International Journal of GEOMATE, April, 2024 Vol.26, Issue 116, pp.85-92
ISSN: 2186-2982 (P), 2186-2990 (O), Japan, DOI: https://doi.org/10.21660/2024.116.4398
Geotechnique, Construction Materials and Environment

AN EFFICIENT FRAMEWORK FOR OPTIMIZATION OF NONLINEAR STEEL TRUSSES WITH CONTINUOUS VARIABLES USING LIGHTGBM

Quoc-Bao Nguyen[1], *Huu-Hue Nguyen[2] and Manh-Hung Ha[3]

[1] Department of Bridge and Tunnel Engineering, Faculty of Bridges and Roads, Hanoi University of Civil Engineering, Vietnam;
[2] Faculty of Civil Engineering, Thuyloi University, Vietnam;
[3] Department of Structure Mechanics, Hanoi University of Civil Engineering, Vietnam

*Corresponding Author, Received: 23 Jan. 2024, Revised: 05 Feb. 2024, Accepted: 6 Feb. 2024

ABSTRACT: Excessive time-consuming has been a great problem when applying metaheuristic algorithms to the optimization of structures using direct analyses, including steel truss structures. In this work, a robust optimization method was proposed to solve this issue. In the proposed method, an efficient variant of the differential evolution algorithm (DE), which was proved to be powerful in searching optimum designs and converged quickly, was employed as the optimizer. An effective framework based on LightGBM classification models was developed to save lots of time-consuming direct analyses required for evaluating constraints. To enhance the performance of LightGBM models, an adaptive parameter, which can reflect the convergence speed of the population, was proposed to prevent the imbalanced classification problem in the training data. Two truss optimizations with continuous design variables were studied, including a 47-bar power line and a 113-bar planar bridge. The results proved that the proposed method yielded better optimum designs than DE/best/1 and EpDE. It also saved more than 60% of time-computing compared to DE/best/1, EpDE, and 2EpDE. Besides that, the proposed framework for using LightGBM models was compatible with metaheuristics having different convergence rates, including DE/best/1, EpDE, and 2EpDE.

Keywords: Optimization; Truss; Direct Analysis; Nonlinear; Machine learning

1. INTRODUCTION

Sizing optimization of steel truss structures has attracted significant interest from researchers. In sizing optimization of a truss, cross-sections of truss elements are optimized to minimize the total costs/mass of the structure. In practice, the cross-section of an element is often chosen from a given discrete list. However, for the study purpose, continuous range can be considered to evaluate the efficiency of the developed algorithm. The constraints of the optimization are the design requirements of the standards, where lots of load combinations are considered. Due to the characteristics of steel material, the analysis of steel trusses requires consideration of the structural nonlinearities. In such cases, nonlinear inelastic analyses are good solutions. The whole structure's nonlinear response and load-carrying capacity are directly captured and the separate member check can be eliminated [1].

Recently, metaheuristic algorithms based on non-gradient have been preferred to gradient-based optimizers since they performed superior for non-discontinuous, non-convex, and highly nonlinear. Publications in the literature proved that metaheuristics are effective with the optimization of various structural types i.e. truss structures [2], modular block walls [3], steel frames subject to static and seismic loads [4], pavement structures [5], etc. Some popular metaheuristics are differential evolution (DE) [6], ant colony [7], and particle swarm optimization (PSO) [8].

In 2018, Truong and Kim [2] proposed an efficient p-best-based DE algorithm (named EpDE) using a p-best method: ‘DE/pbest/1’. This approach allows balancing 2 popular DE mutation strategies: ‘DE/rand/1’ and “DE/best/1’. Numerical results in [2] proved that EpDE found better global optimal solutions and converged more quickly. In 2023, Vu et al. [9] improved EpDE by using two p-best individuals (not only one as integrated with EpDE), called 2EpDE. The results showed that 2EpDE converged more quickly and searched global optimals better than EpDE. However, the study of Vu et al. [9] was limited to steel frames with discrete design variables. The evaluation of 2EpDE robustness for truss structures with continuous design variables is necessary.

Integrating nonlinear inelastic analysis in metaheuristic-based optimization frameworks spends abundant computational efforts since lots of structural analyses are required to evaluate constraints. Consequently, developing efficient methods for reducing the number of time-consuming nonlinear inelastic analyses is an interesting research direction. In conventional approaches, improving the performance of metaheuristic algorithms by increasing the ability to find globally optimal solutions, increasing convergence speed, and reducing calculation time is preferable. Based on this approach, many new metaheuristic algorithms and optimization frameworks based on metaheuristics have been proposed. For example, Pham [10], Ho-Huu et al. [11], and Truong and Kim [2] improved the DE algorithm to develop frameworks for truss optimization that not only found better optimal results but also reduced the number of objective function evaluations. However, the number of structural analyses required is still great.

Recently, using surrogate models based on machine learning (ML) algorithms have been considered a promising solution to the structural optimization problem. Given ML surrogate models, Mai et al. [12] applied a deep neural network (DNN) to optimize truss structures under several load combinations. Mai et al. [13] integrated the DNN surrogate model with DE for optimization of truss structures considering geometrically nonlinearity. Liu and Xia [14] combined DNN and GA for three classical truss problems (size, shape, and size-shape integrated optimizations). Gholizadeh & Mohammadi [15] developed a combination of particle swarm optimization (PSO) and bat algorithm (BA) for reliability-based design optimization (RBDO) of steel frames.

It is well-known that surrogate models can predict with very high accuracy but not absolute. This small error causes many good individuals to be overlooked during the optimization process. From the perspective of using surrogate models as support for the optimization frameworks, Truong et al. [16] developed an efficient optimization method for nonlinear inelastic trusses by combining a binary classification surrogate using light gradient boosting machines (LightGBM) and EpDE, named LightGBM-EpDE. In LightGBM-EpDE, a safety factor t was employed to prevent the removal of good individuals due to surrogate model errors. The authors proposed the value of t increases from 0.9 to 1.0 according to the generation and the increase of the training database. However, the use of such t value may cause the LightGBM classification model to not be efficient if an imbalanced classification problem occurs in the training data. Consequently, although the above framework for LightGBM classification surrogate worked well with EpDE, its efficiency may be significantly reduced when combined with other metaheuristic algorithms, which have a convergence speed much different from EpDE.

The next sections of the paper are as follows. In section 2, the research significance is summarized. Section 3 states the sizing truss optimization problem using nonlinear inelastic analysis. Section 4 introduces the proposed method and Section 5 shows two truss examples to demonstrate the efficiency of the proposed method. Finally, Section 6 draws some conclusions from this work.

2. RESEARCH SIGNIFICANCE

In the current work, an efficient optimization framework for nonlinear truss structures is developed by combining the LightGBM classification model and 2EpDE. 2EpDE plays a role as an optimizer. A framework for using a LightGBM classification surrogate is proposed based on an adaptive safety parameter t that adjusts the value according to the convergence of the population to prevent the imbalanced classification problem in the training data. To evaluate the efficiency of the proposed method, two well-known complexity optimization truss structures are studied, including a 47-bar power line with 27 design variables and a 113-bar planar bridge with 43 design variables.

3. OPTIMIZATION PROBLEM STATEMENT

3.1 Nonlinear Inelastic Analysis of Truss Structures

The incremental equilibrium equation between two configurations for the truss element is written in the terms of virtual displacement principle as [1] [13]:

$$\left([k_E]+[k_G]+[s_1]+[s_2]+[s_3]\right)\{d\}+{}^1f={}^2f \qquad (1)$$

where ${}^1f$ and ${}^2f$ are the element initial nodal forces at previous and current configurations, respectively; $[k_E]$ and $[k_G]$ are elastic and geometric stiffness matrices, respectively; $[s_1]$, $[s_2]$, and $[s_3]$ are higher-order stiffness matrices [18].

The above truss modeling and analysis have been integrated into the Practical Advanced Analysis Program (PAAP) [19] which is applied for structural analysis in this paper. In view of nonlinear inelastic analysis, the ultimate load factor (ULF) (expressed in Eq. (2)) of the whole structure can be found. At this time, the structure is considered safe if ULF > 1.

$$ULF=\frac{R}{S} \qquad (2)$$

where R and S are the structural load-carrying capacity and applied loads, respectively.

3.2 Optimization Problem Statement

Applying the nonlinear inelastic analysis presented in the above section, the sizing optimization of steel truss structures is stated in this section as follows. The objective function is the total mass of the structure that can be expressed as:

$$\text{Minimize} \quad W(\mathrm{X})=\rho\sum_{i=1}^{nm}\left(x_i\sum_{q=1}^{n_i}L_q\right) \qquad (3)$$

$$\mathrm{X}=(x_1,x_2,...,x_{nm}),\ x_i\in[LB_i,UB_i]$$

where $X = (x_1, x_2, .., x_{nm})$ is the vector of design variables, x_i is the cross-sectional area of the element group i^{th}, LB_i and UB_i are given lower and upper bounds of x_i, L_q is the length of the truss element q^{th} of the element group i^{th}, n_i is the number of truss element in the element group i^{th}, and nm is the number of design variables (which is also the number of design variables).

The constraints of the optimization include strength and serviceability constraints that are calculated according to strength and serviceability load combinations, respectively. In strength load combinations, the structural safety is evaluated using ULF as presented in Section 3.1. In serviceability load combinations, nodal displacements are restricted from exceeding the allowable values. All above constraints are expressed as follows:

$$\text{Subject to: } \begin{cases} 1 - ULF_j \le 0 \quad j = 1,.., N_{strength} \\ \dfrac{|d_{k,l}|}{|d^u_{k,l}|} - 1 \le 0 \;\; k = 1,.., N_{service}; l = 1,.., N_{node} \end{cases} \tag{4}$$

where $d_{k,l}$ and $d^u_{k,l}$ are the displacement of the node l^{th} and its allowable value, respectively, according to the serviceability load combination k^{th}.

Since most metaheuristic algorithms were originally proposed for solving unconstrained optimization problems, the penalty method is employed here to convert the above truss optimization into an unconstrained one:

$$W_{un}(X) = \left(1 + \sum_{i=1}^{N_{strength}} \alpha_i^{str} \beta_{1,i} + \sum_{j=1}^{N_{service}} \alpha_j^{service} \beta_{2,j}\right) \times W(X) \tag{5}$$

In which:

$$\beta_{1,i} = \max(1 - ULF_i, 0)$$
$$\beta_{2,j} = \sum_{l=1}^{N_{node}} \max\left(\frac{|d_{j,l}|}{|d^u_{j,l}|} - 1, 0\right) \tag{6}$$

where α_i^{str} and $\alpha_j^{service}$ are the penalty parameters corresponding to the strength load combination i^{th} and the serviceability load combination j^{th}, respectively, and $\beta_{1,i}$ and $\beta_{2,j}$ are the violated magnitude at the strength load combination i^{th} and the serviceability load combination j^{th}, respectively. Using a large value of α makes $W_{un}(X)$ receiving a much greater value if a constraint is violated. Since the optimization process will keep individuals having a smaller objective function in the population, infeasible individuals with very high objective functions easily be removed. In this work, the penalty parameters are chosen to be 10,000.

4. COMBINED METHOD LIGHTGBM-2EPDE

4.1 LightGBM-EpDE Optimization Framework

Observed from Eq. (4), the number of structural analyses for each individual is $(N_{strength} + N_{service})$. Consequently, the total of structural analyses for an optimization process with D design variables, NP individuals in the population, and $Iter_{max}$ generations is $(NP \times Iter_{max} \times (N_{strength} + N_{service}))$. Such structural analyses may require excessive computing time. To overcome this issue, Truong et al. [16] proposed an efficient framework by combining a LightGBM binary classification surrogate and EpDE, named LightGBM-EpDE. The basic concept of LightGBM-EpDE can be summarized as follows:

(1) A LightGBM classification model is employed to evaluate the violation of an individual with constraints. A promising individual is neglected immediately if its predicted unconstrained objective function is worse than the target one. Otherwise, nonlinear inelastic analysis is applied to accurately determine the unconstrained objective function.

(2) The training data is created by collecting from all nonlinear inelastic analyses for individuals. The LightGBM surrogate model is updated constantly if the size of the training data increases a pre-defined value. This approach allows for improving the accuracy of the LightGBM model.

(3) The EpDE algorithm plays the role of the optimizer. LightGBM surrogate models are employed to reduce the number of structural analyses required for constraint evaluation. The characteristics of the EpDE algorithm are maintained, including the capacity to search optimal designs and convergence speed.

(4) A safety parameter t is applied to reduce the predicting error of LightGBM surrogate models. In view of t, the surrogate model LightGBM(t) means that the output is true (or structural is safe) if $ULF \ge t$ but not $ULF \ge 1.0$ as normal. Since $t < 1.0$, more promising individuals may be evaluated using nonlinear inelastic analysis.

It is seen that the efficiency of LightGBM-EpDE is dependent on two key factors: (1) the robustness of EpDE as an optimizer and (2) the effectiveness of LightGBM surrogate model-based framework with the safety factor t. In view of this, in this section, we propose an improved version of LightGBM-EpDE using 2EpDE and an adaptive framework for LightGBM surrogate models, named ALGB2EpDE. In the next sub-sections, LightGBM and 2EpDE are introduced first, then the ALGB2EpDE algorithm is proposed.

4.2 Light Gradient Boosting Machines

LightGBM was introduced by Microsoft in 2017 in the view of a gradient-boosting framework [20].

Several weak learners (decision trees) are ensemble sequentially in LightGBM, where the following tree predicts the error of the previous one. Two novel techniques of LightGBM are exclusive feature bundling (EFB) and gradient-based one-side sampling (GOSS).

EFB is based on the sparse characteristic of high-dimensional data, where several features are mutually exclusive, to regroup exclusive features into an exclusive feature bundle. In view of this, the complexity of the data changes from O(data × feature) to O(data × bundle) with bundle<<feature. Therefore, the training speed is improved while the memory used is reduced.

The GOSS technique is a robust sampling method that can effectively reduce the number of data instances while maintaining the accuracy of decision trees. In GOSS, all data instances having large gradients are kept, and instances with small gradients are randomly dropped (but not all) to retain the decision tree accuracy. The numerical results [20] proved that, with EFB and GOSS, LightGBM outperformed in terms of memory consumption and computational speed.

4.3 Efficient Differential Evolution Algorithm Using Two P-Best Individuals (2EpDE)

The EpDE algorithm was developed by Truong and Kim [2] by integrating a modified 'DE/pbest/1' mutation technique into DE. In EpDE, the trial individual is created in view of a top $100p\%$ individual of the population, X_{pbest}, as follows:

$$\mathrm{U} = \mathrm{X}_{pbest} + F \times \left(\mathrm{X}_{r_1} - \mathrm{X}_{r_2}\right) \tag{7}$$

In which: $$p(j) = A \times NP^{\left(-B\times\frac{j-1}{Iter_{\max}-1}\right)} \tag{8}$$

where A and B are the given parameters for controlling p, j means the generation j^{th}, NP is the number of individuals in the population, and X_{r_1} and X_{r_2} are other indivduals in the population.

Eq. (7) indicates that U inherits the characteristics of X_{pbest}. This may not be efficient in the early stage of the optimization process due to the high dispersion of the population (many X_{pbest} have not good information on optimum areas). To overcome this drawback, Vu et al. [9] proposed an improvement of Eq. (7) (called 'DE/2pbest/1'), where two top $100p\%$ individuals, $\mathrm{X}_{pbest,1}$ and $\mathrm{X}_{pbest,2}$, as follows:

$$\mathrm{U} = 0.5\left(\mathrm{X}_{pbest,1} + \mathrm{X}_{pbest,2}\right) + F \times \left(\mathrm{X}_1 - \mathrm{X}_2\right) \tag{9}$$

As observed from Eq. (13), U now can gather information from two top $100p\%$ individuals, $\mathrm{X}_{pbest,1}$ and $\mathrm{X}_{pbest,2}$. Therefore, using Eq. (13) helps U have more opportunity to get good information than using Eq. (7). The numerical results in [9] also proved that 2EpDE outperformed EpDE in both mathematical examples and steel frames. In particular, 2EpDE found better optimals and converged much faster than EpDE.

4.4 Framework of The Proposed Optimization Method: ALGB2EpDE

The robustness of 2EpDE provides a method for improving LightGBM-EpDE by replacing EpDE with 2EpDE as the optimizer in the optimization framework. Besides that, the efficiency of LightGBM-EpDE also comes from using the factor t, which is calculated as follows:

$$t(j) = 0.9 + 0.1\times\left(1 - \frac{p(j)}{A}\right) = 0.9 + 0.1\times\left(1 - NP^{\left(-B\times\frac{j-1}{Iter_{\max}-1}\right)}\right) \tag{10}$$

According to the report in [20], the optimization process begins by evaluating all constraints using nonlinear inelastic analysis to form the initial database for training LightGBM surrogate models. The surrogate model is built when the size of the database reaches a pre-defined value i.e. 1,000. The database continues updated when a nonlinear inelastic analysis is required for constraint evaluation. And, the surrogate model is rebuilt if the size of the database increases by a given number i.e. 20. The surrogate model using the safety factor t is called LightGBM(t). If $ULF \geq t$, the output is true and vice versa.

Along with the convergence of the optimization, the ULF of U tends to reach around the limit value of constraints i.e. 1.0. Eq. (10) shows that t changes from 0.9 to 1.0 according to the increment of the generation without considering the convergence speed of the optimization. Therefore, if the convergence of the optimization process is much faster than the increment speed of t, most of ULF of trial vectors, which reach around 1.0, will be greater than t. Consequently, new samples added to the training data are considered to be safe, and an imbalanced classification problem occurs. The surrogate models are then not efficient since they may predict all output of ULF to be safe.

To overcome the above issue, we propose a new equation for calculating t which can always ensure a reasonable distribution of classes in the training data. In particular, assuming that the training data is $Datamatrix$ with the N outputs. Absolutely, among N outputs, there are many outputs smaller than 1.0. We assume that there are N_t (N_t<N) outputs smaller than 1.0, and called: $ULF_1, ULF_2, .., ULF_{N_t}$. t is now calculated as follows:

$$t = 0.8 + 0.2\frac{\sum_{i=1}^{N_t} ULF_i}{N_t} \tag{11}$$

Obviously, with Eq. (11) there are lots of samples having outputs smaller than t . So, the imbalanced classification problem is solved. If the population converges quickly, the much later values in the sequence ($ULF_1, ULF_2, .., ULF_{N_t}$) will approach 1.0, and then t increases more quickly. Therefore, Eq. (11) reflects the convergence speed of the population.

5. NUMERICAL EXAMPLES

In this section, two truss structures (a 47-bar power line and a 113-bar planar bridge to represent two common types, including civil structures and bridges, using truss structures) are examined to demonstrate the efficiency of the proposed method. Three algorithms: EpDE, 2EpDE, and DE/best/1 are combined with LightGBM with two options: using Eq. (10) and using Eq. (11). By this way, there are a total of nine optimization frameworks considered, including DE/Best/1, EpDE, 2EpDE, DE/best/1+LightGBM using Eq. (10), EpDE+ LightGBM using Eq. (10), 2EpDE+LightGBM using Eq. (10), DE/best/1+LightGBM using Eq. (11), EpDE+LightGBM using Eq. (11), and 2EpDE+LightGBM using Eq. (11). It should be noted that EpDE+LightGBM using Eq. (10) is the LightGBM-EpDE proposed in Ref. [18] and 2EpDE+LightGBM using Eq. (11) is the proposed method in this work, named as ALGB2EpDE. Furthermore, the multi-comparison technique [2] is integrated into all algorithms to save computational efforts.

The population in algorithms is 30. The A and B parameters in Eq. (8) are 0.5 and 1.0, respectively. In DE/best/1, the parameters F and CR are 0.7 and 0.6, respectively. The total generation is 2,000. The hyperparameters of the LightGBM classification model are taken from [16] as: learning_rate=0.05; n_estimators=1,000; reg_alpha=0.0; reg_lambda=0.0; n_jobs= 5.

5.1 47-Bar Power Line

The 47-bar power line presented in Fig. 1 includes 27 groups of element cross-sectional areas such as: (1) $A_1 = A_3$; (2) $A_2 = A_4$; (3) $A_5 = A_6$; (4) A_7; (5) $A_8 = A_9$; (6) A_{10}; (7) $A_{11} = A_{12}$; (8) $A_{13} = A_{14}$; (9) $A_{15} = A_{16}$; (10) $A_{17} = A_{18}$; (11) $A_{19} = A_{20}$; (12) $A_{21} = A_{22}$; (13) $A_{23} = A_{24}$; (14) $A_{25} = A_{26}$; (15) A_{27}; (16) A_{28}; (17) $A_{29} = A_{30}$; (18) $A_{31} = A_{32}$; (19) A_{33}; (20) $A_{34} = A_{35}$; (21) $A_{36} = A_{37}$; (22) A_{38}; (23) $A_{39} = A_{40}$; (24) $A_{41} = A_{42}$; (25) A_{43}; (26) $A_{44} = A_{45}$; (27) $A_{46} = A_{47}$. The design variables are continuous in the [64.516, 6451.6] (mm^2). The load combination of (1.2DL+0.5LL+1.7W) is investigated where DL, LL, and W are the dead, live, and wind loads, respectively. DL and LL are 70 (kN) and 50 (kN), respectively, at all nodes. W is equal to 30 (kN) at nodes 17 and 22 according to the horizontal axis.

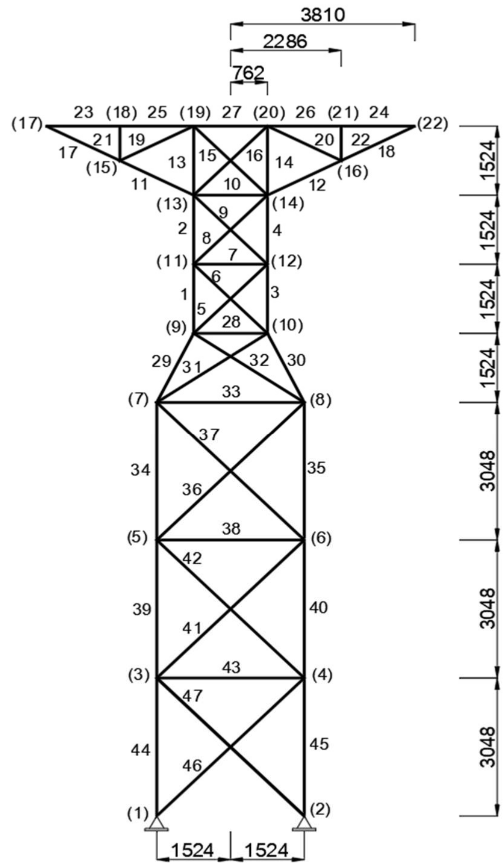

Fig.1 Example 1: 47-bar power line

Table 1 presents the optimal results obtained from 20 independent runs for each algorithm. Figs. 2 and 3 indicate the convergence speed of the algorithms. As observed from these figures, DE/best/1 and 2EpDE converged quite similarly and were much better than EpDE.

The results in Table 1 proved that 2EpDE was slightly better than EpDE while it yielded smaller values of the best, the worst, and the average values of optimum masses found. In particular, the best masses of 2EpDE and EpDE were 1,005.07 (kg) and 1,005.28 (kg), respectively. The DE/best/1 had a worse performance than EpDE and 2EpDE when searching for greater values of the best, the worst, and the average values of the optimum masses. The best mass found using DE/best/1 was 1,006.42 (kg).

To assess the impact of using LightGBM models on the optimal results found by algorithms, three optimization method groups are evaluated, including: (1) (DE/best/1, DE/best/1+LightGBM using Eq. (10), DE/best/1+LightGBM using Eq. (11)), (2) (EpDE, EpDE +LightGBM using Eq. (10), EpDE+LightGBM using Eq. (11)), and (3) (2EpDE, 2EpDE +LightGBM using Eq. (10), 2EpDE+LightGBM using Eq. (11) or ALGB2EpDE).

Table 1 Optimization results of 47-bar power line with different algorithms (*unit: kg*)

Content	DE/best/1	EpDE	2EpDE	Using Eq. (10)			Using Eq. (11)		
				DE/best/1	EpDE	2EpDE	DE/best/1	EpDE	2EpDE
Best mass	1,006.42	1,005.28	1,005.07	1,006.42	1,005.32	1,005.18	1,006.53	1,005.29	1,005.15
Worst mass	1,022.16	1,013.44	1,011.23	1,026.37	1,013.48	1,010.28	1,021.87	1,014.59	1,012.17
Ave. mass	1,010.38	1,006.98	1,006.36	1,013.46	1,006.88	1,006.82	1,011.67	1,006.81	1,006.56
Std. mass	4.109	2.146	1.377	7.499	1.850	1.737	5.136	2.097	1.460
Ave. analyses	42,029	40,344	39,899	32,418	21,877	31,042	17,245	18,477	15,179
Ave. time analysis (s)	59,080	56,705	56,078	45,529	30,666	43,589	24,135	23,041	18,925
Time ratio	312.18%	299.62%	296.31%	240.57%	162.04%	230.32%	127.53%	121.75%	100.00%

In group 1, the optimum designs found using DE/best/1+LightGBM with Eq. (10) and DE/best/1+LightGBM with Eq. (11) were similar to the results of DE/best/1. In particular, the best optimum mass found using DE/best/1, DE/best/1+LightGBM using Eq. (10), and DE/best/1+LightGBM using Eq. (11) were 1,006.42 (kg), 1,006.42 (kg), and 1,006.53 (kg) which were almost the same. The similar results were also found in groups 2 and 3.

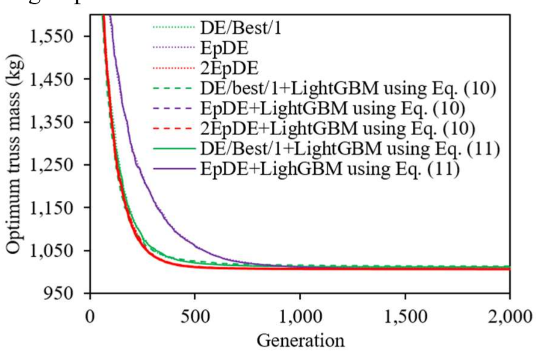

Fig.2 Convergence curves of the average of all runs for the 47-bar power line

The combination of LightGBM surrogate models saved lots of direct analyses of conventional metaheuristic algorithms considered. In group 1, DE/best/1 required an average of 42,029 structural analyses, while DE/best/1+LightGBM using Eq. (10) and DE/best/1+LightGBM using Eq. (11) spent only 32,418 and 17,245 structural analyses, respectively, which equaled to 77.13% and 41.03% that of DE/best/1. However, as can be seen above, DE/best/1+LightGBM using Eq. (11) saved much more structural analyses than DE/best/1+LightGBM using Eq. (10). A similar result was obtained in group 3 where 2EpDE used 39,899 structural analyses while 2EpDE +LightGBM using Eq. (10) and ALGB2EpDE required only 31,042 and 15,179 structural analyses, respectively. The results proved that using the LightGBM model with Eq. (11) saved much more computational effort compared to using Eq. (10). In group 2, EpDE, EpDE +LightGBM using Eq. (10), and EpDE+LightGBM using Eq. (11) required 40,344, 21,877, and 18,477 structural analyses, respectively. It confirmed the efficiency when using LightGBM to reduce the computational effort. However, in this case study, the robustness of using Eqs. (10) and (11) was similar. The above results indicated that the LightGBM model using Eq. (11) worked well with all algorithms DE/best/1, EpDE, and 2EpDE, while the LightGBM model using Eq. (10) seemed to be not efficient with DE/best/1 and 2EpDE.

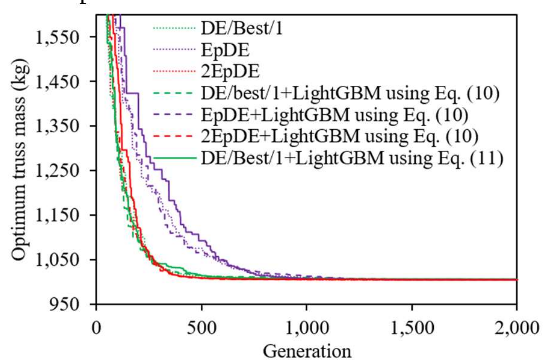

Fig.3 Convergence histories of the best optimal for the 47-bar power line

5.2 113-Bar Planar Truss Bridge

Fig. 4 presents the layout and geometry of the bridge with 34 continuous design variables in the range [3870.96, 22580.6] (mm^2). One load combination, (1.25DL+1.75LL), is studied, where DL and LL are equal to 150 (kN) and 120 (kN), respectively, at all truss joints on the upper chord.

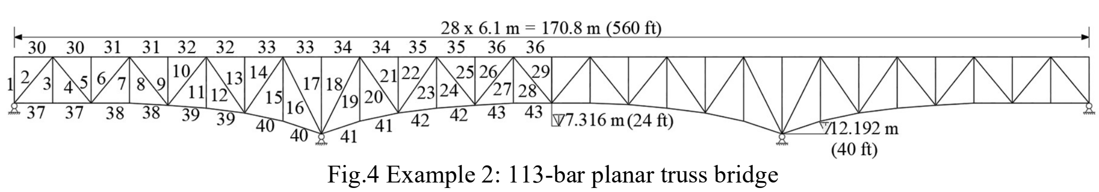

Fig.4 Example 2: 113-bar planar truss bridge

The optimum results are reported in Table 2, while the converged curves are presented in Figs. 5 and 6. As can be seen in these figures, DE/best/1 and 2EpDE converged quite similarly and more quickly than EpDE. Compared to EpDE, 2EpDE performed slightly better with smaller best, worst, and average values of the optimum masses found. DE/best/1 was worse than EpDE and 2EpDE when yielded much greater values of best, worst, and average optimum masses. In particular, the best masses found using DE/best/1, EpDE, and 2EpDE were 35,407.6 (kg), 35,200.6 (kg), and 35,192.6 (kg), respectively.

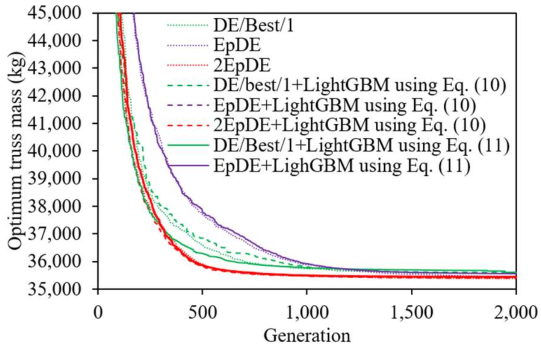

Fig.5 Convergence histories of the average of all runs for 113-bar planar truss bridge

LightGBM surrogate models maintained the efficiency of metaheuristic algorithms considered in searching for the optimum designs. In particular, in group 3, the best masses found using 2EpDE, 2EpDE+LightGBM using Eq. (10), and ALGB2EpDE are similar and equal to 35,192.6 (kg), 35,211.5 (kg), and 35,195.0 (kg), respectively. Secondly, LightGBM surrogate models saved lots of time-consuming direct analyses of DE/best/1, EpDE, and 2EpDE. For example, in group 1, DE/best/1 spent an average of 41,434 structural analyses, while DE/best/1+LightGBM using Eq. (10) and DE/best/1+LightGBM using Eq. (11) required only 28,854 and 18,383 structural analyses, respectively. However, LightGBM surrogate models using Eq. (11) saved more structural analyses than LightGBM surrogate models using Eq. (10) for DE/best/1, EpDE, and 2EpDE, especially for DE/best/1 and 2EpDE. This proved that LightGBM surrogate models using Eq. (11) were better compatible with algorithms.

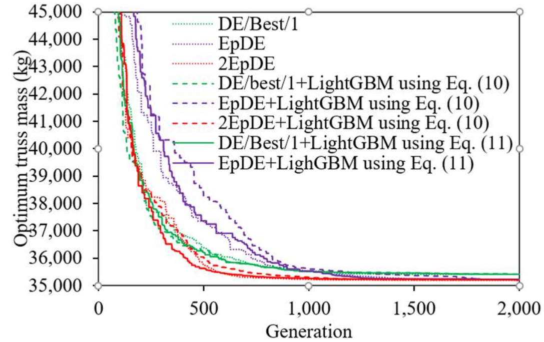

Fig.6 Convergence histories of best optimal for 113-bar planar truss bridge

6. CONCLUSION

The paper developed a robust optimization framework for nonlinear truss structures with continuous design variables. An improved two-pbest-based DE algorithm (2EpDE) was used as the optimizer. 2EpDE showed good performance regarding both convergence speed and optimum design search in the comparison with DE/best/1 and EpDE algorithms. An efficient strategy for using LightGBM classification models was developed. An adaptive parameter t was proposed to (1) prevent the imbalanced classification problem in the training data, (2) prevent the removal of good individuals due to surrogate model errors, and (3) reflect the convergence speed of the population. The results of a 47-bar power line and a 113-bar planar bridge proved the robustness of 2EpDE since it found better optimum designs than DE/best/1 and EpDE. 2EpDE also converged much better than EpDE. The proposed method (ALGB2EpDE) was more powerful than 2EpDE since it not only yielded good optimum designs like 2EpDE but also saved more than 60% of time-computing compared to DE/best/1, EpDE, and 2EpDE. Furthermore, the proposed framework for using LightGBM surrogate models worked well with metaheuristic algorithms that have different convergence speeds, including DE/best/1, EpDE, and

Table 2 Optimization results of 113-bar planar truss bridge with different algorithms (*unit: kg*)

Content	DE/best/1	EpDE	2EpDE	Using Eq. (10)			Using Eq. (11)		
				DE/best/1	EpDE	2EpDE	DE/best/1	EpDE	2EpDE
Best mass	35,407.6	35,200.6	35,192.6	35,411.9	35,213.7	35,211.5	35,411.9	35,228.4	35,195.0
Worst mass	35,904.6	35,765.2	35,653.4	35,903.1	35,723.1	35,646.0	35,902.1	35,770.4	35,653.5
Ave. mass	35,567.7	35,467.3	35,384.1	35,604.2	35,478.7	35,403.3	35,609.3	35,560.2	35,477.1
Std. mass	198.12	185.72	180.32	214.25	208.47	163.68	182.38	170.17	158.37
Ave. analyses	41,434	39,363	38,105	28,854	20,255	27,144	18,383	17,165	16,413
Ave. time analysis (s)	72,089	68,465	66,264	50,074	35,026	47,082	31,749	29,619	28,303
Time ratio	254.71%	241.90%	234.13%	176.92%	123.75%	166.35%	112.18%	104.65%	100.00%

2EpDE. Therefore, it can be considered a good technique for integrating metaheuristic algorithms.

7. ACKNOWLEDGMENTS

This research is funded by the Ministry of Education and Training under grand number B2022-XDA-07.

8. REFERENCES

[1] Thai H.T. and Kim S.E., Large deflection inelastic analysis of space trusses using generalized displacement control method. Journal of Constructional Steel Research, vol. 65, issues 10-11, 2009, pp. 1987-1994.
[2] Truong V.H. and Kim S.E., Reliability-based design optimization of nonlinear inelastic trusses using improved differential evolution algorithm. Advances in Engineering Software, vol. 121, 2018, pp. 59-74.
[3] Chatpattananan V., Prayongphan S., Kongsomboon T. and Ratanavaraha V., Designing a trapezoidal modular block wall with nonlinear optimization. International Journal of GEOMATE, vol. 22, issue 90, 2022, pp. 40-48.
[4] Mehdi M.A., Cherradi T., Bouyahyaoui A., Karkouri S.E., and Qachar A., Evolution study of the pavement structural indicator based on evenness and deflection results using a gis tool mapping. International Journal of GEOMATE, vol. 23, issue 95, 2022, pp. 144-153.
[5] Truong V.H. and Kim S.E., A robust method for optimization of semi-rigid steel frames subject to seismic loading. Journal of Constructional Steel Research, vol. 145, 2018, pp. 184-195.
[6] Storn R. and Price K., Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, vol. 11, no. 4, 1997, pp. 341-359.
[7] Srichandum S. and Pothiya S., Multiple plants multiple sites ready mixed concrete planning using improved ant colony optimization. International Journal of GEOMATE, vol. 19, issue 72, 2020, pp. 88-95.
[8] Aziz M.K.B.M., Yusof F., Daud Z.M., Yusop Z. and Kasno M.K., Optimization design of rain gauge network in Johor by using geostatistics and particle swarm optimization. International Journal of GEOMATE, vol. 11, issue 25, 2016, pp. 2422-2428.
[9] Vu Q.A., Cao T.S., Nguyen H.H., Truong V.H. and Ha M.H., An efficient differential evolution-based method for optimization of steel frame structures using direct analysis. Structures, vol. 51, 2023, pp. 67-78.
[10] Pham H.A., Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison. Advances in Engineering Software, vol. 102, 2016, pp. 142-154.
[11] Ho-Huu V., Nguyen-Thoi T., Vo-Duy T. and Nguyen-Trang T., An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. Computers & Structures, vol. 165, 2016, pp. 59-75.
[12] Mai H.T., Lieu Q.X., Kang J. and Lee J., A novel deep unsupervised learning-based framework for optimization of truss structures. Engineering with Computers, vol. 39, 2023, pp. 2585-2608.
[13] Mai H.T., Kang J. and Lee J., A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior. Finite Elements in Analysis and Design, vol. 196, 2023, pp. 103572.
[14] Liu J.P. and Xia J., A hybrid intelligent genetic algorithm for truss optimization based on deep neutral network. Swarm and Evolutionary Computation, vol. 73, 2022, pp. 101120.
[15] Gholizadeh S. and Mohammadi M., Reliability-based seismic optimization of steel frames by metaheuristics and neural networks. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, vol. 3, issue 1, 2016, pp. 04016013.
[16] Truong V.H., Tangaramvong S. and Papazafeiropoulos G., An efficient LightGBM-based differential evolution method for nonlinear inelastic truss optimization. Expert Systems with Applications, vol. 237, Part B, 2024, pp. 121530.
[17] Blandford G., Progressive failure analysis of inelastic space truss structures. Computers & Structures, vol. 58, issue 5, 1996, pp. 981-990.
[18] Yang Y.B. and Kuo S.R., Theory and analysis of nonlinear framed structures. Prentice Hall, 1994, pp. 1-450.
[19] Yang Y.B. and Shieh M.S., Solution method for nonlinear problems with multiple critical points. AIAA Journal, vol. 28, no. 2, 1990, pp. 2110-2116.
[20] Thai H.T. and Kim S.E., Practical advanced analysis software for nonlinear inelastic analysis of space steel structures. Advances in Engineering Software, vol. 40, issue 9, 2009, pp. 786-797.
[21] Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q. and Liu T.Y., LightGBM: A highly efficient gradient boosting decision tree. 31st Conference on Neural Information Processing Systems (NIPS2017), LongBeach, CA, USA, 2017, pp. 1-9.