Journal of Science and Technology in Civil Engineering, HUCE, 2025, 19 (1): 131–141

A ROBUST XGBOOST-BASED MULTI-OBJECTIVE OPTIMIZATION ALGORITHM FOR NONLINEAR TRUSS STRUCTURES

Manh-Cuong Nguyen[a], Manh-Hung Ha[b], Ngoc-Thang Nguyen[a], Van-Thuat Dinh[b], Viet-Hung Truong[a,*]

[a] *Faculty of Civil Engineering, Thuyloi University, 175 Tay Son, Dong Da, Hanoi, Vietnam*
[b] *Faculty of Building and Industrial Construction, Hanoi University of Civil Engineering, 55 Giai Phong road, Hanoi, Vietnam*

Article history:
Received 27/02/2025, Revised 07/3/2025, Accepted 25/3/2025

Abstract

This paper presents MOEA/D-EpDE_XGBoost, a novel multi-objective optimization (MOO) algorithm designed for efficient and accurate design optimization of nonlinear inelastic steel truss structures. The algorithm integrates a gradient boosting machine learning model (XGBoost) with a dynamic resource allocation multi-objective evolutionary algorithm (MOEA/D-DRA) and an improved pbest-based Differential Evolution (EpDE) algorithm. XGBoost serves as a surrogate model for computationally expensive finite element analyses (FEA), significantly reducing computational costs while maintaining solution accuracy. The performance of MOEA/D-EpDE_XGBoost is compared against five other established MOO algorithms (NSGA2, SPEA2, GDE3, MOEA/D, and a standard ME algorithm) using a 47-bar powerline truss benchmark problem. Results demonstrate that the proposed algorithm achieves superior convergence, diversity, and computational efficiency compared to existing algorithms, while maintaining solution quality.

Keywords: multi-objective optimization; inelastic analysis; metaheuristic; XGBoost; MOEA/D; EpDE.

https://doi.org/10.31814/stce.huce2025-19(1)-11

1. Introduction

The design optimization of truss structures, especially when considering nonlinear inelastic material behavior, presents a significant computational challenge [1–3]. Traditional methods, heavily reliant on computationally expensive finite element analyses (FEA) to determine structural responses under diverse loading conditions, become prohibitively time-consuming [1–3]. This is particularly true within the framework of multi-objective optimization (MOO), where the goal is to identify a set of optimal designs that effectively balance competing design criteria, rather than a single, absolute optimum [4–6]. The inherent complexities of nonlinear inelastic analysis, compounded by the need for extensive exploration of the design space necessary to locate Pareto optimal solutions, result in substantial computational costs. As a result, traditional MOO algorithms are often impractical for real-world applications involving large-scale truss structures. This necessitates the development of innovative and efficient optimization strategies that effectively manage computational resources while maintaining solution quality.

A promising avenue for improving the efficiency of MOO in this context is the integration of machine learning (ML) algorithms. ML models can serve as surrogate models, efficiently approximating the complex relationship between design variables and structural responses. This reduces reliance on computationally intensive FEA. By training an ML model on a relatively small dataset of FEA results,

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

engineers can then rapidly predict structural responses for a multitude of designs, leading to a significant reduction in overall computational cost [7–11]. However, the effectiveness of this ML-based surrogate modeling approach hinges critically on the choice of an appropriate ML algorithm capable of accurately capturing the nonlinearities inherent in both material behavior and structural response, while maintaining computational efficiency.

This research utilizes XGBoost [12], a robust and highly efficient gradient boosting machine learning algorithm, to create accurate surrogate models for the computationally expensive FEA. XGBoost's ability to effectively handle complex datasets and nonlinear relationships, coupled with its demonstrated high accuracy and efficiency, makes it a highly suitable candidate for this application. This ML surrogate model is then integrated within a novel MOO algorithm, MOEA/D-EpDE, designed to efficiently and effectively address the multi-objective optimization problem [6]. MOEA/D-EpDE combines the the multi-objective evolutionary algorithm based on decomposition with dynamical resource allocation called MOEA/D_DRA [13] and an improved pbest-based Differential Evolution (EpDE) algorithm [3] to achieve a superior balance between solution quality and computational efficiency.

The MOEA/D algorithm [14] employs a decomposition-based approach, transforming the complex multi-objective optimization problem into a set of simpler single-objective subproblems. This strategy offers several advantages: more efficient exploration of the design space, improved computational efficiency through distributed computational load, and the ability to maintain solution diversity and high quality [15–17]. The selection of EpDE [3, 6, 7] to solve these single-objective subproblems is strategic due to its dynamic mutation strategy. This strategy effectively balances exploration and exploitation, facilitating faster convergence to the Pareto front while preserving diversity in the solution set.

In essence, this paper introduces MOEA/D-EpDE_XGBoost, a novel algorithm that effectively integrates a powerful machine learning model (XGBoost) with an advanced metaheuristic algorithm (EpDE) within the MOEA/D_DRA framework to address the computational challenges of nonlinear inelastic truss optimization. This integrated approach is expected to yield substantial improvements in both efficiency and solution quality compared to traditional methods, providing a robust and powerful tool for engineers to perform efficient multi-objective design optimization. Subsequent sections will detail the algorithm's methodology, present results from a bi-objective optimization of a 47-bar power line, and discuss the algorithm's performance and implications for future research in structural optimization.

2. Bi-Objective Optimization of Nonlinear Inelastic Truss Structures

The design of truss structures often involves competing objectives, demanding a balanced approach to optimize both cost-effectiveness and structural integrity. This study addresses this challenge by formulating a bi-objective optimization problem for nonlinear inelastic truss structures. This approach acknowledges the inherent complexities of material behavior under significant loading and geometric nonlinearities, going beyond simplified linear elastic assumptions. The two primary objectives are:

2.1. Minimization of structural mass

The first objective function focuses on minimizing the total mass (or equivalently, cost, assuming a linear relationship between mass and material cost) of the structure. This is achieved by optimizing the cross-sectional areas of individual truss members. Reducing material usage is crucial for economic

viability and minimizing environmental impact. The objective function is mathematically represented as:

$$\text{Minimize } F_1(X) = \rho \sum A_{ij} L_{ij} \tag{1}$$

where ρ is the material density, A_{ij} is the cross-sectional area of the j-th member in the i-th member group, L_{ij} is the length of the j-th member in the i-th member group, i indexes the member groups, and j indexes the members within each group. This objective directly encourages the selection of smaller cross-sections, however, it's crucial to note that this can potentially compromise structural performance if not properly constrained.

2.2. Minimization of maximum displacement

The second objective focuses on ensuring structural integrity and safety by minimizing the maximum displacement under specified loading conditions. Excessive displacement can lead to structural failure, loss of functionality, or unacceptable levels of vibration. This objective is expressed mathematically as:

$$\text{Minimize } F_2(X) = \max(|\delta_k|) \tag{2}$$

where δ_k represents the displacement of node k, and k indexes the nodes in the structure. This objective implicitly promotes the use of larger cross-sections and a more robust structural design to withstand loading and maintain acceptable deformation limits.

2.3. Constraint handling

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

Strength Constraints: These constraints ensure that the structural load-carrying capacity R is greater than applied loading S. These constraints are typically evaluated through nonlinear finite element analysis, accounting for material inelasticity and geometric nonlinearities in the form:

$$C^{str} = 1 - \frac{R}{S} = 1 - ULF \leq 0 \tag{3}$$

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

Geometric Constraints: These constraints limit the displacement of the nodes as:

$$C^{ser} = \frac{|\Delta|}{\Delta^u} - 1 \leq 0 \tag{4}$$

where Δ and Δ^u are a nodal displacement and its allowable value, respectively.

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:

$$F_1^{un}(X) = F_1(X) \times \left(1 + \sum_{m=1}^{N_{str}} \alpha_{str,m} \max\left(C_m^{str}, 0\right) + \sum_{l=1}^{N_{ser}} \alpha_{ser,l} \sum_{k=1}^{N_{node}} \max\left(C_{l,k}^{ser}, 0\right)\right) \tag{5}$$

$$F_2^{un}(X) = F_2(X) \times \left(1 + \sum_{m=1}^{N_{str}} \alpha_{str,m} \max\left(C_m^{str}, 0\right) + \sum_{l=1}^{N_{ser}} \alpha_{ser,l} \sum_{k=1}^{N_{node}} \max\left(C_{l,k}^{ser}, 0\right)\right) \tag{6}$$

where $\alpha_{str,m}$ and $\alpha_{ser,l}$ are penalty parameters. A sufficiently large value of α ensures that infeasible solutions are penalized heavily, favoring feasible alternatives. In this work, α is defined as 10,000.

2.4. Direct analysis model for steel trusses

This paper employs a direct analysis model for steel truss structures to accurately predict structural response under various loading conditions. This approach is necessary because both strength and serviceability requirements necessitate a detailed consideration of the structural behavior. Unlike simplified linear elastic models, this direct analysis incorporates nonlinear inelastic material behavior and geometric nonlinearities, particularly crucial for assessing strength under significant loads. For serviceability limit states, where deformations are generally smaller and remain within the elastic range, a nonlinear elastic analysis is sufficient.

The core of the direct analysis hinges on the Blandford [18] stress-strain constitutive model (shown in Fig. 1). This model meticulously captures the complex behavior of steel, including elastic and inelastic post-buckling, as well as unloading characteristics. The model utilizes parameters (X_1, X_2) dependent on the element's slenderness ratio (L/r) to define the transition between elastic and inelastic regions. These parameters, along with the yield stress σ_γ) and yield strain ε_γ), and Euler buckling stress and strain (σ_{cr}, ε_{cr}), fully characterize the material's response under various loading scenarios.

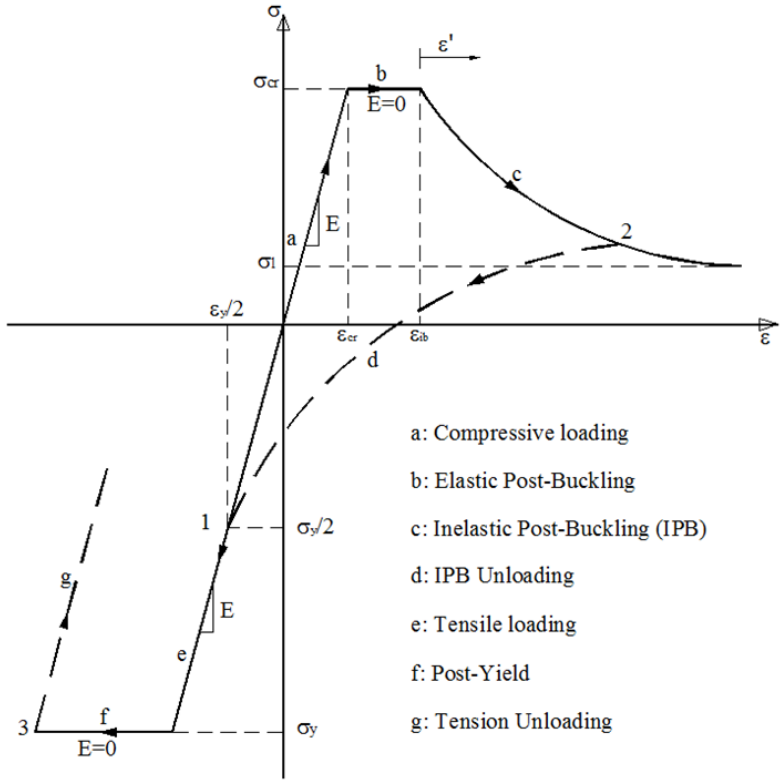


Figure 1. Blandford's stress-strain constitutive model for nonlinear truss elements

To solve the nonlinear equilibrium equations resulting from this constitutive model, the generalized displacement control (GDC) method [19] is implemented. The GDC method offers several advantages: automatic step size adjustment, self-adaptation to loading direction changes, and enhanced stability near critical points. This robust method ensures accurate and efficient solution of the nonlinear system. The direct analysis incorporating the Blandford model and the GDC method is performed using the Practical Advanced Analysis Program (PAAP) [20], a well-established software

package suitable for this type of advanced analysis. The results from this detailed analysis inform the optimization process and ensure the reliability of the final design.

3. Enhancing MOEA/D-EpDE with XGBoost

3.1. MOEA/D-EpDE

MOEA/D-EpDE, presented by Cao et al. [6], offers a novel approach to multi-objective optimization (MOO), particularly effective for complex problems. This algorithm cleverly integrates two powerful techniques: MOEA/D_DRA [13] and EpDE [3]. MOEA/D_DRA provides a decomposition framework, transforming the initial multi-objective problem into a set of simpler, single-objective subproblems. Each subproblem is then independently optimized using the robust EpDE algorithm.

A significant advantage of MOEA/D-DRA is its dynamic resource allocation. Unlike traditional MOEA/D, which distributes resources evenly, MOEA/D-DRA adapts resource allocation based on a complexity coefficient (π_i) that reflects each subproblem's convergence rate. This adaptive strategy focuses computational power on the most promising areas of the search space, leading to increased efficiency

The EpDE algorithm employs a 'DE/pbest/1' mutation strategy, dynamically balancing exploration and exploitation. A dynamically adjusted parameter 'p' controls the selection of the best individuals (pbest) used for mutation, allowing EpDE to adapt to the evolving search landscape. This, combined with controlled crossover rates and scaling factors, enables efficient navigation of complex search spaces. Furthermore, a dynamically updated external archive stores promising Pareto optimal solutions, preventing premature convergence and ensuring a diverse and high-quality final solution set. This combination of features makes MOEA/D-EpDE a highly effective algorithm for challenging MOO problems.

The MOEA/D-EpDE algorithm proceeds as follows:

- **Initialization**: Randomly generate initial population and weight vectors; initialize external archive.
- **Subproblem Selection**: Based on π_i, select subproblems for optimization
- **EpDE Optimization**: Apply EpDE to each selected subproblem, generating trial vectors.
- **Update**: Replace inferior solutions with superior trial vectors; update external archive.
- **Update** π_i: Adjust π_i periodically to adapt resource allocation.
- **Termination**: Continue until termination criteria are met.

MOEA/D-EpDE's integration of dynamic resource allocation and the 'DE/pbest/1' mutation strategy enhances exploration and exploitation, leading to improved convergence, diversity, and superior performance compared to other MOO algorithms [6]. The details of this algorithm can be found in Ref. [6].

3.2. XGBoost-based surrogate for predicting the ultimate load factor of truss structures

This paper utilizes XGBoost, a gradient boosting algorithm, to create a surrogate model for efficiently predicting the ultimate load factor (ULF) of truss structures. XGBoost, pioneered by Chen and Guestrin [12], has demonstrated exceptional performance in various machine learning challenges, including the Higgs Boson competition. Its underlying principle is similar to gradient tree boosting (GTB) [21], iteratively combining weak learning models (decision trees) into a strong ensemble model.

The XGBoost algorithm operates sequentially. Initially, a base learner (a decision tree) is trained to predict the ULF. The difference between the predicted values and the actual ULF values, termed the residual, represents the model's error. To improve accuracy, a second decision tree is trained to

predict this residual, effectively correcting the initial model's errors. This iterative process continues, creating a sequence of trees, where each subsequent tree aims to reduce the residual error of the preceding ensemble. The final prediction is the sum of the predictions from each individual tree in the ensemble.

To enhance the GTB framework, XGBoost incorporates a regularization term into its loss function. This regularization term penalizes model complexity, preventing overfitting. The regularization term includes parameters such as the number of leaves, the score vector, and the complexity of leaves to control the model's complexity. Taylor expansion is used to optimize the loss function, and the final estimated ULF is calculated. The choice of hyperparameters, such as the tree type ('gbtree' in this case), the number of trees (n_estimators), maximum tree depth (max_depth), learning rate (learning_rate), subsample ratio (subsample), L1 regularization (alpha), and L2 regularization (lambda), significantly impacts the model's performance and requires careful consideration. In this study, specific values for these hyperparameters were selected to achieve an optimal balance between accuracy and computational efficiency using Bayesian Optimization [22].

3.3. Integrating XGBoost surrogate model into MOEA/D-EpDE

The MOEA/D-EpDE_XGBoost algorithm, hereafter referred to as MEX, integrates the XGBoost surrogate model into the MOEA/D-EpDE framework to enhance computational efficiency while maintaining solution quality. The MEX algorithm can be structured into these main steps:

Phase 1: Initialization and Problem Decomposition

Initialization: Generate initial population (P^o), weight vectors (λ_i), and initialize the external archive (Ex)

Decomposition: Decompose the multi-objective problem into single-objective subproblems using the Tchebycheff method and weight vectors [13]. Define the neighborhood for each subproblem.

Data Initialization: For each subproblem, store initial individuals and their corresponding outputs (from load combinations) in a data matrix ($Data_i$).

Phase 2: Iterative Optimization

Subproblem Selection: Select a subset of subproblems for optimization based on the dynamic complexity coefficient (π_i)

Solution Generation (EpDE): Generate a potential solution (U) for each selected subproblem using the EpDE algorithm's 'DE/pbest/1' mutation strategy.

Objective Function Evaluation: If sufficient data exists ($Data_i \geq N_{minsize}$), use the XGBoost surrogate model to evaluate the objective function for U. Otherwise, perform nonlinear inelastic analysis.

Update: Update the population, external archive (Ex), and the reference point (Z) based on the evaluated objective functions of U. Update the neighboring set (B_i) if necessary.

Update π_i: Update the complexity coefficient π_i) periodically to adjust resource allocation.

Phase 3: Termination

Termination: Repeat steps 4-8 until a stopping criterion is met (e.g., maximum number of generations or convergence). Select the final Pareto-optimal solution set from the final population or the external archive.

In Step 6, The XGBoost surrogate model for subproblem i 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 $N_s = 50$ samples. A small N_s increases training frequency but may consume excessive computation time. Conversely, a large N_s risks false predictions and local optima. Finally, a safety factor τ, from Truong et al. [9]) is used to reduce XGBoost model error. The term of neighboring set can be found in Ref. [7].

4. Numerical example

This section evaluates the performance of six MOO algorithms—NSGA2, GDE3, SPEA2, MOEA/D, MOEA/D-EpDE (ME), and MOEA/D-EpDE_XGBoost (MEX)—on one planar steel truss structure (47-bar power line) using A992 steel. Member cross-sectional areas, ranging from 645.16 mm^2 to 6451.6 mm^2, served as design variables. All algorithms were implemented in Python, using a population size of 100 and 300 iterations with archive sizes for ME and MEX of 200. A maximum of 300 iterations was selected based on preliminary convergence studies, which indicated that performance metrics (e.g., IGD+ and HV) stabilized within 5% of their final values after approximately 250 iterations across all tested algorithms, ensuring sufficient exploration while managing computational cost. Structural analysis was conducted using the PAAP program [20]. Performance was assessed using generational distance (GD), GD+, IGD+, and hypervolume (HV) [6], and evaluated on a computer with a Core i7-8700 3.2GHz processor and 32GB RAM. The robustness of XGBoost, previously validated in [9, 10, 22, 23], is not further explored in this work. Its hyperparameters are selected as: booster = 'gbtree', n_estimators = 500, learning_rate = 0.05, and subsample = 0.5. The hyperparameters of XGBoost were determined using Bayesian Optimization, which iteratively balances exploration and exploitation to minimize the mean squared error over a predefined hyperparameter search space. This probabilistic approach efficiently identifies optimal settings by modeling the objective function with a surrogate (Gaussian Process) and updating it based on prior evaluations, as detailed in Ref. [22].

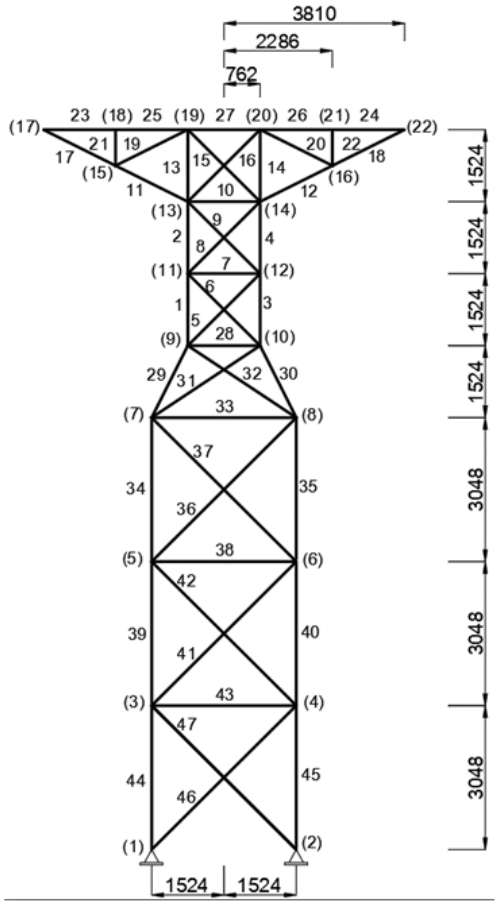


Figure 2. 47-bar powerline truss

Fig. 2 illustrates a 47-bar powerline truss model, incorporating 27 distinct member cross-sections. Three loading scenarios were analyzed: $1.2D + 1.6L + 0.5W$, $1.2D + 1.6L$, and $1.0D + 0.5L + 0.7W$, where D is the dead load (70 kN at all nodes), L is the live load (50 kN at all nodes), and W is the wind load (30 kN at nodes 17 and 22 along the X-axis). Optimization targeted minimum structural mass (objective function 1) and minimized lateral drift (objective function 2). Objective function 2 was defined as the root-sum-square of horizontal deflections (x_i) at nodes i = 3, 5, 7, 9, 11, 13, 15, and 17, constrained by a maximum allowable drift of h/400 (where h is the truss story height). Each optimization algorithm was executed independently for 20 runs.

Since the true Pareto front for this MOO problem is unknown, an approximation was generated using the results from all optimization runs. Fig. 3 displays this approximate Pareto front.

Fig. 4 shows the performance of the considered algorithms using GD, GD+, IGD+, and HV metrics. The ME algorithm achieved the best IGD+ and HV scores. Specifically, ME's average IGD+

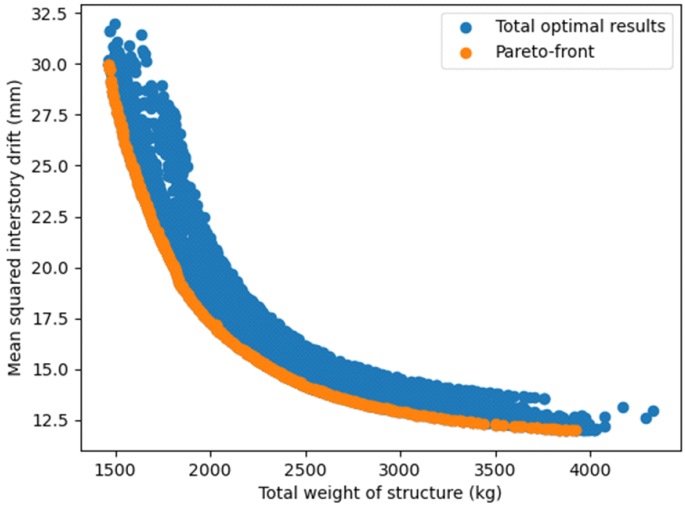
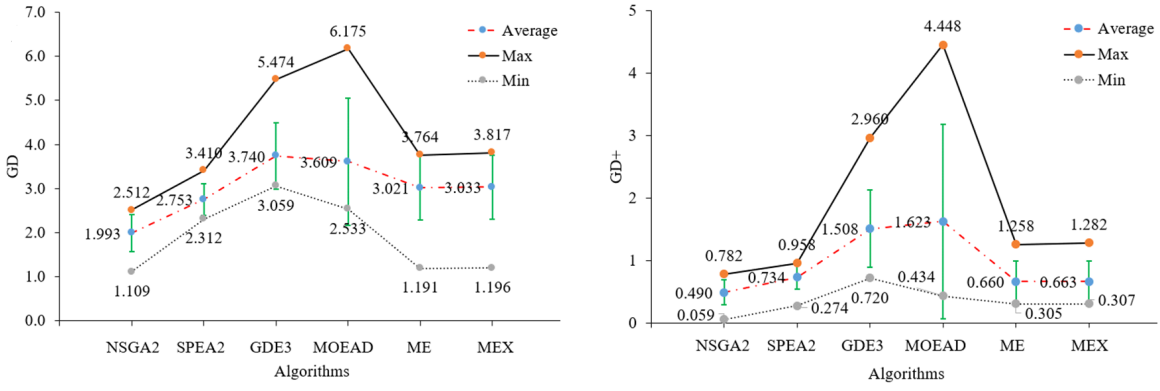


Figure 3. Approximate pareto-front

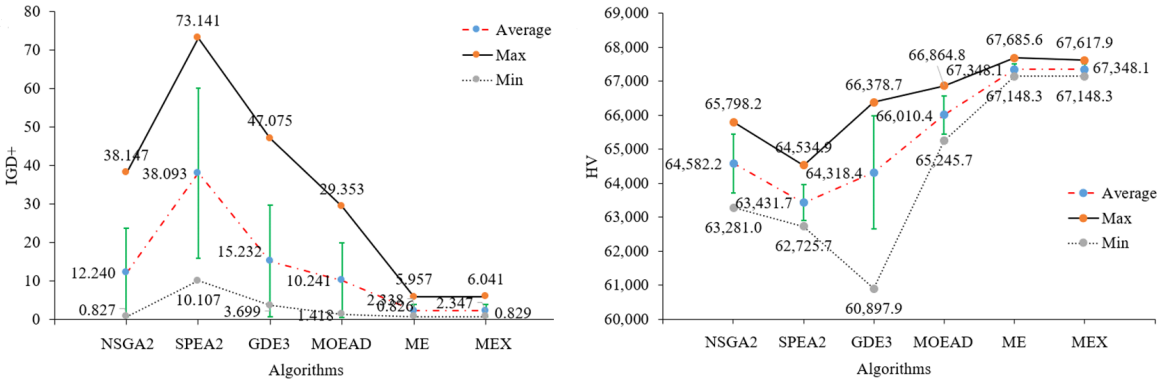
value (2.338) significantly outperformed those of NSGA2 (12.240), SPEA2 (38.093), GDE3 (15.232), and MOEA/D (10.241), indicating superior convergence and diversity. Similarly, ME's average HV score (67,381.1) exceeded those of NSGA-II (64,582.2), SPEA2 (63,431.7), GDE3 (64,318.4), and MOEA/D (66,010.4). While ME also showed superior GD and GD+ scores compared to GDE3 and MOEA/D, it had slightly higher scores than NSGA2. It is important to note that GD and GD+ primarily assess proximity to the Pareto front, not solution diversity.

The MEX algorithm, an enhanced version of ME incorporating an XGBoost model to reduce computational cost, aimed to maintain ME's performance while improving efficiency. Fig. 4 shows that the differences between MEX and ME in GD (3.033 vs. 3.021), GD+ (0.663 vs. 0.660), IGD+ (2.347 vs. 2.338), and HV (67348.1 vs. 67148.3) are deemed negligible, with relative differences less than 1% and p-values exceeding 0.05 (via paired t-tests across 20 runs), indicating no statistically significant distinction



(a) GD (b) GD+

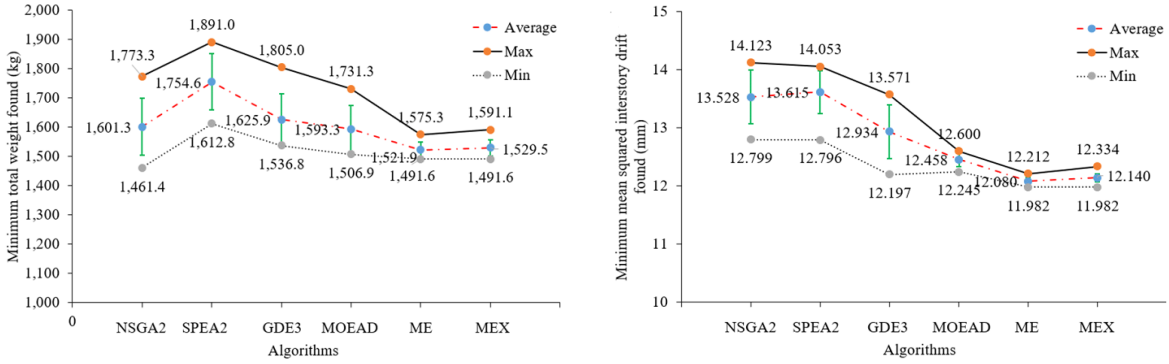
Fig. 5 presents the optimal solutions obtained for both objective functions. The ME method achieved the lowest total structural weight (1491.6 kg), SPEA2 (1612.8 kg), GDE3 (1536.8 kg), and MOEA/D (1506.9 kg), although slightly exceeding NSGA-2's result (1461.4 kg) (see Fig. 5(a)). Importantly, the ME's average minimum weight (1521.9 kg) was substantially lower than that of



(c) IGD+ (d) HV

Figure 4. Indicators of MOO algorithms

the other algorithms. In terms of the second objective function (maximum displacement), the ME method yielded the lowest average, best, and worst values (12.080 mm, 11.982 mm, and 12.212 mm, respectively). These results highlight the ME algorithm's robustness and its ability to generate a well-distributed set of optimal solutions. The proposed method (MEX) demonstrated performance closely matching that of the ME algorithm, achieving identical minimum values for both objective functions. While MEX showed slightly higher average values than ME, the differences are not statistically significant and demonstrate that MEX maintains the robust performance of the ME algorithm.



(a) First objective (b) Second objective

Figure 5. Minimum objective functions found regarding

Table 1 compares the computational demands of the various algorithms. NSGA-II, SPEA2, GDE3, MOEA/D, and the standard ME algorithm, without any analysis reduction techniques, each required a maximum of 120,000 nonlinear inelastic analyses, corresponding to an approximate computation time of 168,000 seconds. In contrast, the MEX algorithm, leveraging XGBoost surrogate models, performed only 52,362 structural analyses and required 1430 training times for XGBoost model. This resulted in a significantly reduced average computation time of 77,311 seconds for MEX—approximately 46% of the time required by the ME algorithm. Therefore, MEX achieved a computational time savings of roughly 54% compared to the standard ME approach.

Table 1. Time analysis of algorithms for the 47-bar powerline

Algorithm	Structural analysis	XGBoost model building	Time analysis (second)	Ratio
NSGA2, SPEA2, GDE3, MOEA/D	120,000	0	168,000	100.0%
ME	120,000	0	168,000	100.0%
MEX	52,362	1430	77,311	46.02%

5. Conclusions

This research introduced MOEA/D-EpDE_XGBoost, a novel algorithm that effectively combines an advanced metaheuristic algorithm with a gradient boosting machine learning model. The algorithm was evaluated against standard MOO algorithms using a benchmark truss structure problem under nonlinear inelastic conditions. The results demonstrate that MOEA/D-EpDE_XGBoost offers superior convergence, diversity, and computational efficiency compared to existing algorithms. Specifically, MOEA/D-EpDE_XGBoost achieved a 54% reduction in computation time compared to the standard ME algorithm while maintaining a comparable solution quality. The enhanced performance highlights the algorithm's potential as a powerful and efficient tool for multi-objective design optimization of complex nonlinear inelastic structures, particularly within resource-constrained environments. Future work will explore the algorithm's applicability to even larger-scale structures and other types of optimization problems.

Acknowledgements

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

References

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

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

[3] 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 (JSTCE) - HUCE*, 14(2):65–75.

[4] 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.

[5] Cao, T.-S., Nguyen, T.-T.-T., Nguyen, V.-S., Truong, V.-H., Nguyen, H.-H. (2023). Performance of six metaheuristic algorithms for multi-objective optimization of nonlinear inelastic steel trusses. *Buildings*, 13(4):868.

[6] Cao, T.-S., Pham, H.-A., Truong, V.-H. (2025). An efficient algorithm for multi-objective structural optimization problems using an improved pbest-based differential evolution algorithm. *Advances in Engineering Software*, 197:103752.

[7] Truong, V.-H., Tangaramvong, S., Papazafeiropoulos, G. (2024). An efficient LightGBM-based differential evolution method for nonlinear inelastic truss optimization. *Expert Systems with Applications*, 237 (B):121530.

[8] 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.

[9] Truong, V.-H., Cao, T.-S., Tangaramvong, S. (2024). A robust machine learning-based framework for handling time-consuming constraints for bi-objective optimization of nonlinear steel structures. *Structures*, 62:106226.

[10] 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.

[11] 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.

[12] 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, San Francisco, CA, USA, ACM, 785–794.

[13] Zhang, Q., Liu, W., Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE Congress on Evolutionary Computation*, Trondheim, Norway, IEEE, 203–208.

[14] Zhang, Q., Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

[15] Zheng, W., Tan, Y., Fang, Z., Li, S. (2017). An improved MOEA/D with optimal DE schemes for many-objective optimization problems. *Algorithms*, 10(3):86.

[16] 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.

[17] von Lücken, C., Brizuela, C. A., Barán, B. (2022). Clustering-based multipopulation approaches in MOEA/D for many-objective problems. *Computational Optimization and Applications*, 81(3):789–828.

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

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

[20] 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.

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

[22] 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.

[23] 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.