in CM is added to the current population, and the same

number of the current worst CBs are deleted. Next, the CBs are sorted according to the associated masses in ascending order. In the next step, CBs are divided into two distinct groups: stationary and moving. The first half of the objects ( $i = 1, 2, \dots, \frac{nCB}{2}$ ) are considered as the stationary objects, while the next half of them are assumed to be moving objects ( $i = \frac{nCB}{2} + 1, \frac{nCB}{2} + 2, \dots, nCB$ ). For colliding, moving objects move toward to the corresponding stationary objects. The velocities of stationary and moving objects before collision ( $v_i$ ) and after collision ( $v_i'$ ) can be obtained respectively by the following equations:

$$v_i = 0; \quad i = 1, 2, \dots, \frac{nCB}{2}, \quad (4)$$

$$v_i = CB_{i-\frac{nCB}{2}} - CB_i; \quad i = \frac{nCB}{2} + 1, \frac{nCB}{2} + 2, \dots, nCB, \quad (5)$$

$$v_i' = \frac{\left( \frac{m_{i+\frac{nCB}{2}} + \varepsilon m_{i-\frac{nCB}{2}}}{m_i + m_{i-\frac{nCB}{2}}} \right) v_{i+\frac{nCB}{2}}}{m_i + m_{i-\frac{nCB}{2}}}; \quad i = 1, 2, \dots, \frac{nCB}{2}, \quad (6)$$

$$v_i' = \frac{\left( \frac{m_i - \varepsilon m_{i-\frac{nCB}{2}}}{m_i + m_{i-\frac{nCB}{2}}} \right) v_i}{m_i + m_{i-\frac{nCB}{2}}}; \quad i = \frac{nCB}{2} + 1, \frac{nCB}{2} + 2, \dots, nCB, \quad (7)$$

$$\varepsilon = 1 - \frac{it}{MaxNITs}, \quad (8)$$

in which  $\varepsilon$  is the coefficient of restitution (COR) decreasing linearly from unit to zero;  $it$  is the current iteration number of the algorithm;  $MaxNITs$  is the maximum number of algorithm iterations. The ECBO algorithm assumes the current position of the stationary objects as the origin of both stationary and moving objects. Thus, the new position of both stationary and moving objects will be achieved by adding their new velocities to their current positions according to the following equations:

$$CB_{new,i} = CB_{old,i} + rand_i \circ v_i'; \quad i = 1, 2, \dots, \frac{nCB}{2}, \quad (9)$$

$$CB_{new,i} = CB_{old,i-\frac{nCB}{2}} + rand_i \circ v_i'; \quad (10)$$

$$i = \frac{nCB}{2} + 1, \frac{nCB}{2} + 2, \dots, nCB,$$

where  $rand_i$  generates a uniformly distributed random vector in which each component is in the range of  $[-1, 1]$  and the sign " $\circ$ " is the element-by-element multiplication between two vectors.

In the next step of the ECBO algorithm, a mechanism is considered to escape from local optima. To this end, a uniformly distributed random number like  $rn_i$  is generated in the range of  $(0, 1)$  for each  $CB_i$ . This randomly generated number is then compared with a parameter like  $pro$  within  $(0, 1)$ . If  $rn_i < pro$ , one randomly selected component of  $i$ th CB is regenerated, and its value is changed by:

$$CB_{ij} = CB_{j,min} + rand_{ij} \times (CB_{j,max} - CB_{j,min}), \quad (11)$$

where  $CB_{ij}$  is the  $i$ th design variable of the  $i$ th CB;  $CB_{j,min}$  and  $CB_{j,max}$  represent the lower and upper bounds of the  $j$ th design variable, respectively.

The optimization process will be terminated if the maximum number of function evaluations ( $MaxNFES$ ) as a stopping criterion of the ECBO algorithm is reached.

The pseudo-code of the ECBO algorithm is given in Algorithm 2.

### 2.3 Shuffled Shepherd Optimization Algorithm (SSOA)

In 2020, Kaveh and Zaerrega [27] developed a new multi-population metaheuristic algorithm, namely Shuffled Shepherd Optimization Algorithm (SSOA). This algorithm is inspired by the herding behavior of shepherds in nature. Like the TLBO and ECBO algorithms, SSOA starts with randomly generated solutions, each of which is named as Sheep ( $S$ ). Sheep are divided into the  $nh$  herds using the shuffling process. To this end, first of all, all sheep are evaluated and sorted in ascending order based on their objective function values. Next,  $nh$  of the first sorted sheep are selected and assigned randomly to each herd. Thus, each herd has one sheep at the first step of forming herds. After assigning the first sheep for each herd, the sorted sheep from  $nh + 1$  to  $2nh$  are selected and again assigned randomly to the herds. Each herd has two sheep at the end of forming herds in this step. This process continues until

---

#### Algorithm 2 Pseudo-code of the ECBO

---

Set the algorithm parameters:  $nCB$ ,  $MaxNFES$ , size of  $CM$ , and  $pro$   
Generate the initial  $CBs$  randomly in the search space  
Evaluate the initial  $CBs$

**while**  $NFES < MaxNFES$

    Calculate the mass of each  $CB$  using Eq. (3)

    Update the  $CM$

    Update the population

    Sort the population in ascending order

    Create the groups

    Calculate the velocity of each  $CBs$  using Eq. (6) and (7)

    Generate the new  $CBs$  using Eq. (9) and (10)

    Apply escape from local optima mechanism using Eq. (11)

    Evaluate the new  $CBs$

**end while**

Report the best  $CB$  found by the ECBO algorithm

---

all sheep are given to the herds. All herds have an equal number of sheep, and the best and worst sheep of each herd are the first and last members of the corresponding herd. After forming the herds and assigning the sheep to them, the step size can be calculated for each sheep. To this end, sheep with better and worse objective function values than the considered sheep are selected randomly from the same herd. In the SSOA, the considered, better, and worse sheep of the corresponding herd are respectively called the shepherd ( $S_{i,j}$ ), horse ( $S_{i,h}$ ), and sheep ( $S_{i,s}$ ). In order to guide the sheep toward the horse, the shepherd changes his position to the sheep and then moves toward the horse. This concept can be mathematically stated as follows:

$$stepsize_{i,j} = stepsize_{i,j}^{sheep} + stepsize_{i,j}^{horse} \quad (12)$$

$$i = 1, 2, \dots, nh, \quad j = 1, 2, \dots, nS / nh,$$

where  $nS$  and  $nh$  are the number of the sheep and herds, respectively. Moreover,  $stepsize_{i,j}^{sheep}$  and  $stepsize_{i,j}^{horse}$  are calculated as:

$$stepsize_{i,j}^{sheep} = \alpha \times rand_1 \circ (S_{i,s} - S_{i,j}), \quad (13)$$

$$stepsize_{i,j}^{horse} = \beta \times rand_2 \circ (S_{i,h} - S_{i,j}), \quad (14)$$

where  $rand_1$  and  $rand_2$  are random vectors in which each component is in the interval [0,1];  $\alpha$  and  $\beta$  are control parameters. They are used to control the exploration and exploitation rates of the SSOA, respectively. These controlling parameters are defined as follows:

$$\alpha = \alpha_{max} \times \left( 1 - \frac{it}{MaxNITs} \right), \quad (15)$$

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) \times \frac{it}{MaxNITs}, \quad (16)$$

where  $it$  and  $MaxNITs$  are respectively the current number of iteration and the maximum number of iterations;  $\alpha_{max}$ ,  $\beta_{max}$ , and  $\beta_{min}$  are parameters of the algorithm defined by the user. Eqs. (15) and (16) show that if the number of algorithm iterations is increased,  $\alpha$  decreases linearly from  $\alpha_{max}$  to zero, and  $\beta$  increases linearly from  $\beta_{min}$  to  $\beta_{max}$ . Decreasing  $\alpha$  and increasing  $\beta$  provide a good counterbalance between exploration and exploitation capabilities during the course of the optimization process. After calculating the stepsize for the sheep of all herds, the new position of each sheep is as follows:

$$S_{i,j}^{new} = S_{i,j}^{old} + stepsize_{i,j}, \quad (17)$$

In the next step, the replacement strategy is applied between  $S_{i,j}^{new}$  and  $S_{i,j}^{old}$ , and the best position of sheep is going to

the next round of iteration. This process is repeated for the sheep of all herd.

Thereafter, the formed herds are merged together, and the sheep are sorted in ascending order based on the objective function values. Again, the sheep are divided into  $nh$  herds using the shuffling process. The afore-mentioned process is repeated in the cyclic body of the algorithm until the algorithm is terminated. Similar to the TLBO and ECBO algorithms, the  $MaxNFES$  is considered as the stopping criterion of the SSOA.

The pseudo-code of the SSOA algorithm is given in Algorithm 3.

#### 2.4 Water Strider Algorithm (WSA)

Water Strider Algorithm (WSA), developed by Kaveh and Dadras Eslamlou [28], is a new nature-inspired meta-heuristic algorithm. This population-based swarm intelligent algorithm mimics territorial behavior, intelligent ripple communication, mating style, feeding mechanisms, and succession of water striders. WSA is initialized with a population of randomly generated solutions in the search space. In this algorithm, each candidate solution is considered as a Water Strider ( $WS$ ). After the initialization phase of the algorithm, the territories are established. Each territory has at least one mature male (keystone) and few female bugs. For assigning  $nws$  number of  $WSs$  to the  $nt$  number of territories, the following method is performed. In the first step of establishing territories,  $WSs$  are evaluated and sorted in ascending order of their objective function values. Then, the population of  $WSs$  is orderly divided into  $nws/nt$  groups. Next, the first  $WS$  of each group is selected and assigned orderly to the first territory. At the end of the first step, the first territory has  $nws/nt$   $WSs$ . In the next step, the second  $WS$  of each group is chosen and placed orderly in the second territories. At the end of this step, the second territory has the same number of  $WSs$  as the first territory.

---

#### Algorithm 3 Pseudo-code of the SSOA

---

```

Set the algorithm parameters:  $nS$ ,  $nh$ ,  $\alpha_{max}$ ,  $\beta_{min}$ ,  $\beta_{max}$ , and  $MaxNFES$ 
Generate the initial sheep randomly in the search space
Evaluate the initial sheep
while  $NFES < MaxNFES$ 
    Apply shuffling proses
    Calculate the stepsize using Eq. (12)
    Generate the new sheep using Eq. (17)
    Evaluate the new sheep
    Apply the replacement strategy between the new and old sheep
    Merge all herds into the single population
end while
Report the best  $S$  found by the SSOA algorithm

```

---

The process of establishing territory continues until all  $WSs$  are assigned to the territories. At the end of last step, the territories have an equal number of  $WSs$ . It is clear that the first and last  $WSs$  are respectively the best and worst agents within a specified territory, and they are respectively considered as female and male (keystone).  $WSs$  are updated cyclically to search the optimum within three sequential steps: mating, feeding, and finally succession of keystone. Each of these steps is briefly stated as follows:

#### 2.4.1 Mating

In order to mate, the keystone sends a ripple to an objective female. She responds to him by sending attraction or repulsive ripple signals. Since this answer is unknown, the probability of mating is defined. For simplicity, this probability is assumed to be 50%. The following equation gives an equal probability of mating and repelling and updates the position of the keystone:

$$WS_i^{c+1} = \begin{cases} WS_i^c + R \circ rand & \text{if } rand < p \\ WS_i^c + R \circ (1 + rand) & \text{otherwise} \end{cases}, \quad (18)$$

where  $WS_i^c$  denotes to the position of the  $i$ th  $WS$  in the  $c$ th cycle;  $rand$  is a random vector which each component is generated between 0 and 1,  $R$  is a vector obtained as follows:

$$R = WS_F^{c-1} - WS_i^{c-1}, \quad (19)$$

in which  $WS_F^{c-1}$  and  $WS_i^{c-1}$  represent the position of female and male  $WSs$  in the  $(c-1)$ th cycle, respectively.

#### 2.4.2 Feeding

The keystone uses up a lot of energy whether the mating process occurs successfully or not. In the next step, he needs a recovery by searching for food sources. For finding the food supplies, the objective function is evaluated. If the objective function value is better than that obtained in the mating process, he has already achieved enough food. On the contrary, if the objective function of the keystone is worse than that of in the previous state, the keystone should move toward the best  $WS$  of the lake ( $WS_{BL}$ ) to look for food availability by the following equation:

$$WS_i^{c+1} = WS_i^c - 2 \times rand_i \circ (WS_{BL}^c - WS_i^c), \quad (20)$$

#### 2.4.3 Succession of keystone

In this phase, if the quality of the keystone is worse than the previous state, the keystone will die since he cannot find food. Thus, a new keystone is regenerated randomly in the lake as follows:

$$WS_i^{c+1} = WS_{j,min}^c + rand \circ (WS_{j,max}^c - WS_{j,min}^c). \quad (21)$$

In the iterative body of the WSA, establishing territories with the aforementioned steps is repeated until the stopping criterion of the algorithm is met. Similar to the above-mentioned algorithms, the  $MaxNFEs$  is considered as the stopping criterion of the WSA. For further clarity, the pseudo-code of the WSA algorithm is given in Algorithm 4.

### 3 Statement of the discrete optimization problem

The optimization problem definition for the optimal design of the frame structures can be stated as follows:

$$\begin{aligned} \text{Find } \{X\} &= [x_1, x_2, x_3, \dots, x_n], \\ \text{To minimize: } &W(\{X\}), \end{aligned} \quad (22)$$

in which  $\{X\}$  is the vector of design variable;  $n$  is the number of design variables, and  $W(\{X\})$  is the total weight of the steel frame structures. We know that the presented algorithms (i.e., TLBO, ECBO, SSOA, and WSA) were developed for continuous search space. However, such algorithms can be simply used for discrete optimization problems. In this paper, we used the rounding function to convert the vector of continuous design variables to discrete design variables. In other words, the vector of design variables (i.e.,  $\{X\} = [x_1, x_2, x_3, \dots, x_n]$  in Eq. (22)) generated by the algorithm rounds to the nearest available value using the rounding function.

The objective function presented in Eq. (22) is minimized subjected to the following design constraints:

---

#### Algorithm 4 Pseudo-code of the WSA

---

```

Set the algorithm parameters:  $nws$ ,  $nt$ ,  $p$ , and  $MaxNFEs$ 
Generate the initial  $WSs$  randomly in the search space
Evaluate the initial  $WSs$ 
while  $NFEs < MaxNFEs$ 
    Establish territories
    Calculate the new position of keystone using Eq. (18)
    Evaluate him
    if the old keystone better than the new one
        Calculate the other new keystone using Eq. (20)
        Evaluate him
        if old keystone better than new one again
            Regenerate the keystone randomly in the search space
            Evaluate him
            Replace the new keystone with the old one
        else
            Replace the new keystone with the old one
        end if
    end if
    else
        Replace the new keystone with the old one
    end if
end while
Report the best  $WS$  found by the WSA algorithm
    
```

---

- $G_1$ : Check the stability constraint
- $G_2$ : Check the buckling constraints
- $G_3$ : Check the strength constraints (23)
- $G_4$ : Check the maximum vertical displacement
- $G_5$ : Check the maximum horizontal displacement

### 3.1 Checking design constraints of the problem

In the following subsections, the procedure of checking design constraints given in Eq. (23) is examined.

#### 3.1.1 Checking constraint $G_1$

To check the stability of the structure, constraint  $G_1$  is employed. This constraint is defined based on ANSI/AISC 360-16 provisions [29]. For checking the stability of a structure, the stability index under  $P$ -delta effects which provide extra forces in the members is calculated according to the following equation:

$$\theta = \frac{P_x \Delta I_e}{V_x h_{sx} C_d}, \quad (24)$$

in which  $\theta$  is the coefficient of stability;  $P_x$  denotes to the total vertical design load above level  $x$  with a maximum load factor of 1.0 (kip or kN);  $\Delta$  represents the design story drift happening concurrently with  $V_x$ ;  $I_e$  is the importance factor;  $V_x$  is the seismic shear force acting between levels  $x$  and  $x - 1$ ;  $h_{sx}$  is the story height below level  $x$ , and  $C_d$  is the deflection amplification factor. The upper bound of  $\theta$  is determined as:

$$\theta_{max} = \frac{0.5}{\beta C_d} \leq 0.25, \quad (25)$$

where  $\beta$  is the ratio of shear demand to shear capacity for the story between levels  $x$  and  $x - 1$ . It is worth mentioning that  $\beta$  is allowed to be conservatively taken as 1.0. If  $\theta \leq 1.0$ ,  $P$ -delta effects are not required to be considered. If  $1.0 < \theta \leq \theta_{max}$ , the incremental factor related to the  $P$ -delta effects on member forces and displacements should be determined by the rational analysis. Alternatively, it is allowed to multiply member forces and displacements by  $1.0/(1 - \theta)$ . Otherwise, if  $\theta > \theta_{max}$ , the structure is potentially unstable and should be redesigned [29, 30].

#### 3.1.2 Checking constraint $G_2$

This constraint checks the buckling of the structural elements. According to the ANSI/AISC 341-16 [30] requirements for designing slender compression members, the logical and practical width-to-thickness ratios (i.e., Eqs. (26) and (27)) must be fulfilled so that the buckling does not occur.

$$b_f / t_f \leq 18, \quad (26)$$

$$h / t_w \leq 0.4E / F_y \leq 260. \quad (27)$$

#### 3.1.3 Checking constraint $G_3$

This constraint checks the strength of all sections according to the requirements of ANSI/AISC 360-16 [29]:

$$\begin{aligned} \frac{P_u}{2\phi_c \bar{P}_n} + \frac{M_u}{\phi_b \bar{M}_n} - 1 &\leq 0; \text{ if } \frac{P_u}{\phi_c \bar{P}_n} < 0.2, \\ \frac{P_u}{\phi_c \bar{P}_n} + \frac{8}{9} \left( \frac{M_u}{\phi_b \bar{M}_n} \right) - 1 &\leq 0; \text{ if } \frac{P_u}{\phi_c \bar{P}_n} \geq 0.2, \end{aligned} \quad (28)$$

where  $P_u$  is the required strength (tension or compression);  $\bar{P}_n$  is the nominal axial strength (tension or compression);  $\phi_c$  is the resistance factor ( $\phi_c = 0.9$  for tension,  $\phi_c = 0.85$  for compression);  $M_u$  is the required flexural strength;  $\bar{M}_n$  represents the nominal flexural strength, and  $\phi_b$  is the flexural resistance reduction factor ( $\phi_b = 0.90$ ). The following equation calculates the nominal tensile strength:

$$P_n = A_g \times F_y, \quad (29)$$

in which  $A_g$  is the gross cross-sectional area of the member and  $F_y$  is specified minimum yield stress. The nominal compressive strength of a member is computed as:

$$P_n = A_g \times F_{cr}, \quad (30)$$

$$F_{cr} = \left( 0.658 \frac{F_y}{F_e} \right) F_y; \text{ for } \frac{KL}{r} \leq 4.71 \sqrt{\frac{E}{F_y}}, \quad (31)$$

$$F_{cr} = 0.877 \times F_e; \text{ for } \frac{KL}{r} > 4.71 \sqrt{\frac{E}{F_y}}, \quad (32)$$

$$F_e = \frac{\pi^2 E}{\left( \frac{KL}{r} \right)^2}, \quad (33)$$

where  $F_{cr}$  is the critical stress of the member,  $F_e$  is elastic buckling stress;  $E$  is the modulus of elasticity,  $r$  is the radius of gyration,  $L$  is the laterally unbraced length of the member, and  $k$  is the effective length factor.

#### 3.1.4 Checking constraint $G_4$

The maximum vertical displacement is checked by this constraint as follows:

$$\frac{\Delta_V}{L} - R_V \leq 0, \quad (34)$$

in which  $\Delta_v$  is the maximum vertical displacement of apex in CRF or roof in the PRF;  $L$  is the span length of the CRF or PRF, and  $R_v$  represents the allowable vertical displacement and equal to 1/360 and 1/240 under the dead and live loadings, respectively.

### 3.1.5 Checking constraint $G_5$

This constraint checks the maximum horizontal displacement by the following equation:

$$\frac{\Delta_H}{H} - R_H \leq 0, \quad (35)$$

where  $\Delta_H$  is the maximum horizontal displacement of the eaves point in the CRF or PRF;  $H$  is the height of the column, and  $R_H$  represents the allowable horizontal displacement and equal to  $H/200$  under all loadings.

To handle all the constraints mentioned above, the penalty approach is used. In this regard, if these design constraints are not violated, the value of the penalty will be equal to zero. On the contrary, if each of the design constraints is violated, the penalty is considered for the objective function as follows:

$$f_{penalty}(\{X\}) = W(\{X\}) \times \left( 1 + \sum_{i=1}^{nte} p_i \right), \quad (36)$$

where  $nte$  is the number of structural elements, and  $p_i$  is the penalty for the  $i$ th member, which can be calculated as follows:

$$p_i = \begin{cases} 1 & \text{if } G1 \text{ is violated} \\ 1 & \text{if } G2 \text{ is violated} \\ r & \text{if } G3 \text{ is violated} \\ 1 & \text{if } G4 \text{ is violated} \\ 1 & \text{if } G5 \text{ is violated} \\ 0 & \text{otherwise} \end{cases}, \quad (37)$$

in which  $r > 1$  is the stress ratio.

## 3.2 Optimum design of the structures using SAP2000-OAPI

The optimum design of the structure is an optimization problem where the solution can be the optimal size for the structural members (sizing optimization), optimal coordinate for the structure nodes (shape optimization), or optimal size and connectivity between structural members (topology optimization). In this paper, we are going to find the optimal size for structural members of the PRF and CRF. This problem can be solved as an optimization

task using Eq. (36). For the purpose of finding the value of the vector  $\{X\}$  in Eq. (36), metaheuristic algorithm as a powerful and reliable optimization tool is selected to minimize  $f_{penalty}(\{X\})$ .

For analyzing the structural model and obtaining some necessary information to check design constraints, we link SAP2000 and MATLAB software via the Open Application Programming Interface (OAPI) feature. The flowchart of finding the optimal size for structural members based on SAP2000-OAPI and the metaheuristic algorithm is presented in Fig. 1. According to this figure, first of all, the metaheuristic algorithm's parameters, such as population size and maximum number of function evaluations as termination criterion, are set. After that, the population of candidate solutions (algorithm agents) is initialized randomly. The randomly initialized solutions are generated in the continuous form. However, as mentioned before, here, we deal with discrete search space in which the design variables of the optimization problem are selected from the discrete set. Accordingly, the rounding function is employed to convert the generated solutions from continuous to discrete, and the solutions round to the nearest discrete available value. Thus, the obtained values are discrete ones that compose our initial sections. At the same time, the model of the structure is also initialized in SAP2000. Using the OAPI feature, we access the SAP2000 software through MATLAB to change the sections of the initial model. By considering this, the initial sections obtained from MATLAB are assigned to members modeled in SAP2000. The model is then analyzed to get the forces of the members. In the design stage, the necessary information required to check the problem's design constraints (as mentioned in Section 3.1) is determined. By obtaining the required information, these constraints are checked in MATLAB environments. If each design constraint is violated, it will be penalized according to Eq. (37). It should be noted that the value of  $W(\{X\})$  is obtained from the SAP 2000. By determining the value of  $W(\{X\})$  and  $\sum_{i=1}^{nte} p_i$  in Eq. (36), the value of the objective function, which indicates the total weight of the structure, is obtained. After that, we will go to the main loop of the metaheuristic algorithm, which is implemented iteratively until the stopping criterion is satisfied. In each iteration, the vector of design variable  $\{X\}$  is updated by the metaheuristic algorithm. In other words, since we deal with the population-based metaheuristic algorithm, each algorithm agent generates new values for the components of the vector  $\{X\}$  in each iteration. Then, each value of this vector is converted to the discrete value by rounding function, and the

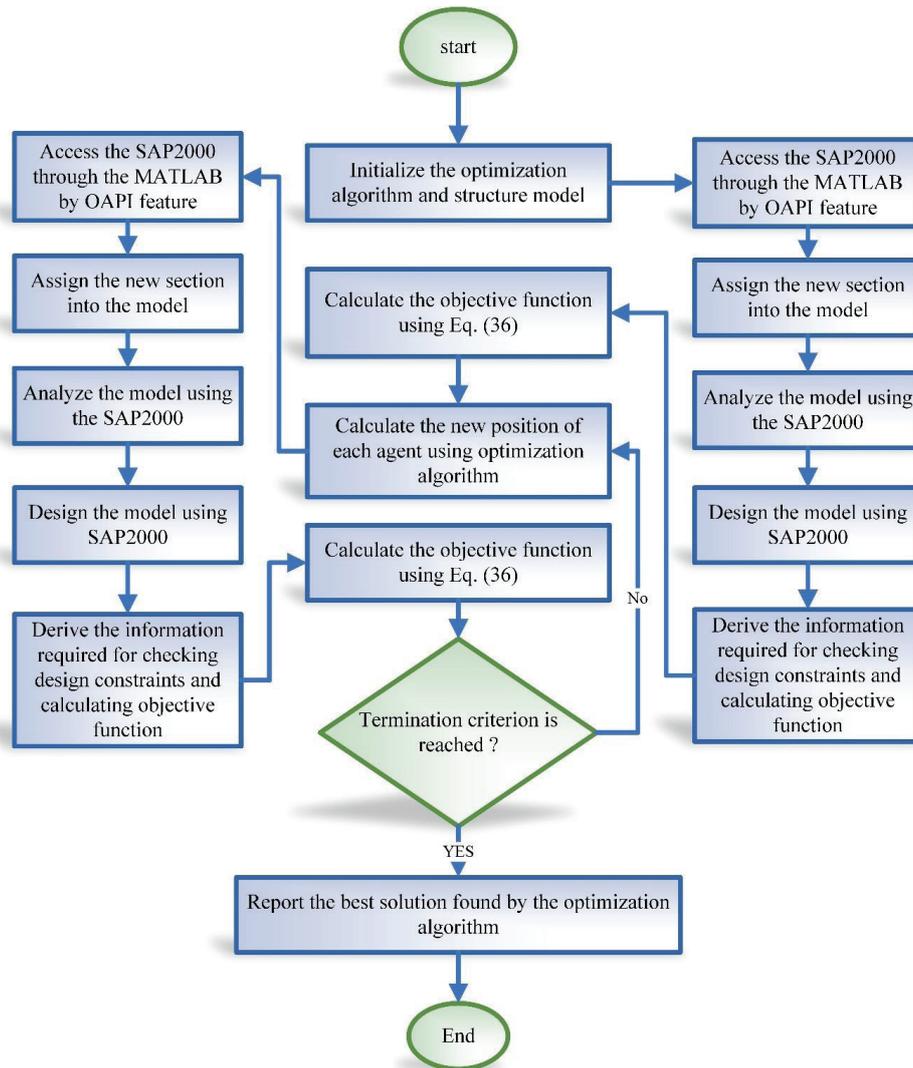


Fig. 1 The flowchart of finding the optimal size of structural members based on SAP2000-OAPI and the metaheuristic algorithm

process mentioned above is repeated until the new value for the objective function is found. This process continues in the cyclic body of the algorithm for all agents, and it will be terminated when the stopping criterion of the algorithm is reached. Finally, the best values for the components of the vector  $\{X\}$  is reported as the best solution found by the metaheuristic algorithm.

#### 4 Structural loading

##### 4.1 Load combinations

Some load combinations must be considered for the design of steel portal frames. According to the ASCE/SEI 7-10 [31] requirements, the following equation is assumed for designing members of the frames. These combinations are defined for vertical loads (i.e., dead, live, and snow) and lateral loads (i.e., seismic and wind).

1.  $1.4D$
  2.  $1.2D + 1.6L + 0.5(S \text{ or } R)$
  3.  $1.2D + 1.6(S \text{ or } R) + (L \text{ or } 0.8W)$
  4.  $1.2D + 1.0W + L + 0.5(S \text{ or } R)$
  5.  $(1.2 + 0.2S_{DS})D + E + L + 0.2S$
  6.  $0.9D + 1.0W$
  7.  $(0.9 + 0.2S_{DS})D + E$
- (38)

##### 4.2 Vertical loads

###### 4.2.1 Dead loads

The dead and collateral loads (D) consist of the self-weight of the structure and the weight of the roof purlins and panels with a mass equal to  $14.65 \text{ kg/m}^2$ , as shown in Table 1. In this case, the effects of collateral loads are not considered, so it is assumed zero.

**Table 1** The dead load parameters

Dead load (kg/m <sup>2</sup> )	14.65
Loading width (per m)	6.0
Frame distributed dead load (kg/m)	87.85

**4.2.2 Live loads**

According to the ASCE/SEI 7-10, the live loads acting on the roof beams are almost 100 kg/m<sup>2</sup>. The summary of live loads is given in Table 2.

**4.2.3 Snow load**

There are two different types of snow load in these structures. The balanced and unbalanced snow loads. The flat snow load ( $P_f$ ) and the balanced snow load ( $P_{s0}$ ) are calculated as follows:

$$P_f = 0.7C_eC_tI_sP_g, \tag{39}$$

$$P_{s0} = C_sP_f, \tag{40}$$

in which  $C_e$  is the exposure factor,  $C_t$  is the thermal factor,  $I_s$  is the importance factor,  $P_g$  is the ground snow load, and  $C_s$  is the roof slope factor. Table 3 gives the details of the necessary parameters for calculating the snow load. In our design examples, the roof slope is taken a value lower than 30 degrees. Thus, the value of  $C_s$  will be obtained equal to 1.0 according to the code. Fig. 2 shows how the distribution of balanced and unbalanced snow loads is calculated.

**4.3 Lateral loads**

**4.3.1 Seismic load**

For calculating the seismic load, first, the seismic base shear is calculated:

$$V = C_sW, \tag{41}$$

where  $W$  is the effective seismic weight, and  $C_s$  is the seismic response coefficient obtained as follows:

**Table 2** The live load parameters

Live load (kg/m <sup>2</sup> )	100
Loading width (per m)	6.0
Frame distributed live load (kg/m)	590

**Table 3** The snow load parameters

$C_e$	1.0
$C_t$	1.0
$I_s$	1.0
$P_g$ (kg/m <sup>2</sup> )	98
$P_f$ (kg/m <sup>2</sup> )	69
$C_s$	1.0

$$C_s = \frac{S_{DS}}{RSI_e}, \tag{42}$$

where  $S_{DS}$  is the design spectral response acceleration parameter in short period range,  $R$  is the response modification factor, and  $I_e$  is the importance factor. Since the site of the investigated design examples is placed in Clay County of Kansas in the United States of America, the summarized calculation of the  $C_s$  is given in Table 4.

**4.3.2 Wind load**

In order to calculate the wind load for low-rise buildings, first, the wind pressure should be obtained:

$$q_z = 0.613K_ZK_{Zt}K_dV^2 \left( N / m^2 \right), \tag{43}$$

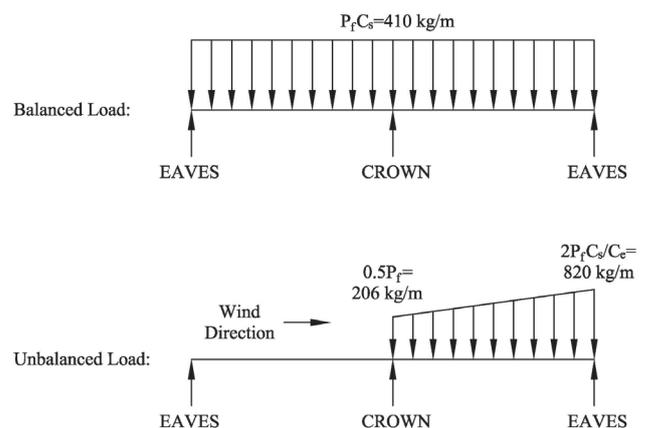
in which  $K_Z$  represents the velocity pressure exposure coefficient,  $K_{Zt}$  denotes to the topographic factor,  $K_d$  is the wind directionality factor, and the  $V$  is the basic wind speed.  $q_h$  is the velocity pressure at height  $h$  (average height of roof) obtained as follows:

$$q_h = 0.613K_hK_{Zt}K_dV^2 \left( N / m^2 \right). \tag{44}$$

According to Table 27.3-1 in ASCE 7-10,  $K_h$  is obtained as follows:

$$K_h = 2.01 \left( \frac{h}{274.32} \right)^{9.5} \text{ in metric,} \tag{45}$$

where  $h$  is the average height of the roof, and  $K_h$  is a function of the average height of the roof so that its value is changed by altering the slope of the roof.



**Fig. 2** Calculation of balanced and unbalanced snow load

**Table 4** The summarized calculation of

$S_{DS}$	0.2768
$R$	4.5
$I_e$	1
$C_s = \frac{S_{DS}}{RSI_e}$	0.041

The design wind pressures for the frame system of an enclosed and partially enclosed rigid building at all heights are calculated as below:

$$P = qGC_p - q_i(GC_{pi}), \quad (46)$$

in which  $q$  is velocity pressure ( $\text{kg/m}^2$ );  $G$  is the gust-effect factor;  $C_p$  is the external pressure coefficient calculated according to Table 5 for PRFs and Table 6 for CRFs;  $q_i$  is velocity pressure for internal pressure determination, and  $GC_{pi}$  is the internal pressure coefficient. For  $q$  and  $q_i$ , we have:

- $q = q_z$  for windward walls evaluated at height above the ground.
- $q = q_h$  for leeward walls, side walls, and roofs evaluated at height  $h$ .
- $q_i = q_z$  for the positive internal pressure evaluation in partially enclosed buildings in which the height  $z$  is defined as the level of the highest opening in the building that could affect the positive internal pressure. For positive internal pressure evaluation ( $q_i$ ) may conservatively be evaluated at height  $h$ .
- $q_i = q_h$  for the windward wall, side walls, leeward wall, and roof of enclosed buildings and negative internal pressure evaluation in partially enclosed buildings.

Table 7 summarizes the details of wind load parameters that need for calculating wind load parameters of the examined design examples.

### 5 Design examples

In this section, two design examples of symmetric portal frames including steel CRF and PRF with different span lengths are examined. The design of the portal frames is mostly performed for a two-dimensional frame due to repeating a series of transverse frames braced longitudinally. The span length of the portal frames for the first and second design examples are considered as 16.0 m ( $L = 16.0$  m) and 32.0 m ( $L = 32.0$  m), respectively. The geometrical shape of the portal frames for the first and second design examples are illustrated in Figs. 3 and 4, respectively. The columns and rafters of the design examples are web-tapered I-section with the same flange width, and the inside and outside of the flange thickness also have an equal value. In the analysis and design of the portal frames, the connections between columns and base plates are considered as pin-jointed. However, the connections between columns and rafters are assumed as

**Table 5** The coefficient of in two orthogonal directions of wind for PRFs

No. Case	The directions of wind	$C_p$
Transverse wind direction (Case 1)	Windward wall	0.8
	Windward roof	-0.7
	Leeward roof	-0.5
Transverse wind direction (Case 2)	Leeward wall	-0.5
	Windward wall	0.8
	Windward roof	-0.18
	Leeward roof	-0.5
	Leeward wall	-0.5

**Table 6** The coefficient of in two orthogonal directions of wind for CRFs

Rise-to-Span Ratio, $r$	Windward quarter	Center half	Leeward quarter
$0 < r < 0.2$	-0.9	$-0.7 - r$	-0.5
$0.2 \leq r < 0.3$	$1.5r - 0.3$	$-0.7 - r$	-0.5
$0.3 \leq r \leq 0.6$	$2.75r - 0.7$	$-0.7 - r$	-0.5

\*  $r$  is the rise-to-span ratio

**Table 7** Details of wind load parameters for each design example

wind load parameters	The first design example	The second design example
$K_z$	0.89	0.89
$K_{zt}$	1.0	1.0
$K_d$	0.85	0.85
$V$	90 mph (40.234 m/s)	90 mph (40.234 m/s)
$h$	26.57 ft	53.14 ft
$G$	0.85	0.85
$GC_{pi}$	$\pm 0.18$	$\pm 0.18$
$q_z$	15.834 psf	15.834 psf
$q_h$	16.232 psf	18.782 psf
$q_i$	16.232 psf	18.782 psf
$r$	0.0093	0.0093

moment-resisting. The nodal geometry of the members is given based on the neutral axis of the members. The design examples are located in Clay County from Kansas in the United States of America.

TLBO, ECBO, SSOA, and WSA algorithms are considered for finding the best, mean, and worst optimum weight of the design examples, and the obtained results of the algorithms are compared. For all algorithms, population size and the maximum number of function evaluations ( $MaxNFEs$ ) are equal to 20 and 4000, respectively. The internal parameters of ECBO, SSOA, and WSA are according to the literature: ECBO, size of  $CM = 2$  and  $pro = 0.3$ ; SSOA,  $nh = 4$ ,  $\alpha_{max} = 1$ ,  $\beta_{min} = 2$ , and  $\beta_{max} = 3$ ; and WSA,  $nt = 10$ , and  $p = 0.5$ . According to the Fig. 5, 13 and

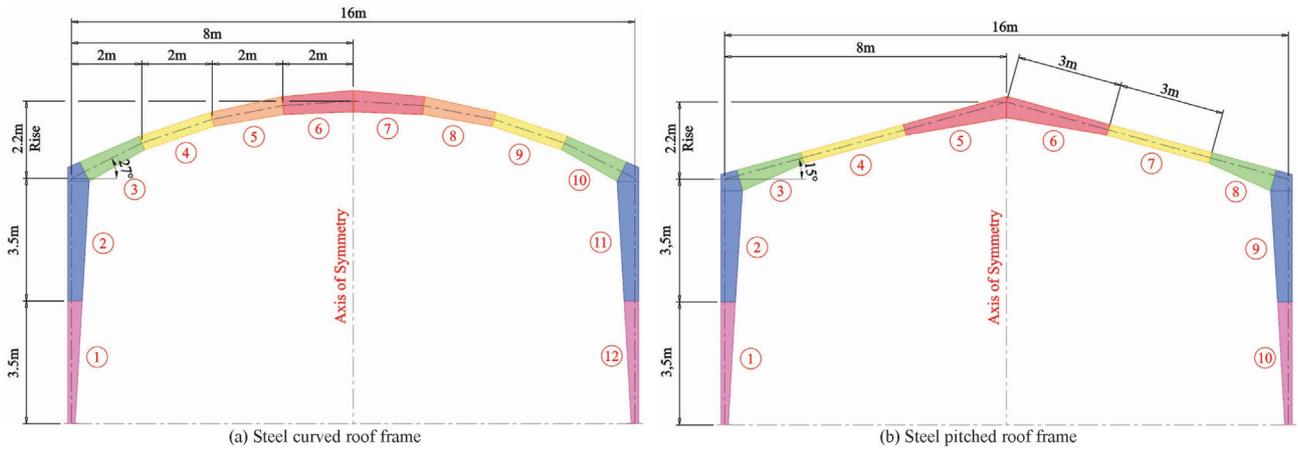


Fig. 3 Geometrical shape of the first design example with  $L = 16.0$  m

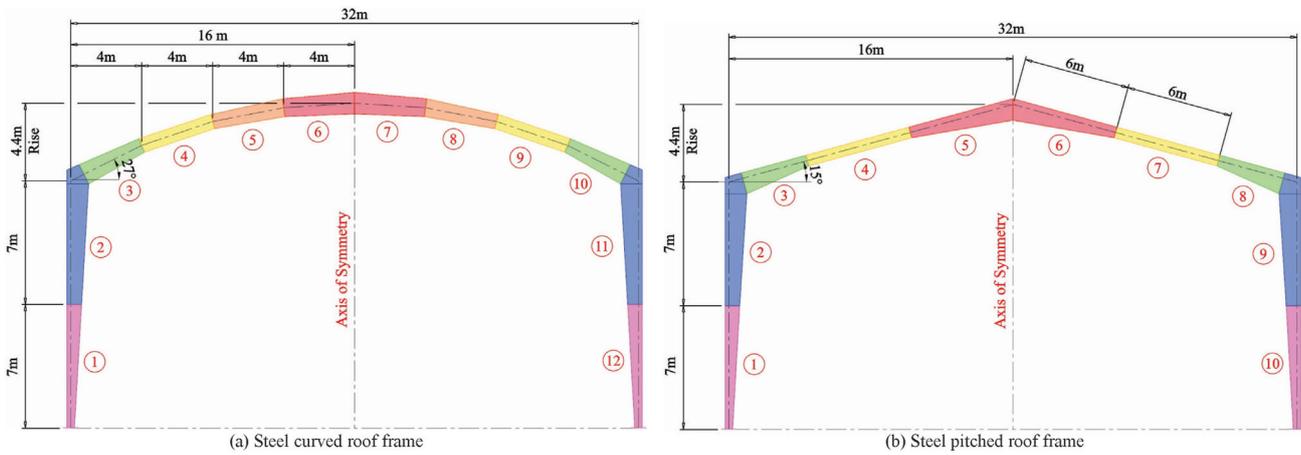


Fig. 4 Geometrical shape of the second design example with  $L = 32.0$  m

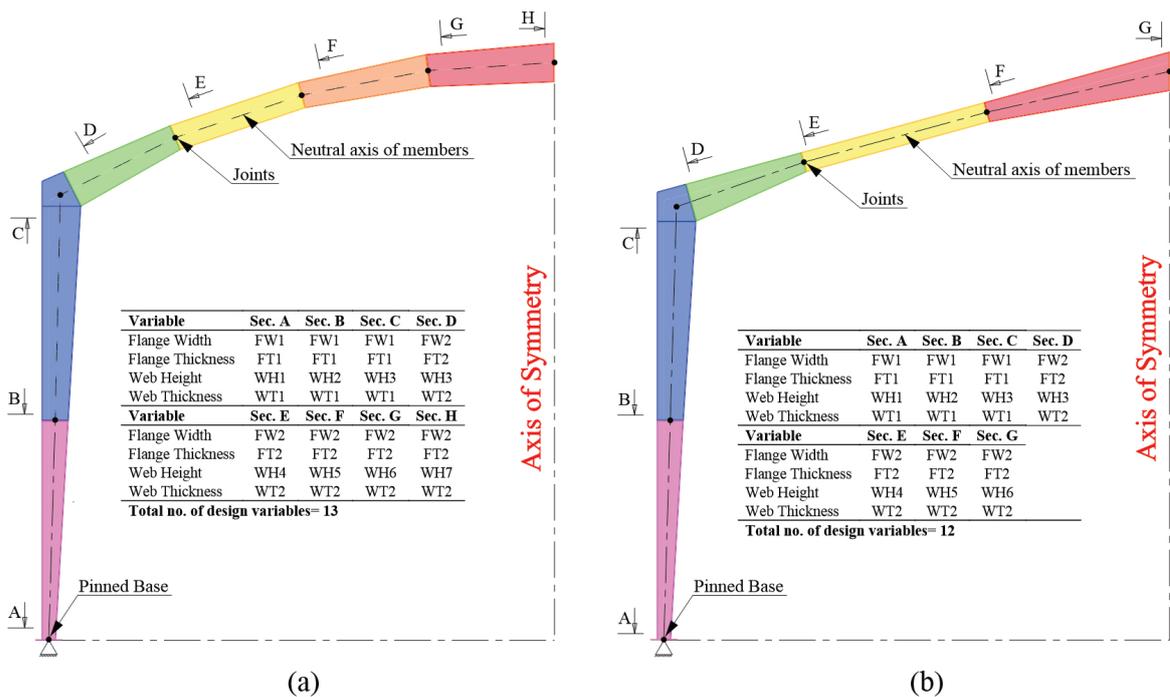


Fig. 5 The considered design variables for the optimal design of steel (a) CRF and (b) PRF

12 design variables are considered for the optimal design of steel CRFs and PRFs, respectively. The characteristics of the design variables are shown in this figure. For both investigated examples, the values for the thickness of web and flange, the web height, and flange width should be selected from the discrete set as given in Table 8.

The characteristic and number of design variables are fixed when the span length of the portal frames is increased from 16.0 m to 32.0 m. In this article, the material density, modulus of elasticity, yield stress, and poisson ratio of the all examined design examples are  $\rho = 7850 \text{ kg/m}^3$ ,  $E = 2.1 \times 10^6 \text{ kg/m}^2$ ,  $F_y = 2520 \text{ kg/m}^2$ , and  $\nu = 0.3$ , respectively. The final results of the metaheuristic algorithm in each independent run are not the same with those found in other executions. Because they are stochastic solvers. In this regard, 20 independent runs are executed for each case study of this article. The considered algorithms are coded in MATLAB environment, and the analysis and design of the examples are performed based on the existing commercial software SAP2000-OAPI.

**5.1 Discussion and results for the frames with  $L = 16.0 \text{ m}$**

Table 9 compares the optimization results acquired by TLBO, ECBO, SSOA, and WSA algorithms for the first design example. The optimization results are optimal sections and statistical measures including best, worst, mean, and standard deviation. Each algorithm was run 20 times to get statistically meaningful results. From Table 9, the mean weights attained by TLBO, ECBO, SSOA, and WSA for CRF are 1789.07 kg, 1798.14 kg, 1773.54 kg, and 1803.21 kg, respectively. As seen, the mean weight found by SSOA outperformed the mean weight obtained by other algorithms. For the PRF, these weights are 2064.78 kg, 2012.28 kg, 1993.09 kg, and 2037.9 kg, respectively. Again, this result reveals that SSOA found the mean weight lighter than other utilized algorithms. In terms of finding the best optimum weight for the CRF, ECBO is ranked first, and the best optimum weight found by SSOA is slightly inferior to the results found by ECBO. However, in PRF, ECBO, SSOA, and WSA found an equal value (i.e., 1882.39 kg) for the best optimum weight. Figs. 6 and 7 present the final

**Table 8** The values of design variables for investigated portal frames

Design examples	The first design example	The second design example
	CRF and PRF with $L = 16.0$	CRF and PRF with $L = 32.0$
Thickness of web and flange	$T = \{8, 10, 12, 15, 20, 22, 25\}$	$T = \{10, 12, 15, 20, 22, 25, 30, 35, 40\}$
Web height	$WH = \{200, 210, 220, \dots, 690, 700\}$	$WH = \{200, 210, 220, \dots, 1490, 1500\}$
Flange width	$FW = \{200, 210, 220, \dots, 490, 500\}$	$FW = \{200, 210, 220, \dots, 590, 600\}$

**Table 9** Comparison of results of different optimization methods for the first design example

Design variables	CRF				PRF			
	TLBO	ECBO	SSOA	WSA	TLBO	ECBO	SSOA	WSA
WT1	8	8	8	8	8	8	8	10
WT2	8	8	8	8	8	8	8	8
FT1	15	15	15	15	15	15	15	20
FT2	8	8	8	8	8	8	8	8
FW1	220	220	220	210	210	210	210	200
FW2	200	200	200	200	310	310	310	290
WH1	200	200	200	200	200	200	200	210
WH2	200	200	200	200	220	200	200	200
WH3	490	490	490	420	440	480	480	600
WH4	200	210	210	200	210	200	200	200
WH5	250	230	240	440	400	360	360	200
WH6	360	380	350	480	480	430	430	250
WH7	390	350	400	500	-	-	-	-
Best Weight (kg)	1672.14	1669.71	1670.99	1707.76	1901.81	1882.39	1882.39	1882.39
Worst Weight (kg)	1910.92	1913.52	1868.29	1866.49	2282.42	2281.88	2281.34	2281.85
Mean Weight (kg)	1789.07	1798.14	1773.54	1803.21	2064.78	2012.28	1993.09	2037.9
Standard Deviation	69.627	76.2658	66.727	57.9758	70.6406	87.7254	92.999	127.52

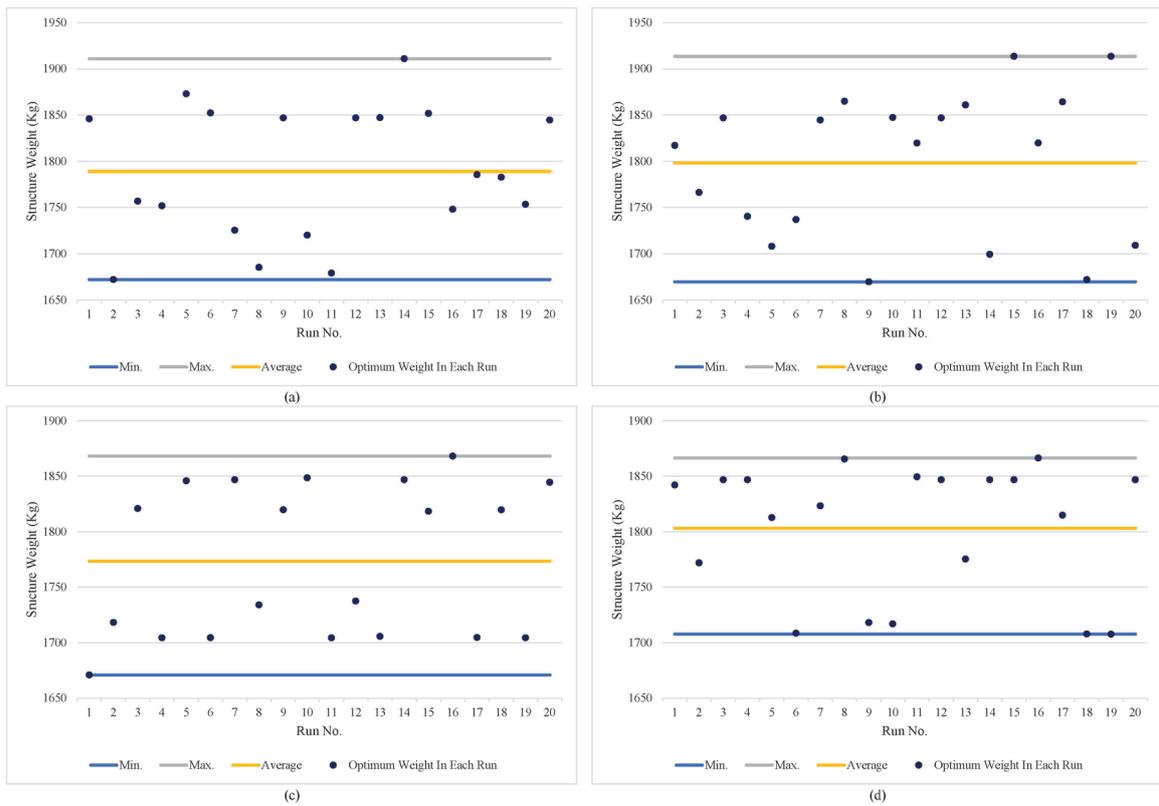


Fig. 6 The final structural weight in each independent run for the steel CRF of the first design example obtained by (a) TLBO, (b) ECBO, (c) SSOA, and (d) WSA

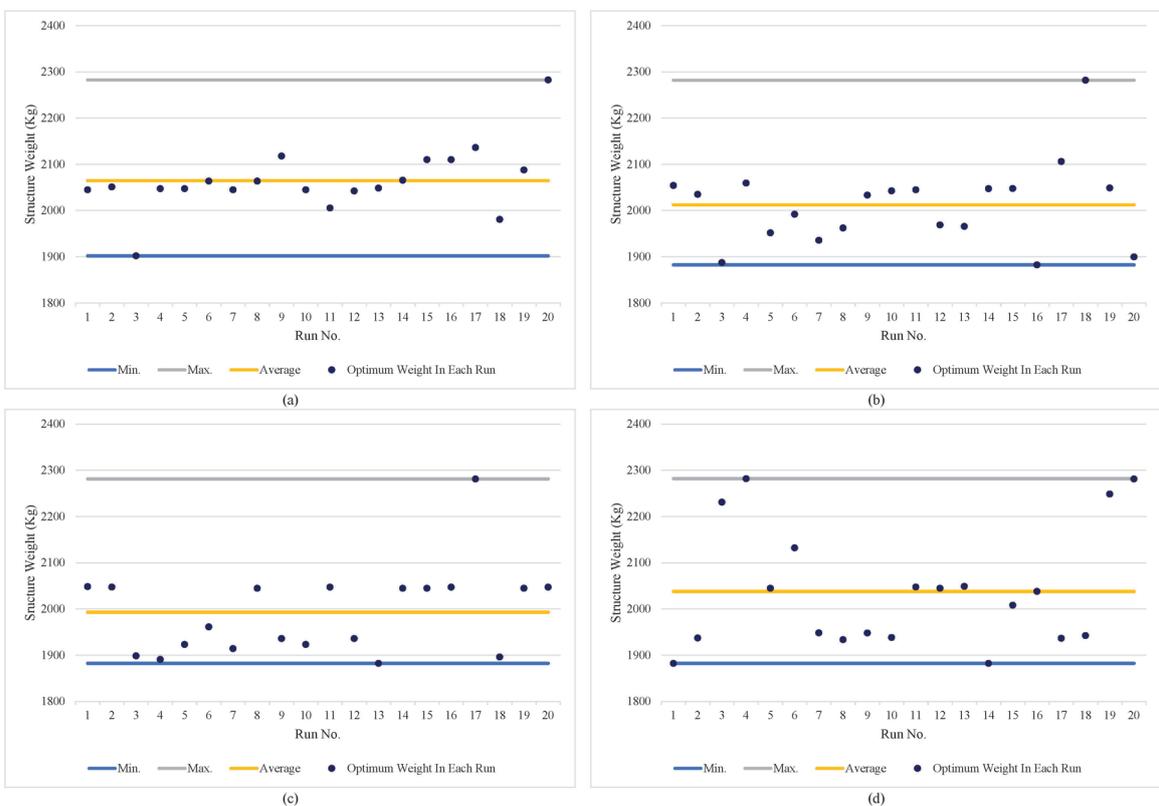
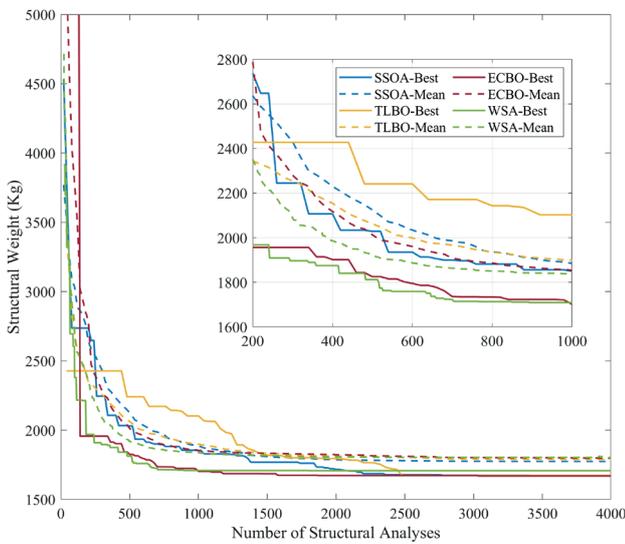
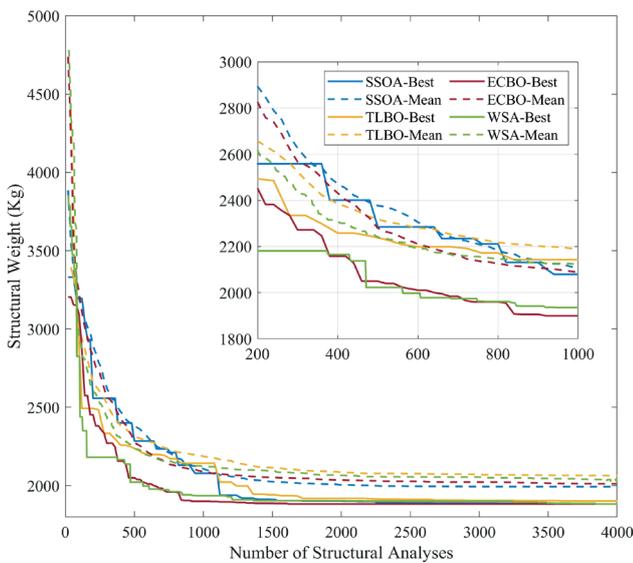


Fig. 7 The final structural weight in each independent run for the steel PRF of the first design example obtained by (a) TLBO, (b) ECBO, (c) SSOA, and (d) WSA

structural weights found by the investigated algorithms in 20 independent runs for CRF and PRF, respectively. Figs. 8 and 9 show the convergence histories of TLBO, ECBO, SSOA, and WSA for the CRF and PRF, respectively. A zoomed section is added to the convergence histories to simplify the comparison of the utilized algorithms. A close examination of these figures illustrates that the convergence speed of WSA and ECBO algorithms is considerably more than TLBO and SSOA in the early iterations. However, SSOA reached the lowest average weight



**Fig. 8** Comparison of the convergence histories from TLBO, ECBO, SSOA, and WSA metaheuristic algorithms for the steel CRF of the first design example

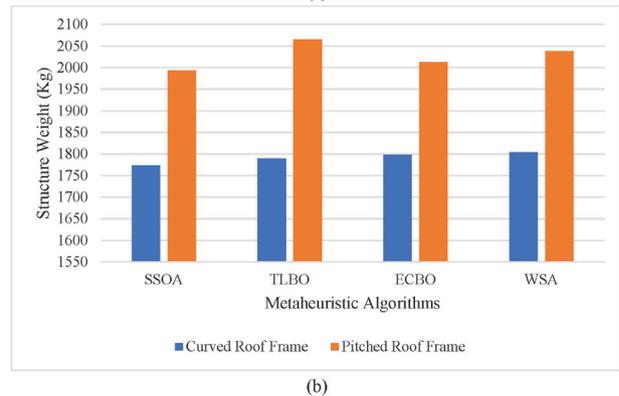
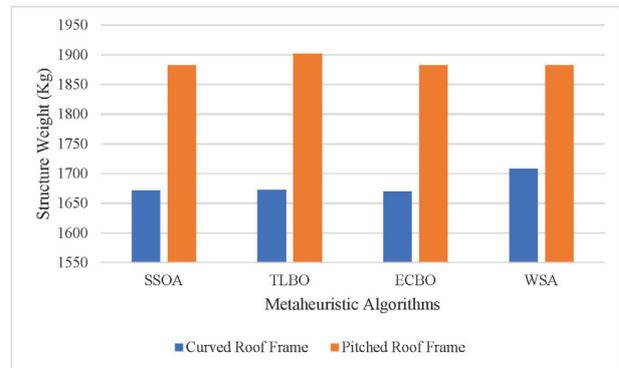


**Fig. 9** Comparison of the convergence histories from TLBO, ECBO, SSOA, and WSA metaheuristic algorithms for the steel PRF of the first design example

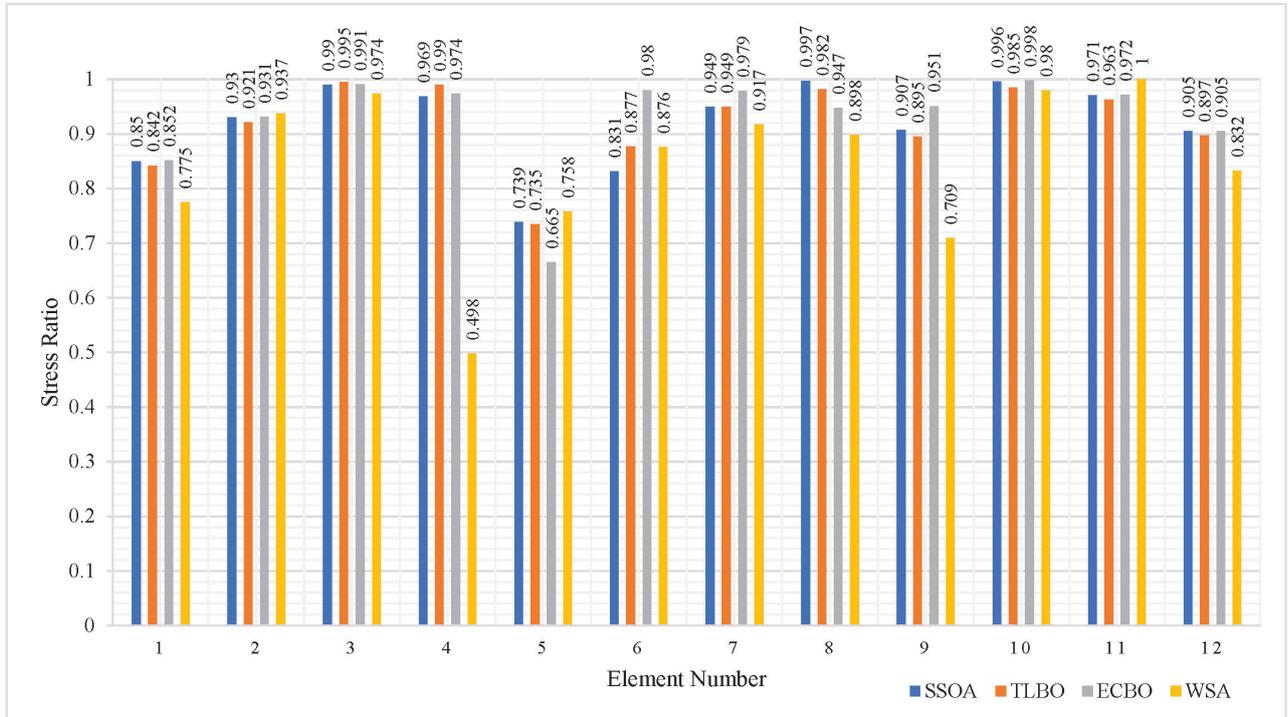
in both frames at the end iterations of the optimization process. Fig. 10 presents a comparison between the structural weight of CRF and PRF. As can be seen, the CRF with the same height and span gives a lower structural weight than PRF. Thus, the optimized results reveal that CRF is more economical than PRF. The obtained stress ratios of members of both frames for the best optimal design utilizing optimization algorithms are reported in Fig. 11.

**5.2 Discussion and results for the frames with  $L = 32.0$  m**

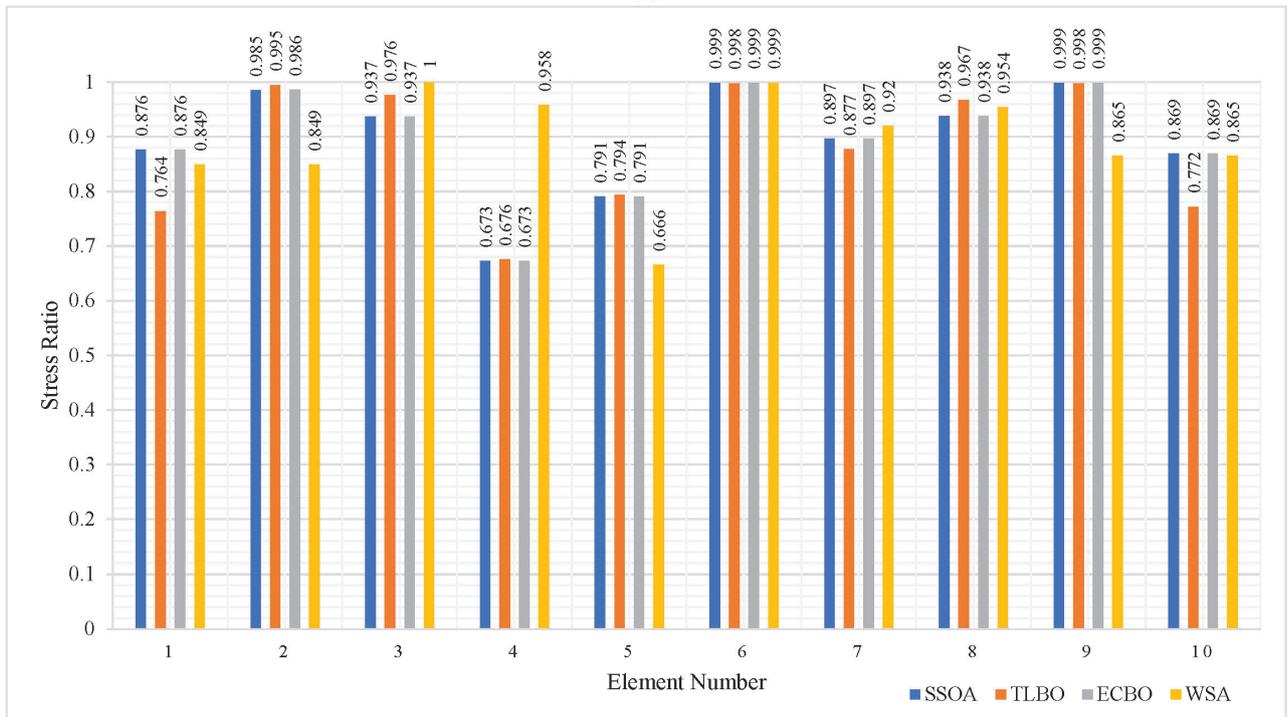
The optimized weight of the second design example acquired by TLBO, ECBO, SSOA, and WSA algorithms are compared in Table 10. As can be seen from this table, the average weights of CRF attained by TLBO, ECBO, SSOA, and WSA algorithms are respectively 7783.62 kg, 7731.3 kg, 7530.4 kg, and 7675.39 kg, while the average weights of PRF found by these algorithms are 12280.9 kg, 12182.9 kg, 11507.5 kg, and 12057.5 kg, respectively. Thus, it can be concluded that the average weight of both frames obtained by the SSOA outperformed other investigated algorithms. Moreover, the best weight of CRF is obtained by the TLBO, while the best weight of PRF is found by



**Fig. 10** Comparison of the portal frames of the first design example in terms of (a) the best structural weight and (b) the average structural weight



(a)

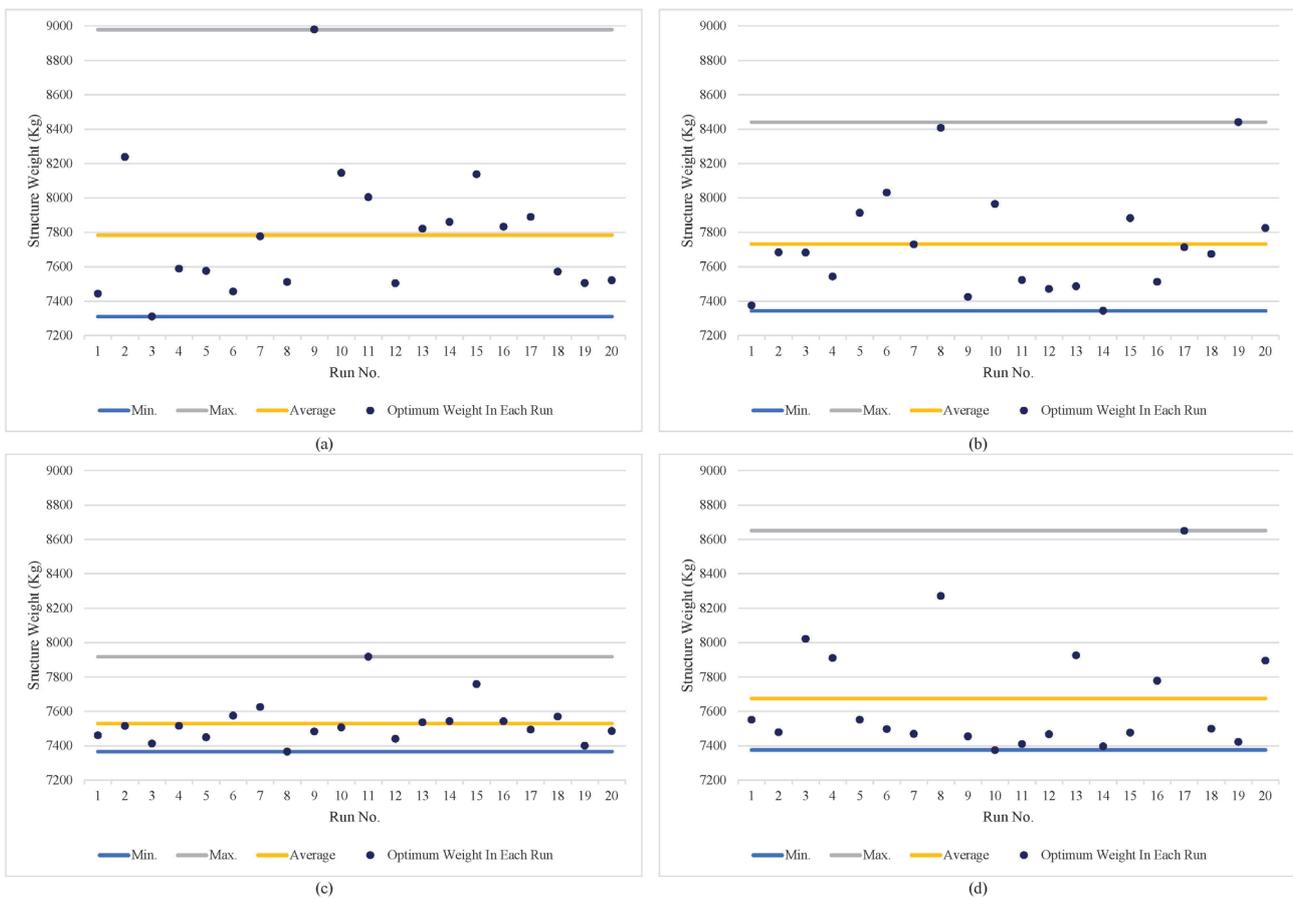


(b)

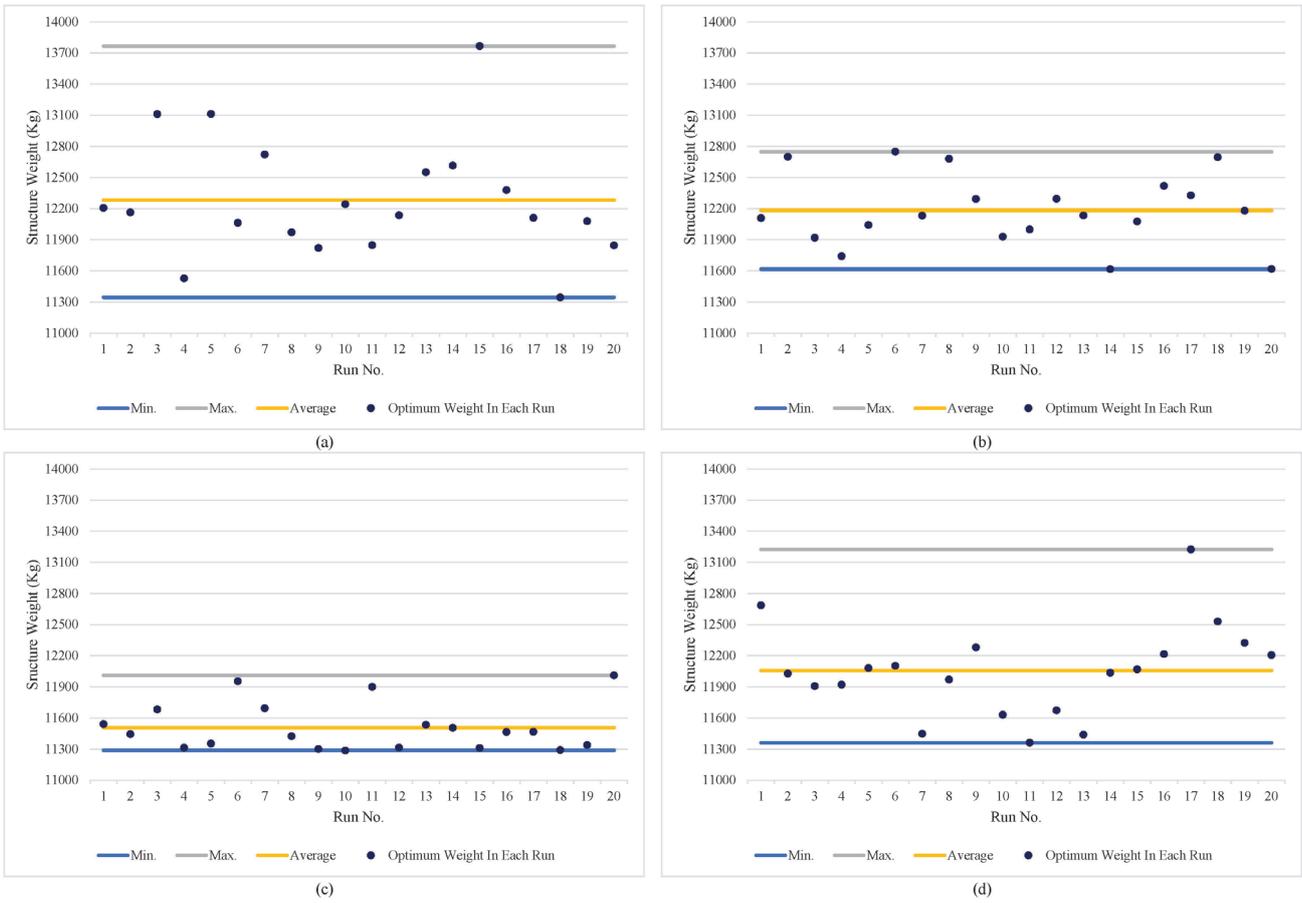
**Fig. 11** The obtained stress ratios from different optimization algorithms for the best optimum design of the first design example: (a) CRF and (b) PRF

**Table 10** Comparison of results of different optimization methods for the second design example

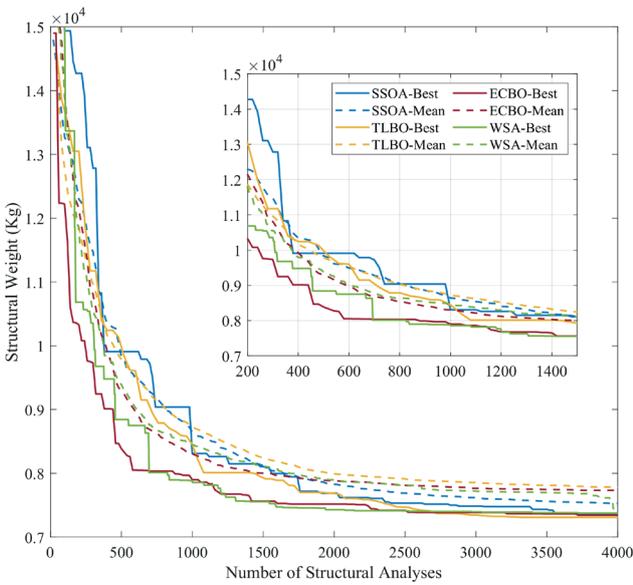
Design variables	CRF				PRF			
	TLBO	ECBO	SSOA	WSA	TLBO	ECBO	SSOA	WSA
WT1	10	10	10	10	12	20	10	15
WT2	10	10	10	10	10	10	10	10
FT1	15	15	15	15	30	30	30	30
FT2	12	12	12	12	15	15	15	15
FW1	480	480	480	470	360	350	340	360
FW2	250	250	250	230	540	510	560	560
WH1	200	200	210	220	200	200	240	200
WH2	460	440	460	500	390	330	300	290
WH3	1160	1100	1180	1290	700	1120	540	870
WH4	610	550	620	660	200	300	310	200
WH5	220	290	230	230	730	350	990	420
WH6	470	580	490	520	940	340	1210	590
WH7	560	670	580	600	-	-	-	-
Best Weight (kg)	7310.62	7343.64	7366.46	7375.23	11343.9	11616.9	11287.6	11362.9
Worst Weight (kg)	8979.15	8441.08	7918.12	8650.9	13766.7	12748.7	12011.7	13225.2
Mean Weight (kg)	7783.62	7731.3	7530.4	7675.39	12280.9	12182.9	11507.5	12057.5
Standard Deviation	376.416	300.34	122.213	332.611	559.753	334.92	222.721	434.038



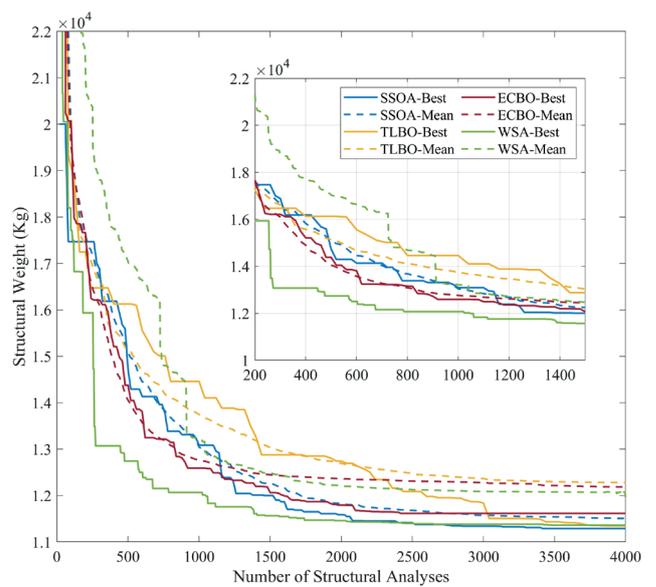
**Fig. 12** The final structural weight in each independent run for the steel CRF of the second design example obtained by (a) TLBO, (b) ECBO, (c) SSOA, and (d) WSA



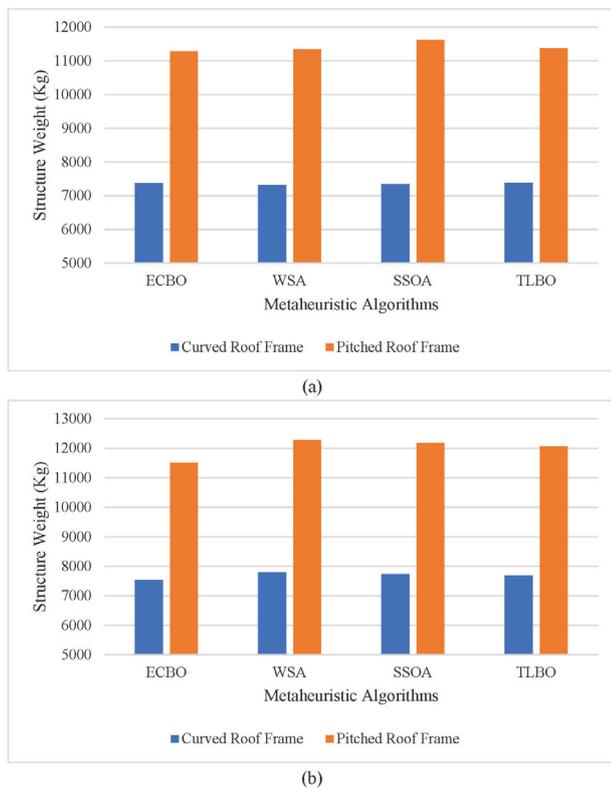
**Fig. 13** The final structural weight in each independent run for the steel PRF of the second design example obtained by (a) TLBO, (b) ECBO, (c) SSOA, and (d) WSA



**Fig. 14** Comparison of the convergence histories from TLBO, ECBO, SSOA, and WSA metaheuristic algorithms for the steel CRF of the second design example



**Fig. 15** Comparison of the convergence histories from TLBO, ECBO, SSOA, and WSA metaheuristic algorithms for the steel PRF of the second design example



**Fig. 16** Comparison of the portal frames of the second design example in terms of (a) the best structural weight and (b) the average structural weight

SSOA again. The final structural weights of steel CRF and PRF obtained by TLBO, ECBO, SSOA, and WSA algorithms in 20 independent runs are depicted in Figs. 12 and 13, respectively. Convergence curves of the utilized algorithms for the CRF and PRF are compared in Figs. 14 and 15, respectively. Like the previous design example, although the convergence rate of ECBO and WSA are more than TLBO and SSOA in the early iterations, SSOA outperforms other algorithms in terms of finding the average weight in both frames. A comparison between the structural weight of CRF and PRF is performed in Fig. 16. Again, the results reveal that the CRF is better than the PRF with the same height and span from the economic perspective. Fig. 17 reports the obtained stress ratio of members of both portal frames for the best execution of the considered algorithms.

## 6 Conclusions

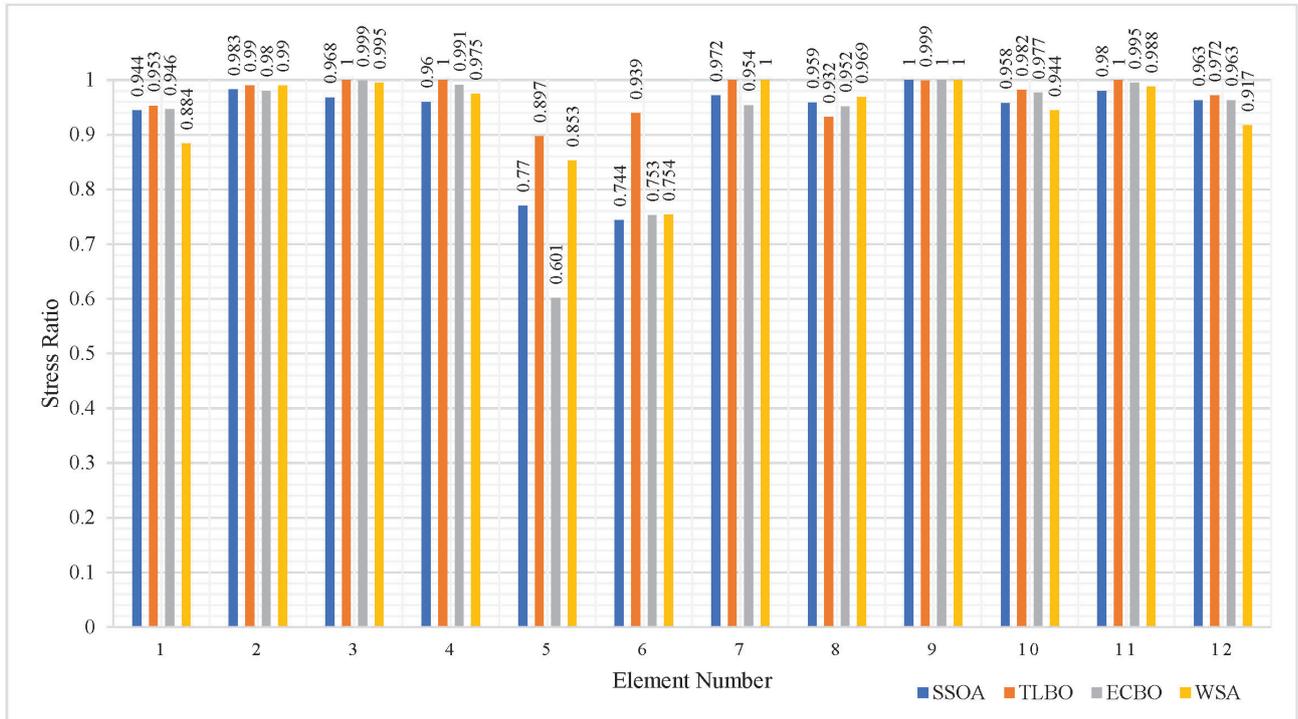
Optimal design of two types of portal frames comprising planar steel Curved Roof Frame (CRF) and Pitched Roof Frame (PRF) with tapered I-section members are studied in this article. The optimal design aims to minimize the weight of these frame structures while satisfying some design constraints based on the requirements of ANSI/AISC 360-16 and ASCE 7-10. In order to optimize the design of these types of portal frames, the existing commercial software SAP2000 and MATLAB are linked via Open Application Programming Interface (OAPI) feature. Two main objectives are examined in this paper. The first one is related to comparing the optimization results of CRF and PRF structures with the same dimensions for heights and spans. In this case, the frame structures are compared in two different span lengths, which are 16.0 m and 32.0 m. The second objective of the present research is to investigate the efficiency of four population-based metaheuristic optimization algorithms for optimal design of the CRF and PRF structures. The investigated metaheuristics are Teaching-learning-based optimization (TLBO), Enhanced Colliding Bodies Optimization (ECBO), Shuffled Shepherd Optimization Algorithm (SSOA), and Water Strider Algorithm (WSA). The obtained results enable us to draw the following conclusions:

- CRF structures are considerably more economical than PRF structures in both investigated design examples, and the optimized weights obtained in CRF structures are lower than those found in PRF structures.
- The SSOA metaheuristic is more efficient than other considered algorithms due to finding the lightest weight in the average of runs. Thus, it is highly recommended for the optimal design of these types of portal frames.

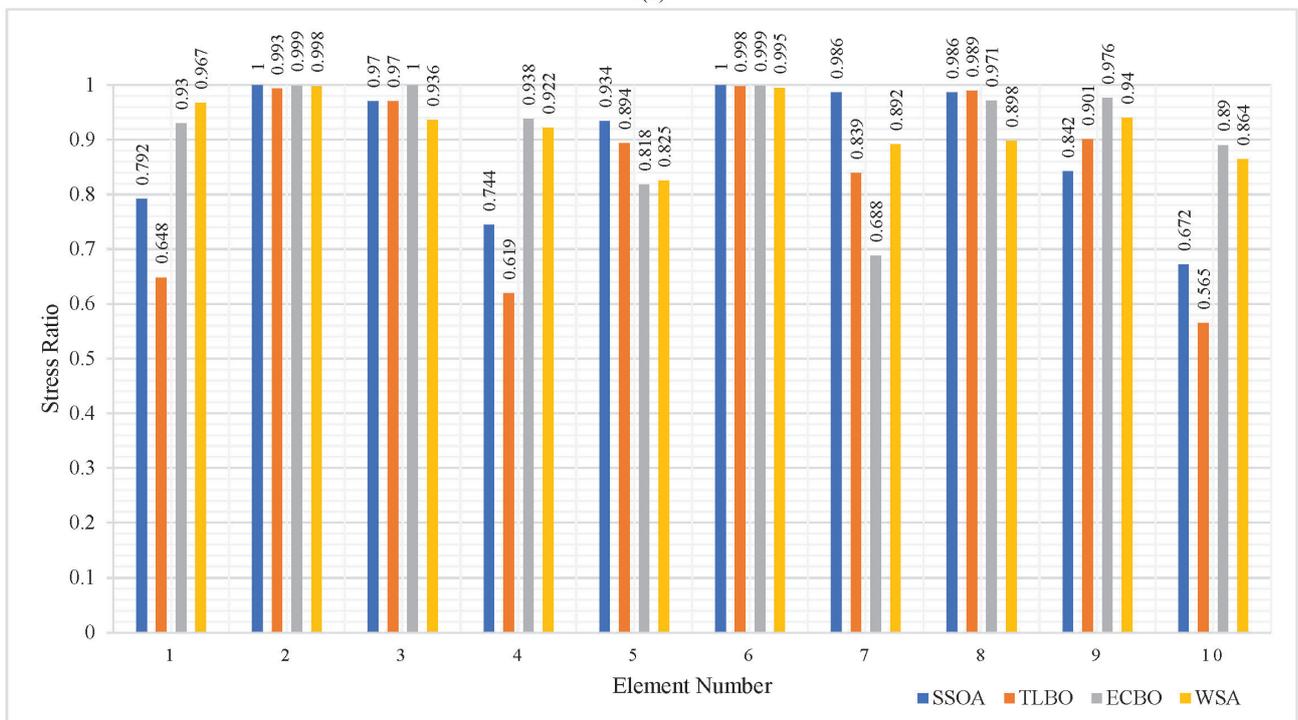
## Declaration of interests

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.

The authors declare the following financial interests/ personal relationships which may be considered as potential competing interests.



(a)



(b)

Fig. 17 The obtained stress ratios from different optimization algorithms for the best optimum design of the second design example: (a) CRF and (b) PRF

## References

- [1] Horridge, J. F., Morris, L. J. "Comparative costs of single-storey steel framed structures", *The Institution of Structural Engineers*, 7, pp. 177–181, 1986.
- [2] Fraser, D. J. "Design of tapered member portal frames", *Journal of Constructional Steel Research*, 3(3), pp. 20–26, 1983.  
[https://doi.org/10.1016/0143-974X\(83\)90003-2](https://doi.org/10.1016/0143-974X(83)90003-2)
- [3] Pintér, E., Lengyel, A., Lógó, J. "Structural Topology Optimization with Stress Constraint Considering Loading Uncertainties", *Periodica Polytechnica Civil Engineering*, 59(4), pp. 559–565, 2015.  
<https://doi.org/10.3311/PPci.8848>
- [4] Blachowski, B., Tazowski, P., Lógó, J. "Yield limited optimal topology design of elastoplastic structures", *Structural and Multi-disciplinary Optimization*, 61(5), pp. 1953–1976, 2020.  
<https://doi.org/10.1007/s00158-019-02447-9>
- [5] Tazowski, P., Blachowski, B., Lógó, J. "Topology optimization of elasto-plastic structures under reliability constraints: A first order approach", *Computers & Structures*, 243, Article number: 106406, 2021.  
<https://doi.org/10.1016/j.compstruc.2020.106406>
- [6] Lógó, J., Ismail, H. "Milestones in the 150-Year History of Topology Optimization: A Review", *Computer Assisted Methods in Engineering and Science*, 27(2–3), pp. 97–132, 2020.  
<https://doi.org/10.24423/comes.296>
- [7] Talbi, E.-G. "Metaheuristics: From Design to Implementation", John Wiley & Sons, Hoboken, NJ, USA, 2009.  
<https://doi.org/10.1002/9780470496916>
- [8] Mirjalili, S., Lewis, A. "The Whale Optimization Algorithm", *Advances in Engineering Software*, 95, pp. 51–67, 2016.  
<https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [9] Pijarski, P., Kacejko, P. "A new metaheuristic optimization method: the algorithm of the innovative gunner (AIG)", *Engineering Optimization*, 51(12), pp. 2049–2068, 2019.  
<https://doi.org/10.1080/0305215X.2019.1565282>
- [10] Anita, Yadav, A. "AEFA: Artificial electric field algorithm for global optimization", *Swarm and Evolutionary Computation*, 48, pp. 93–108, 2019.  
<https://doi.org/10.1016/j.swevo.2019.03.013>
- [11] Mohammadi, D., Abd Elaziz, M., Moghdani, R., Demir, E., Mirjalili, S. "Quantum Henry gas solubility optimization algorithm for global optimization", *Engineering with Computers*, 2021.  
<https://doi.org/10.1007/s00366-021-01347-1>
- [12] Mei, L., Wang, Q. "Structural Optimization in Civil Engineering: A Literature Review", *Buildings*, 11(2), Article number: 66, 2021.  
<https://doi.org/10.3390/buildings11020066>
- [13] Premkumar, M., Jangir, P., Sowmya, R. "MOGBO: A new Multiobjective Gradient-Based Optimizer for real-world structural optimization problems", *Knowledge-Based Systems*, 218, Article number: 106856, 2021.  
<https://doi.org/10.1016/j.knosys.2021.106856>
- [14] Alkayem, N. F., Cao, M. "Damage identification in three-dimensional structures using single-objective evolutionary algorithms and finite element model updating: evaluation and comparison", *Engineering Optimization*, 50(10), pp. 1695–1714, 2018.  
<https://doi.org/10.1080/0305215X.2017.1414206>
- [15] Beheshti Aval, S. B., Mohebian, P. "A Novel Optimization Algorithm Based on Modal Force Information for Structural Damage Identification", *International Journal of Structural Stability and Dynamics*, 21(7), Article number: 2150100, 2021.  
<https://doi.org/10.1142/S0219455421501005>
- [16] Kaveh, A., Akbari, H., Hosseini, S. M. "Plasma Generation Optimization for Optimal Design of Reinforced Concrete Cantilever Retaining Wall Structures", *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 2021.  
<https://doi.org/10.1007/s40996-020-00566-8>
- [17] Sharma, S., Saha, A. K., Lohar, G. "Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm", *Engineering with Computers*, 2021.  
<https://doi.org/10.1007/s00366-021-01294-x>
- [18] Saka, M. P. "Optimum design of pitched roof steel frames with haunched rafters by genetic algorithm", *Computers & Structures*, 81(18–19), pp. 1967–1978, 2003.  
[https://doi.org/10.1016/S0045-7949\(03\)00216-5](https://doi.org/10.1016/S0045-7949(03)00216-5)
- [19] Phan, D. T., Lim, J. B. P., Ming, C. S. Y., Tanyimboh, T., Issa, H., Sha, W. "Optimization of Cold-Formed Steel Portal Frame Topography Using Real-Coded Genetic Algorithm", *Procedia Engineering*, 14, pp. 724–733, 2011.  
<https://doi.org/10.1016/j.proeng.2011.07.092>
- [20] Phan, D. T., Lim, J. B. P., Tanyimboh, T. T., Sha, W. "An efficient genetic algorithm for the design optimization of cold-formed steel portal frame buildings", 15(5), p. 519–538, 2013.  
<https://doi.org/10.12989/scs.2013.15.5.519>
- [21] Phan, D. T., Lim, J. B. P., Tanyimboh, T. T., Wrzesien, A. M., Sha, W., Lawson, R. M. "Optimal design of cold-formed steel portal frames for stressed-skin action using genetic algorithm", *Engineering Structures*, 93, pp. 36–49, 2015.  
<https://doi.org/10.1016/j.engstruct.2015.02.037>
- [22] McKinstry, R., Lim, J. B. P., Tanyimboh, T. T., Phan, D. T., Sha, W. "Optimal design of long-span steel portal frames using fabricated beams", *Journal of Constructional Steel Research*, 104, pp. 104–114, 2015.  
<https://doi.org/10.1016/j.jcsr.2014.10.010>
- [23] Kaveh, A., Mahdavi, V. R., Kamalinejad, M. "Optimal design of pitched roof frames with tapered members using ECBO algorithm", *Smart Structures and Systems*, 19(6), pp. 643–652, 2017.  
<https://doi.org/10.12989/sss.2017.19.6.643>
- [24] Kaveh, A., Hoseini Vaez, S. R., Hosseini, P., Bakhtyari, M. "Optimal Design of Steel Curved Roof Frames by Enhanced Vibrating Particles System Algorithm", *Periodica Polytechnica Civil Engineering*, 63(4), pp. 947–960, 2019.  
<https://doi.org/10.3311/PPci.14812>
- [25] Rao, R. V., Savsani, V. J., Vakharia, D. P. "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, 43(3), pp. 303–315, 2011.  
<https://doi.org/10.1016/j.cad.2010.12.015>
- [26] Kaveh, A., Ilchi Ghazaan, M. "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", *Advances in Engineering Software*, 77, pp. 66–75, 2014.  
<https://doi.org/10.1016/j.advengsoft.2014.08.003>

- [27] Kaveh, A., Zaerreza, A. "Shuffled shepherd optimization method: a new Meta-heuristic algorithm", *Engineering Computations*, 37(7), pp. 2357–2389, 2020.  
<https://doi.org/10.1108/EC-10-2019-0481>
- [28] Kaveh, A., Dadras Eslamlou, A. "Water strider algorithm: A new metaheuristic and applications", *Structures*, 25, pp. 520–541, 2020.  
<https://doi.org/10.1016/j.istruc.2020.03.033>
- [29] AISC "ANSI/AISC 360-16 Specification for Structural Steel Buildings", American Institute of Steel Construction, Chicago, IL, USA, 2016.
- [30] AISC "ANSI/AISC 341-16 Seismic Provisions for Structural Steel Buildings", American Institute of Steel Construction, Chicago, IL, USA, 2016.
- [31] ASCE/SEI "ASCE/SEI 7-10 Minimum Design Loads for Buildings and Other Structures", American Institute of Steel Construction, Chicago, IL, USA, 2010.