Structural and Multidisciplinary Optimization (2023) 66:13
https://doi.org/10.1007/s00158-022-03435-2

RESEARCH PAPER

Variable functioning and its application to large scale steel frame design optimization

Amir H. Gandomi[1] · Kalyanmoy Deb[2] · Ronald C. Averill[3] · Shahryar Rahnamayan[4] · Mohammad Nabi Omidvar[5]

Received: 26 July 2021 / Revised: 17 September 2022 / Accepted: 7 October 2022 / Published online: 27 December 2022

Abstract
To solve complex real-world problems, heuristics and concept-based approaches can be used to incorporate information into the problem. In this study, a concept-based approach called variable functioning (Fx) is introduced to reduce the optimization variables and narrow down the search space. In this method, the relationships among one or more subsets of variables are defined with functions using information prior to optimization; thus, the function variables are optimized instead of modifying the variables in the search process. By using the problem structure analysis technique and engineering expert knowledge, the Fx method is used to enhance the steel frame design optimization process as a complex real-world problem. Herein, the proposed approach was coupled with particle swarm optimization and differential evolution algorithms then applied for three case studies. The algorithms are applied to optimize the case studies by considering the relationships among column cross-section areas. The results show that Fx can significantly improve both the convergence rate and the final design of a frame structure, even if it is only used for seeding.

Keywords Engineering optimization · Problem structure · Gray-box optimization · Variable interaction analysis · Evolutionary computation

1 Introduction

Real-world optimization problems are often complex and difficult to solve due to factors like dimensionality, nonlinearity, and existence of complex constraints. To deal with this level of complexity, many approaches, such as dimensionality reduction, approximation, and problem decomposition, are common. Another way of dealing with complex problems is incorporating expert knowledge into the optimization process. For example, in the case of problem decomposition, knowledge of problem structure is needed, which might be known *a priori* or might be discovered automatically with relevant analysis techniques (Mei et al. 2016).

Recently, gray-box optimization has been coined to refer to the optimization process of problems for which the structure is known (Santana 2017). This is in contrast to black-box optimization, where zero knowledge of the problem is assumed. Although this might be a reasonable assumption for simulation type problems, it is not a realistic assumption for a wide range of optimization problems. For example, the use of a deterministic crossover operator, which respects the problem structure, allows optimization of cast scheduling problems with up to a billion variables (Deb and Myburgh 2017). Similar approaches have been used successfully with Traveling Salesman Problems (Whitley et al. 2010) and pseudo-boolean problems (Tintos et al. 2015).

In the context of evolutionary algorithms, two general approaches have been suggested to adapt an algorithm to the known characteristics of a given problem (De Jong 1988): 1. To change the representation of the problem such that

Responsible Editor: Palaniappan Ramu

✉ Amir H. Gandomi
gandomi@uts.edu.au

[1] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, Australia

[2] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

[3] Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA

[4] Department of Electrical, Computer and Software Engineering, University of Ontario Institute of Technology, Oshawa, ON, Canada

[5] School of Computing and Leeds University Business School, University of Leeds, Leeds, UK

the traditional variation operators remain applicable; and 2. To devise new variation operators to work with the original representation of the problem perceived as “natural”. In this work, we used the former approach to solve steel frame design optimization problems.

Frame design is one of the most popular optimization problems in structural engineering (Saka 2007). It is considered to be a complex problem considering the involvement of elaborate finite element models and existence of several mechanical and geometric constraints relating to maximum and minimum stress, buckling, story and roof drifts. Furthermore, the existence of discrete or mixed-type variables, due to the need for predefined cross-sections in the frame construction industry, also adds to its complexity.

Due to the wide practical applications and versatility of frame design optimization, an effective method of solving such problems can significantly reduce the construction cost. As a result, many researchers have attempted to optimize frame structure design as a complex, discrete problem using a variety of methods (Lamberti and Pappalettere 2011), including non-deterministic and stochastic algorithms (Hasançebi et al. 2010; Azad and Hasançebi 2015). The objective of frame design is to minimize the frame weight (relating to cost) subject to complex nonlinear constraints. For the steel frame structures, the design variables usually include the cross-sections of the beams and columns. In practice, these members must be chosen from a standardized set of cross-sections, which makes the problem discrete.

In this paper, we propose a method called variable functioning (*Fx*) to change the problem representation with the aim of controlling its complexity and taming the curse of dimensionality by reducing the number of decision variables. In this method, the structural information of the problem is extracted using state-of-the-art variable interaction analysis methods from which functional mapping is performed to map the input space to a lower dimensional space based on the identified interaction patterns of original input variables. For the purposes of this paper, we adopted differential grouping (DG2) (Omidvar et al. 2017) to find the problem structure in the form of nonlinear relationships among the decision variables. This information can be visualized with heat-maps and/or variable interaction graphs to give further insight about the nature of the problem. The variable interaction information contains interesting patterns, which are otherwise hard to discover even by the experts. These patterns are then used by the experts to devise a functional map that transforms the original complex problem into a simpler problem with fewer decision variables. Once the problem is reformulated, the resulting transformed problem can be optimized using any suitable optimizer.

Additional information can be embedded in an optimization process in a heuristic way to simplify or improve the process. Moreover, the heuristics can be very effective if they are applied correctly. There are some existing heuristics in the frame design optimization process, such as considering the symmetry of a problem to decrease the number of variables and number of finite elements in the model (Talatahari et al. 2015). Additionally, fabrication conditions should be imposed on the construction of structural elements. For instance, the same beam/column cross-section should be used for *N* consecutive stories, resulting in a reduction in the number of problem variables (Talatahari et al. 2015). However, unlike variable functions proposed in this paper, these methods are problem-specific and cannot be generalized to a wider range of problems. Variable functioning, on the other hand, is based on automatic variable interaction analysis, which can be applied to a wide range of problems. It is worth mentioning that the proposed method is compatible with these ad hoc techniques and can be considered complementary.

The proposed approach is simple and can be coupled with any optimization algorithm, such as particle swarm optimization (PSO) and differential evolution (DE) algorithms. The proposed method is explained along with an illustrative example and then applied to three steel frame design optimization problems. The results show that the proposed approach can significantly improve both the convergence rate and the final solution of frame design optimization problems, even if it is only used in the initialization step and not through entire search optimization process.

The rest of this study is organized as follows. Section 2 presents the formulation of the steel frame design problem along with the optimization algorithms and constraint handling schemes used in this study. The proposed variable functioning approach and its application to steel frame design problem are explained in Sect. 3 by means of an illustrative example. Section 4 is devoted to the three case studies used for benchmarking. Finally, conclusions, discussions, and future insights are presented in Sect. 5.

2 Steel frame design optimization process

2.1 Problem formulation

The main variables in a steel frame design optimization problem are usually member sections, which are grouped based on the fabrication conditions and symmetry of the structure. Therefore, the optimization variables in a steel frame structure are the cross-sections of each group, as $\mathbf{x} = (x_1, \ldots, x_{n_g})$, where n_g is the number of member groups. As it is mentioned, the objective of frame design problems is usually the minimization of the frame weight (W), which can be formulated as:

$$\underset{\mathbf{x}\in\Omega}{\text{minimize}}\;\; W(\mathbf{x}) = \sum_{i=1}^{n_g} \rho \left(\sum_{j=1}^{n_m} L_{i,j} \right) x_i, \tag{1}$$

where ρ is the material density; $L_{i,j}$ is the length of jth element in the ith group; ng is the number of groups; n_m is the number of members in the ith group; and Ω is the search space for the variables (cross-sections). Note that although other variables can be considered for steel frame design optimization, this study only considers cross-sections as design variables. Even though the weight of the steel frames is considered as the objective function in this work, a more detailed optimization approach could take into account the cost-efficiency of the design examples. Despite the fact that the main concern of the current study is reducing the search space, it can also be readily applied to problems where minimizing the cost is considered the objective (Pavlovčič et al. 2004). In practice, cross-sections must be chosen from a predefined and standardized set of cross-sections (e.g., W-shapes), which makes the problem discrete. This problem is generally subject to stress constraints, maximum lateral displacement, and inter-story displacement constraints (Talatahari et al. 2015). The stress constraints can be formulated as follows:

$$v_i^\sigma = \left| \frac{\sigma_i}{\sigma_i^a} \right| - 1 \geq 0, \forall i \in \{1, \ldots, n_m\} \tag{2}$$

where σ_i and σ_i^a are the maximum stress and allowable stress in the ith member, respectively. Therefore, the number of stress constraints is equal to the number of members (elements) in the problem. The maximum lateral displacement and inter-story displacement constraints can be respectively formulated as follows:

$$v^\Delta = \frac{\Delta_T}{H} - R \geq 0, \tag{3}$$

$$v_j^d = \frac{d_j}{h_j} - R_I \geq 0, \tag{4}$$

where Δ_T is the maximum lateral displacement; H is the height of the frame structure; R is the maximum drift index; d_j is the inter-story drift; h_j is the height of the jth floor; ns is the total number of stories; and R_I is the inter-story drift index permitted by the standard design code in engineering practice. The allowed inter-story drift index is taken as 1/300, based on the American Institute of Steel Construction (AISC) design code (AISC 2011). For the Load and Resistance Factor Design (LRFD), the interaction formula constraints (AISC 2001, Equation H1-1a,b) are formulated as follow:

$$v_i^I = \begin{cases} \dfrac{P_u}{2\phi_c P_n} + \left(\dfrac{M_{ux}}{\phi_b M_{nx}} + \dfrac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0, & \dfrac{P_u}{\phi_c P_n} < 0.2 \\ \dfrac{P_u}{\phi_c P_n} + \dfrac{8}{9}\left(\dfrac{M_{ux}}{\phi_b M_{nx}} + \dfrac{M_{uy}}{\phi_b M_{ny}} \right) - 1 \leq 0, & \dfrac{P_u}{\phi_c P_n} \geq 0.2, \end{cases} \tag{5}$$

where P_u is the required strength (tension or compression); P_n is the nominal axial strength (tension or compression); ϕ_c is the resistance factor ($\phi_c = 0.9$ for tension, $\phi_c = 0.85$ for compression); M_{ux} and M_{uy} are the required flexural strengths in the x and y directions, respectively; M_{nx} and M_{ny} are the nominal flexural strengths in the x and y directions (for two-dimensional structures, $M_{ny} = 0$), respectively; and ϕ_b is the flexural resistance reduction factor ($\phi_b = 0.9$). The effective length factors of members (K) are required to compute the allowable compression and Euler buckling stresses.

2.2 Algorithms

Metaheuristics are the global and stochastic optimization algorithms generally inspired by nature. Based on the sources of information, metaheuristics can be divided into two classes: swarm intelligence (Slowik and Kwasnicka 2018) algorithms that mimic a swarm behavior, and evolutionary computation (Eiben and Smith 2015) algorithms that use evolutionary mechanism, such as crossover, mutation, and selection. Both classes of metaheuristics have been widely used for simulation optimization (such as FE analysis). Also, they have been successfully applied in complex frame design optimization problems (Saka 2007; Gholizadeh and Poorhoseini 2016; Ghasemi and Farshchin 2011).

In this study, the particle swarm optimization (PSO) algorithm is used as a classical swarm intelligence, and a differential evolution (DE) algorithm is applied for frame optimization as an acclaimed evolutionary algorithm to couple with the proposed variable functioning approach. It should be noted that finding the best algorithm(s) is not the purpose of this study and these two algorithms have been used to represent two classes of metaheuristics.

The PSO algorithm, initially suggested by Kennedy (1995), mimics the social behavior of bird flocks and fish schools. As one of the best-established swarm intelligence algorithms, PSO has been applied to many structural optimization problems. Since it is population-based, particles forage the search space to find the best solutions. DE Storn and Price (1997) is a population-based evolutionary optimization algorithm, which uses three operators (selection, mutation, and crossover) to lead the solution toward the global optimum. PSO and DE algorithms were used in this study for steel frame design optimization. Note that this study does not propose a new optimization algorithm. Instead, the aim of this study is to show how the proposed method works with global optimization algorithms to reduce the search space, or more specifically the design space of steel frame design.

2.3 Constraint handling

The constraint handling is based on feasibility rules proposed by Deb (2000) as follows:

1. If both solutions are feasible, the one with the better objective function value is preferred.
2. A feasible solution is preferred to an infeasible one.
3. If both solutions are infeasible, the one with a smaller amount of constraint violation is preferable. The amount of constraint violation is also normalized according to Becerra and Coello (2006):

$$G(\mathbf{x}) = \sum_{i=1}^{n_c} \frac{g_i(\mathbf{x})}{g_{\max,i}}, \tag{6}$$

where n_c is the number of constraints; $g_i(\mathbf{x})$ is the ith constraint of the problem; and $g_{\max\ i}$ is the largest violation of the ith constraint so far. Based on the rules, a fitness function is proposed by Deb (2000) to penalize the solutions that violate the constraint(s). The penalized fitness function is formulated as follows:

$$f_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } G(\mathbf{x}) \geq 0 \\ f_{\max} + G(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{7}$$

where $f_{\max}$ is the objective function value of the worst feasible solution in the population. Using this strategy, infeasible solutions are only compared based on their normalized constraint violation. Note that other constraint handling schemes, such as the Automatic Dynamic Penalisation method (e.g., Montemurro et al. (2013)), which may perform better but needs to be evaluated since it depends on the optimization algorithm, problem as well as the implementation details of the proposed approach. In both DE and PSO algorithms, solutions may go out of the defined boundaries. In DE, if a component of a solution violates either upper or lower bounds, it is returned to the violated bound. In PSO, however, if the solution is returned to the boundary, it has a high probability of violating the boundary again. This is because the inertia of the previous motion contributes to the current motion. Therefore, when a particle component violates a boundary, its related velocity component (**V**) is reversed to return the solution to the boundary and also overcome the problem. This boundary constrained handling can be formulated as follows:

$$v_{i,j}^{(t+1)} = \begin{cases} v_{i,j}^{(t+1)} & \text{feasible} \\ -v_{i,j}^{(t+1)} & \text{infeasible.} \end{cases} \tag{8}$$

3 Variable functioning

3.1 Methodology

Most real-world problems are complex and often include with black/gray box models. Optimization of such problems is difficult because the systems typically lack much information. In such cases, embedding information and knowledge can be very helpful to boost the optimization process. This information can be obtained from different sources, such as statistical tests, engineers' knowledge, expert systems, etc.

Such knowledge can be utilized as additional information even before the search process and can be adapted to significantly improve the optimization process. In this section, an approach is proposed and explained for incorporating information from different sources to narrow down the search space and potentially reduce the problem dimension. The next section particularly focuses on applying the proposed approach on a steel frame design optimization problem.

Here, engineering domain knowledge is embedded into the steel frame optimization process and can also be used in similar problems.

In an optimization problem, one (or more) objective function(s) should be optimized with respect to some variable vector $\mathbf{x} = (x_1, \dots, x_n)$, which can be simply formulated as:

$$\underset{\mathbf{x}\in\Omega}{\text{Optimize}}\, f(x_1, \dots, x_n). \tag{9}$$

Considering the relationship among a set of variables, a general relationship could be defined using a mathematical function $\zeta : \mathbb{R}^m \to \mathbb{R}^q$. For example, if the first q variables, $(x_1, \dots, x_q)$, have a functional relationship that can be expressed with ζ, they can be replaced by a function with m variables $\mathbf{y} = (y_1, \dots, y_m)$. Now, the optimization problem can be represented as:

$$\underset{\mathbf{x}\in\Omega_x \wedge \mathbf{y}\in\Omega_y}{\text{optimize}}\ f(\zeta(\mathbf{y}), x_{q+1}, \dots, x_n), \tag{10}$$

or simply

$$\underset{\mathbf{x}\in\Omega_x \wedge \mathbf{y}\in\Omega_y}{\text{optimize}}\ \hat{f}(y_1, \dots, y_m, x_{q+1}, \dots, x_n). \tag{11}$$

where m is the number of variables in the functioning relationship (ζ). Using the new formulation ($\hat{f}$), the optimization results always satisfy the defined relationship for the first q variables. Now, if $m < q$, the problem dimension is also reduced, which reduces the search space and thus simplifies the problem. If more than one set of variables has functioning relationships, the generalized form of the approach for any number of functions can be formulated as follow:

$$\underset{\mathbf{x}\in\Omega_{\mathbf{x}}\wedge\mathbf{y}\in\Omega_{\mathbf{y}}}{\text{optimize}}\, f(\zeta_1(\mathbf{y}_1),\ldots,\zeta_s(\mathbf{y}_s),x_{q+1},\ldots,x_n), \tag{12}$$

where s is the number of functioning relationships and each functioning relationship $\zeta_i(\mathbf{y}_i) : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{q_i}$. From the general formulation, $\sum_{j=1}^{s} m_j < \sum_{j=1}^{s} q_j$ indicates the problem dimension is reduced and is equal to their difference, $\sum_{j=1}^{s} (q_j - m_j)$. The proposed approach can be coupled with any generalized optimization algorithm, as it only alters optimization problem formulation.

Several approaches have been proposed recently to group decision variables of a black-box problem, such as adaptive monotonicity checking (Munetomo and Goldberg 1999; Chen et al. 2010), variable partitioning (Ray and Yao 2009), and min/max variance decomposition (Liu and Tang 2013). However, all these methods still have a low grouping accuracy. This drawback has resulted in several recent studies on variable interaction learning (Chen et al. 2010), statistical learning decomposition (Sun et al. 2012), and meta-modeling decomposition (Mahdavi et al. 2014). For instance, (Omidvar et al. 2014) proposed differential grouping (DG) to determine the nonseparable groups. This method was later extended as DG2 (Omidvar et al. 2017), which has shown superior performance concerning grouping accuracy methods and does not have any parameters to tune (Omidvar et al. 2017).

3.2 Variable functioning for steel frame structures

Structural engineering knowledge and concepts are typically used for the formulation of structural optimization problems. For instance, to consider the fabrication conditions in steel frame optimization problems, the beam and column sections are grouped into two/three consecutive stories. In this section, we aim to find ways to embed information and knowledge to find relationship between column cross-sections as the main variable in steel frame design optimization. A single stepped column under a lateral load is shown in Fig. 1. The objective of this problem is minimization of the column weight to satisfy the maximum stress constraint in each segment, and the variables are the radii of the cross-sections.

One way of finding variable interactions in a black box model is performing the differential grouping test (Omidvar et al. 2014). This method was recently improved by Omidvar et al. (2017), named DG2, as mentioned above. Herein, the DG2 method was used to find interaction between variables, and the results for a column with 50 segments is visualized in Fig. 2. From this heat map, it is clear that variables have high dependency on the first variable (cross-section of the base segment).

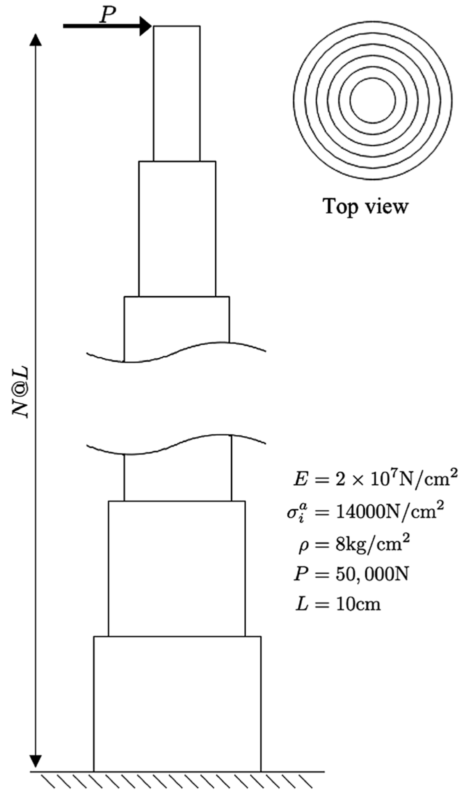

P
Top view
$N@L$
$E = 2 \times 10^7 \mathrm{N/cm}^2$
$\sigma_i^a = 14000 \mathrm{N/cm}^2$
$\rho = 8 \mathrm{kg/cm}^2$
$P = 50{,}000 \mathrm{N}$
$L = 10 \mathrm{cm}$

Fig. 1 Stepped column design problem with circular cross-sections under lateral load

From an engineering point-of-view, there are several reasons why the cross-section of a floor should be less than or equal to a cross-section in the lower floor. The first reason concerns the physics of the problem, whereby each column cross-section carries the cumulative axial loads above it. Additionally, installing a cross-section on a smaller cross-section in a column is not practical and notably difficult, as it cannot transfer the load correctly. Therefore, the cross-section area should be monotonically decreased as the story number increases. In other words, the story height and column cross-section area have an inverse relationship.

The following points are considered to relate the cross-section area of a column by means of their heights:

- dependencies of all cross-sections to the first one (based on the DG2 test results illustrated in Fig. 2); and
- monotonically decreasing column cross-section area with increasing heights (based on engineering points of view);

The respective function is defined as:

$$A(h) = \frac{A(0)}{\alpha^h} \tag{13}$$

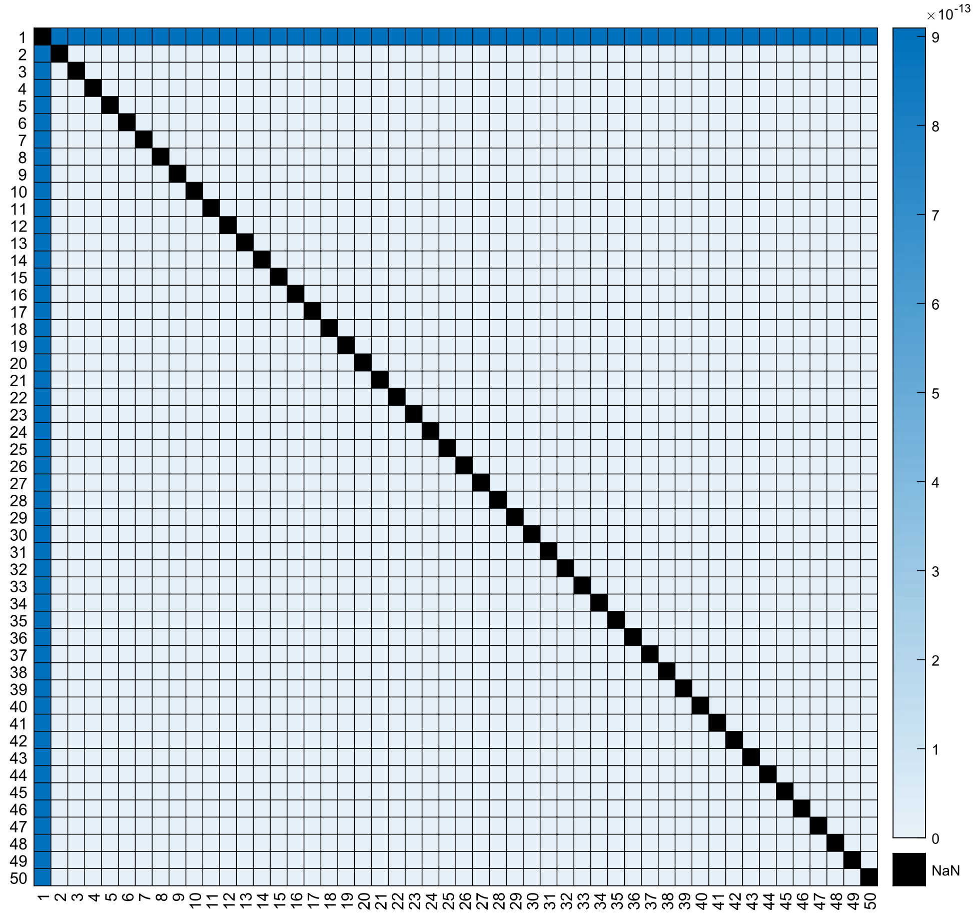


Fig. 2 Interaction matrix

where $A(0)$ is the cross-section area of the base segment; α is a new variable; h is the height of a cross-section from the base (or ground); and $A(h)$ is the cross-section area of the column located at the height of h. For the base cross-section, $h = 0$, $\alpha = 1$, and from (13), $A(h) = A(0)$. In this equation, all the variables (cross-section areas) are related to the first variable by the new variable, α, and cross-section height. Therefore, all of a column's cross-section areas can be related to each other using a function with just two variables: the cross-section area of the base column, $A(0)$, and α. To monotonically reduce the cress-section areas while increasing the height, α should be greater than 1. Furthermore, the largest value of α is obtained when the lowest possible cross-section area is located at the end of the column (highest cross-section), which can be calculated from the following equation:

$$\alpha_{\max} = \sqrt[h_u]{\frac{A_{\max}}{A_{\min}}}, \tag{14}$$

where h_u is the height of the uppermost column section from the ground; and $A_{\min}$ and $A_{\max}$ are the smallest and largest cross-section areas among the predefined sections, respectively. Based on the above discussion, the boundaries of the new variable are $\Omega_\alpha : \alpha \in [1, \alpha_{\max}]$. Using this approach, all the variables (cross-section areas) above the base are replaced with α in the new formulation. It should be noted that this function can be used for different sets of columns. In other words, a problem with s columns and n cross-sections, $f(A_1, \ldots, A_n)$, is replaced with $f(A^1(h), \ldots, A^s(h), A_{i+1}, \ldots, A_n)$ after defining s functions. Note that other rules may apply for other problems like weak beam-strong column rule for the seismic regions that are not low-risk.

3.3 Illustrative example

A simple example for applying the proposed approach is a single stepped column under a lateral load, which is created here to illustrate the proposed method. Here, the objective is to minimize the column weight, subject to satisfying the maximum stress constraint in each segment, and variables are the radii of the sections (here radii are used instead of area). Then, the optimization problem is formulated as follows:

$$\underset{r \in \Omega_r}{\text{minimize}}\, f(\mathbf{r}) = \rho L \sum_{i=1}^{N} \pi r_i^2, \tag{15}$$

where N is the number of segments, which is equal to 50; $\mathbf{r} \in \{r_1, \dots, r_N\}$; r_i is the radius of the ith segment subject to $r \in [3, 50]^{50} cm$; ρ is the density; and L is the length of each segment. Subject to:

$$g(r) = \sigma_i - \sigma_i^a \leq 0, \tag{16}$$

where σ_i is the maximum bending stress at the bottom of the ith segment; and σ_i^a is the allowable stress. To implement the proposed variable functioning strategy, the variables are converted as:

$$\{r_1, \dots, r_N\} \mapsto \{r_1, \alpha\}$$

The height vector of the cross-section is now equal to:

$$h(i) = (i-1)L$$

where h_u is $h_{\max} = (N-1)L = 49L = 490\text{cm}$, and therefore, the α boundary can be obtained as:

$$\alpha_{\max} \sqrt[490]{\frac{50}{3}} = 1.0057581$$

The α bounds should be: $\Omega_\alpha : \alpha \in [1, 1.00575817]$ All the segments' radii are related to the radius of the first segment and the α value. For this problem, ten solutions with $r_1 = 30\text{cm}$ are presented in Fig. 3. The straight line at the top shows $\alpha = 1$, and the lowest line (brightest line) indicates the solution with α equal to $\alpha_{\max}$. In this case, all the possible solutions will be placed within these two lines. Based on the original formulation, each cross-section radius should be within 3 and 50 cm. Since $r_1 = 30\text{cm}$, all radii should be less than 30 cm. Also, components of a solution have an order (shown as lines in Fig. 3) and are not distributed in the whole search space. Therefore, it is clear how much this approach narrows down the search space, which can enhance the search process.

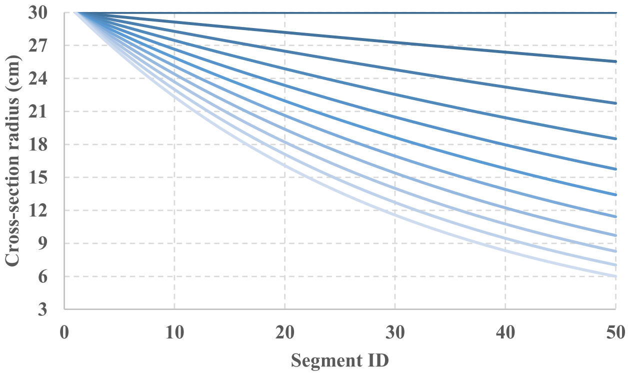


Fig. 3 Sample solutions for the cross-section of the base column equal to 30 cm

After reformulating the current problem, the new objective function can be redefined as follows:

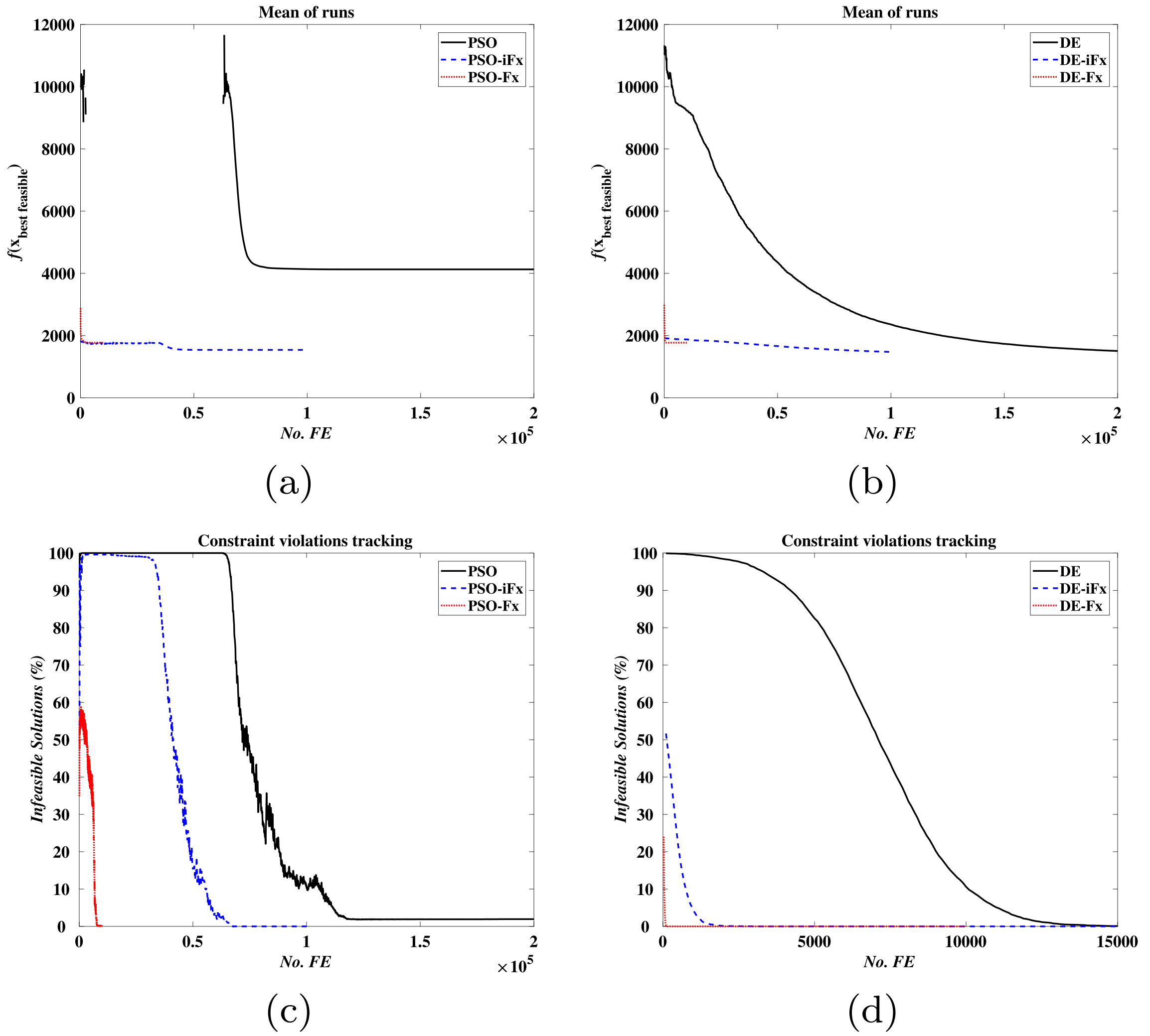
$$\begin{aligned} \underset{r_1 \in \Omega_r \wedge \alpha \in \Omega_\alpha}{\text{minimize}} \quad & f(r_1, \alpha) = \rho L \sum_{i=1}^{N} \pi \left(\frac{r_1}{\alpha^{h(i)}} \right)^2 \\ \text{subject to} \quad & g(r_1, \alpha) = \sigma_i - \sigma_i^a \leq 0. \end{aligned}$$

Using the new formula, the problem with 50 variables was converted to a problem with only two variables and was solved using both PSO and DE. Each algorithm was used for optimizing this problem via three strategies:

1. Optimization algorithm for solving the problem without reformulation
2. Optimization algorithm for solving the problem with reformulation only for initialization (iFx)
3. Optimization algorithm for solving the problem with reformulation (Fx)

Due to the random nature of the optimization process, each strategy was ran 51 times, and the performance plots are presented in Fig. 4. At first, it is obvious that the PSO and DE algorithms have completely different results in optimizing a constrained problem due to their different natures. Also, it should be noted that, because of the selection operator in DE, a feasible solution cannot be replaced with an infeasible solution. Thus, the number of infeasible solutions cannot increase, and the DE histories are smooth. Conversely, since the PSO algorithm does not have such a selection operator, a feasible solution can be replaced with an infeasible solution. As a result, the number of infeasible solutions may increase during the search process.

From Fig. 4a and b, it can be seen that iFx and Fx strategies have better starting points. This should be the main reason why they converge more quickly and to better values in comparison to the original algorithm without the variable functioning approach. These two performance plots show that the third strategy can converge after a few iterations. Since the Fx strategy reduces the problem dimension from 50 to 2, it converges very fast. Although the third strategy converges very quickly, its final solution is slightly worse than the second strategy for this problem, due to the fact that the candidate solutions are forced to stay with the defined function during all iterations. From Fig. 4c and d, all candidate solutions to the algorithms with the proposed strategy quickly converge on the feasible region, and their convergence rates are better than those of the iFx and Fx strategies. The PSO-iFx candidate solutions move toward the feasible region faster than those of PSO, which suggests that initialization using the proposed



(a)

(b)

(c)

(d)

Fig. 4 History results (performance plots) of the optimization algorithms for stepped column design problem. **a** mean convergence histories of PSO runs, **b** mean convergence histories of DE runs, **c** histories of infeasible candidate solutions rate for PSO runs, **d** histories of infeasible candidate solutions rate for DE runs

approach could be effective in PSO. Compared to DE, the DE-iFx candidate solutions converge to the feasible region much faster. For this stepped column example, it is clear that the PSO and DE results significantly improved after using the proposed approach. However, the steel frame design optimization is more complex, and cross-section areas should be selected from predefined sections. The next section presents several case studies on steel frame design optimization to demonstrate the applicability of the proposed method.

4 Numerical case studies

Design of three steel frame structures are optimized in this section as case studies. In the structural engineering literature, frame structures are usually defined by the number of stories and bays (Talatahari et al. 2015). These benchmarks for 1-bay 8-story, 3-bay 15-story, and 3-bay 24-story are considered the best-designed frame case studies in this study (Talatahari et al. 2015). The level

Table 1 Population size of benchmark steel frames

Problem	Variable Functioning		
	None	*iFx*	*Fx*
1-bay 8-story frame	25	25	20
3-bay 15-story frame	40	40	25
3-bay 24-story frame	60	60	25

of interaction between each pair of variables for these three case studies are visualized in Fig. 6, which confirms that all column cross-sections depend on the base column cross-section. These steel frame design problems were optimized by both PSO and DE algorithms. Like the illustrative example, three strategies were implemented in each of the optimization algorithms. Because of the random nature of the optimization process, each strategy was ran 51 times to obtain meaningful results. After trial and error, with the initial population up to 100, a range of 20 to 60 was found to be best for the benchmark steel frame problems with and without the variable functioning approach, as illustrated in Table 1.

The maximum number of function evaluations, or the number of finite element (FE) analyses, was considered to be the stopping criterion, which can be found from the convergence history plots. The structural analysis was performed in MATLAB using the matrix stiffness method implementation of FE.

4.1 Design of a 1-bay, 8-story frame

The 1-bay 8-story problem is one of the benchmark structural engineering problems (Gandomi and Yang 2011) that has been widely used in the literature (e.g., Juliani and Gomes (2022)). The configuration of this frame structure, including the applied loads, is shown in Fig. 5. After considering the fabrication conditions affecting the construction of the frame structure, the same beam/column cross-sections were used for the two following stories. The values of both the beam and column element groups were chosen from all 267 W-shapes. In this case, the roof drift is the only performance constraint, which should be less than 5.08 cm. The modulus of elasticity (E) of the steel was taken as 200 GPa.

The proposed approach was used to reformulate the problem variables (column cross-section areas). Here, four column cross-section areas were replaced with the base cross-section area and α; therefore, the number of column variables was decreased to two variables. PSO and DE algorithms were applied to solve this benchmark problem with the three defined strategies. The convergence histories of PSO and DE algorithms with 3000 FEs for Fx and 5000 FEs for the other strategies and the mean of 51 runs are presented in Fig. 7. From the shown histories, it can be seen that the algorithms using *iFx* and *Fx* strategies started the search from a better fitness compared to the algorithm without variable functioning, which demonstrates that the initialization itself (*iFx*) can significantly improve the search

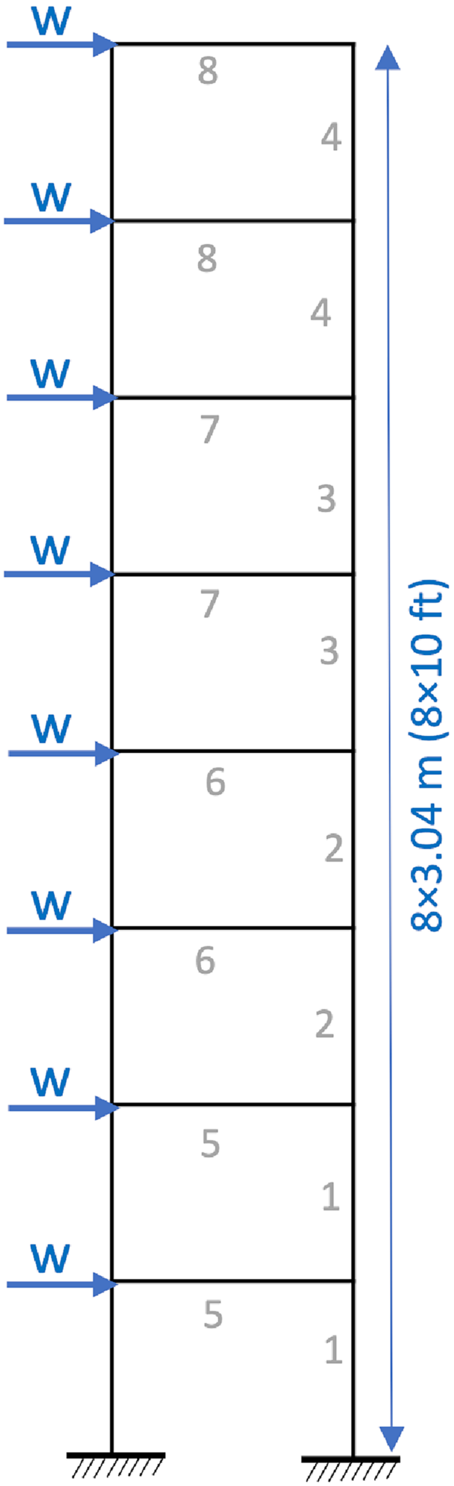
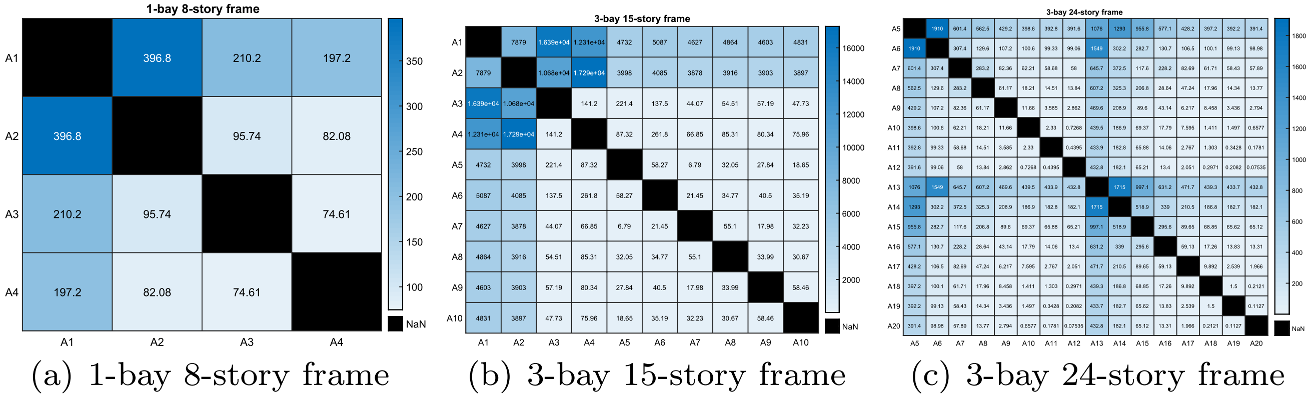
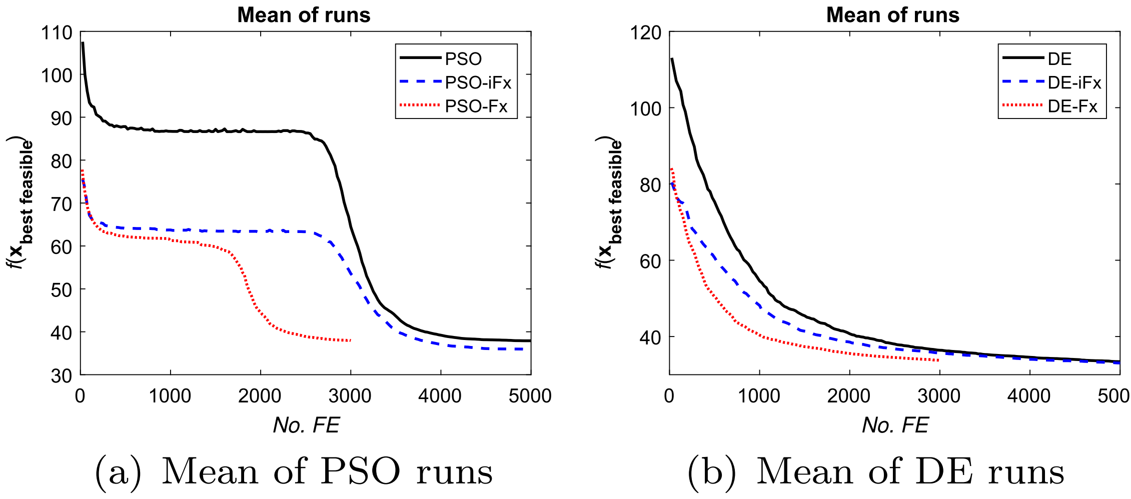


Fig. 5 Topology of the 1-bay 8-story frame



(a) 1-bay 8-story frame (b) 3-bay 15-story frame (c) 3-bay 24-story frame

Fig. 6 Variable interaction matrix of various steel frames

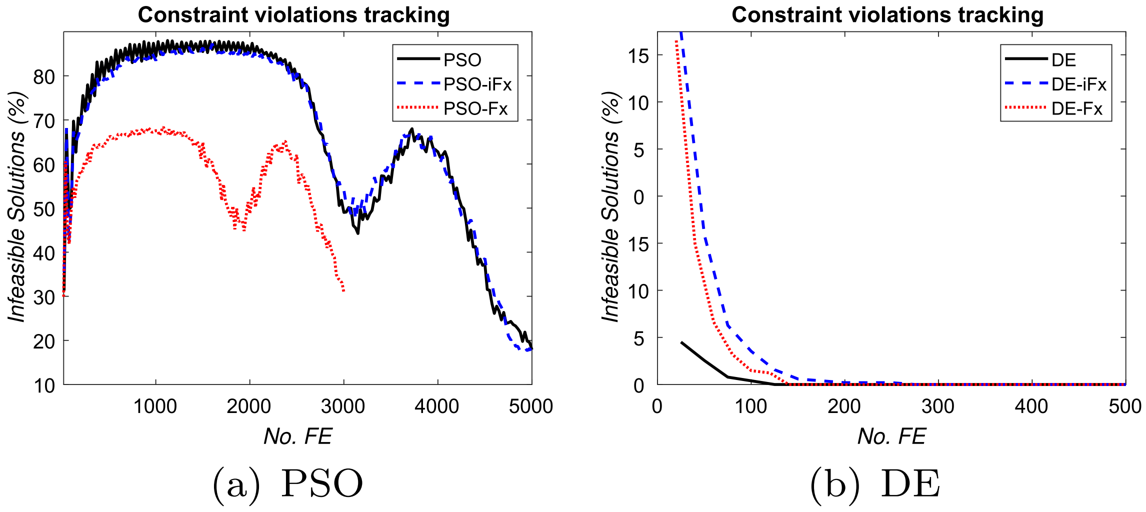


(a) Mean of PSO runs (b) Mean of DE runs

Fig. 7 Convergence histories of the 1-bay 8-story frame optimization problem using PSO and DE algorithms

process. Note that each of these methods converged to a different solution, and the final cross-sections of any two strategies were not the same. The PSO convergence history (Fig. 7a) clearly indicates that *iFx* helps the algorithm to converge more quickly and to better solutions. Using the proposed approach during the search process (*Fx*) significantly improved the convergence rate of PSO in comparison with the other two strategies. The DE convergence histories also have the same trend as those of PSO histories. Although the convergence rates improved after using DE-*iFx* and DE-*Fx*, the improvements were not as significant as those of PSO. Note that the results of the PSO-*Fx* and DE-*Fx* algorithms were obtained after 3,000 FEs, while the others were received after 5,000 FEs. Therefore, the third strategy (*Fx*) improved the convergence rates significantly for this case.

Figure 8 presents the number of infeasible solutions during the PSO and DE iterations, respectively. From Figs. 8a and 7a, it can be seen that PSO mostly performs exploration for about the first half of iterations when the best feasible solutions and violation rates remained almost constant after a few iterations. Fig. 8a also shows that the percentage of infeasible solutions in PSO and PSO-*iFx* is similar. This means that, despite different initializations, both algorithms have the same constraint violation histories. In Fig. 8a, the PSO-*Fx* results have the same pattern as the other methods with a smaller scale (fewer FEs), which confirms the advantages of the *Fx* strategy over the other two strategies in reaching more feasible solutions. Figure 8b illustrates an entirely different paradigm, where the constraint violation of DE is less than that of DE-*iFx* and DE-*Fx*. Also, DE-*iFx* and DE-*Fx* infeasible solutions histories are very similar and differ from the DE history. This is because the DE-*iFx* and DE-*Fx* strategies generate more infeasible solutions in early iterations compared to the DE strategy (approximately 10%).



(a) PSO (b) DE

Fig. 8 Infeasible solutions histories in 1-bay 8-story frame optimization problem

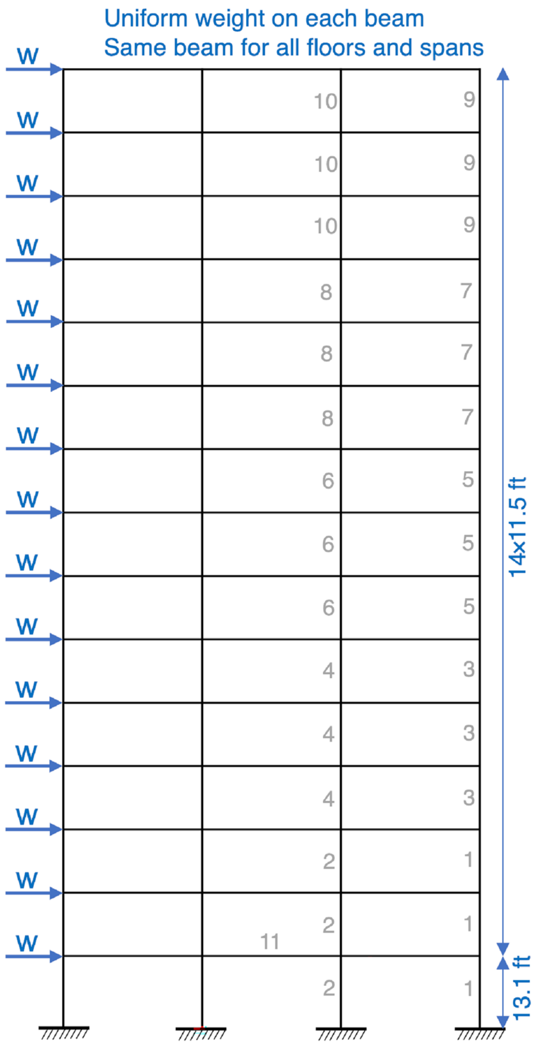
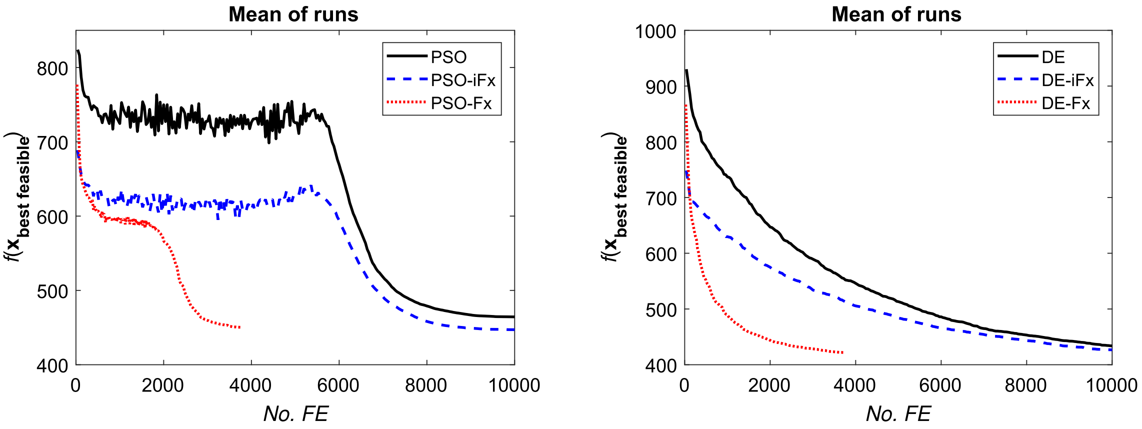

Uniform weight on each beam
Same beam for all floors and spans
W
10
9
8
7
6
5
4
3
2
1
11
14x11.5 ft
13.1 ft

Fig. 9 Topology of the 3-bay 15-story frame

4.2 Design of a 3-bay, 15-story frame

The 3-bay 15-story frame structure is shown in Fig. 9, including the configuration and applied loads. This steel frame optimization problem has been used by many researchers as a benchmark (Mosharmovahhed and Moharrami 2021). The AISC combined strength and displacement limit (sway of the roof is restricted to 23.5 cm) IA considered as the constraint for optimizing the frame weight. The used steel has E = 200 GPa and a yield stress (Fy) of 248.2 MPa. After considering the fabrication conditions for the construction of the frame, the same column cross-sections were used for the three following stories. Because of the symmetry of the frame structure, two sets of columns were considered here: inner columns and outer columns. Each column set was replaced with one function to apply the proposed approach, and as a result, the number of variables decreased from 11 to 5 after using the *Fx* approach.

The 3-bay 15-story frame problem was solved by PSO and DE algorithms with different strategies. The maximum number of FEs was set to 4000 for the Fx strategy and to 10,000 for the other strategies. Also, it should be clarified that each of these methods converge to a different solution, which is expected due to the vast search space of this case study. The convergence histories of the PSO and DE algorithms for the mean of runs are presented in Fig. 10. From Fig. 10a, it can be seen that PSO-*iFx* convergence slightly improved in contrast with PSO. However, it is clear that PSO-*iFx* notably improved the average results (shown in Fig. 10a). Using the proposed approach during the whole iterations (*Fx*) significantly improved the convergence of PSO for this problem in comparison with the other two strategies. Similar to the PSO results, it can be seen that the *iFx* strategy improved the DE algorithm, and using the



(a) Mean of PSO runs (b) Mean of DE runs

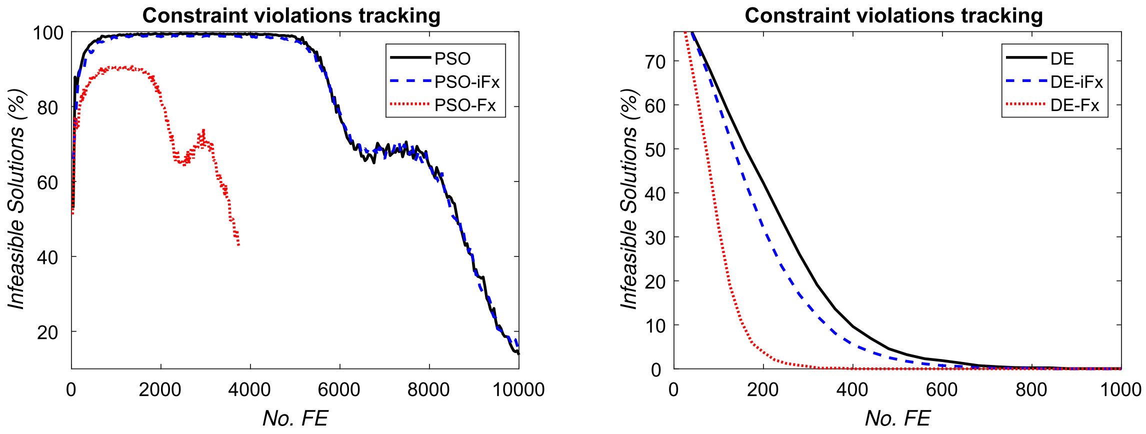
Fig. 10 Convergence histories of the 3-bay 15-story frame optimization problem using PSO and DE algorithms

approach during all iterations (*Fx*) significantly improved the convergence history (Fig. 10b). Therefore, it is clear that the proposed approach can notably improve the results for this steel frame optimization problem.

Figure 10a indicates that the PSO-*Fx* and PSO-*iFx* methods need fewer than 3000 and 8000 FEs, respectively, to reach the objective value obtained by PSO after 10,000 FEs. Figure 10b similarly shows that the DE-*Fx* and DE-*iFx* methods reached better objective values with lesser FEs. In Fig. 10b, the improvement is significant for DE-*iFx*, which only needs about a quarter of the FEs compared to the DE methods to reach the same objective value.

Figure 11 displays the number of infeasible solutions during iterations in PSO and DE algorithms. From Fig. 11a, it can be seen that the percentages of infeasible solutions of PSO and PSO-*iFx* during optimization processes have a similar pattern. This suggests that, despite different initializations, these two algorithms have the same constraint violation histories. In Fig. 11a, the PSO-*Fx* results have the same pattern as the other methods with fewer FEs. This confirms the advantages of the proposed approach for finding more feasible solutions when it is used during iterations. However, using the proposed method for the initialization (shown in Fig. 11b) slightly increased the number of feasible solutions, in comparison with the DE strategy. Once again, using the proposed approach during iterations significantly enhanced the convergence to the feasible region.

It can be seen that using the proposed method for either the initialization or whole iterations results in cost savings. For the PSO algorithm, the *iFx* strategy lead to the best results with a 3.6% improvement of the median over the 51 PSO runs. However, using *Fx* during the whole search process improved the median by only 2.5%. The DE results show that the *Fx* strategy could produce the



(a) PSO (b) DE

Fig. 11 Infeasible solutions histories in 3-bay 15-story frame optimization problem

greatest improvement (3.1%) of the median. Interestingly, the latter strategy only used 3750 FEs, which is far fewer than the other strategies that used 10,000 FEs. It should be noted that about a 2% improvement of the median was obtained when only the seeding approach was used with the DE algorithm.

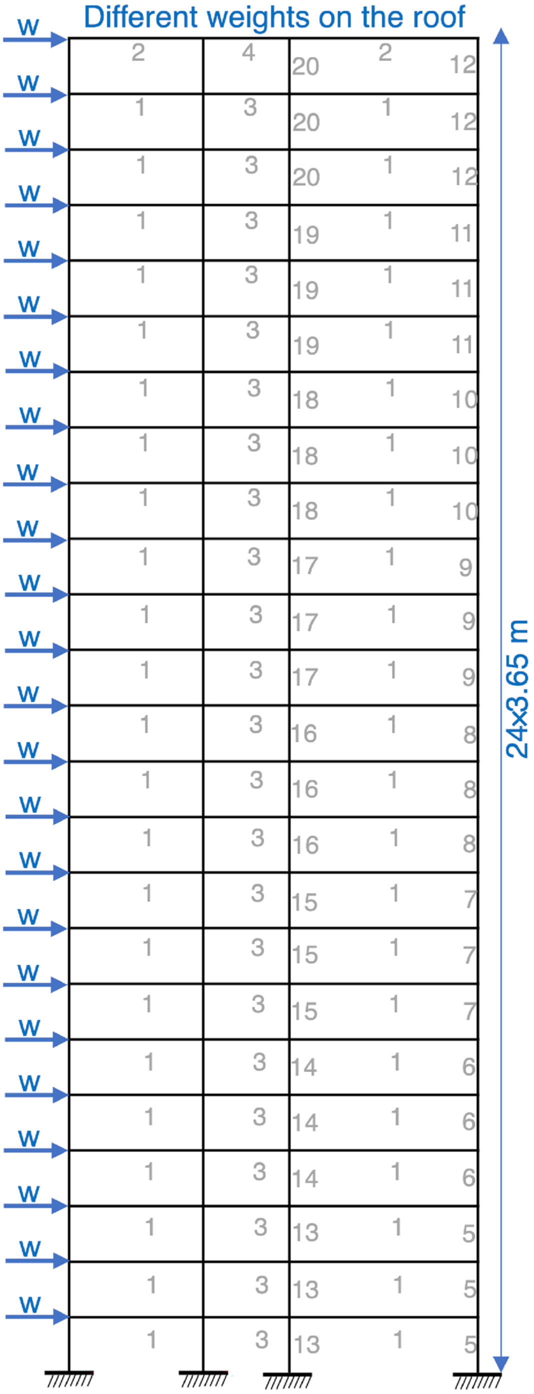


Fig. 12 Topology of the 3-bay 24-story frame

4.3 Design of a 3-bay 24-story frame

The topology of the 3-bay 24-story frame, along with the service loading conditions, are shown in Fig. 12. This tall steel frame problem originally designed by Davison and Adams (1974) has 168 members. This case study was later optimized by many researcher as a challenging steel frame optimization problem Bigham and Gholizadeh (2020). After imposing the fabrication conditions on the construction of the frame, the same beam cross-section was used in the first and third bay of all floors, except the roof beams. The interior columns were combined into one set, and the exterior columns were combined in another set of three following stories, yielding 20 design variables: 4 beam and 16 column groups. This steel frame structure was designed based on the LRFD specifications under inter-story drift constraints. The steel material has $E = 205\,\text{GPa}$ and $Fy = 230.3\,\text{MPa}$. In this problem, column element groups should be chosen from W14 sections (37 W-shapes), while beam element groups could be any of the 267 W-shapes.

One function was assigned to each of the column sets, interior column and exterior column sets, and thus, the number of variables for the columns decreased from 16 to 4, which is significant dimension reduction. The 3-bay 24-story steel frame design problem was solved by PSO and DE algorithms with the three defined strategies, where the Fx and other strategies had a maximum number of iterations of 5000 and 15,000, respectively. The convergence histories of PSO and DE algorithms for the mean of runs are presented in Fig. 13. From these histories, it is clear that both the *iFx* and *Fx* strategies improved the convergence rates of PSO and DE algorithms significantly and also converged to better solutions. Comparing the strategies using the proposed approach, *Fx* had more improvements in the convergence rate than *iFx*. Each of these methods converged to a different solution, which was expected because of the huge search space of the case study.

Figure 13a illustrates that the PSO-*Fx* and PSO-*iFx* methods need less than 4000 and 10,000 FEs to reach the objective value of PSO after 15,000 FEs, respectively. Figure 13b similarly indicates that both DE-*Fx* and DE-*iFx* methods reached better objective values using fewer FEs. In Fig. 13b, the improvement is significant for DE-*iFx* since it only needs about a quarter of the FEs in comparison with DE methods to reach the same objective value.

Figure 14 displays the percentage of infeasible solutions in the histories of PSO and DE algorithms, respectively. From Fig. 14a, it can be seen that the percentages of infeasible solutions of PSO and PSO-*iFx* during iterations are almost identical. In Fig. 14a, the solutions of the PSO algorithm moved toward the feasible area in fewer FEs, which confirms the advantages of the proposed approach for finding more feasible solutions during the entire search process. However, Fig. 14b shows that the proposed approach for the

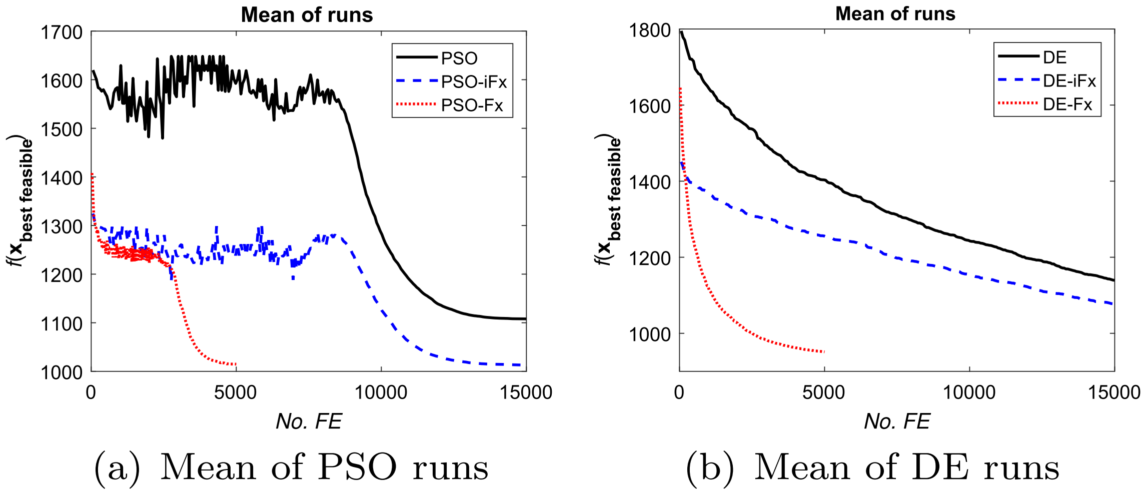


Fig. 13 Convergence histories of the 3-bay 24-story frame optimization problem using PSO and DE algorithms

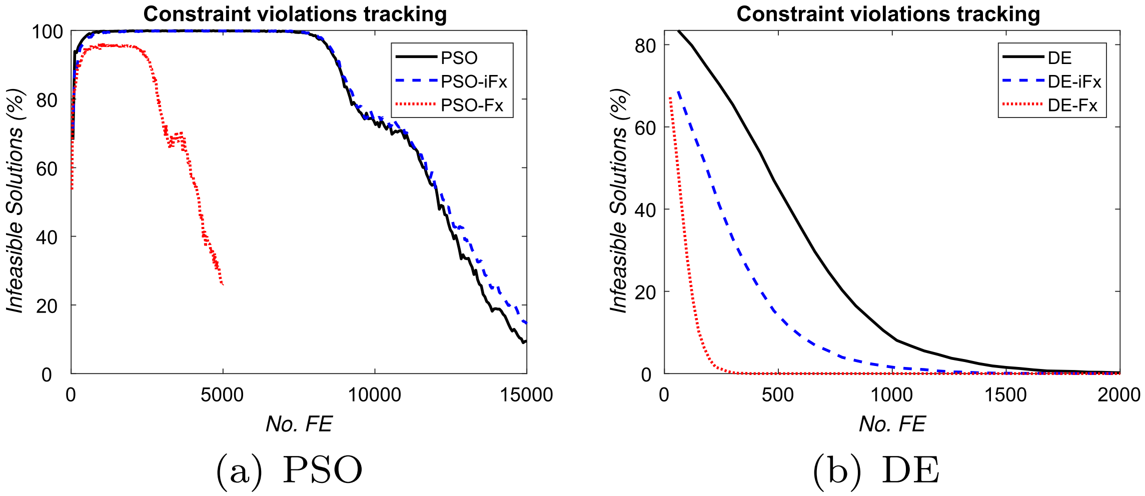
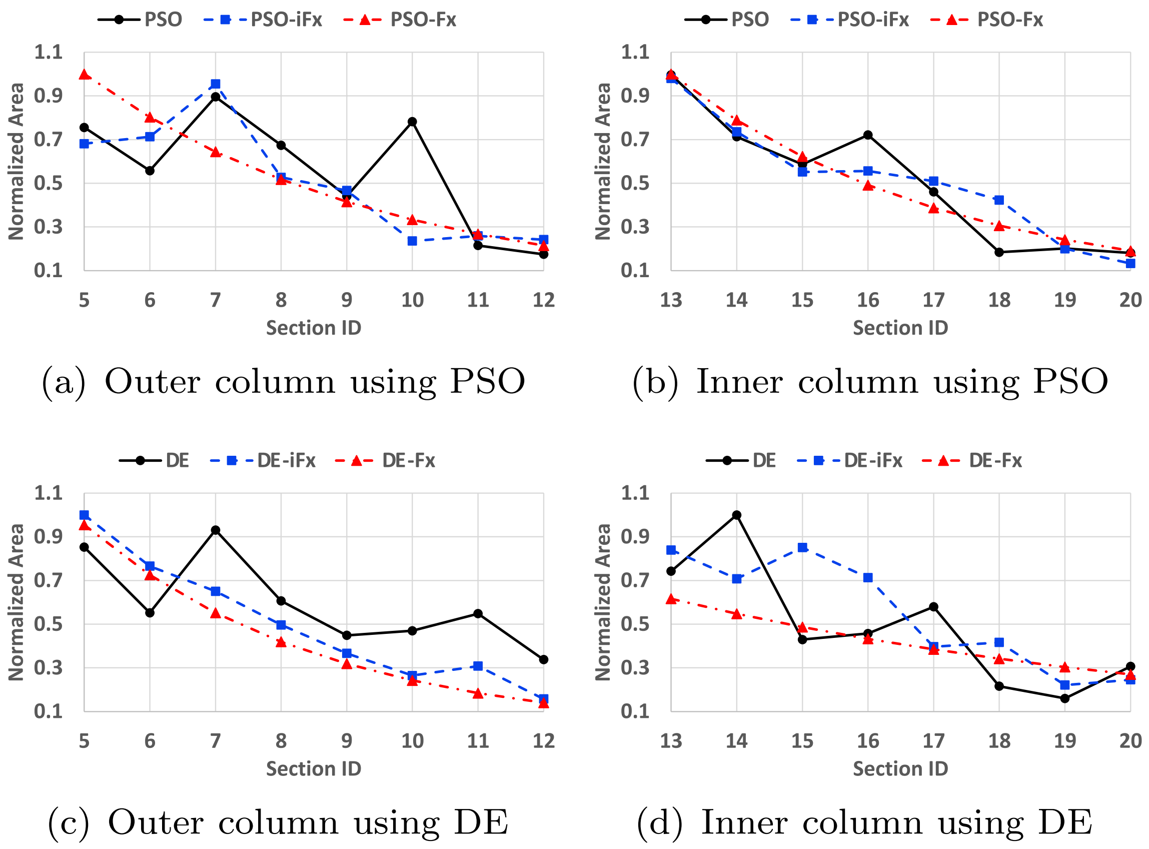


Fig. 14 Infeasible solutions histories in 3-bay 23-story frame optimization problem

initialization notably increased the number of feasible solutions (*iFx*). Using this approach during whole iterations (*Fx*) can even improve the number of feasible solutions, and all solutions become feasible at early iterations.

It can be seen that *iFx* and *Fx* strategies resulted in better designs. For the PSO algorithm, both *iFx* and *Fx* strategies improved the median results by about 9%. When using only the function for seeding of the DE algorithm (*iFx*), the median improved by about 5.9%. Moreover, the *Fx* strategy is much more effective and improved the median results of DE by 16.8% with only one third of FEs used in other strategies.

The normalized outer and inner column cross-section areas of the best solutions using PSO and DE algorithms are presented in Fig. 15. As previously discussed (section 4.2), a column's cross-section areas should be reduced with increasing height based on the physics of the problem and engineering knowledge. In Fig. 15a, it can be seen that only PSO- (deleted something here on accident) has the expected trend in the results for the outer column. Figure 15b shows that the optimum design of PSO without the proposed approach does not have the desired trend in the inner column cross-section areas, and the final design of PSO-*iFx* slightly violates this condition. However, this is not the case for PSO-*Fx* since the expected trend is considered by the variable functioning. In other words, all the solutions are forced to satisfy the condition, and therefore, the algorithms only search the solutions that have the trend and monotonically decrease with increasing height. From Fig. 15c and d, DE and DE-*iFx* do not exhibit this condition in the outer and the inner columns. Therefore, the DE-*Fx* results are even more practical, as they satisfy the conditions.



(a) Outer column using PSO

(b) Inner column using PSO

(c) Outer column using DE

(d) Inner column using DE

Fig. 15 Optimum design of columns cross-section areas for the 3-bay 24-story frame structure

5 Conclusion and discussions

In this study, a concept-based approach is proposed to replace sets of variables with their functions by incorporating different sources of information. For the steel frame design optimization problems, the structure of the variables (cross-section areas) was explored using a grouping method, then a function was defined to relate each column cross-section areas by means of their heights. A general equation was developed to relate a column's cross-section areas using a differential grouping and engineering points-of-view. Here, the proposed approach was coupled with two well-known global optimization algorithms (DE and PSO) and implemented in a 50-stepped column design problem. The designs of three complex steel frame structures with different numbers of bays and stories were further optimized as numerical case studies. The proposed approach was coupled with the optimization algorithms using two different strategies: 1. Only for initialization; and 2. During the entire iterations. The results suggest that the proposed method can improve the results when used only for initialization and can significantly improve the convergence rates and final solutions when used during the iterations, especially for the tall and more complex frame design optimization problems. Using the proposed approach during initialization showed to be significantly effective in all cases and improved the results for the studied steel frame structures. Therefore, it can be used for seeding of the frame design optimization problem instead of randomly generating the initial solutions. Although the relationship proposed for the column sections may not be exact, it achieved competitive results and, most of the time, produced better results compared with an algorithm that does not use the proposed approach. In terms of numbers of function evaluations, the proposed approach can significantly improve the convergence rate and reduce the number of required finite element analyses (a time-consuming process for real-world problems) when used during the entire search process. It should be noted that this approach also considers the practicality aspect of a column cross-section areas, which may not be achieved with other strategies. Since the proposed function was effectively applied to columns of moment-resisting steel frames, we suggest future research focus on applying and modifying it to different types of frame systems (Camp and Assadollahi 2013) and real-world structures (Azad 2021; Montemurro et al. 2012). Also, the proposed approach can be coupled with curve fitting (Costa et al. 2018) and surrogate models (Audoux et al. 2020). Since the variable functioning aims to narrow down the search space in the optimization problem, in problems where it is possible to introduce a variable functioning, a similar superior efficiency of the optimization process can be expected.

Appendix

See Tables 2 and 3.

Table 2 Summary of notations used in this study

Symbol / Acronym	Definition / Description
A	Cross-section area
E	Elastic modulus
H	Height of the frame structure
K	Effective length factors of members
L	Length
M_{nx}	Required nominal flexural strengths in the x-direction
M_{ny}	Required nominal flexural strengths in the y-direction
M_{ux}	Required flexural strengths in the x-direction
M_{uy}	Required flexural strengths in the y-direction
P	Force
P_n	The nominal axial strength (tension or compression)
P_u	The required strength (tension or compression)
R	Maximum drift index
RI	Inter-story drift index permitted by standard design code
$\mathbf{V}$	Velocity component matrix
W	Frame weight
$\mathbf{x}$	Design variable
d	Inter-story drift
f	Objective function
$f_{\max}$	Objective function value of the worst feasible solution in the population
g	Constraint function
$g_{\max}$	The largest known violation of a constraint
h	Height from the base (or ground)
h_j	Height of jth floor
l	Length of each segment
n_{c}	Number of constraints
n_{g}	Number of groups
n_{m}	Number of members
ns	Total number of stories
r	Radius
Δ_T	The maximum lateral displacement
Ω	Search space
ϕ_b	Flexural resistance reduction factor
ϕ_c	Resistance factor (tension or compression)
ρ	Material density
σ_i	Maximum stress in the ith member
σ_i^a	Allowable stress in the ith member
ζ	A multivariate mathematical function from $\mathbb{R}^m$ to $\mathbb{R}^q$

Table 3 Summary of acronyms used in this study

Acronym	Description
AISC	American Institute of Steel Construction
DE	Differential evolution
DG	Differential grouping
DG2	Extended differential grouping
FE	Finite element
Fx	Variable functioning
LRFD	Load and Resistance Factor Design
PSO	Particle swarm optimization
iFx	Initialization by variable functioning

Acknowledgements Authors are grateful to Associate Professor Saeid K Azad of the Atilim University for his constructive comments and fruitful suggestions. This research was supported in part by the National Science Foundation (NSF) under Cooperative Agreement DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. Funding was in part provided by National Science Foundation (DBI-0939454).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results Optimization algorithm codes and data for replication can be provided up on request.

References

Audoux Y, Montemurro M, Pailhes J (2020) Non-uniform rational basis spline hyper-surfaces for metamodelling. Comput Methods Appl Mech Eng 364:112918

Azad SK (2021) Design optimization of real-size steel frames using monitored convergence curve. Struct Multidisc Optim 63(1):267–288

Azad SK, Hasançebi O (2015) Computationally efficient discrete sizing of steel frames via guided stochastic search heuristic. Comput Struct 156:12–28

Becerra RL, Coello CAC (2006) Cultured differential evolution for constrained optimization. Comput Methods Appl Mech Eng 195(33–36):4303–4322

Bigham A, Gholizadeh S (2020) Topology optimization of nonlinear single-layer domes by an improved electro-search algorithm and its performance analysis using statistical tests. Struct Multidisc Optim 62(4):1821–1848

Camp CV, Assadollahi A (2013) Co 2 and cost optimization of reinforced concrete footings using a hybrid big bang-big crunch algorithm. Struct Multidisc Optim 48(2):411–426

Chen W, Weise T, Yang Z, Tang K (2010) Large-scale global optimization using cooperative coevolution with variable interaction learning. In: International conference on parallel problem solving from nature, Springer, pp 300–309

Costa G, Montemurro M, Pailhès J (2018) A general hybrid optimization strategy for curve fitting in the non-uniform rational basis spline framework. J Optim Theory Appl 176(1):225–251

Davison JH, Adams PF (1974) Stability of braced and unbraced frames. J Struct Div 100(2):319–334

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):311–338

Deb K, Myburgh C (2017) A population-based fast algorithm for a billion-dimensional resource allocation problem with integer variables. Eur J Oper Res 261(2):460–474

Eiben AE, Smith J (2015) From evolutionary computation to the evolution of things. Nature 521(7553):476

Gandomi AH, Yang XS (2011) Benchmark problems in structural optimization. Computational optimization, methods and algorithms. Springer, New York, pp 259–281

Ghasemi MR, Farshchin M (2011) Ant colony optimisation-based multiobjective frame design under seismic conditions. Proc Inst Civ Eng-Struct Build 164(6):421–432

Gholizadeh S, Poorhoseini H (2016) Seismic layout optimization of steel braced frames by an improved dolphin echolocation algorithm. Struct Multidisc Optim 54(4):1011–1029

Hasançebi O, Çarbaş S, Doğan E, Erdal F, Saka M (2010) Comparison of non-deterministic search techniques in the optimum design of real size steel frames. Comput struct 88(17–18):1033–1048

De Jong K (1988) Learning with genetic algorithms: an overview. Mach Learn 3(2–3):121–138

Juliani MA, Gomes WJ (2022) An efficient Kriging-based framework for computationally demanding constrained structural optimization problems. Struct Multidisc Optim 65(1):1–16

Kennedy R (1995) J. and eberhart, particle swarm optimization. In: Proceedings of IEEE international conference on neural networks IV, pages, vol 1000

Lamberti L, Pappalettere C (2011) Metaheuristic design optimization of skeletal structures: a review. Comput Technol Rev 4(1):1–32

Liu J, Tang K (2013) Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In: International conference on intelligent data engineering and automated learning. Springer, pp 350–357

Mahdavi S, Shiri ME, Rahnamayan S (2014) (2014) Cooperative coevolution with a new decomposition method for large-scale optimization. IEEE Congress on evolutionary computation (CEC). IEEE, pp 1285–1292

Mei Y, Omidvar MN, Li X, Yao X (2016) A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. ACM Trans Math Softw 42(2):13

Montemurro M, Vincenti A, Vannucci P (2012) A two-level procedure for the global optimum design of composite modular structures-application to the design of an aircraft wing. J Optim Theory Appl 155(1):24–53

Montemurro M, Vincenti A, Vannucci P (2013) The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. Comput Methods Appl Mech Eng 256:70–87

Mosharmovahhed M, Moharrami H (2021) Design optimization of moment frame structures by the method of inscribed hyperspheres. Struct Multidisc Optim 64(1):335–348

Munetomo M, Goldberg DE (1999) Identifying linkage groups by nonlinearity/non-monotonicity detection. Proc Genet Evolut Comput Conf 1:433–440

Omidvar MN, Li X, Mei Y, Yao X (2014) Cooperative co-evolution with differential grouping for large scale optimization. IEEE Trans Evol Comput 18(3):378–393

Omidvar MN, Yang M, Mei Y, Li X, Yao X (2017) DG2: a faster and more accurate differential grouping for large-scale black-box optimization. IEEE Trans Evol Comput 21(6):929–942

Pavlovčič L, Krajnc A, Beg D (2004) Cost function analysis in the structural optimization of steel frames. Struct Multidisc Optim 28(4):286–295

Ray T, Yao X (2009) A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning. In: IEEE Congress on evolutionary computation CEC'09. IEEE, pp 983–989

Saka M (2007) Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. Civ Eng Comput 6:105–147

Santana R (2017) Gray-box optimization and factorized distribution algorithms: where two worlds collide. http://arxiv.org/abs/1707.03093

Slowik A, Kwasnicka H (2018) Nature inspired methods and their industry applications-swarm intelligence algorithms. IEEE Trans Ind Inf 14(3):1004–1015

Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Sun L, Yoshida S, Cheng X, Liang Y (2012) A cooperative particle swarm optimizer with statistical variable interdependence learning. Inf Sci 186(1):20–39

Talatahari S, Gandomi AH, Yang XS, Deb S (2015) Optimum design of frame structures using the eagle strategy with differential evolution. Eng Struct 91:16–25

Tintos R, Whitley D, Chicano F (2015) Partition crossover for pseudo-boolean optimization. In: Proceedings of the 2015 ACM conference on foundations of genetic algorithms XIII, ACM, pp 137–149

Whitley D, Hains D, Howe A (2010) A hybrid genetic algorithm for the traveling salesman problem using generalized partition crossover. In: International conference on parallel problem solving from nature. Springer, pp 566–575

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.