Journal of Science and Technology in Civil Engineering, HUCE, 2025, 19 (2): 48–61

MULTI-OBJECTIVE OPTIMIZATION FOR NONLINEAR STEEL FRAMES USING A PARAMETER-LESS MOO ALGORITHM AND XGBOOST

Manh-Cuong Nguyen[a], Thi-Thu-Hien Le[b], Manh-Tien Le[c], Quoc-Anh Vu[d], Ngoc-Thang Nguyen[a], Viet-Hung Truong[a,*]

[a]*Faculty of Civil Engineering, Thuyloi University, 175 Tay Son road, Dong Da, Hanoi, Vietnam*
[b]*Faculty of Water Resources Engineering, Thuyloi University, 175 Tay Son road, Dong Da, Hanoi, Vietnam*
[c]*Vietnam Medical Technology Equipment Joint Stock Company, 09 Phuong Nam road, Thanh Xuan district, Hanoi, Vietnam*
[d]*Faculty of Civil Engineering, Hanoi Architectural University, Km10 Nguyen Trai road, Hanoi, Vietnam*

Article history:
Received 04/4/2025, Revised 04/5/2025, Accepted 13/5/2025

Abstract

This study proposes a hybrid RDMO-XGBoost method for bi-objective optimization of nonlinear steel frames, targeting minimal structural mass and top-story displacement under static loads. Traditional linear elastic models fail to capture the nonlinear inelastic behavior of steel frames, such as yielding and buckling, necessitating advanced optimization strategies. The Rao-DE Multi-Objective Optimization (RDMO) algorithm, combined with Differential Evolution, ensures diverse and accurate Pareto solutions without parameter tuning. Integrated with XGBoost, a machine learning tool, the method predicts frame responses rapidly, reducing reliance on time-intensive finite element analysis. Applied to a two-story frame, RDMO-XGBoost reduced computation time by over 55% (from 50,400 to 22,370 seconds) compared to generalized differential evolution 3 (GDE3) and RDMO, while maintaining comparable performance in convergence, coverage, and diversity. Anchor point results confirm its effectiveness, achieving lower mass (5148.891 kg) and displacement (0.807 mm), demonstrating its potential for efficient and robust steel frame design optimization.

Keywords: multi-objective optimization; inelastic analysis; steel frame; XGBoost; RDMO; Rao.

https://doi.org/10.31814/stce.huce2025-19(2)-04

1. Introduction

Designing steel structures, particularly steel frames, which is a fundamental task in structural engineering, involves balancing cost, strength, and safety under diverse loading conditions [1]. Traditional methods, which typically rely on linear elastic assumptions, fail to capture the nonlinear inelastic behavior that steel frames exhibit when subjected to heavy loads, such as yielding or buckling [2]. This limitation poses significant challenges in multi-objective optimization (MOO), where the objectives of minimizing structural weight and controlling excessive deformations must be addressed concurrently to ensure practical and resilient frame designs [3]. Since assessing the nonlinear inelastic responses of steel frames through finite element analysis (FEA) requires substantial time and computational resources, particularly for complex frame systems, there emerges a pressing need for faster, yet reliable, optimization strategies [4, 5].

Recent research has underscored the potential of metaheuristic algorithms as powerful tools for tackling these challenges, delivering effective solutions to intricate structural problems involving nonlinear inelastic behavior [6]. Algorithms such as Differential Evolution (DE), Harmony Search (HS),

*Corresponding author. *E-mail address:* truongviethung@tlu.edu.vn (Truong, V.-H.)

and Particle Swarm Optimization (PSO), which systematically explore a wide range of design possibilities, demonstrate strong performance even when confronted with the complex nonlinear constraints inherent in steel frame optimization [7–10]. In MOO, where the focus shifts from seeking a single optimal solution to generating a Pareto set that balances competing goals like weight reduction and displacement control under inelastic conditions, additional complexities arise [11]. Because this process demands extensive evaluations to span the design space, and because configuring these algorithms to handle nonlinear inelastic responses can be intricate, traditional approaches often falter in maintaining computational efficiency.

To address these challenges, this study introduces a hybrid approach that integrates the Rao-DE Multi-Objective Optimization algorithm (RDMO) with XGBoost, a robust machine learning tool engineered for rapid predictions [6, 12, 13]. RDMO, which merges the simplicity of the Rao-1 algorithm with the robust search capabilitis of Differential Evolution (DE), utilizes a ranking system to assess solutions while eliminating the need for additional parameter tuning [7, 14, 15]. By incorporating multiple high- and low-ranked solutions, RDMO improves both the diversity and precision of outcomes, surpassing tools like Differential Evolution for Multi-Objective Optimization (DEMO), generalized differential evolution 3 (GDE3), and Multi-Objective Evolutionary Algorithm with Inverted Generational Distance and Non-dominated Sorting (MOEA-IGD-NS) in evaluations of solution variety and proximity to optimal results [16–19]. However, when applied to steel frames exhibiting nonlinear inelastic behavior under heavy loads, RDMO's dependence on repeated finite element analysis (FEA) evaluations hampers its efficiency, highlighting the need for a more streamlined approach [20].

Incorporating machine learning, specifically Extreme Gradient Boosting (XGBoost), mitigates this inefficiency by predicting the nonlinear inelastic responses of steel frames without requiring continuous FEA assessments. Developed by Chen and Guestrin, XGBoost excels at rapidly identifying patterns in data, making it well-suited for estimating critical parameters such as load capacities or deformations with limited initial data [14, 21, 22]. Previous studies, which evaluated its performance on structures with inelastic behavior, confirm its capacity to accelerate optimization processes while preserving result accuracy [23]. This method extends the success of MOEA/D-EpDE_XGBoost (MEX), a framework that reduced computation time by over 50% compared to its baseline, indicating that combining RDMO with XGBoost could achieve comparable efficiency gains for steel frame optimization [3, 24].

Given that steel frames under significant loads display complex nonlinear inelastic behavior—such as yielding and buckling—accurate analysis is essential, and conventional linear models fail to fully capture these responses [25]. The Blandford model, which monitors stretching and buckling, integrates with the Generalized Displacement Control (GDC) method to provide reliable solutions for these phenomena, as facilitated by software like the Practical Advanced Analysis Program (PAAP) [26–28]. Nevertheless, because conducting such detailed analyses repeatedly for large-scale frame designs consumes excessive computational resources, the proposed RDMO-XGBoost approach employs XGBoost to estimate frame responses after a few actual tests, iteratively refining predictions throughout the optimization process [29, 30]. This combination aims to deliver faster results without compromising precision.

The optimization focuses on two primary objectives: reducing the weight of steel frames, which lowers costs, and constraining horizontal deformations, which ensures structural integrity under inelastic conditions [31–33]. By combining RDMO's efficient search mechanism with XGBoost's swift predictive capabilities, this method seeks to outperform existing tools in terms of speed, solution di-

versity, and computational efficiency, with its effectiveness to be validated on steel frame designs [34]. Previous research, which showcased the benefits of integrating metaheuristics with machine learning, underpins this effort to enhance optimization techniques for steel frames exhibiting nonlinear inelastic behavior [35].

2. Bi-objective optimization of steel frames

2.1. Objective functions

Designing steel frames requires balancing cost efficiency and structural stability, often through multi-objective optimization. This study formulates a bi-objective problem to address the nonlinear inelastic behavior of steel frames under static loads, surpassing basic linear elastic models. The approach targets two goals: minimizing the total mass (and thus cost, linked directly to material use) by adjusting member cross-sections, enhancing economic and environmental outcomes. The first objective function is expressed as:

$$\text{Minimize} \quad T(\mathrm{X}) = \rho \sum_{i=1}^{nm} \left(A(x_i) \sum_{q=1}^{n_i} L_q \right) \tag{1}$$

where ρ represents the material density, while $\mathrm{X} = (x_1, x_2, \ldots, x_{nm})$ ($x_i \in [1, UB_i]$) denotes a vector design variables. Each variable specifies the cross-sectional area assigned to a particular group of structural elements. x_i is an integer number ranging from 1 to UB_i, indicating the sequential index of a W-shaped section selected from the design catalog, where signifies the total number of available sections. Additionally, n_i corresponds to the count of frame components within the member group, $A(x_i)$ denotes the cross-sectional area of the elements in the member group, and L_q indicates the length of q member in that group.

The second goal aims to maintain the safety and stability of steel frames by reducing peak horizontal displacement at the top point under given loads. Large deformations may cause failure, impaired performance, or excessive vibrations. This is mathematically formulated as:

$$\text{Minimize} \quad F_2(\mathrm{X}) = |\Delta| \tag{2}$$

where Δ represents the horizontal displacement of the structual top. This objective implicitly promotes the use of larger cross-sections and a more robust structural design to withstand loading and maintain acceptable deformation limits.

2.2. Constraint handling

The optimization process must incorporate several constraints to ensure structural feasibility and adherence to design standards. These constraints include:

Constructability constraints: Constructability constraints ensure steel frame designs remain practical by preventing unrealistic optimization outcomes. These rules require that an upper column's section depth be less than that of its lower column. Additionally, for beam-to-column flange connections, the beam's flange width must be narrower than the column's flange width. For beam-to-column web connections, the column's web height must exceed the beam's flange width. These conditions are expressed mathematically as:

$$C_{i,1}^{con}(\mathrm{X}) = \left(\frac{D_c^{up}}{D_c^{low}} \right)_i - 1 \leq 0, \quad i = 1, \ldots, n_{c-c} \tag{3}$$

$$C_{i,2}^{con}(\mathrm{X}) = \left(\frac{b_{bf}}{b_{cf}}\right)_i - 1 \leq 0, \quad i = 1, \ldots, n_{b-c1} \tag{4}$$

$$C_{i,3}^{con}(\mathrm{X}) = \left(\frac{b_{bf2}}{T_c}\right)_i - 1 \leq 0, \quad i = 1, \ldots, n_{b-c2} \tag{5}$$

where n_{c-c}, n_{b-c1}, and n_{b-c2} represent the counts of column-to-column, beam-to-column flange, and beam-to-column web connections, respectively; D_c^{low} and D_c^{up} are the depths of lower and upper columns; and, b_{cf} and b_{bf} are the column and beam flange widths, respectively, at a beam-to-column flange connection; b_{bf2} and T_c are the beam flange width and column web height. Fig. 1 illustrates the construction of a beam-to-column connection.

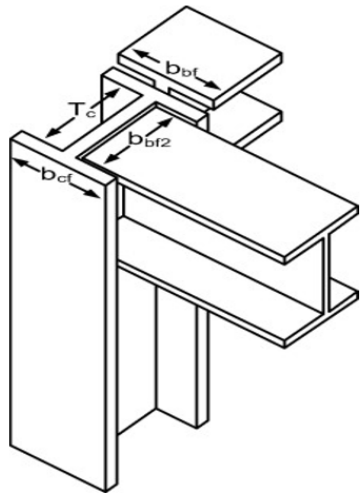


Figure 1. Constructability of a beam-to-column connection

Strength constraints: These limitations guarantee that the structure's load-bearing capacity, denoted as R, exceeds the applied load, S. They are generally assessed using nonlinear finite element analysis, which incorporates material inelasticity and geometric nonlinearities, expressed as:

$$C^{str} = 1 - \frac{R}{S} = 1 - ULF \leq 0 \tag{6}$$

where $ULF = \dfrac{R}{S}$ is the ultimate load factor.

Geometric constraints: These constraints limit the the horizontal displacement of the structual top as:

$$C^{ser} = \frac{|\Delta|}{\Delta^u} - 1 \leq 0 \tag{7}$$

where Δ^u is the allowable value of Δ.

The constraints are incorporated using a penalty function approach. This method adds penalty terms to the objective functions proportional to the violation of the constraints. The penalty terms increase as the constraint violation increases, guiding the optimization algorithm towards feasible solutions as follows:

$$F_1^{un}(\mathrm{X}) = F_1(\mathrm{X}) \times \alpha \left(\begin{array}{l} 1 + \sum_{j=1}^{n_{con}} \left(\max\left(C_{i,1}^{con}, 0\right) + \max\left(C_{i,2}^{con}, 0\right) + \max\left(C_{i,3}^{con}, 0\right) \right) \\ + \sum_{m=1}^{N_{str}} \max\left(C_m^{str}, 0\right) + \sum_{k=1}^{N_{node}} \max\left(C_{l,k}^{ser}, 0\right) \end{array} \right) \tag{8}$$

$$F_2^{un}(\mathrm{X}) = F_2(\mathrm{X}) \times \alpha \left(\begin{array}{l} 1 + \sum_{j=1}^{n_{con}} \left(\max\left(C_{i,1}^{con}, 0\right) + \max\left(C_{i,2}^{con}, 0\right) + \max\left(C_{i,3}^{con}, 0\right) \right) \\ + \sum_{m=1}^{N_{str}} \max\left(C_m^{str}, 0\right) + \sum_{k=1}^{N_{node}} \max\left(C_{l,k}^{ser}, 0\right) \end{array} \right) \tag{9}$$

where α is the penalty parameter, which is used to add a heavy penalty to infeasible solutions. Since an individual having the greater objectives is removed in the optimization process, infeasible solutions will be eliminated from the population. In the current study, α is set to 10,000.

2.3. Direct analysis for steel frames

To evaluate the constraints presented in the above section, nonlinear inelastic analysis is applied using the Practical Advanced Analysis Program (PAAP), which can provide a robust framework for modeling and analyzing steel frames, capturing their complex responses more accurately than simplified approaches. As outlined in Thai and Kim (2009), PAAP integrates advanced numerical techniques to account for both material inelasticity and geometric nonlinearities, offering a practical alternative to computationally intensive FEA [27]. Central to PAAP's capability is its use of a beam-column model, which simulates the interaction between beams and columns in steel frames. This model discretizes frame members into only 1 to 2 elements that account for axial, flexural, and shear deformations, incorporating inelastic effects like plastic hinge formation at critical sections under extreme loads [27]. By employing stability functions and updated Lagrangian formulations, the beam-column approach in PAAP precisely captures second-order effects and load redistribution post-yielding, enhancing the accuracy of displacement and stress predictions [25, 27]. In PAAP, the Generalized Displacement Control (GDC) method, proposed by Yang and Shieh (1990), is used to efficiently solve nonlinear problems with multiple critical points [26, 27].

In the context of MOO for steel frames, as explored in various studies, PAAP's ability to model nonlinear inelastic behavior supports the development of hybrid optimization approaches like RDMO-XGBoost [6]. While PAAP provides reliable baseline analyses, its repeated use for large-scale designs can be time-consuming, prompting the integration of machine learning tools like XGBoost to predict responses rapidly after initial PAAP evaluations [29]. This combination aligns with the optimization goals of minimizing frame weight and controlling displacement, as seen in Nguyen et al. and Truong et al., enhancing both efficiency and accuracy [3]. By leveraging PAAP, researchers can validate designs against constructability constraints—such as ensuring upper columns have shallower depths than lower ones—while addressing the practical demands of cost, safety, and environmental impact [5, 31]. Overall, PAAP serves as a cornerstone for advancing the analysis and optimization of steel frames exhibiting nonlinear inelastic behavior, bridging traditional engineering with modern computational strategies.

3. The proposed RDMO-XGBoost method

3.1. RDMO

The RDMO algorithm, introduced by Nguyen, Pham, and Truong, merges the straightforward Rao-1 method with the powerful search capabilities of DE to address MOO challenges, such as optimizing truss structures for weight and displacement [6]. Its core principle lies in producing a diverse set of optimal solutions (Pareto front) by balancing conflicting goals without the burden of tuning complex control parameters, a common drawback in many MOO techniques. Unlike traditional Rao-1, which relies solely on the best and worst solutions, RDMO enhances exploration by using multiple high- and low-ranked solutions, evaluated via non-dominated sorting and crowding distance measures. This approach, combined with a DE crossover strategy, ensures a robust trade-off between solution diversity and convergence speed, outperforming established algorithms like DEMO and GDE3 in terms of coverage and accuracy, as demonstrated in tests on five truss structures.

Key advantages of RDMO include its simplicity—no algorithm-specific parameters need setting—making it user-friendly and adaptable, and its robustness, shown through consistent performance across various truss examples. The main formula updates a solution's position as follows:

$$\mathbf{x}_k^{new} = \mathbf{x}_k + \mathbf{r}_1 \left(\mathbf{x}_{rbest} - \mathbf{x}_{rworst}\right) \tag{10}$$

where $\mathbf{x}_{rbest}$ and $\mathbf{x}_{rworst}$ are randomly selected solutions among the best-ranked solutions and the worst-ranked solutions, respectively, defined by non-dominated sorting.

3.2. Integrating XGBoost surrogate model into RDMO with discrete variables

XGBoost is a powerful machine learning algorithm designed for fast and accurate predictions, widely used in optimization tasks like those for steel frames. Its core principle involves building an ensemble of decision trees sequentially, where each tree corrects the errors of its predecessors by minimizing a loss function, enhanced by gradient boosting techniques. Unlike traditional methods, XGBoost incorporates regularization to prevent overfitting and uses a scalable, tree-based structure to handle complex data patterns efficiently. Developed by Chen and Guestrin, it excels in speed due to parallel processing and optimized memory usage, making it ideal for predicting nonlinear inelastic behavior in structural analysis [13]. Key advantages include its high accuracy, ability to manage sparse data, and flexibility with various objective functions, offering significant computational savings when paired with optimization algorithms like RDMO, as noted in recent studies [3]. XGBoost improves the GTB framework by adding a regularization term to its loss function, curbing overfitting by penalizing model complexity. This term factors in the number of leaves, score vector, and leaf complexity. It uses Taylor expansion to refine the loss function and estimate the final ULF. Hyperparameters like tree type ('gbtree'), number of trees, max depth, learning rate, subsample ratio, and L1/L2 regularization (alpha, lambda) heavily influence performance. In this study, these were fine-tuned via Bayesian Optimization to balance accuracy and efficiency [29].

In this study, we propose a combination of RDMO and XGBoost for bi-objective optimization of steel frames with discrete variables, named as RDMO_XGBoost. The main step of RDMO_XGBoost are as follows:

Step 1: Begin

- Create an initial population **P** consisting of **NP** random solutions within the defined search space, where design variables are discrete and selected from the W-shaped section list (AISC-LRFD). Each solution xi x_i xi is a vector of integers representing section indices:

- $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, where $x_{ij} \in \{1, 2, \ldots, S_j\}$, and S_j is the number of sections for variable j.

- Use FEA evaluate all individuals.

- Apply non-dominated sorting to rank all solutions in P, forming a group B of top-ranked solutions (rank = 0) and a group W of the lowest-ranked solutions.

Step 2: Evolutionary process

- The RDMO mutation formula $u_i = \text{Round}(x_i + \text{random}()(x_{rbest} - x_{rworst}))$ is used with rounding to enforce discreteness. x_{rbest} and x_{rworst} are randomly chosen in **B** and **W**, respectively. Then, DE crossover is applied for finally creating new solutions **S**.

Step 2.1: When the conditions for using the XGBoost model are not yet satisfied:

Use FEA evaluate all individuals in **S**.

Apply non-dominated sorting and crowding distance for **Q** = **S** + **P** to form the new population **P'**.

Apply non-dominated sorting to rank all solutions in **P'**, forming a group **B** of top-ranked solutions (rank = 0) and a group W of the lowest-ranked solutions.

Step 2.2: When the conditions for using the XGBoost model are satisfied:

Use XGBoost predict constraints of individuals in **S** in order to find approximate objectives of individuals in **S**.

Apply non-dominated sorting and crowding distance for **Q** = **S** + **P** to form the new population **P'_pre**.

Identify and re-evaluate objectives of individuals in *S* which are chosen into **P'_pre** in the above step, defined as **S'**.

Apply non-dominated sorting and crowding distance for **Q** = **S'** + **P** to form the new population **P'**.

Apply non-dominated sorting to rank all solutions in **P**, forming a group **B** of top-ranked solutions (rank = 0) and a group W of the lowest-ranked solutions.

Step 3: Termination

A standard stopping criterion ensures convergence, common to both RDMO and metaheuristic methods.

Upon completion, the final population **P** forms the Pareto set of optimal solutions for the multi-objective optimization problem.

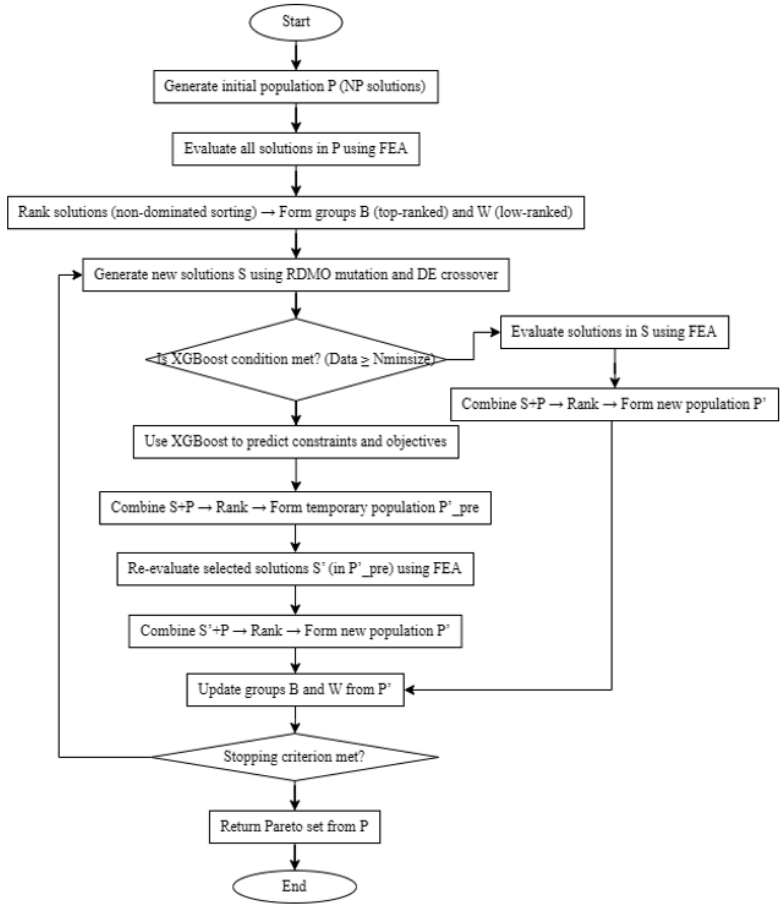


Figure 2. The flowchart of RDMO_XGBoost

In Step 2.2, the conditions for using the XGBoost model are defined as follows. The XGBoost surrogate model is initially built when $Data_i$ reaches a minimum sample size ($N_{minsize}$ = 1000), ensuring

acceptable accuracy. This model is rebuilt when $Data_i$ increases by N_s = 20 samples. Furthermore, a safety factor (τ, from Truong et al. [3]) is used to reduce XGBoost model error. In particular, the safety factor addresses potential inaccuracies in the XGBoost predictions, which may arise due to the complexity of nonlinear structural behavior, limited training data, or model overfitting. Specifically, τ is used to adjust the predicted values of critical parameters, such as the ultimate load factor (ULF) or displacements, to ensure conservative estimates that prioritize structural safety and feasibility.

In addition, in the proposed RDMO-XGBoost algorithm, XGBoost is employed as a surrogate model to predict the nonlinear inelastic responses of steel frames, significantly reducing the computational burden of repeated finite element analyses (FEA) while maintaining prediction accuracy. Specifically, XGBoost is used to estimate critical parameters associated with the constraint conditions of the bi-objective optimization problem, which include ultimate load factor of the structure in strength constraints and the horizontal displacement of the structual top in geometric constraints, as outlined in Section 2.2 of the manuscript. To enhance prediction accuracy and model robustness, a separate XGBoost model is trained for each distinct constraint parameter, allowing the algorithm to capture the unique relationships between design variables and individual constraint responses. For the XGBoost model applied to steel frames, the input variables corresponding to one type of frame element section consist of 16 geometric characteristics of that section. Therefore, the total number of input variables for the training model will be 16×D, where D is the number of design variables in the optimization problem. Details on setting up the ML model for estimating the behavior of steel frames can be found in Ref. [36]. Fig. 2 presents the procedure of RDMO_XGBoost algorithm.

4. Numerical example

This example optimizes a two-story, three-dimensional frame, as depicted in Fig. 3, targeting two objectives: (1) minimizing the structure's total mass and (2) reducing the top story's horizontal displacement. The frame's beams and columns are divided into eight groups, serving as design variables. Columns draw from 97 sections (W10, W12, W14, W16), while beams utilize 267 sections (W10–W44), yielding a design space exceeding 6.52E+17 combinations. Sections are ordered by cross-sectional area, with variables indicating their positions in this space.

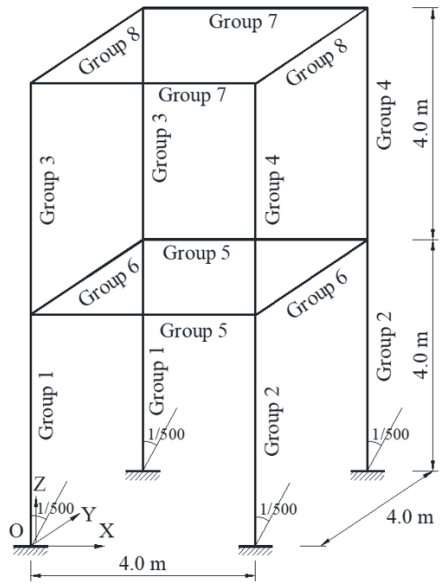


Figure 3. Two-story space frame

The material, A992 steel, has a yield strength of 344.7 MPa, an elastic modulus of 200,000 MPa, and a density of 7,850 kg/m^3. Wind loads of 50 kN act in the X-direction at front-face beam-column joints. Beams bear vertical dead and live loads of 37.5 kN/m and 25 kN/m, respectively. Three load combinations are applied: 1.2D + 1.6L, 1.2D + 0.5L + 1.6W, and 1.0D + 0.5L + 0.7W (D: dead, L: live, W: wind), with an inter-story drift limit of h/400.

Three multi-objective optimization (MOO) algorithms—GDE3, RDMO, and RDMO_XGBoost—are assessed. Implemented in Python, each uses a population of 50 over 300 iterations. Structural analysis relies on the PAAP program [27]. Performance metrics include GD+, IGD+, hypervolume

(HV), and spread, computed on a Core i7-8700 3.2 GHz system with 32 GB RAM. The GD+ metric measures the convergence of the obtained Pareto front $\mathrm{PF}_{\mathrm{obtained}}$ to the true or approximated Pareto front $\mathrm{PF}_{\mathrm{true}}$. It calculates the average distance from each point in $\mathrm{PF}_{\mathrm{obtained}}$ to its nearest point in $\mathrm{PF}_{\mathrm{true}}$, emphasizing how close the solutions are to the optimal set. A lower GD+ value indicates better convergence. The formula is:

$$\mathrm{GD+} = \frac{1}{|\mathrm{PF}_{\mathrm{obtained}}|}\left(\sum_{i=1}^{|\mathrm{PF}_{\mathrm{obtained}}|} d_i^{+2}\right)^{1/2} \tag{11}$$

where $|\mathrm{PF}_{\mathrm{obtained}}|$ is the number of solutions in the obtained Pareto front, $d_i^+ = \max\{a_i - z_i, 0\}$ represents the modified distance from a_i in $\mathrm{PF}_{\mathrm{obtained}}$ to its nearest reference point in $\mathrm{PF}_{\mathrm{true}}$ with the corresponding value z_i. The IGD+ metric evaluates both convergence and coverage by measuring the average distance from each point in $\mathrm{PF}_{\mathrm{true}}$ to its nearest point in $\mathrm{PF}_{\mathrm{obtained}}$. A lower IGD+ value indicates that the obtained solutions are both close to and well-distributed across the true Pareto front. The formula is:

$$\mathrm{IGD+} = \frac{1}{|\mathrm{PF}_{\mathrm{true}}|}\left(\sum_{i=1}^{|\mathrm{PF}_{\mathrm{true}}|} d_i^{+2}\right)^{1/2} \tag{12}$$

where $|\mathrm{PF}_{\mathrm{true}}|$ is the number of solutions in the true Pareto front, $d_i^+ = \max\{a_i - z_i, 0\}$ represents the modified distance from z_i in $\mathrm{PF}_{\mathrm{true}}$ to the closest solution in $\mathrm{PF}_{\mathrm{obtained}}$ with the corresponding value a_i. HV calculates the area/volume, which is dominated by the provided set of solutions with respect to a reference point. The Spread metric assesses the diversity and uniformity of the solutions in $\mathrm{PF}_{\mathrm{obtained}}$. It measures the distribution of solutions along the Pareto front by calculating the average deviation of distances between consecutive solutions relative to the ideal spacing. A lower Spread value indicates a more uniform distribution. The formula is:

$$\mathrm{Spread} = \left(\sum_{i=1}^{m} d_i^e + \sum_{x \in \mathrm{PF}_{\mathrm{obtained}}} \left|d\left(x\right) - d_{avg}\right|\right) / \left(\sum_{i=1}^{m} d_i^e + \mathrm{N}_{\mathrm{obtained}} \times d_{avg}\right) \tag{13}$$

where d_i^e is the distance between the extreme solutions of $\mathrm{PF}_{\mathrm{obtained}}$ and $\mathrm{PF}_{\mathrm{true}}$ for the i-th objective, $d(x)$ is the Euclidean distance from solution x to its nearest neighbor in $\mathrm{PF}_{\mathrm{obtained}}$, d_{avg} is the average distance between consecutive solutions, and $\mathrm{N}_{\mathrm{obtained}}$ is the number of solutions in $\mathrm{PF}_{\mathrm{obtained}}$.

Each algorithm ran 20 independent trials. The XGBoost variant, previously validated in [3, 22, 29], employs fixed hyperparameters: booster = 'gbtree', n_estimators = 1000, learning_rate = 0.05, and subsample = 0.5, with its robustness assumed and not re-evaluated here. The XGBoost model for nonlinear inelastic steel frames can be found in Ref. [23]. This analysis highlights the algorithms' effectiveness in balancing mass and displacement while navigating a vast design space under realistic loading conditions. Because the actual Pareto front for this multi-objective optimization (MOO) problem remains unidentified, an estimated version was created based on the outcomes of all optimization attempts. This approximated Pareto front is illustrated in Fig. 4.

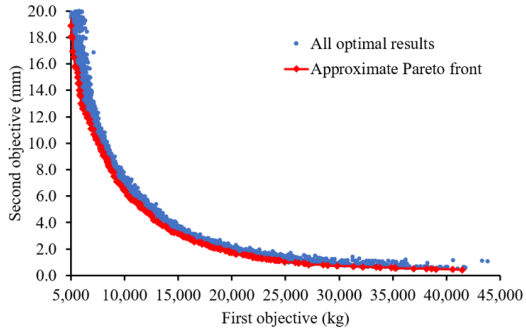
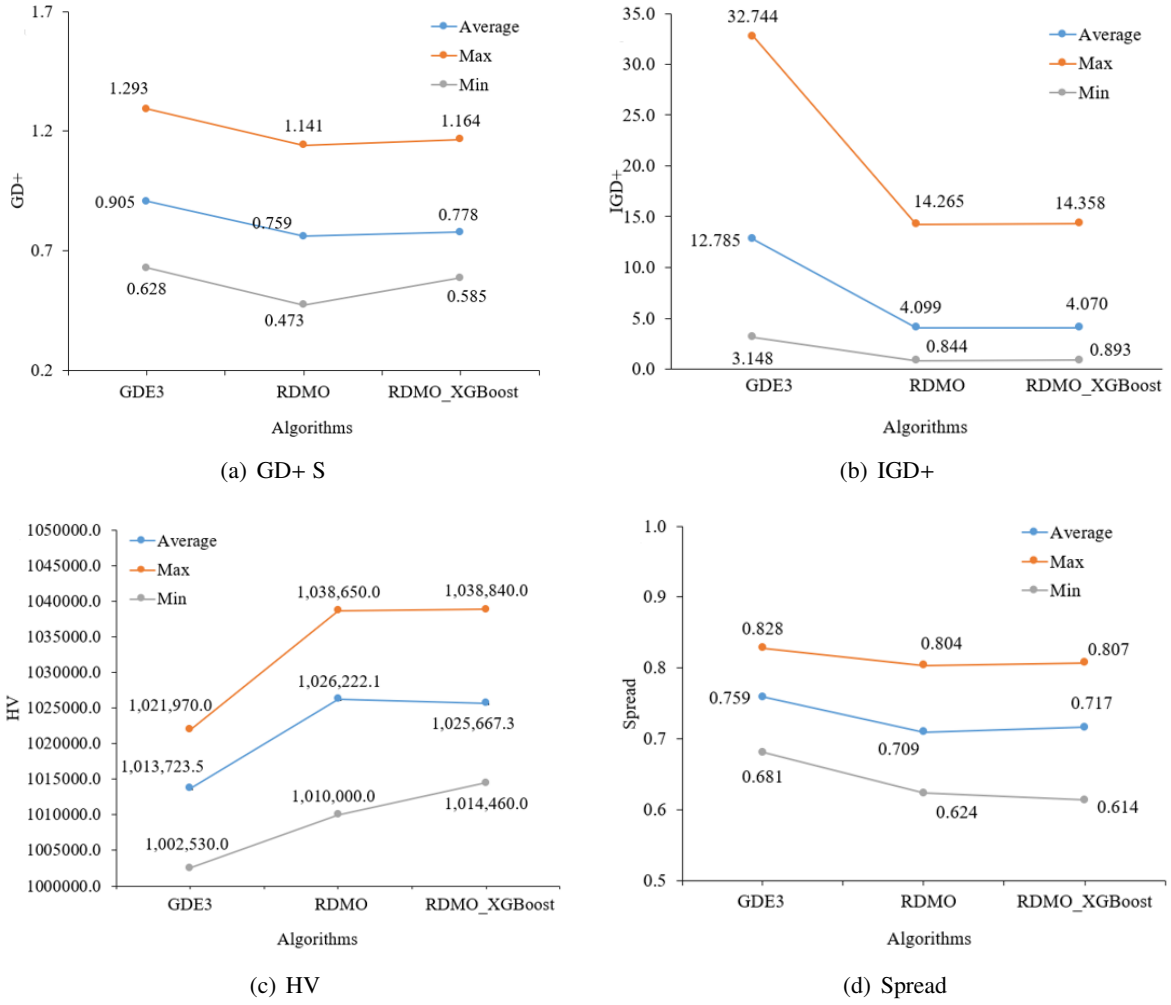


Figure 4. Approximate pareto-front

Fig. 5 shows the performance of the considered algorithms using GD+, IGD+, HV, and spread metrics. The performance of three multi-objective optimization algorithms—GDE3, RDMO, and RDMO_XGBoost—is evaluated using GD+, IGD+, HV, and Spread metrics, revealing distinct strengths and trade-offs.



(a) GD+ S (b) IGD+

(c) HV (d) Spread

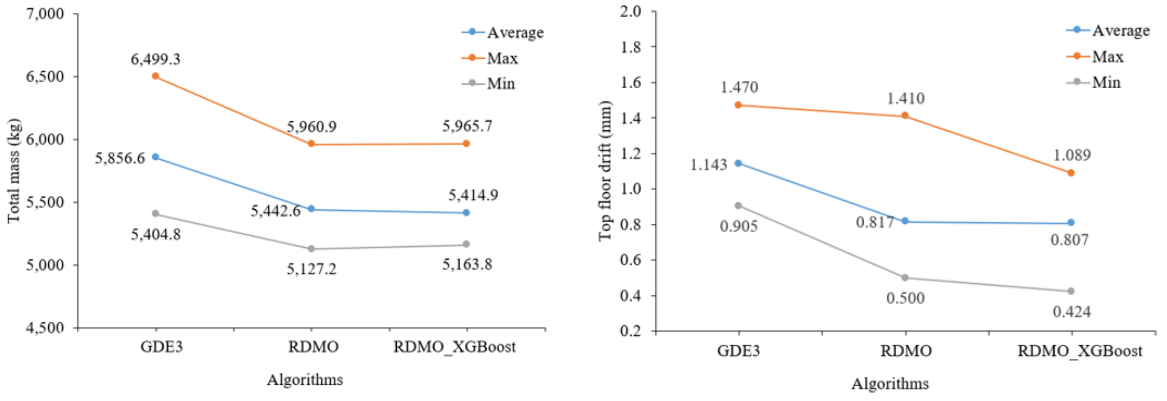
Figure 5. Indicators of MOO algorithms

The convergence and coverage performance of the algorithms are evaluated based on GD+, IGD+, and HV. As can be seen in Fig. 5, GDE3 exhibits the highest GD+ (Max: 1.293, Ave: 0.905), indicating poorer convergence to the Pareto front compared to RDMO (Max: 1.141, Ave: 0.759) and RDMO_XGBoost (Max: 1.164, Ave: 0.778). Similarly, GDE3's IGD+ (Max: 32.744, Ave: 12.785) is significantly higher than RDMO (Max: 14.265, Ave: 4.070) and RDMO_XGBoost (Max: 14.358, Ave: 4.081), suggesting weaker coverage of the Pareto front. However, GDE3 achieves a higher HV (Max: 1,021,970, Ave: 1,013,723.5), reflecting better overall solution quality compared to RDMO (Max: 1,038,650, Ave: 1,026,222.1) and RDMO_XGBoost (Max: 1,038,840, Ave: 1,025,667.3). This indicates that while GDE3 struggles with convergence, it excels in capturing a larger solution space, balancing mass and displacement effectively.

Regarding diversity of the algorithms, Spread values show GDE3 (Max: 0.828, Ave: 0.759)

has less uniform solution distribution than RDMO (Max: 0.804, Ave: 0.709) and RDMO_XGBoost (Max: 0.807, Ave: 0.717), suggesting RDMO variants provide more evenly distributed solutions. Stability, measured by standard deviation (Std), is lowest for RDMO across all metrics (e.g., GD+ Std: 0.146, IGD+ Std: 3.003), followed closely by RDMO_XGBoost (GD+ Std: 0.146, IGD+ Std: 4.081), while GDE3 shows higher variability (GD+ Std: 0.212, IGD+ Std: 8.439). Comparing RDMO and RDMO_XGBoost directly, their GD+ values are nearly identical (Ave: 0.759 vs. 0.778), and IGD+ scores are very close (Ave: 4.070 vs. 4.081), indicating that XGBoost integration does not compromise convergence or coverage. Spread values (Ave: 0.709 vs. 0.717) and HV (Ave: 1,026,222.1 vs. 1,025,667.3) further confirm that RDMO_XGBoost maintains comparable diversity and solution quality, proving that XGBoost does not degrade RDMO's performance. Consequently, RDMO and RDMO_XGBoost outperform GDE3 in convergence, coverage, and diversity, with RDMO almost similar to RDMO_XGBoost. GDE3, despite its higher HV, is less reliable for uniform and consistent optimization.

The results in Fig. 6 compare the performance of GDE3, RDMO, and RDMO_XGBoost for two anchor points: Anchor 1 (minimizing Objective 1, total structural mass) and Anchor 2 (minimizing Objective 2, top-story displacement). For Anchor 1, RDMO achieves the lowest mass (Min: 5127.200 kg, Ave: 5442.584 kg), followed closely by RDMO_XGBoost (Min: 5148.891 kg, Ave: 5414.072 kg), while GDE3 lags behind (Min: 5856.615 kg, Ave: 6493.900 kg). For Anchor 2, RDMO_XGBoost yields the smallest displacement (Min: 0.807 mm, Ave: 0.824 mm), slightly outperforming RDMO (Min: 0.817 mm, Ave: 0.834 mm), with GDE3 showing the highest values (Min: 1.143 mm, Ave: 1.171 mm). The close performance between RDMO and RDMO_XGBoost—evident in their minimal differences in both objectives (e.g., mass Ave difference: 28.512 kg, displacement Ave difference: 0.010 mm)—demonstrates that integrating XGBoost does not degrade RDMO's effectiveness, maintaining its robustness in optimizing both mass and displacement.



(a) First objective

(b) Second objective

Figure 6. Minimum values

Table 1 presents the computational efficiency of GDE3, RDMO, and RDMO_XGBoost for optimizing the two-story frame. GDE3 and RDMO, without surrogate assistance, demand 45,000 structural analyses, resulting in a total computation time of 50,400 seconds (100% ratio). Conversely, RDMO_XGBoost reduces the structural analysis count to 18,045, with an additional 1,521 seconds for XGBoost model building, yielding a total time of 22,370 seconds—equivalent to 44.39% of the baseline. This indicates that RDMO_XGBoost achieves a substantial time saving of over 55% compared to GDE3 and RDMO, highlighting the effectiveness of incorporating XGBoost surrogate models in

enhancing computational efficiency without compromising optimization outcomes.

Table 1. Time analysis for two-story frame

Optimization algorithm	FEA runs	XGBoost model training runs	Computing time (second)	Ratio
GDE3, RDMO	45,000	0	50,400	100.00%
RDMO_XGBoost	18,045	1,521	22,370	44.39%

5. Conclusions

This study successfully developed and validated a hybrid RDMO-XGBoost approach for multi-objective optimization of nonlinear steel frames, focusing on minimizing structural mass and top-story displacement. The integration of RDMO with XGBoost significantly enhanced computational efficiency, reducing analysis time by over 55% (from 50,400 to 22,370 seconds) compared to GDE3 and standalone RDMO, while maintaining solution quality. Performance evaluations revealed that RDMO and RDMO_XGBoost outperformed GDE3 in convergence (GD+ Ave: 0.759 and 0.778 vs. 0.905), coverage (IGD+ Ave: 4.070 and 4.081 vs. 12.785), and diversity (Spread Ave: 0.709 and 0.717 vs. 0.759), with RDMO_XGBoost showing nearly identical effectiveness to RDMO, confirming that XGBoost integration does not compromise RDMO's robustness. Anchor point analysis further highlighted their superiority, achieving lower mass (RDMO: 5127.200 kg, RDMO_XGBoost: 5148.891 kg) and displacement (RDMO_XGBoost: 0.807 mm, RDMO: 0.817 mm) compared to GDE3. However, the study is limited by its focus on static loads, potentially overlooking dynamic effects like seismic loading. Future work will explore these dynamic conditions and extend the approach to larger-scale steel structures.

Acknowledgements

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.01-2021.14.

References

[1] Saka, M. P., Hasançebi, O., Eser, H., Geem, Z. W. (2024). Historical evolution of structural optimization techniques for steel skeletal structures including industrial design applications. *Engineering Optimization*, 57(1):69–129.

[2] Li, L., Khandelwal, K. (2017). Topology optimization of geometrically nonlinear trusses with spurious eigenmodes control. *Engineering Structures*, 131:324–344.

[3] Nguyen, M.-C., Ha, M.-H., Nguyen, N.-T., Dinh, V.-T., Truong, V.-H. (2025). A robust XGBoost-based multi-objective algorithm for nonlinear truss structures. *Journal of Science and Technology in Civil Engineering, HUCE*, 19(1):131–141.

[4] Degertekin, S. O. (2012). Improved harmony search algorithms for sizing optimization of truss structures. *Computers & Structures*, 92–93:229–241.

[5] Truong, V.-H., Hung, H. M., Anh, P. H., Hoc, T. D. (2020). Optimization of steel moment frames with panel-zone design using an adaptive differential evolution. *Journal of Science and Technology in Civil Engineering (STCE) - NUCE*, 14(2):65–75.

[6] Nguyen, M.-C., Pham, H.-A., Truong, V.-H. (2025). An efficient multi-objective algorithm based on Rao and differential evolution for solving bi-objective truss optimization. *Engineering Optimization*, 1–31.

[7] Storn, R., Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

[8] Geem, Z. W., Kim, J. H., Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2):60–68.

[9] Fourie, P. C., Groenwold, A. A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, 23(4):259–267.

[10] Kaveh, A., BolandGerami, A. (2016). Optimal design of large-scale space steel frames using cascade enhanced colliding body optimization. *Structural and Multidisciplinary Optimization*, 55(1):237–256.

[11] Kaveh, A., Laknejadi, K. (2011). A hybrid multi-objective particle swarm optimization and decision-making procedure for optimal design of truss structures. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 35(C2):137–154.

[12] Truong, V.-H., Tangaramvong, S., Pham, H.-A., Nguyen, M.-C., Su, R. (2025). An efficient archive-based parameter-free multi-objective Rao-DE algorithm for bi-objective optimization of truss structures. *Computers & Structures*, 308:107647.

[13] Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, 785–794.

[14] Rao, R. V. (2020). Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *International Journal of Industrial Engineering Computations*, 11:107–130.

[15] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*, Springer Berlin Heidelberg, 849–858.

[16] Kaveh, A., Mahdavi, V. R. (2018). Multi-objective colliding bodies optimization algorithm for design of trusses. *Journal of Computational Design and Engineering*, 6(1):49–59.

[17] Robič, T., Filipič, B. (2005). DEMO: Differential evolution for multiobjective optimization. In *The Third International Conference on Evolutionary Multi-Criterion Optimization*, Springer Berlin Heidelberg, 520–533.

[18] Kukkonen, S., Lampinen, J. (2005). GDE3: The third evolution step of generalized differential evolution. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, IEEE, 443–450.

[19] Tian, Y., Zhang, X., Cheng, R., Jin, Y. (2016). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 5222–5229.

[20] Pham, H.-A., Tran, T.-D. (2022). Optimal truss sizing by modified Rao algorithm combined with feasible boundary search method. *Expert Systems with Applications*, 191:116337.

[21] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

[22] Nguyen, H.-H., Truong, V.-H. (2024). Machine Learning-based prediction of seismic lateral deflection of steel trusses using nonlinear time-history analysis. *Structures*, 69:107369.

[23] Truong, V.-H., Pham, H.-A., Tangaramvong, S. (2025). An efficient method for nonlinear inelastic truss optimization based on improved k-nearest neighbor comparison and Rao algorithm. *Structures*, 71:108158.

[24] Cao, T.-S., Pham, H.-A., Truong, V.-H. (2024). An efficient algorithm for multi-objective structural optimization problems using an improved pbest-based differential evolution algorithm. *Advances in Engineering Software*, 197:103752.

[25] Blandford, G. E. (1996). Progressive failure analysis of inelastic space truss structures. *Computers & Structures*, 58(5):981–990.

[26] Yang, Y.-B., Shieh, M.-S. (1990). Solution method for nonlinear problems with multiple critical points. *AIAA Journal*, 28(12):2110–2116.

[27] Thai, H.-T., Kim, S.-E. (2009). Practical advanced analysis software for nonlinear inelastic analysis of space steel structures. *Advances in Engineering Software*, 40(9):786–797.

[28] Pham, H.-A., Dang, V.-H., Vu, T.-C., Nguyen, B.-D. (2024). An efficient k-NN-based Rao optimization method for optimal discrete sizing of truss structures. *Applied Soft Computing*, 154:111373.

[29] Le, D.-N., Pham, T.-H., Papazafeiropoulos, G., Kong, Z., Tran, V.-L., Vu, Q.-V. (2024). Hybrid machine learning with Bayesian optimization methods for prediction of patch load resistance of unstiffened plate girders. *Probabilistic Engineering Mechanics*, 76:103624.

[30] Ho-Huu, V., Hartjes, S., Visser, H. G., Curran, R. (2018). An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization. *Expert Systems with Applications*, 92:430–446.

[31] Kaveh, A., Laknejadi, K. (2012). A hybrid evolutionary graph-based multi-objective algorithm for layout optimization of truss structures. *Acta Mechanica*, 224(2):343–364.

[32] Gholizadeh, S., Fattahi, F. (2019). Multi-objective design optimization of steel moment frames considering seismic collapse safety. *Engineering with Computers*, 37(2):1315–1328.

[33] Kaveh, A., Ilchi Ghazaan, M. (2019). A new VPS-based algorithm for multi-objective optimization problems. *Engineering with Computers*, 36(3):1029–1040.

[34] Panagant, N., Pholdee, N., Bureerat, S., Yildiz, A. R., Mirjalili, S. (2021). A comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems. *Archives of Computational Methods in Engineering*, 28(5):4031–4047.

[35] Kong, Z., Le, D.-N., Pham, T.-H., Pologanathan, K., Papazafeiropoulos, G., Vu, Q.-V. (2024). Hybrid machine learning with optimization algorithm and resampling methods for patch load resistance prediction of unstiffened and stiffened plate girders. *Expert Systems with Applications*, 249:123806.

[36] Truong, V. H., Tangaramvong, S., Nguyen, M. C., Pham, H. A. (2025). Machine learning-based safety assessment of steel frames under seismic loadings using nonlinear time-history analysis. *Steel and Composite Structures*, 54(4):295–312.