Research Article

Two Strategies to Improve the Differential Evolution Algorithm for Optimizing Design of Truss Structures

Ching-Yun Kao ¹, Shih-Lin Hung ², and Budy Setiawan²

¹Department of Applied Geoinformatics, Chia Nan University of Pharmacy & Science, No. 60, Sec. 1, Erren Rd., Tainan 71710, Taiwan

²Department of Civil Engineering, National Chiao Tung University, No. 1001, University Rd., Hsinchu 300, Taiwan

Correspondence should be addressed to Shih-Lin Hung; slhung@mail.nctu.edu.tw

Received 8 December 2019; Accepted 30 April 2020; Published 20 July 2020

Academic Editor: Giuseppe Quaranta

Copyright © 2020 Ching-Yun Kao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of differential evolution (DE) mostly depends on mutation operator. Inappropriate configurations of mutation strategies and control parameters can cause stagnation due to over exploration or premature convergence due to over exploitation. Balancing exploration and exploitation is crucial for an effective DE algorithm. This work presents an enhanced DE (EDE) for truss design that utilizes two new strategies, namely, *integrated mutation* and *adaptive mutation factor* strategies, to obtain a good balance between the exploration and exploitation of DE. Three mutation strategies (*DE/rand/1*, *DE/best/2*, and *DE/rand-to-best/1*) are combined in the *integrated mutation* strategy to increase the diversity of random search and avoid premature convergence to a local minimum. The *adaptive mutation factor* strategy systematically adapts the mutation factor from a large value to a small value to avoid premature convergence in the early searching period and to increase convergence to the global optimum solution in the later searching period. The outstanding performance of the proposed EDE is demonstrated through optimization of five truss structures.

1. Introduction

Structural design optimization is a critical and challenging topic and has attracted considerable attention in the last few decades. Optimization enables designers to generate many desirable designs while saving money and time [1].

Metaheuristic has been widely used over the last few decades to solve complex optimization problems because of its simplicity, ease of implementation, and ability to avoid local optima and it deals with derivative free problems [2, 3]. Some methods of metaheuristic have been extensively utilized to solve structural optimization problems. Such methods include genetic algorithm (GA) [4–6], ant colony optimization (ACO) [7–9], artificial bee colony (ABC) [10], particle swarm optimization (PSO) [11–19], big bang–big crunch (BB-BC) [20], and differential evolution (DE) algorithms [21, 22]. These methods belong to two most predominant and successful classes or directions in

metaheuristic, swarm-based algorithms and evolutionary algorithms [2, 3, 23].

Swarm-based algorithms, such as ACO, ABC, and PSO algorithms, are inspired by collective behavior in animals. They encompass the implementation of collective intelligence of groups of simple agents that are based on the behavior of real-world insect swarms, as a problem solving tool. ACO proposed by Marco Dorigo [24] is a probabilistic technique used to solve computational problems that can be reduced to finding good paths through graphs. It is inspired by the behavior of ants in finding paths from the colony to food. It has strong robustness and good distributed calculative mechanism. Besides, it is easy to combine with other methods and shows good performance in resolving complex optimization problems. However, ACO has a longer search time than other methods and tends to terminate at a nonoptimal solution. ABC introduced by Karaboga [25] is an optimization algorithm motivated by the intelligent

behavior of honey bees. It is simple and only uses common control parameters, such as colony size and maximum cycle number. However, it shows slow convergence speed during searching. The PSO algorithm, which is a probabilistic and iterative approach, finds the optimal position in the search space by simulating the behavior of a flock of foraging birds. The PSO algorithm has fewer parameters and is easier to implement than the GA. It also has a higher convergence rate than other evolutionary algorithms in solving several problems [26]. Although the PSO algorithm can rapidly converge in the early searching stage, premature convergence may cause particle searching to fall into a local optimum. Most modifications on the simple PSO have been made to improve its convergence rate and to increase the swarm diversity. He et al. [27] introduced the concept of passive congregation that affects the particle velocity in accordance with the positions of other randomly selected particles. A particle swarm optimizer with passive congregation (PSOPC) can improve the search efficiency and the probability of finding the optimal solution. Kaveh and Talatahari [28] presented a heuristic particle swarm ant colony optimization (HPSACO) for the optimum design of trusses. This algorithm is based on PSOPC, ACO, and harmony search scheme. Their comparison results showed that HPSACO has better efficiency and robustness than other PSO-based algorithms and has a higher convergence rate than PSO and PSOPC. Lu et al. [29] proposed an augmented PSO (AugPSO) algorithm with an increased convergence rate in early search and increased diversity that does not fall into a local optimum. The two major strategies used in the AugPSO algorithm are heuristic-inspired boundary shifting and mutation-like particle position resetting. The two strategies are inspired by a heuristic and mutation scheme in GAs. The boundary shifting approach forces particles to move to the boundary between feasible and infeasible regions for increasing the convergence rate in searching. The particle position resetting approach aims to increase the diversity of particles and to prevent the solution of particles from falling into local minima. Numerical analyses showed that the AugPSO algorithm is more robust than the PSO and PSOPC algorithms.

Evolutionary algorithms such as GA, BB-BC, and DE are inspired by natural evolution. They are population-based stochastic search algorithms performing with best-to-survive criteria. Each algorithm commences by creating an initial population of feasible solutions and iteratively evolves from generation to generation toward the best solution. In successive iterations of the algorithm, fitness-based selection occurs within the population of solutions. Better solutions are preferentially selected for survival into the next generation of solutions [23]. The GA proposed by Holland [30] is the most well-known branch among existing evolutionary algorithms. It follows the principles of the Charles Darwin's theory of survival of the fittest. The three principal genetic operators in the GA involve selection, crossover, and mutation. Although a GA has many positive features, the convergence of binary GA is slow, and the result may not be the optimal solution. GAs are unsuitable for solving constraint optimization problems [23]. The BB-BC algorithm developed by Erol and Eksin [31] is inspired from evolution theories of the universe, namely, the big bang and big crunch theory. The BB-BC algorithm consists of two parts,

namely, the big bang, where candidate solutions are randomly distributed over the search space, and the big crunch, where a contraction operation estimates a weighted average or center of mass for the population. The BB-BC algorithm has been applied to various optimization problems in different fields. However, the BB-BC algorithm is easily trapped in local optima, similar to most other heuristic optimization algorithms [32]. DE proposed by Storn and Price [33] has yielded promising results for solving complex optimization problems. It is well known for its simple structure, ease of application, quality of solution, and robustness [34]. Similar to other evolutionary algorithms, DE simulates the natural evolution via mutation, crossover, and selection to evolve a population of initially random solutions into an optimal solution [35]. The main difference between the GA and DE is that mutation is the result of small perturbations to the genes of an individual in the GA, whereas mutation is the result of arithmetic combinations of individuals in DE. Although DE is recognized for its simplicity and efficiency, Mohamed et al. [36] noted that DE has the following shortcomings: the convergence rate of DE is low; premature convergence in which the search process is trapped in a local optimum may cause DE to become progressively less diverse; there is a stagnation problem, in which the search process occasionally stops proceeding toward the global optimum although the population has not converged to any other point.

A metaheuristic algorithm globally explores the problem space and locally searches in the neighborhoods of the existing solutions to obtain new and better solutions. For the metaheuristic algorithm in solution space, balancing exploration and exploitation is crucial for an effective optimization algorithm. The former indicates the ability of the algorithm to discover new search areas, while the latter focuses on finding the best solution in a promising region of the search space. The performance of a metaheuristic algorithm is problem-dependent [2, 3].

This study focuses on performance improvement of DE for truss optimization problems. Mutation operator plays a crucial role in the DE algorithm [37]. However, inappropriate configurations of mutation strategies and control parameters (population size NP, mutation factor F , and crossover rate CR) can cause stagnation due to over exploration or premature convergence due to over exploitation [38]. Many researchers have suggested new techniques to improve the original DE [39–41]. These proposed modifications on DE are adjusting control parameters in an adaptive or self-adaptive manner and developing new mutations rule. In fact, the parameters of DE are problem-dependent and it is difficult to adjust them for different problems [42]. To improve the performance of original DE for truss optimization problem, an enhanced DE (EDE) algorithm is proposed in this work to obtain a good balance between the exploration and exploitation of DE. The two major strategies used in the EDE algorithm are *integrated mutation* and *adaptive mutation factor* strategies. Some of the developed mutation strategies are fit for global search with good exploration ability (e.g., mutation strategy $DE/rand/1$), and others are good at local search with good exploitation ability (e.g., mutation strategy $DE/best/2$) [43]. Three mutation strategies ($DE/rand/1$, $DE/best/2$, and

DE/rand-to-best/1) are combined in the *integrated mutation* strategy to increase the diversity of random search (the exploration ability in solution space) and avoid premature convergence to a local minimum. Each individual in the population uses one of the three mutation strategies. For gradient-based analytical approach, two steps are involved in each search step. The first step is finding an appropriate search direction, a step-descent gradient vector, in N -dimension solution space. Analogy in the DE algorithm, this step is finding an appropriate differential vector in mutation and crossover operators. The second step is a line search approach that finds a step length in that gradient direction (in a 1D problem). Similarly, this step sets a mutation factor for each individual in the DE algorithm. Inspired by gradient-based analytical approach, the *adaptive mutation factor* strategy systematically adapts the mutation factor from a large value to a small value on the basis of the typical convergence curves of truss optimization [4–22] to avoid premature convergence in the early searching period and to increase convergence to the global optimum solution in the later searching period. Five truss examples were presented to demonstrate the effectiveness and efficiency of the proposed EDE algorithm in this study. The performance of EDE was compared with that of original DE and other metaheuristic algorithms in the literature.

2. Problem Formulation

The optimization problem aims to optimize the design by minimizing the total weight of all members of a structure while satisfying the displacement and stress constraints on the design variables and structural responses. The general form is expressed as follows:

$$\begin{aligned} \min \quad & O(X) \\ \text{subject to} \quad & h_j(X) \leq e_j \quad \text{for } j = 1, 2, \dots, J \\ & x_n^l \leq x_n \leq x_n^u \quad \text{for } n = 1, 2, \dots, N, \end{aligned} \quad (1)$$

where objective function $O(X)$ is the total weight of the truss. Function $h_j(X)$ and e_j are the j th inequality constraint function and its predefined specified threshold, respectively. Decision variable vector $\mathbf{X}$ is composed of N design variables, $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_N]^T$, which are the cross-sectional areas of the bar members of a truss. These design variables define the design space; the objective function is a “surface” of N dimensions embedded in a space of $N+1$ dimensions. Design variable x_n represents the cross-sectional area of the n th member of a truss. The lower and upper limits on design variable x_n are x_n^l and x_n^u , respectively. An available solution region (problem search space) for the optimization problem is defined in terms of these limits. Inequality constraints can then be applied to reduce the size of the feasible region. In this work, these N design variables are clustered into NG groups, and each group may contain a variant number of members ($g_1, g_2, \dots, g_{NG}$). The bar members in the same group have identical cross-sectional areas. Thus, the number of design variables reduces from N to NG . Decision variable vector $\mathbf{X}$ can be rewritten as follows:

$$\mathbf{X} = \begin{Bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{NG} \end{Bmatrix}, \text{ for } \begin{cases} X_1 = x_1 = \dots = x_{g_1}, \\ X_2 = x_{g_1+1} = \dots = x_{g_1+g_2}, \\ \vdots \\ X_{NG} = x_{(\sum_{i=1}^{NG-1} g_i)+1}, \dots, x_N. \end{cases} \quad (2)$$

3. DE Algorithm

Storn and Price [33] coined the term DE. The basic idea that underlies DE is that the difference between two vectors yields a difference vector that can be utilized with a scaling factor to diversify the search space. The general DE procedure consists of four stages, namely, initialization, mutation, crossover, and selection.

3.1. Initialization. An initial population with NP individuals is generated through random sampling from the search space. Each individual is a vector containing NG design variables $\mathbf{X}_i = [X_{1,i} \ X_{2,i} \ \dots \ X_{NG,i}]^T$. To prevent the initial vectors from falling into an infeasible area, the initial population of cross-sectional areas of truss members is generated by selecting random values between the lower (half of the upper bound) and upper bounds:

$$X_{j,i}^{(1)} = X_j^L + \text{rand}[0, 1](X_j^U - X_j^L) = \frac{X_j^U + \text{rand}[0, 1]X_j^U}{2}, \quad (3)$$

for $i = 1, 2, \dots, NP$; $j = 1, 2, \dots, NG$,

where $\text{rand}[0, 1]$ represents a uniformly distributed random value between zero and one, X_j^L and X_j^U are the minimum and maximum cross-sectional areas of the j th group members, respectively, NP is the population size of DE, and NG is the number of element groups clustered from N design variables. Superscript (1) of $X_{j,i}^{(1)}$ denotes the first generation.

3.2. Mutation. For each vector $\mathbf{X}_i^{(G)}$, a mutated vector $\mathbf{v}_i^{(G)}$ is generated:

$$\mathbf{v}_i^{(G)} = \mathbf{X}_{r1}^{(G)} + F(\mathbf{X}_{r2}^{(G)} - \mathbf{X}_{r3}^{(G)}), \quad (4)$$

where $F > 0$ and integers $r1, r2, r3 \in \{1, 2, \dots, NP\}$ are randomly generated such that $r1 \neq r2 \neq r3 \neq i$. F is the mutation factor, which controls the amplification of the difference vector $(\mathbf{X}_{r2}^{(G)} - \mathbf{X}_{r3}^{(G)})$. Superscript (G) denotes the G th generation.

3.3. Crossover. To increase the diversity of parameter vectors, a trial vector $\mathbf{u}_i^{(G)} = [u_{1,i}^{(G)} \ u_{2,i}^{(G)} \ \dots \ u_{NG,i}^{(G)}]^T$ is formed for each element $X_{j,i}^{(G)}$:

$$u_{j,i}^{(G)} = \begin{cases} v_{j,i}^{(G)}, & \text{for } (\text{rand}[0, 1] \leq CR \mid j = j_{\text{rand}}), \\ X_{j,i}^{(G)}, & \text{otherwise,} \end{cases} \quad (5)$$

where $i = 1, 2, \dots, NP$; $j = 1, 2, \dots, NG$; CR is the crossover rate, which is drawn from the range of zero to one and preset

by the user; j_{rand} is an integer selected from 1 to NG. The criterion that a random value is less than or equal to CR can trigger the crossover for each trial vector. The rate is randomly selected for each trial vector.

Before the selection step, each trial vector $\mathbf{u}_i^{(G)}$ is checked to determine whether it violates the relevant constraints or not. If it does, then the trial vector will be rejected and replaced with the original vector of the current generation, and

$$\mathbf{u}_i^{(G)} = \begin{cases} \mathbf{u}_i^{(G)}, & \text{if } \sigma_j^L \leq \sigma_j \leq \sigma_j^U \wedge \delta_c^L \leq \delta_c \leq \delta_c^U, \\ \mathbf{X}_i^{(G)}, & \text{otherwise,} \end{cases} \quad (6)$$

where $i = 1, 2, \dots, \text{NP}$; $j = 1, 2, \dots, \text{NG}$; σ_j^L and σ_j^U are the minimum and maximum allowable stresses of the j th member group, respectively; $c = 1, 2, \dots, \text{NC}$, and NC is the number of joints or connections; δ_c^L and δ_c^U are the minimum and maximum allowable displacements of the c th connection, respectively.

3.4. Selection. The performances of the trial and original vectors in the selection operation are compared. The better one is selected and passed to the next generation. The trial vector should satisfy the predefined constraints. The performance is determined using objective function O , which is the total weight of the structure under consideration. The new population for the next generation is formed as

$$\mathbf{X}_i^{(G+1)} = \begin{cases} \mathbf{u}_i^{(G)}, & \text{if } (O(\mathbf{u}_i^{(G)}) < O(\mathbf{X}_i^{(G)})), \\ \mathbf{X}_i^{(G)}, & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbf{X}_i^{(G+1)}$ and $\mathbf{X}_i^{(G)}$ are the i th individuals of the population for the next generation (generation $G + 1$) and the current generation (generation G), respectively, and $\mathbf{u}_i^{(G)}$ is the trial vector. The mutation, crossover, and selection steps are repeated until a specified number of generations is reached.

4. EDE Algorithm

4.1. Adaptive Mutation Factor. The DE algorithm with a fixed mutation factor has several shortcomings; for example, a low mutation factor yields a slow convergence rate, and a high mutation factor causes stagnation problem and fails to yield a good result because the search space is extremely large. The first strategy adaptively updates the mutation factor between the minimum and maximum mutation factors in each generation. In the early searching period, a large mutation factor is required to search with great diversity and to avoid premature convergence. In the later searching period, a low mutation factor is utilized to increase convergence for reaching the global optimum solution. The adaptive mutation factor function is developed on the basis of the typical convergence curves of truss optimization. Figure 1 plots the power function. For each generation G , a mutation factor $F^{(G)}$ is generated using a power function with degree a and multiplier F_u :

FIGURE 1: Adaptive mutation factor curve.

$$F^{(G)} = F_u \times G^a, \quad (8)$$

and the degree of power function a is given by

$$a = \frac{\ln F_l - \ln F_u}{\ln G_{\max}}, \quad (9)$$

where $G = 1, 2, \dots, G_{\max}$, and $G_{\max}$ is the total number of generations. F_u and F_l are the user-defined maximum and minimum mutation factors, respectively, and their values are between zero and one.

4.2. Integrated Mutation Strategy. The integrated mutation strategy uses several mutation strategies to increase the convergence rate and search diversity in the parameter space. The three following mutation strategies are selected on the basis of the preliminary study:

(a) DE/rand/1

$$\mathbf{v}_i^{(G)} = \mathbf{X}_{r1}^{(G)} + F^{(G)}(\mathbf{X}_{r2}^{(G)} - \mathbf{X}_{r3}^{(G)}). \quad (10)$$

(b) DE/best/2

$$\mathbf{v}_i^{(G)} = \mathbf{X}_{\text{best}}^{(G)} + F^{(G)}(\mathbf{X}_{r1}^{(G)} - \mathbf{X}_{r2}^{(G)}) + F^{(G)}(\mathbf{X}_{r3}^{(G)} - \mathbf{X}_{r4}^{(G)}). \quad (11)$$

(c) DE/rand-to-best/1

$$\mathbf{v}_i^{(G)} = \mathbf{X}_{r1}^{(G)} + F^{(G)}(\mathbf{X}_{\text{best}}^{(G)} - \mathbf{X}_i^{(G)}) + F^{(G)}(\mathbf{X}_{r2}^{(G)} - \mathbf{X}_{r3}^{(G)}), \quad (12)$$

where $\mathbf{v}_i^{(G)}$ is the mutated vector, $\mathbf{X}_{r1}^{(G)}$, $\mathbf{X}_{r2}^{(G)}$, $\mathbf{X}_{r3}^{(G)}$, and $\mathbf{X}_{r4}^{(G)}$ are random vectors, $F^{(G)}$ is the adaptive mutation factor, $\mathbf{X}_{\text{best}}^{(G)}$ is the best vector of the current generation, and $\mathbf{X}_i^{(G)}$ is the i th vector of the current generation. The three strategies are chosen because each of them has unique characteristics and advantages.

According to Qiang and Mitchell [43], the three strategies are performed as follows:

- (a) *DE/rand/1* is the original strategy and has a stronger exploration capability but converges more slowly than the strategies that use the best solution from the parent generation.
- (b) *DE/best/2* uses the best solution in the parent population and converges rapidly to the optimal solution, but it may encounter the stagnation problem, which refers to becoming stuck at a local minimum during multimodal function optimization.
- (c) *DE/rand-to-best/1* compromises between the exploitation of the best solution and exploration of the random parameter space. It has highly diverse mutant vectors that are obtained from a randomly selected parent vector.

Ao and Chi [37] experimentally investigated differential evolution strategies (ESs). Their experimental results revealed that strategies that use the best vector typically have a high convergence speed but easily become stuck at a local optimum, whereas the *DE/rand* strategies converge faster to the global optimum. Therefore, the above strategies are combined to overcome the shortcomings of all types of strategy and to integrate their advantages. Each vector $i = 1, 2, \dots, NP$ uses one of these strategies in the following sequence. For example, the first vector ($i = 1$) uses *DE/rand/1*, the second vector ($i = 2$) uses *DE/best/2*, and the third vector ($i = 3$) uses *DE/rand-to-best/1*. The fourth vector ($i = 4$) returns by applying *DE/rand/1*.

4.3. Optimization. Figure 2 presents the overall optimization process, which includes initialization, mutation, crossover, constraints handling, and selection and is implemented as follows:

- (a) Define the truss parameters, including joint coordinates, member connections, load cases, and constraints.
- (b) Define the DE parameters NG , NP , F , F_u , F_b , CR , and G_{max} .
- (c) Generate an $NG \times NP$ matrix of the initial population by randomly generating vector elements between $0.5X^U$ and X^U .
- (d) Calculate degree of the power function a and adaptive mutation factor $F^{(G)} = F_u \times G^a$.
- (e) Define mutation strategies for each vector i on the basis of the *integrated mutation strategy*:
  - if $i = 1, 4, 7, \dots \rightarrow$ employ *DE/rand/1*,
  - if $i = 2, 5, 8, \dots \rightarrow$ employ *DE/best/2*,
  - if $i = 3, 6, 9, \dots \rightarrow$ employ *DE/rand - to - best/1*.

(13)

- (f) Generate NP mutated vectors $\mathbf{v}_i^{(G)}$ and check if the cross-sectional areas in $\mathbf{v}_i^{(G)}$ are beyond the boundary.

- (g) Crossover step: generate NP trial vectors.
- (h) Calculate stress and displacement by performing a matrix analysis of structures.
- (i) Check whether each of the trial vectors violates the constraints; if it does, then replace it by the original vector or the parent solution.
- (j) Selection step: select the best solution from the original and trial vectors on the basis of the objective function.
- (k) Sort the solutions and specify the best vector.
- (l) Repeat “step f” to “step k” for the next generation until the maximum number of generations is reached.
- (m) Generate the optimal solution.

The program is terminated when a defined number of generations G_{max} is reached and the final solution is generated. The final result is the total weight of the structure.

5. Numerical Study

Various benchmarks of truss structures were studied to evaluate the performance of the proposed EDE method: (1) a 10-bar truss, (2) a 25-bar truss, (3) a 72-bar truss, (4) a 120-bar truss, and (5) a 942-bar truss. The performance of the proposed method was compared with those of other methods in the literature to verify its feasibility. To test the individual effect of the two strategies, *adaptive mutation factor* strategy and *integrated mutation* strategy, used in the EDE algorithm to improve DE algorithm, the following versions of algorithm are also compared with EDE:

- (1) The same EDE version without *adaptive mutation factor* strategy, called EDE-1, is experimentally investigated to test the individual effect of *adaptive mutation factor* strategy on the performance of EDE algorithm.
- (2) The same EDE version without *DE/rand/1* mutation, called EDE-2, is experimentally investigated to test the individual effect of *integrated mutation strategy* on the performance of EDE algorithm.
- (3) The same EDE version without *DE/best/2* mutation, called EDE-3, is experimentally investigated to test the individual effect of *integrated mutation strategy* on the performance of EDE algorithm.
- (4) The same EDE version without *DE/rand-to-best/1* mutation, called EDE-4, is experimentally investigated to test the individual effect of the proposed *integrated mutation strategy* on the performance of EDE algorithm.

5.1. Ten-Bar Truss. The truss is a cantilevered truss with a pinned support and a roller support. Figure 3 displays the configuration of the 10-bar truss and the loading condition. All members are assumed to be made from the same material with an elastic modulus of $E = 10,000,000$ psi and a density of

FIGURE 2: Flow chart of the optimization process.

0.10 lb/in³. The design variables are the cross-sectional areas of all members. The minimum and maximum cross-sectional areas of the members are set to 0.1 and 35.0 in², respectively. The maximum deflection of any node in either direction must not exceed ± 2 in, and the maximum allowable stress of all members is set to $\pm 25,000$ psi.

The DE parameters that need to be set are NP, F, CR, and $G_{\max}$. In this study, NP is set to a value between 5 NG and 8 NG based on suggestions of Storn and Price [44] and Gamperle et al. [45], and NP is set to 50 in this example (NG is equal to 10 in this example). After determination of NP,

$G_{\max}$ is set to a value making the number of analyses N_{analyses} (the product of the number of population and the number of generations or $NP \times G_{\max}$) of DE less than those of other methods in the literature for performance comparison. $G_{\max}$ is set to 200 in this example. After determination of NP and $G_{\max}$, F and CR are determined by a two-step method. First, CR is set to 0.1 and F is from 0.1 to 1.0 every 0.1 to test DE to determine adequate F based on the suggestion of Storn and Price [44] that crossover rate CR = 0.1 is an initial good choice. DE with a certain F value is tested for five runs. Figure 4 shows the effects of various values of F on average

FIGURE 3: Configuration of the 10-bar truss (reproduced from Lu et al. [29], under the Creative Commons Attribution License/public domain).

FIGURE 4: Effects of various values of F on average weight of designs for the 10-bar truss using DE with $CR=0.1$.

weight of designs for the 10-bar truss using DE with $CR=0.1$. Herein, mutation factor F is set to 0.5 since it provided minimum average weight (5209.233 lb). Second, CR is determined on the basis of predetermined F . F is the predetermined value and CR is from 0.1 to 1.0 every 0.1 to test DE to determine adequate CR . DE with a certain CR value is tested for five runs. Figure 5 shows the effects of various values of CR on average weight of designs for the 10-bar truss using DE with $F=0.5$. Herein, CR is set to 0.9 since it provided minimum average weight (5063.911 lb). The EDE parameters that need to be set are NP , F_b , F_u , CR , and G_{max} . NP and G_{max} are the same as those of DE. F_l and F_u are determined based on the results of setting F in DE. Figure 4 shows that average weight is smaller when F is between 0.3 and 1.0. Therefore, F_u and F_l are set to 0.3 and 1.0, respectively. After determining F_l and F_u , CR is set to be from 0.1 to 1.0 every 0.1 to test EDE to determine adequate CR . EDE with a certain CR value is tested for five runs. Figure 6 shows the effects of various values of CR on average weight of designs for the 10-bar truss using EDE with $F_l=0.3$ and $F_u=1.0$. It shows that $CR=0.8$ provided minimum average weight (5061.369 lb). Thus, CR is set to 0.8 in EDE. The EDE-1 parameters (NP , F , CR , and G_{max}) are the same as those of DE. The parameters of EDE-2, EDE-3, and EDE-4 (NP , F_b , F_u , CR , and G_{max}) are the same

FIGURE 5: Effects of various values of CR on average weight of designs for the 10-bar truss using DE with $F=0.5$.

FIGURE 6: Effects of various values of CR on average weight of designs for the 10-bar truss using EDE with $F_l=0.3$ and $F_u=1.0$.

as those of EDE. The same method is also used to determine parameters (NP , F , F_b , F_u , CR , and G_{max}) in the following examples.

Table 1 compares designs for the 10-bar truss using the proposed algorithms with other optimization techniques. It is noted that the design runs of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 are set to 30 for all truss design examples in this study, since the normal approximation for design parameter (truss weight) will generally be good if the design runs are larger and equal to 30. Considering that the solution of HPSACO [28] violated the relevant constraints, the best truss design of EDE (5060.896 lb) is lighter than those of DE (5061.578 lb) and other four optimization techniques in the literature. The mean (5061.734 lb) of design weight of EDE is also smaller than those of DE and the other three optimization techniques in the literature (PSO, PSOPC, and AugPSO). The standard deviation of design weight of EDE (2.877 lb) is larger than that of DE but smaller than those of the other three optimization techniques in the literature (PSO, PSOPC, and AugPSO). Moreover, EDE required fewer analyses compared with GA [5]. Results revealed that EDE is more robust than other heuristic methods in solving this benchmark problem. Since the difference between mean (or median) weights of DE and EDE-1 is larger than that of EDE and EDE-1, the effect of *integrated mutation* strategy is larger than that of *adaptive mutation factor* strategy on the performance of EDE in this example. Comparing DE, EDE,

TABLE 1: Performance comparison for the 10-bar truss.

Element group	Member	Camp et al. [5] GA	Kaveh and Talatahari [28] HPSACO	Cross-sectional areas (in ²)								This work			
				Lu et al. [29]											
				PSO	PSOPC	AugPSO	DE	EDE	EDE-1	EDE-2	EDE-3	EDE-4			
1	1	28.92	30.307	20.149	25.923	30.457	30.272	30.542	30.559	30.501	30.721	30.570			
2	2	0.1	0.1	0.1	0.39	0.1	0.1	0.1	0.1	0.1	0.1	0.1			
3	3	24.07	23.434	32.233	23.247	23.584	23.425	23.203	23.317	23.304	23.175	23.232			
4	4	13.96	15.505	14.831	18.208	15.029	14.842	15.233	15.193	15.261	15.295	15.172			
5	5	0.1	0.1	0.1	0.108	0.1	0.1	0.1	0.1	0.1	0.1	0.1			
6	6	0.56	0.5241	0.116	0.1	0.564	0.559	0.530	0.526	0.547	0.551	0.556			
7	7	7.69	7.4365	8.349	9.007	7.42	7.445	7.456	7.448	7.439	7.466	7.448			
8	8	21.95	21.079	28.039	26.629	20.987	21.128	21.019	21.020	20.961	21.023	20.969			
9	9	22.09	21.229	22.909	18.736	21.524	21.745	21.539	21.486	21.540	21.361	21.582			
10	10	0.1	0.1	3.066	0.196	0.1	0.1	0.1	0.1	0.1	0.1	0.1			
Weight _{best} (lb)		5076.31	*5056.56	5,606.036	5,225.282	5,061.209	5061.578	5060.896	5060.951	5060.899	5060.987	5060.900			
Weight _{worst} (lb)		—	—	7,008.815	5,854.163	5,179.516	5066.422	5076.892	5076.791	5076.687	5062.822	5076.907			
Weight _{mean} (lb)		—	—	6,256.895	5,425.074	5,103.484	5063.490	5061.734	5061.719	5061.878	5061.517	5061.930			
Weight _{median} (lb)		—	—	—	—	—	5062.980	5061.098	5061.097	5061.120	5061.342	5061.207			
Weight _{stddev} (lb)		—	—	370.071	175.764	31.755	1.310	2.877	2.864	2.885	0.500	2.901			
N _{analyses}		15000	—	—	—	—	10000	10000	10000	10000	10000	10000			

Note. * indicates that the result violates constraint.

EDE-1, EDE-2, EDE-3, and EDE-4, EDE provided minimum best design weight (5060.896 lb) and EDE-1 provided minimum median weight (5061.097 lb). In fact, the median weight of EDE (5061.098 lb) is almost the same as that of EDE-1. However, EDE-3 provided minimum mean (5061.517 lb) and standard deviation (0.500 lb) of design weight. Notably, comparing EDE, EDE-1, EDE-2, EDE-3, and EDE-4, the standard deviation of design weight of EDE-3 (0.500 lb) is minimum but the best design weight of EDE-3 (5060.987 lb) is maximum, which implies that a DE algorithm without using *DE/best/2* strategy provides a more reliable solution but has less opportunity to search for the global optimum solution. Figure 7 compares convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4. EDE performed better and converged much faster than DE, and EDE-1 converged fastest in the early search period. The DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms required approximately 77, 56, 57, 66, 66, and 73 generations (3850, 2800, 2850, 3300, 3300, and 3650 analyses), respectively, to converge to a solution.

5.2. Twenty-Five-Bar Truss. The 25-bar transmission tower displayed in Figure 8 is one of the benchmarks utilized in structural optimization by various numerical techniques. The material density is 0.1 lb/in³, and modulus of elasticity is 10,000,000 psi. The 25 members are categorized into the following eight groups, namely, (1) A_1 , (2) A_2 – A_5 , (3) A_6 – A_9 , (4) A_{10} – A_{11} , (5) A_{12} – A_{13} , (6) A_{14} – A_{17} , (7) A_{18} – A_{21} , and (8) A_{22} – A_{25} .

Table 2 presents the stress constraints on each group of elements. The structure is subjected to the loading condition specified in Table 3. Allowable displacements of each node in the x , y , and z directions are limited to a maximum value of 0.35 in. For each group of members, the minimum and maximum cross-sectional areas were set to 0.01 and 3.4 in², respectively.

In this example, the parameters of DE and EDE-1 are set to $NP = 50$, $F = 0.6$, $CR = 0.9$, and $G_{\max} = 160$. The parameters of EDE, EDE-2, EDE-3, and EDE-4 are set to $NP = 50$, $F_u = 1.0$, $F_l = 0.3$, $CR = 0.9$, and $G_{\max} = 160$. Table 4 compares designs for the 25-bar truss using the proposed algorithms with other optimization techniques. The best truss design of EDE (545.163 lb) is lighter than those of DE (545.319 lb) and five other optimization techniques in the literature. The mean (545.166 lb) and standard deviation (0.007 lb) of design weight of EDE are also smaller than those of DE and other optimization techniques in the literature. Moreover, EDE required the fewest analyses compared with PSO [12], ACO [8], and BB-BC [20]. Results revealed that EDE is more robust than other heuristic methods in solving this benchmark problem. Since the difference between mean (or median) weights of DE and EDE-1 is smaller than that of EDE and EDE-1, the effect of *integrated mutation* strategy is smaller than that of *adaptive mutation factor* strategy on the performance of EDE in this example. Comparing DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4, minimum best design weight (545.163 lb) was provided by EDE, EDE-2,

FIGURE 7: Convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 for the 10-bar truss.

and EDE-4, and EDE provided minimum mean weight (545.166 lb), median weight (545.164 lb), and standard deviation of design weight (0.007 lb). The results showed that the EDE algorithm exhibited an improved computational efficiency and consistent performance compared with all other algorithms for this example. As presented in Figure 9, the optimization of 25-bar truss using EDE converged to the optimal solution more quickly than that using DE, and EDE-4 converged fastest in the early search period. The DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms required approximately 100, 80, 87, 80, 80, and 80 generations (5000, 4000, 4350, 4000, 4000, and 4000 analyses), respectively, to converge to a solution.

5.3. Seventy-Two-Bar Truss. The 72-bar truss is a benchmark problem in 3D truss optimization and has been tackled by several researchers using various methods. Figure 10 presents the configuration of the 72-bar space truss, including its nodes and the corresponding schemes for numbering groups of elements. The members of this truss are grouped into 16 categories. Table 5 presents the two independent loading conditions applied to the 72-bar space truss. All of the structural members were assumed to be made of the same material with an elastic modulus of 10^7 psi and a density of 0.1 lb/in³. Displacements of the uppermost joints in the x and y directions are limited to be a maximum value of 0.25 in, satisfying the constraint in load case 1. Under load case 2, the maximum displacement of top floor in the z direction satisfies the displacement constraint. The maximum allowable stress is ± 25000 psi. The minimum and maximum cross-sectional areas are set to 0.1 and 3.0 in², respectively.

In this example, the parameters of DE and EDE-1 are set to $NP = 60$, $F = 0.4$, $CR = 0.9$, and $G_{\max} = 200$. The parameters of EDE, EDE-2, EDE-3, and EDE-4 are set to $NP = 60$, $F_u = 0.6$, $F_l = 0.3$, $CR = 0.9$, and $G_{\max} = 200$. Table 6 compares designs for the 72-bar truss using the proposed algorithms with other optimization techniques. The best truss design of EDE (379.645 lb) is lighter than

FIGURE 8: Configuration of the 25-bar truss (reproduced from Sonmez [10], under the Creative Commons Attribution License/public domain).

TABLE 2: Allowable stresses for the 25-bar truss.

Member group	Members	Compression (psi)	Tension (psi)
1	1	35092	35000
2	2, 3, 4, 5	11590	35000
3	6, 7, 8, 9	17305	35000
4	10, 11	35092	35000
5	12, 13	35092	35000
6	14, 15, 16, 17	6759	35000
7	18, 19, 20, 21	6959	35000
8	22, 23, 24, 25	11082	35000

TABLE 3: The loading condition of the 25-bar truss.

Node	Case 1			Case 2		
	F_x (kip)	F_y (kip)	F_z (kip)	F_x (kip)	F_y (kip)	F_z (kip)
1	0	20	-5	1	-10	-10
2	0	-20	-5	0	-10	-10
3	0	0	0	0.5	0	0
6	0	0	0	0.6	0	0

those of DE (379.939 lb) and four other optimization techniques in the literature. The mean (379.807 lb) and standard deviation (0.184 lb) of design weight of EDE are also smaller than those of DE and other optimization techniques in the literature. Moreover, EDE required fewer analyses compared with GA [6], ACO [8], and BB-BC phase 1 [20]. Results revealed that EDE is more robust than other heuristic methods in solving this benchmark problem. Since the difference between mean (or median)

weights of DE and EDE-1 is larger than that of EDE and EDE-1, the effect of *integrated mutation* strategy is larger than that of *adaptive mutation factor* strategy on the performance of EDE in this example. Comparing DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4, EDE provided minimum best design weight (379.645 lb) and mean weight (379.807 lb), and EDE-2 provided minimum median weight (379.734 lb). Although the standard deviation of design weight of EDE (0.184 lb) is a little larger

TABLE 4: Performance comparison for the 25-bar truss under the two load cases.

Element group	Member	Cross-sectional areas (in ²)										
		Schutte and Groenwold [12]	Camp and Bichon [8]	Camp [20]		Sonmez [10]			This work			
		PSO	ACO	BB-BC phase 1	BB-BC phase 2	ABC	DE	EDE	EDE-1	EDE-2	EDE-3	EDE-4
1	1	0.01	0.01	0.01	0.01	0.011	0.01	0.01	0.011	0.01	0.01	0.01
2	2,3,4,5	2.122	2	2.092	2.092	1.979	1.969	1.986	1.972	1.986	1.987	1.987
3	6,7,8,9	2.893	2.966	2.964	2.964	3.003	3.015	2.995	3.003	2.995	2.994	2.994
4	10,11	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.011	0.01	0.01	0.01
5	12,13	0.01	0.012	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
6	14,15,16,17	0.671	0.689	0.689	0.689	0.69	0.697	0.684	0.690	0.684	0.683	0.684
7	18,19,20,21	1.611	1.679	1.601	1.601	1.679	1.682	1.677	1.686	1.677	1.677	1.677
8	22,23,24,25	2.717	2.668	2.686	2.686	2.652	2.641	2.662	2.650	2.662	2.663	2.662
	Weight _{best} (lb)	545.21	545.53	545.48	545.38	545.19	545.319	545.163	545.245	545.163	545.164	545.163
	Weight _{worst} (lb)	—	—	—	—	—	546.096	545.196	545.570	546.906	545.210	545.441
	Weight _{mean} (lb)	546.84	546.34	546.4	545.78	—	545.514	545.166	545.386	545.224	545.169	545.180
	Weight _{median} (lb)	—	—	—	—	—	545.462	545.164	545.362	545.164	545.167	545.167
	Weight _{stdev} (lb)	1.478	0.94	0.653	0.491	—	0.204	0.007	0.092	0.318	0.009	0.051
	N _{analyses}	9596	16500	9746	10820	—	8000	8000	8000	8000	8000	8000

FIGURE 9: Convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 for the 25-bar truss.

than that of EDE-1 (0.108 lb), the results still showed that the EDE algorithm had excellent computational efficiency and consistency for this example. As presented in Figure 11, EDE optimized the 72-bar truss faster than DE, and EDE-2 converged fastest in the early search period. The DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms required approximately 126, 88, 92, 99, 114, and 83 generations (7560, 5280, 5520, 5940, 6840, and 4980 analyses), respectively, to converge to a solution.

5.4. One-Hundred-Twenty-Bar Truss. The 120-bar spatial truss is a dome-shaped truss and has become a benchmark in truss optimization. Figure 12 presents the configuration and group numbering schemes of the 120-bar truss. The structural members are symmetrically organized into seven groups of elements. All of the structural members were assumed to be made of the same material with an elastic modulus of 30,450,000 psi and a

FIGURE 10: Configuration of the 72-bar truss (reproduced from Zeng and Li [19], under the Creative Commons Attribution License/public domain).

TABLE 5: The loading condition of the 72-bar truss.

Node	Case 1			Case 2		
	F_x (kip)	F_y (kip)	F_z (kip)	F_x (kip)	F_y (kip)	F_z (kip)
17	5	5	-5	0	0	-5
18	0	0	0	0	0	-5
19	0	0	0	0	0	-5
20	0	0	0	0	0	-5

density of 0.288 lb/in³. The minimum and maximum cross-sectional areas of all members are set to 0.775 and 20.0 in², respectively. The yielding stress of steel is taken

TABLE 6: Performance comparison of the 72-bar truss.

Element group	Member	Cross-sectional areas (in ²)									
		Cao [6]	Camp and Bichon [8]	Camp [20]		This work					
		GA	ACO	BB-BC phase 1	BB-BC phase 2	DE	EDE	EDE-1	EDE-2	EDE-3	EDE-4
1	1–4	1.8562	1.948	1.9004	1.8577	1.9296	1.8770	1.9129	1.8924	1.8718	1.9469
2	5–12	0.4933	0.508	0.5252	0.5059	0.4989	0.5190	0.5153	0.5073	0.5114	0.5146
3	13–16	0.1	0.101	0.1	0.1	0.1021	0.1	0.1	0.1	0.1	0.1
4	17, 18	0.1	0.102	0.1	0.1	0.1	0.1	0.1001	0.1	0.1003	0.1
5	19–22	1.283	1.303	1.3134	1.2476	1.2747	1.2641	1.2698	1.2800	1.2692	1.2468
6	23–30	0.5028	0.511	0.4801	0.5269	0.5153	0.5055	0.5069	0.5167	0.5063	0.5092
7	31–34	0.1	0.101	0.1	0.1	0.1001	0.1	0.1	0.1	0.1001	0.1001
8	35, 36	0.1	0.1	0.1	0.1012	0.1	0.1	0.1	0.1	0.1	0.1
9	37–40	0.5177	0.561	0.5254	0.5209	0.5406	0.5201	0.5260	0.5054	0.5468	0.5251
10	41–48	0.5227	0.492	0.5267	0.5172	0.5061	0.5163	0.5138	0.5165	0.5217	0.5128
11	49–52	0.1	0.1	0.1016	0.1004	0.1008	0.1	0.1001	0.1	0.1004	0.1001
12	53, 54	0.1049	0.107	0.1253	0.1005	0.1018	0.1002	0.1001	0.1	0.1004	0.1
13	55–58	0.1557	0.156	0.1558	0.1565	0.1561	0.1564	0.1564	0.1568	0.1563	0.1565
14	59–66	0.5501	0.55	0.5456	0.5507	0.5477	0.5488	0.5488	0.5442	0.5445	0.5457
15	67–70	0.3981	0.39	0.4314	0.3922	0.4199	0.4170	0.4007	0.4089	0.4086	0.3997
16	71, 72	0.6749	0.592	0.5231	0.5922	0.5718	0.5640	0.5685	0.5791	0.5756	0.5721
Weight _{best} (lb)		380.32	380.24	380.46	379.85	379.939	379.645	379.656	379.652	379.708	379.694
Weight _{worst} (lb)		–	–	–	–	386.970	380.476	380.063	383.078	380.356	381.039
Weight _{mean} (lb)		–	383.16	384.75	382.08	380.618	379.807	379.808	379.910	379.954	380.008
Weight _{median} (lb)		–	–	–	–	380.342	379.764	379.774	379.734	379.894	379.916
Weight _{stdev} (lb)		–	3.66	2.434	1.912	1.251	0.184	0.108	0.618	0.159	0.332
N _{analyses}		15000	18500	12679	6942	12000	12000	12000	12000	12000	12000

FIGURE 11: Convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 for the 72-bar truss.

as 58,000 psi. The stress and displacement constraints are presented as follows:

- (a) According to the American Institute of Steel Construction's allowable strength design [46], the allowable stress satisfies

$$\begin{cases} \sigma_i^+ = 0.6F_y, & \text{for } \sigma_i \geq 0, \\ \sigma_i^-, & \text{for } \sigma_i < 0, \end{cases} \quad (14)$$

where σ_i^- is calculated using the slenderness ratio:

$$\sigma_i^- = \begin{cases} \frac{[(1 - \lambda_i^2/2C_C^2)F_y]}{(5/3 + 3\lambda_i/8C_C - \lambda_i^3/8C_C^3)}, & \text{for } \lambda_i < C_C, \\ \frac{12\pi^2 E}{23\lambda_i^2}, & \text{for } \lambda_i \geq C_C, \end{cases} \quad (15)$$

where E is the modulus of elasticity, and F_y is the yielding stress of steel. The slenderness ratio that separates the elastic from inelastic buckling regions C_C is calculated:

$$C_C = \sqrt{2\pi^2 E/F_y}. \quad (16)$$

FIGURE 12: Configuration of the 120-bar truss (reproduced from Lee and Geem [1], under the Creative Commons Attribution License/public domain).

Slenderness ratio λ_i satisfies

$$\lambda_i = kL_i/r_i, \quad (17)$$

where k is the effective length factor, L_i is the length of a member, the radius of gyration $r_i = 0.4993A_i^{0.6777}$ for the pipe sections, and A_i is the cross-sectional area of the pipe.

- (b) Maximum displacement of all nodes in the x , y , and z directions is ± 0.1969 in.

The 120-bar dome has 49 joints, with 37 unsupported joints and 12 hinged supports. The truss is subjected to vertical loading only at its unsupported joints in each layer ring. The loads are -13.49 kip at node 1, -6.744 kip at nodes 2 to 14, and -2.248 kip at other unsupported nodes (nodes 15 to 37).

In this example, the parameters of DE and EDE-1 are set to $NP = 50$, $F = 0.4$, $CR = 0.9$, and $G_{\max} = 200$. The parameters of EDE, EDE-2, EDE-3, and EDE-4 are set to $NP = 50$, $F_u = 0.5$, $F_l = 0.4$, $CR = 0.9$, and $G_{\max} = 200$. Table 7 compares designs for the 120-bar truss using the proposed algorithms with other optimization techniques. The best truss design of EDE (20665.883 lb) is lighter than those of DE (20666.393 lb) and three other optimization techniques in the literature. The mean (20666.137 lb) and standard deviation (0.488 lb) of design weight of EDE are also smaller than those of DE and three other optimization techniques in the literature. Results revealed that EDE is more robust than other heuristic methods in solving this benchmark problem. Since the difference between mean (or median) weights of DE and EDE-1 is larger than that of EDE and EDE-1, the effect of *integrated mutation* strategy is larger than that of *adaptive mutation factor* strategy on the performance of EDE in this

TABLE 7: Performance comparison of the 120-bar truss.

Element group	Cross-sectional areas (in ²)								
	PSO	Lu et al. [29] PSOPC	AugPSO	DE	EDE	This work			
						EDE-1	EDE-2	EDE-3	EDE-4
1	3.331	3.325	3.287	3.287	3.288	3.288	3.288	3.288	3.288
2	5.397	5.972	3.486	3.572	3.545	3.549	3.556	3.560	3.555
3	4.513	4.505	4.256	4.255	4.254	4.254	4.254	4.254	4.254
4	3.272	3.026	2.752	2.754	2.753	2.753	2.753	2.753	2.753
5	1.674	0.914	1.353	1.301	1.317	1.315	1.311	1.309	1.311
6	6.459	4.153	3.507	3.508	3.507	3.507	3.507	3.507	3.507
7	3.548	2.415	2.411	2.404	2.406	2.405	2.405	2.405	2.405
Weight _{best} (lb)	26728.980	22654.325	20675.545	20666.393	20665.883	20665.825	20665.861	20665.944	20665.884
Weight _{worst} (lb)	53777.510	38471.212	21678.621	20716.242	20668.554	20673.359	20669.575	20667.394	20671.517
Weight _{mean} (lb)	31401.182	26424.703	21175.514	20671.274	20666.137	20666.462	20666.391	20666.309	20666.580
Weight _{median} (lb)	—	—	—	20667.828	20665.989	20665.986	20666.074	20666.153	20666.231
Weight _{stdev} (lb)	5653.241	4103.610	259.105	10.423	0.488	1.400	0.882	0.356	1.235
N _{analyses}	—	—	—	10000	10000	10000	10000	10000	10000

example. Comparing DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4, EDE-1 provided minimum best design weight (20665.825 lb), EDE provided minimum mean weight (20666.137 lb), EDE-1 provided minimum median weight (20665.986 lb), and EDE-3 provided minimum standard deviation (0.356 lb) of design weight. In fact, the median weight of EDE (20665.989 lb) is only a little larger than that of EDE-1, and the standard deviation of design weight of EDE (0.488 lb) is only a little larger than that of EDE-3 (0.356 lb); the results still showed that the EDE algorithm had excellent computational efficiency and consistency for this example. As presented in Figure 13, EDE optimized the 120-bar truss faster than DE, and EDE converged fastest in the early search period. The DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms required approximately 82, 60, 70, 66, 67, and 87 generations (4100, 3000, 3500, 3300, 3350, and 4350 analyses), respectively, to converge to a solution.

5.5. Nine-Hundred and Forty-Two-Bar Truss. The 942-bar spatial truss is a 26-story tower truss proposed by several researchers. Figures 14 and 15 display the geometry and element numbering of the truss. The allowable displacement is limited to ± 15 in, and the allowable stress is limited to ± 25 ksi. The allowable cross-sectional area of all members is between 1 and 200 in². The members of this truss are grouped into 59 categories. The tower is subject to a single loading condition consisting of horizontal and vertical loads, as follows: (i) the vertical loads in the z direction are -3.0 , -6.0 , and -9.0 kips at each node in the first, second, and third sections, respectively, (ii) the lateral loads in the y direction are 1.0 kip at all nodes of the tower, and (iii) the lateral loads in the x direction are 1.5 and 1.0 kips at each node on the left and right sides of the tower, respectively.

In this example, the parameters of DE and EDE-1 are set to NP = 200, $F = 0.3$, CR = 0.7, and $G_{\max} = 500$. The parameters of EDE, EDE-2, EDE-3, and EDE-4 are set to NP = 200, $F_u = 0.7$, $F_l = 0.3$, CR = 0.8, and $G_{\max} = 500$. Table 8 compares designs for the 942-bar truss using the proposed algorithms with other optimization techniques. The best truss design of EDE (132441 lb) is lighter than those of DE (142436 lb) and

FIGURE 13: Convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 for the 120-bar truss.

Adaptive ESs [47]. The mean (133153 lb) and standard deviation (483.1 lb) of design weight of EDE are also smaller than those of DE. Moreover, EDE required fewer analyses compared with Adaptive ESs [47]. Results revealed that EDE is more robust than DE and Adaptive ESs. Since the difference between mean (or median) weights of DE and EDE-1 is larger than that of EDE and EDE-1, the effect of *integrated mutation* strategy is larger than that of *adaptive mutation factor* strategy on the performance of EDE in this example. Comparing DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4, EDE provided minimum best design weight (132441 lb), mean weight (133153 lb), and median weight (133018 lb). The standard deviation of design weight of EDE (483.1 lb) is only a little larger than that of EDE-3 (396.9 lb) and smaller than others; the results showed that the EDE algorithm had excellent computational efficiency and consistency for this example. As presented in Figure 16, EDE optimized the 120-bar truss faster in the later search period than DE, and EDE-1 converged fastest in the early search period. The DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms required approximately 400, 300, 260, 300, 300, and 400 generations (80000, 60000, 52000, 60000, 60000, and 80000 analyses), respectively, to converge to a solution.

FIGURE 14: Layout of the 942-bar truss (reproduced from Hasancebi [47], under the Creative Commons Attribution License/public domain).

FIGURE 15: Element numbering and sizes of the 942-bar truss (reproduced from Hasancebi [47], under the Creative Commons Attribution License/public domain).

TABLE 8: Performance comparison of the 942-bar truss.

Element group	Members	Cross-sectional areas (in ²)						
		Hasancebi [47] Adaptive ESs	This work					
			DE	EDE	EDE-1	EDE-2	EDE-3	EDE-4
1	1, 2	1.020	5.503	1.867	1.177	1.805	1.527	1.351
2	3–10	1.037	3.807	0.325	0.108	0.130	0.246	0.239
3	11–18	2.943	3.738	3.533	4.120	3.730	3.657	2.752
4	19–34	1.920	2.442	1.982	2.248	1.870	1.870	1.923
5	35–46	1.025	11.269	10.283	9.062	10.473	11.110	11.668
6	47–58	14.961	10.952	10.717	10.379	10.717	10.027	10.362
7	59–82	3.074	10.436	9.427	9.878	9.378	10.401	10.465
8	83–86	6.780	8.689	4.620	4.464	2.708	2.808	5.141
9	87–94	18.580	6.797	8.100	9.911	7.250	6.362	6.107
10	95–98	2.415	23.975	23.242	22.531	23.263	21.361	29.080

TABLE 8: Continued.

Element group	Members	Cross-sectional areas (in ²)						
		Hasancebi [47]	This work					
		Adaptive ESs	DE	EDE	EDE-1	EDE-2	EDE-3	EDE-4
11	99–106	6.584	8.455	7.191	5.441	7.568	7.317	7.038
12	107–122	6.291	4.776	5.839	5.761	5.615	5.001	4.661
13	123–130	15.383	12.470	15.171	16.285	16.669	17.421	17.639
14	131–162	2.100	1.998	2.056	2.139	2.038	2.128	2.139
15	163–170	6.021	2.646	4.374	3.360	3.977	4.023	3.667
16	171–186	1.022	0.100	0.104	0.222	0.100	0.363	0.524
17	187–194	23.099	21.578	22.630	21.533	22.433	21.486	21.802
18	195–226	2.889	2.529	2.323	2.528	2.515	2.486	2.331
19	227–234	7.960	6.817	7.478	8.255	7.111	7.515	7.643
20	235–258	1.008	0.644	0.551	0.779	0.448	0.543	0.379
21	259–270	28.548	32.349	27.738	26.512	28.049	27.327	29.069
22	271–318	3.349	3.304	2.959	2.931	2.776	2.894	2.895
23	319–330	16.144	17.146	15.878	15.792	15.216	15.574	16.259
24	331–338	24.822	22.083	24.093	23.860	22.865	21.512	22.837
25	339–342	38.401	52.242	35.819	34.526	34.530	34.837	29.940
26	343–350	3.787	1.786	1.717	3.466	4.145	1.520	2.794
27	351–358	12.320	19.338	11.374	9.421	9.966	12.545	9.453
28	359–366	17.036	13.275	14.507	12.802	13.606	15.593	13.035
29	367–382	14.733	15.048	12.021	13.493	13.284	13.351	13.338
30	383–390	15.031	12.601	15.345	14.397	14.915	14.688	15.111
31	391–398	38.597	26.285	34.848	35.191	36.275	36.871	36.072
32	399–430	3.511	3.564	3.145	3.099	2.947	3.235	2.926
33	431–446	2.997	5.679	2.083	1.803	2.368	2.216	1.968
34	447–462	3.060	5.278	2.340	2.349	2.565	2.539	2.410
35	463–486	1.086	1.602	0.123	0.185	0.100	0.190	0.227
36	487–498	1.462	0.990	0.213	0.356	0.421	0.100	0.117
37	499–510	59.433	52.600	55.823	56.099	52.331	52.739	53.967
38	511–558	3.632	2.594	2.963	2.928	3.036	3.266	3.046
39	559–582	1.887	2.196	2.320	2.323	2.188	2.317	2.320
40	583–606	4.072	3.954	2.892	2.854	2.953	2.615	2.787
41	607–630	1.595	0.979	0.244	0.121	0.267	0.257	0.104
42	631–642	3.671	0.911	1.134	1.841	1.382	1.284	0.815
43	643–654	79.511	74.845	73.370	76.727	73.712	72.484	72.741
44	655–702	3.394	3.735	3.076	3.041	3.114	2.884	2.907
45	703–726	1.581	1.843	2.076	2.093	1.946	2.006	2.079
46	727–750	4.204	2.666	3.074	3.079	3.313	3.419	3.300
47	751–774	1.329	1.317	0.308	0.725	0.417	0.416	0.483
48	775–786	2.242	0.100	0.277	0.451	0.577	0.553	0.256
49	787–798	96.886	78.724	86.644	86.760	84.354	86.335	88.369
50	799–846	3.710	3.145	3.107	3.195	3.130	3.142	3.619
51	847–870	1.055	8.557	2.075	0.658	2.384	2.469	2.599
52	871–894	4.566	4.121	3.539	2.949	3.452	3.830	3.869
53	895–902	9.606	30.069	9.719	7.361	12.883	9.837	10.473
54	903–906	2.984	12.180	8.382	7.519	8.500	8.553	10.070
55	907–910	45.917	54.295	40.526	40.681	40.973	43.239	39.841
56	911–918	1.000	0.535	0.129	0.201	0.271	0.171	0.132
57	919–926	62.426	31.736	56.362	59.574	55.438	53.716	52.565
58	927–934	2.977	4.960	4.725	2.888	3.893	5.330	4.996
59	935–942	1.000	8.553	0.122	0.225	2.893	0.763	0.335
	Weight _{best} (lb)	141,241	142,436	132,441	132,529	132,713	132,658	132,905
	Weight _{worst} (lb)	—	151,351	134,404	135,764	135,694	134,170	135,632
	Weight _{mean} (lb)	—	145,986	133,153	133,386	133,436	133,243	133,942
	Weight _{median} (lb)	—	145,639	133,018	133,279	133,280	133,161	133,698
	Weight _{stdev} (lb)	—	2292.7	483.1	613.0	709.5	396.9	652.3
	N _{analyses}	150,000	100,000	100,000	100,000	100,000	100,000	100,000

FIGURE 16: Convergence rates of DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 for the 942-bar truss.

TABLE 9: Average ranks for different algorithms across all five truss problems.

Algorithm	10-bar truss	25-bar truss	72-bar truss	120-bar truss	942-bar truss	Mean ranking	Rank
DE	5.73	5.70	5.73	5.80	6	5.79	6
EDE	2.63	2.20	2.37	2.40	2.17	2.35	1
EDE-1	2.80	5.20	2.67	2.77	3.17	3.32	3
EDE-2	2.90	2.27	2.53	2.87	3.03	2.72	2
EDE-3	3.77	3.03	4.03	3.70	2.47	3.40	4
EDE-4	3.17	2.60	3.67	3.47	4.17	3.42	5
Friedman P value	0.000	0.000	0.000	0.000	0.000	0.000	

5.6. Numerical Results and Statistical Comparisons of the Proposed Algorithms. In this section, the statistical analysis results of applying Friedman test [48–50] between DE, EDE, EDE-1, EDE-2, EDE-3, and EDE-4 algorithms will be discussed. Table 9 lists the average ranks according to Friedman test for the six algorithms. The best ranks are shown in bold. From this table, we can see that P values computed through Friedman test for all five truss problems are less than 0.05. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. EDE ranked best in all five truss problems. Regarding mean ranking, EDE gets the first ranking followed by EDE-2, EDE-1, EDE-3, EDE-4, and DE. This observation confirms the positive effect of *integrated mutation* and *adaptive mutation factor* strategies on the EDE algorithm for truss optimization.

6. Conclusions

This work proposed an EDE algorithm for truss design, which improves the performance of the original DE by modifying the mutation operator using two new strategies, namely, *adaptive mutation factor* and *integrated mutation* strategies. *Adaptive mutation factor* strategy systematically adapts the mutation factor from a large value to a small value on the basis of the typical convergence curves of truss optimization to avoid premature convergence in the early searching period and to increase convergence to the global optimum solution in the later searching period. *Integrated mutation* strategy combined

three mutation strategies, *DE/rand/1*, *DE/best/2*, and *DE/rand-to-best/1*, to increase the diversity of random search and avoid premature convergence to a local minimum. The effectiveness of the proposed EDE was demonstrated by using it to solve the *10-bar truss*, *25-bar truss*, *72-bar truss*, *120-bar truss*, and *942-bar truss* optimization problems. The following important conclusions are drawn from the results.

- (1) EDE yielded results that competed favorably with those generated using original DE and other metaheuristic algorithms (the GA, PSO, PSOPC, AugPSO, ACO, BB-BC, and ABC algorithms) in the literature. Furthermore, EDE provided an extraordinary result within fewer analyses than required by other methods. EDE is highly competitive in terms of robustness, stability, and quality of the solution obtained.
- (2) Compared with EDE-1, EDE-2, EDE-3, and EDE-4, EDE provided minimum best design weight and mean (or median) design weight, and the standard deviation of design weight of EDE is small for most truss optimization problems. Moreover, EDE gets the first ranking followed by EDE-2, EDE-1, EDE-3, and EDE-4 by the Friedman test. This observation confirms the positive effect of *integrated mutation* and *adaptive mutation factor* strategies on the EDE algorithm for truss optimization.
- (3) Mutation factor F and crossover rate CR greatly influence the effectiveness, efficiency, and robustness of the DE algorithm. The choice of F_l and F_u in the proposed EDE algorithm is subjective. Further

research could investigate how to determine F_l and F_u more objectively. Moreover, EDE uses a constant CR in this study; the proposed EDE algorithm with an adaptive CR could be further researched.

- (4) This study focuses on comparing effectiveness of the proposed EDE algorithm with those of original DE and other metaheuristic algorithms in the literature for solving truss optimization problems. The effectiveness comparison of EDE and other advanced DE variants and application of EDE to solve other problems could be further researched.

Data Availability

The datasets generated and/or analysed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers & Structures*, vol. 82, no. 9–10, pp. 781–798, 2004.
- [2] A. Kumar, R. K. Misra, D. Singh, S. Mishra, and S. Das, "The spherical search algorithm for bound-constrained global optimization problems," *Applied Soft Computing*, vol. 85, p. 105734, 2019.
- [3] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 34, 2019.
- [4] S.-J. Wu and P.-T. Chow, "Steady-state genetic algorithms for discrete optimization of trusses," *Computers & Structures*, vol. 56, no. 6, pp. 979–991, 1995.
- [5] C. Camp, S. Pezeshk, and G. Cao, "Optimized design of two-dimensional structures using a genetic algorithm," *Journal of Structural Engineering*, vol. 124, no. 5, pp. 551–559, 1998.
- [6] G. Cao, "Optimized design of framed structures using a genetic algorithm," Ph.D. thesis, University of Memphis, Memphis, Tenn, USA, 1996.
- [7] V. C. Charles and J. B. Barron, "Design of space trusses using ant colony optimization," *Journal of Structural Engineering –ASCE*, vol. 130, pp. 741–751, 2004.
- [8] C. V. Camp and B. J. Bichon, "Design of space trusses using ant colony optimization," *Journal of Structural Engineering*, vol. 130, no. 5, pp. 741–751, 2004.
- [9] A. Kaveh and S. Shojaei, "Optimal design of skeletal structures using ant colony optimization," *International Journal for Numerical Methods in Engineering*, vol. 70, no. 5, pp. 563–581, 2007.
- [10] M. Sonmez, "Artificial bee colony algorithm for optimization of truss structures," *Applied Soft Computing*, vol. 11, no. 2, pp. 2406–2418, 2011.
- [11] P. C. Fourie and A. A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259–267, 2002.
- [12] J. F. Schutte and A. A. Groenwold, "Sizing design of truss structures using particle swarms," *Structural and Multidisciplinary Optimization*, vol. 25, no. 4, pp. 261–269, 2003.
- [13] L. J. Li, F. M. Ren, F. Liu, and Q. H. Wu, "An improved particle swarm optimization method and its application in civil engineering," in *Proceedings of the 5th International Conference on Engineering Computational Technology*, New York, NY, USA, September 2006.
- [14] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," *Computers & Structures*, vol. 85, no. 7–8, pp. 340–349, 2007.
- [15] S.-K. S. Fan and Y.-Y. Chiu, "A decreasing inertia weight particle swarm optimizer," *Engineering Optimization*, vol. 39, no. 2, pp. 203–228, 2007.
- [16] S.-K. S. Fan and J.-M. Chang, "A parallel particle swarm optimization algorithm for multi-objective optimization problems," *Engineering Optimization*, vol. 41, no. 7, pp. 673–697, 2009.
- [17] S.-K. S. Fan and J.-M. Chang, "Dynamic multi-swarm particle swarm optimizer using parallel PC cluster systems for global optimization of large-scale multimodal functions," *Engineering Optimization*, vol. 42, no. 5, pp. 431–451, 2010.
- [18] M. Montazeri-Gh, S. Jafari, and M. R. Ilkhani, "Application of particle swarm optimization in gas turbine engine fuel controller gain tuning," *Engineering Optimization*, vol. 44, no. 2, pp. 225–240, 2012.
- [19] S. K. Zeng and L. J. Li, "Particle swarm-group search algorithm and its application to spatial structural design with discrete variables," *International Journal of Optimization in Civil Engineering*, vol. 2, no. 4, pp. 443–458, 2012.
- [20] C. V. Camp, "Design of space trusses using big bang-big crunch optimization," *Journal of Structural Engineering*, vol. 133, no. 7, pp. 999–1008, 2007.
- [21] C.-Y. Wu and K.-Y. Tseng, "Truss structure optimization using adaptive multi-population differential evolution," *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, pp. 575–590, 2010.
- [22] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang, "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables," *Computers & Structures*, vol. 165, pp. 59–75, 2016.
- [23] S. Binita and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 137–151, 2012.
- [24] M. Dorigo, "Optimization, learning and natural algorithms," Politecnico di Milano, Italy, Ph.D. dissertation, in Italian, 1992.
- [25] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, New York, NY, USA, 2005.
- [26] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA, USA, 2001.
- [27] S. He, Q. H. Wu, J. Y. Wen, J. R. Saunders, and R. C. Paton, "A particle swarm optimizer with passive congregation," *Bio-systems*, vol. 78, no. 1–3, pp. 135–147, 2004.
- [28] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers & Structures*, vol. 87, no. 5–6, pp. 267–283, 2009.
- [29] Y. C. Lu, J. C. Jan, S. L. Hung, and G. H. Hung, "Enhancing particle swarm optimization algorithm using two new

- strategies for optimizing design of truss structures,” *Engineering Optimization*, vol. 45, no. 10, pp. 1251–1271, 2013.
- [30] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [31] O. K. Erol and I. Eksin, “A new optimization method: Big Bang-Big Crunch,” *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.
- [32] A. Rezaee Jordehi, “A chaotic-based big bang-big crunch algorithm for solving global optimisation problems,” *Neural Computing and Applications*, vol. 25, no. 6, pp. 1329–1335, 2014.
- [33] R. Storn and K. Price, “*Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, Technical Report No. TR-95-012, International Computer Science Institute, Berkley, CA, USA, 1995.
- [34] H. Tang, S. Xue, and C. Fan, “Differential evolution strategy for structural system identification,” *Computers & Structures*, vol. 86, no. 21–22, pp. 2004–2012, 2008.
- [35] S. Das, S. S. Mullick, and P. N. Suganthan, “Recent advances in differential evolution—an updated survey,” *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [36] A. W. Mohamed, H. Z. Sabry, and M. Khorshid, “An alternative differential evolution algorithm for global optimization,” *Journal of Advanced Research*, vol. 3, no. 2, pp. 149–165, 2012.
- [37] Y. Ao and H. Chi, “Experimental study on differential evolution strategies,” *Proceedings of 2009 WRI Global Congress on Intelligent Systems*, vol. 2, pp. 19–24, 2009.
- [38] L. Tang, Y. Dong, and J. Liu, “Differential evolution with an individual-dependent mechanism,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 560–574, 2015.
- [39] H. Peng and Z. Wu, “Heterozygous differential evolution with Taguchi local search,” *Soft Computing*, vol. 19, no. 11, pp. 3273–3291, 2015.
- [40] H. Peng, Z. Wu, and C. Deng, “Enhancing differential evolution with commensal learning and uniform local search,” *Chinese Journal of Electronics*, vol. 26, no. 4, pp. 725–733, 2017.
- [41] H. Peng, Z. Guo, C. Deng, and Z. Wu, “Enhancing differential evolution with random neighbors based strategy,” *Journal of Computational Science*, vol. 26, pp. 501–511, 2018.
- [42] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, “Differential evolution using a neighborhood-based mutation operator,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [43] J. Qiang and C. Mitchell, *An Adaptive Unified Differential Evolution Algorithm for Global Optimization*, LBNL Report LBNL-6853E, Berlin, Germany, 2015.
- [44] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [45] R. Gamperle, S. D. Muller, and P. Koumoutsakos, “A parameter study for differential evolution,” *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, vol. 10, pp. 293–298, 2002.
- [46] AISC-ASD (American Institute of Steel Construction-Allowable Stress Design), *Manual of Steel Construction-Allowable Stress Design*, AISC-ASD (American Institute of Steel Construction-Allowable Stress Design), Berlin, Germany, 1989.
- [47] O. Hasançebi, “Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures,” *Computers & Structures*, vol. 86, no. 1–2, pp. 119–132, 2008.
- [48] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [49] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, “novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization,” *Swarm and Evolutionary Computation*, vol. 50, 2019.
- [50] A. K. Mohamed and A. W. Mohamed, “Real-parameter unconstrained optimization based on enhanced AGDE algorithm,” in *Machine Learning Paradigms: Theory and Application*, A. Hassanien, Ed., Vol. 801, Springer, Berlin, Germany, 2019.