Advances in Computational Design, *Vol. 1, No. 3 (2016) 275-296*
DOI: http://dx.doi.org/10.12989/acd.2016.1.3.275

Discrete optimal sizing of truss using adaptive directional differential evolution

Anh H. Pham*

Department of Structural Mechanics, National University of Civil Engineering, 55 Giai Phong Road, Hanoi, Vietnam

(Received June 22, 2016, Revised July 19, 2016, Accepted July 21, 2016)

Abstract. This article presents an adaptive directional differential evolution (ADDE) algorithm and its application in solving discrete sizing truss optimization problems. The algorithm is featured by a new self-adaptation approach and a simple directional strategy. In the adaptation approach, the mutation operator is adjusted in accordance with the change of population diversity, which can well balance between global exploration and local exploitation as well as locate the promising solutions. The directional strategy is based on the order relation between two difference solutions chosen for mutation and can bias the search direction for increasing the possibility of finding improved solutions. In addition, a new scaling factor is introduced as a vector of uniform random variables to maintain the diversity without crossover operation. Numerical results show that the optimal solutions of ADDE are as good as or better than those from some modern metaheuristics in the literature, while ADDE often uses fewer structural analyses.

Keywords: adaptive directional differential evolution; population diversity; truss sizing optimization; discrete variables

1. Introduction

The goal of structural optimization is to obtain appropriate form for a structure so that it is safe and economical. Structural optimization can be classified as sizing optimization (finding optimal size of structural members), shaping optimization (obtaining the optimal form for the structure) and topology optimization (optimal size and connectivity between structural members). These have been an extensive research area both in modeling and development of optimization methods.

Optimal sizing design of truss structure is an important field within structural optimization. Truss sizing optimization is known as a difficult optimization problem because of non-linear constraints and non-convex feasible region, which requires appropriate optimization techniques. Moreover, the design variables (the cross-section areas) are usually discrete values which can be selected from a list of available values provided by manufacturers. These inherent characteristics of the problem do not favor conventional gradient-based techniques. Developing efficiently alternative methods for truss optimization with discrete design variables remains one of the interesting subjects for many researchers.

*Corresponding author, Ph.D., E-mail: anhpham.nuce@gmail.com

http://www.techno-press.com/journals/acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

Nowadays, metaheuristic algorithms have become more and more popular for truss optimization with discrete design variables (Stolpe 2015). Methods in this field include genetic algorithms (GA) (Rajeev and Krishnamoorthy 1992, Hajela and Lin 1992), Tabu search (Bennage and Dhingra 1995a), simulated annealing (Bennage and Dhingra 1995b), ant colony optimization (ACO) (Bland 2001, Camp and Bichon 2004), harmony search (HS) algorithm (Lee *et al*. 2005), big bang-big crunch (Camp 2007), particle swarm optimization (PSO) (Li *et al*. 2009), differential evolution (Wang *et al*. 2009), and several modified or hybrid variants of them (see Stolpe 2015). Besides, new algorithms proposed for truss design with discrete design variables appear in the literature on a regular basis. Some of recent algorithms are mine blast algorithm (MBA) (Sadollah *et al*. 2012), refined big bang-big crunch algorithm (Hasançebi and Azad 2014), guided stochastic search (GSS) (Azad *et al*. 2014, Azad and Hasançebi 2015), teaching-learning-based optimization (TLBO) (Camp and Farshchin 2014), supervised charged system search (Kaveh and Ahmadi 2014), colliding bodies optimization (CBO) (Kaveh and Mahdavi 2014), Enhanced colliding bodies optimization (ECBO) (Kaveh and Ghazaan 2014 and 2015), elitist self-adaptive step-size search (ESASS) (Azad and Hasançebi 2014), adaptive dimensional search (ADS) (Hasançebi and Azad 2015), improved mine blast algorithm (IMBA) and water cycle algorithm (WCA) (Sadollah *et al*. 2015). Although metaheuristics can find promising solutions, they often require a high number of function evaluations. As such, performance enhancement of metaheuristics to obtain sufficiently good result with reasonable computational cost is thus always the issue (Azad *et al*. 2013, Bureerat and Pholdee 2015).

Among various metaheuristics, differential evolution (DE) (Storn and Price 1997) is a simple one and has shown to be efficient for numerous optimization problems from diverse domains of science and technology (Das and Suganthan 2011). Relatively few applications of DE on optimization of truss structures with discrete variables have appeared in the literature. Wang *et al*. (2009) reported a very first study of DE for optimization of truss with continuous and discrete variables. Krempser *et al*. (2012) introduced the SMDE which is combination of surrogate models and DE for sizing optimization of truss. Recently, Ho-Huu *et al.* (2016) proposed a felicitous approach which adaptively employs multiple mutation operators in their adaptive elitist differential evolution algorithm, aeDE. Similar to other metaheuristics, exploration/exploitation balance is a key feature to control the performance of a DE algorithm (Das *et al*. 2009). Several DE variants have been proposed to deal with this issue and achieved better performance on various problems. However, simplicity of DE has been usually decreased in many of those modified DE versions.

To enhance the performance of DE for solving discrete sizing truss optimization problems while maintaining the simplicity, three modifications are introduced in this article. The modifications include: 1) self-adaptive mutation operation based on the change in population diversity for balancing global exploration and local exploitation; 2) simple directional variation rule for increasing the possibility of finding an improved solution; and 3) random scaling factor to maintain the diversity without crossover operation. Combining with a simple rounding technique for treatment of discrete variable, a new algorithm called adaptive directional differential evolution (ADDE) is proposed. Four benchmark problems of discrete sizing truss optimization are used to investigate the performance of ADDE. The results are compared with those of some competitive modern metaheuristics, including three other existing adaptive DE variants.

The rest of this article is organized as follows. In section 2, the formulation of the truss optimization problem and the constraint handling rules are presented. The basic DE is briefly introduced in section 3. Then, the ADDE algorithm is described in section 4. In section 5, the test problems and numerical results are shown and discussed. Conclusions are given in section 6.

2. Truss sizing optimization with discrete variables

2.1 Problem formulation

For the class of truss optimization problems considered in this study, the objective function is the total weight of a truss structure and the design variables are cross-section area of the truss elements. The design constraints are limits on stress in the structural members and/or the nodal displacement. The problem is typically formulated as Eq. (1)

$$\begin{gathered} \text{Minimize } W(\mathbf{A}) = \sum_{e=1}^{M} L_e\, A_e \rho_e, \quad e = 1, 2, \ldots, M \\ \text{subject to } \sigma_{min} \le \sigma_e(\mathbf{A}) \le \sigma_{max} \\ \delta_{min} \le \delta_n(\mathbf{A}) \le \delta_{max} \\ \mathbf{A} = \{A_e\} \in \mathbf{S} = \{A_1, A_2, \ldots, A_P\} \end{gathered} \tag{1}$$

where $\mathbf{A}$ is the vector of design variables, i.e., cross-section areas; $\mathbf{S}$ is the list of P allowable discrete values of cross-section area; $W(\mathbf{A})$ is the weight of the truss; L_e, A_e, ρ_e are the length, the cross-section area and the material density of the e-th element, respectively; M is the number of elements; $\sigma_e(\mathbf{A})$ and $\delta_n(\mathbf{A})$ are the stress in the e-th element and the n-th nodal displacement, respectively. The subscript 'min' and 'max' denote the minimum and maximum limits.

2.2 Constraint handling

To solve the constrained optimization problem as given in Eq. (1), the following constraint handling rules (Deb 2000) are employed:

1. A feasible solution is better than any infeasible one.
2. Between two feasible solutions or two solutions with equal constraint violation, the one having smaller objective function value is better.
3. Between two infeasible solutions, the one having smaller constraint violation is better.

The constraint violation of a solution $\mathbf{A}$ is determined by Eqs. (2) and (3)

$$C(\mathbf{A}) = \sum_{i=1}^{N_c} \max\{0, c_i(\mathbf{A})\} \tag{2}$$

$$c_i(\mathbf{A}) = \frac{g_i(\mathbf{A})}{g_{0,i}} - 1 \tag{3}$$

where $C(\mathbf{A})$ is the constraint violation; N_c is the number of constraints of the optimization problem; $g_i(\mathbf{A})$ is the i-th constraint function (stress or nodal displacement) and $g_{0,i}$ is the permissible values of $g_i(\mathbf{A})$.

3. Basic differential evolution

Differential evolution (DE) introduced by Storn and Price (Storn and Price 1997) is a population-based metaheuristics. DE has been shown to be one of the most efficient direct search methods and suitable for solving optimization problems in many fields (Das and Suganthan 2011).

As other population-based metaheuristics, DE uses a population of NP candidate vectors $\boldsymbol{x}_k (k = 1,2,\dots,NP)$ (called individuals) of the design variables. The population is then restructured by survival individuals evolutionally. First, an initial population is randomly sampled from the solution space as

$$x_{k,i} = x_i^l + rand[0,1] \times \left(x_i^u - x_i^l\right), \;\; i = 1,2,\dots,D \tag{4}$$

where x_i^l and x_i^u are the lower and upper bounds of $x_{k,i}$, respectively; D is the number of design variables of the optimization problem; $rand[0,1]$ is a uniformly distributed random real value in the range [0,1]. Then, each individual $\boldsymbol{x}_k$ (called the target vector) of the current population is compared with a newly generated vector (called the trial vector) and the better will be selected as member for the population of next generation. The evolution proceeds until a termination criterion is reached. The crucial idea behind DE is a scheme for producing trial vectors. Two operators, named as 'mutation' and 'crossover', are used for this purpose and they are described as follows.

Mutation: For each target vector $\boldsymbol{x}_k$, a mutant vector $\boldsymbol{y}$ is first generated. There are various mutation strategies which can be employed to create the mutant vector. Some popular ones are

$$-\,\mathrm{DE/rand/1}: \;\; \boldsymbol{y} = \boldsymbol{x}_{r_1} + F \times \left(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}\right) \tag{5}$$

$$-\,\mathrm{DE/best/1}: \;\; \boldsymbol{y} = \boldsymbol{x}_{best} + F \times \left(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}\right) \tag{6}$$

$$-\mathrm{DE/current-to-rand/1}: \;\; \boldsymbol{y} = \boldsymbol{x}_k + F \times (\boldsymbol{x}_{r_1} - \boldsymbol{x}_k) + F \times \left(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}\right) \tag{7}$$

$$-\mathrm{DE/current-to-best/1}: \;\; \boldsymbol{y} = \boldsymbol{x}_k + F \times (\boldsymbol{x}_{best} - \boldsymbol{x}_k) + F \times \left(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}\right) \tag{8}$$

where $\boldsymbol{x}_{r_1}, \boldsymbol{x}_{r_2}, \boldsymbol{x}_{r_3}$ are three mutually different individuals randomly selected from the current population, i.e., $r_1 \neq r_2 \neq r_3 \neq k$; $\boldsymbol{x}_{best}$ is the current best individual; F is a scaling factor, a real and constant factor usually chosen in the interval $[0,1]$, which controls the amplification of the differential variations.

Crossover: Crossover is introduced to exchange the information of the mutant vector with the target vector $\boldsymbol{x}_k$, creating a trial vector $\boldsymbol{z}$ with its elements determined by

$$z_i = \begin{cases} y_i, & \text{if } (rand[0,1] \leq Cr) \text{ or } (r = i) \\ x_{k,i}, & \text{otherwise} \end{cases} \tag{9}$$

where $i = 1,2,\dots,D$; $rand[0,1]$ is a uniformly distributed random number in [0,1]; r is randomly chosen integer in the interval $[1, D]$ to ensure that the trial vector has at least one parameter value from the mutant vector; Cr is the crossover rate predefined in [0, 1], which control the fraction of parameter values copied from the mutant vector.

It is well-known that mutation strategy plays a vital role in the DE search capability and convergence rate. For example, the 'DE/rand/1' strategy is able to maintain population diversity and global exploration ability. However, its local exploitation ability is regarded weak and its convergence velocity is often too low. By contrast, the 'DE/best/1' and 'DE/current-to-best/1' strategies for instance, which better take advantage of the guiding information of the best individual, have great local exploitation ability and fast convergence velocity. However, they can lose population diversity and suffers from the problem of premature convergence. In order to balance between global exploration and local exploitation, using multiple mutation operators is commonly suggested and still a research focus for improving DE's performance (Mallipeddi *et al*. 2011, Wang *et al*. 2011, Gong *et al*. 2011, Elsayed *et al*. 2011, Takahama and Sakai 2012, Wu *et al*.

2015, Xiang *et al*. 2015, Zamuda *et al*. 2013, Kushida *et al*. 2015).

Moreover, the values of the scaling factor and crossover rate have great impact on the performance of DE. Numerous research works studied the effects of these control parameters and suggested parameter setting or proposed parameter adaptations (Qin and Suganthan 2005, Zhang and Sanderson 2009, Wang *et al*. 2011, Tanabe and Fukunaga 2013). Nevertheless, simplicity of DE has been usually reduced in those adaptive versions (Yang and Yao 2014).

4. The ADDE algorithm

In this section, an enhanced DE, namely ADDE, is proposed considering both efficiency and simplicity. Three modifications are introduced, which are: 1) adaptive 'current-to-pbest/1' mutation for balancing global exploration and local exploitation abilities; 2) simple directional variation rule for increasing the possibility of finding an improved new solution; and 3) random scaling factor for keeping diversity with no crossover operation. The details of these modifications are described in the followings.

4.1 Adaptive 'current-to-pbest' mutation

In truss optimization problem, the objective function (the weight of the structure in this study) is often a unimodal function. The challenge will be caused by a non-convex feasible region. Therefore, global exploration of the search space at the beginning can quickly get close to the region containing an optimum. Later, exploitation of the region nearby the optimum is more demanding than other regions. Under such considerations, in this study, an adaptive mutation scheme is proposed to enhance the performance of DE.

For this purpose, the 'pbest' method 'DE/current-to-pbest/1' used in JADE (Zhang and Sanderson 2009) is employed. In 'DE/current-to-pbest/1', a mutant vector **y** is produced as

$$\boldsymbol{y} = \boldsymbol{x}_k + F \times (\boldsymbol{x}_{pbest} - \boldsymbol{x}_k) + F \times (\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) \tag{10}$$

with $\boldsymbol{x}_{pbest}$ is selected randomly from the top $p \times NP$ ($p \in (0,1]$) individuals. Thus, several good individuals will be utilized to guide the search, making this method more reliable than the 'DE/current-to-best/1' (Zhang and Sanderson 2009).

As a matter of fact, the p value plays an important role in balancing the exploration ability and the exploitation ability of this method. It is desirable that good global exploration is maintained at the beginning of the evolution and fast convergence velocity is achieved at the end of the optimization process. So in this study, the value of p is adaptively adjusted during the search based on the change of population diversity as follows

$$p = \frac{1}{NP} + \left(1 - \frac{1}{NP}\right) \times \frac{DI_t}{DI_0} \tag{11}$$

where DI_t is a diversity index that measure of the diversity of population at the t-th generation; DI_0 is the diversity index of the initial population. The diversity index is defined as

$$DI_t = \frac{1}{NP} \sum_{k=1}^{NP} \sqrt{\sum_{i=1}^{D} \left(\frac{x_{k,i} - x_{C,i}}{x_i^u - x_i^l} \right)^2}; \quad x_{C,i} = \frac{1}{NP} \sum_{k=1}^{NP} x_{k,i} \tag{12}$$

where $x_{C,i}$ is the mean value of the i-th design variable of all solutions in the population. The concept of diversity index was introduced by Kaveh and Zolghadr (2012) to reflect the ratio of the portion of the search space covered by the individuals to the entire search space at each generation. Here, the diversity index is defined to represent the distribution of the individuals around the center of the current population. With this adjustment, in the sooner generations larger values of p are used to favor exploring the domain containing the global optimum, and in later generations smaller values of p are used to enhance exploitation for accelerating the convergence speed. In particular, when $DI_t = DI_0$ (i.e., $p = 1$), the mutation operator 'DE/current-to-rand/1' is utilized; when $DI_t = 0$ (i.e., $p = 1/NP$), the 'DE/current-to-best/1' is performed. This should result in ADDE being suitable for truss optimization problems.

4.2 Directional variation rule

In the mutation operator of Eq. (10), random variations are derived from the difference of two randomly selected different individuals. Consequently, they have no bias to any special search directions. In order to further take advantage of guiding information of the population, the scaled differential variation is multiplied by a 'directed' factor d, i.e.

$$\boldsymbol{y} = \boldsymbol{x}_k + F \times (\boldsymbol{x}_{pbest} - \boldsymbol{x}_k) + d \times F \times \left(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}\right) \tag{13}$$

where d takes either value 1 or -1 depending on the order relation between the two difference vectors $\boldsymbol{x}_{r_2}$ and $\boldsymbol{x}_{r_3}$. Specifically, d is determined as

$$d = \begin{cases} 1, & \text{if } \boldsymbol{x}_{r_2} \text{better than } \boldsymbol{x}_{r_3} \\ -1, & \text{otherwise} \end{cases} \tag{14}$$

This kind of directional mutation has the same concept of the well-known opposition based method presented for improving DE search performance in literatures (Rahnamayan *et al*. 2008, Pholdee *et al*. 2015). This rule guarantees that the scaled differential variation is oriented toward a better vector, thus increasing the possibility of finding an improved solution.

4.3 Modification of scaling factor

In classical DE, the scaling factor F is a constant value often chosen in the interval [0, 1]. In this article, instead of keeping F constant during the search process, for each target vector a set of F values are randomly sampled from uniform distribution in [0, 1] and applied for each component of the mutant vector so as to perturb the base vector by different weights. These random scaling factors attempt to maintain both exploitation (with small F values) and exploration (with large F values) abilities throughout the entire evolution process. Particularly, when a value of F is closed to zero, the corresponding component of the mutant vector is basically identical to the one from the target vector. This case is similar to the crossover operation in DE. Thus, by introducing F as a vector of uniform random variables in [0, 1], one can skip the crossover step.

4.4 Discrete variable handling

To adapt the algorithm for problems with discrete variables, the simple rounding technique (Lampinen and Zenlinka 1999) is applied, i.e., each continuous design variable is rounded to the nearest value in the discrete value set $\mathbf{S}$. This technique was also used in aeDE (Ho-Huu *et al.*

2016) for handling discrete variables in optimization of truss structures.

4.5 Algorithm description

Unlike most other strategy adaptation mechanisms which often use several mutation operators to perform, ADDE adapts itself with only one mutation operator and provides a gradual transition from explorative to exploitative operation during the evolution. This adaptation is based on the change of the population diversity, which is dependent on the problem characteristics. In the next section, it is shown that ADDE can suit different truss optimization problems considered.

The determination of diversity index in ADDE requires additionally computational cost of *NP* calculations for normalized Euclidean distance at each generation. Nevertheless, this additional cost is normally negligible compared with the overall computational cost taken to solve the optimization problem. This is because in structural optimization that the computational cost for function evaluation (involving structural analysis) is so large, and the size of the population is so small, that the time taken for calculating the diversity index will be comparatively small.

The ADDE algorithm is simple and requires only the common control parameters like population size and number of generations for its working and does not require crossover operation. The procedure of ADDE is summarized in **Algorithm 1.**

Algorithm 1: The pseudo-code of ADDE

Define NP, T_{max}, *fitness function, constraints, and allowable discrete values;*
Generate initial population and evaluate fitness and constraints for each individual;
Calculate the diversity index of the initial population, DI_0*;*
$t = 1$;
while $t < T_{max}$ **do**
  Calculate diversity index DI_t *and update p value*;
  for $k = 1$ **to** NP **do**
    Select $\boldsymbol{x}_{pbest}$ *randomly from top* $100p\%$ *individuals in the population;*
    Select randomly two different vectors in the population, $\boldsymbol{x}_{r_2}$ *and* $\boldsymbol{x}_{r_3}$*;*
    if $\boldsymbol{x}_{r_2}$ *is better than* $\boldsymbol{x}_{r_3}$ **then**
      $d = 1$;
    else
      $d = -1$;
    end if
    for $i = 1$ **to** D **do**
      $F = rand[0,1]$;
      $x_{new,i} = x_{k,i} + F \times (x_{pbest,i} - x_{k,i}) + d \times F \times \left(x_{r_2,i} - x_{r_3,i}\right)$;
      Round-off the generated values to the closest discrete value;
    end for
    if $\boldsymbol{x}_{new}$ *is better than* $\boldsymbol{x}_k$ **then**
      $\boldsymbol{x}_k = \boldsymbol{x}_{new}$;
    end if
  end for
  $t = t + 1$;
end while

Table 1 Design conditions of the test problems

	10-bar truss	52-bar truss	72-bar truss	200-bar truss
Design variables	A_i, i=1,2,…,10	G_i, i=1,2,…,12	G_i, i=1,2,...,16	G_i, i=1,2,...,29
Stress constraints	$-25 \le \sigma_e \le 25$ ksi	$-180 \le \sigma_e \le 180$ MPa	$-25 \le \sigma_e \le 25$ ksi	$-10 \le \sigma_e \le 10$ ksi
Displacement constraints	$-2 \le \delta_n \le 2$ in	-	$-0.25 \le \delta_n \le 0.25$ in	-
Material density	0.1 lb/in^3	7860 kg/m^3	0.1 lb/in^3	0.283 lb/in^3
Modulus of elasticity	10^4 ksi	207 GPa	10^4 ksi	30000 ksi

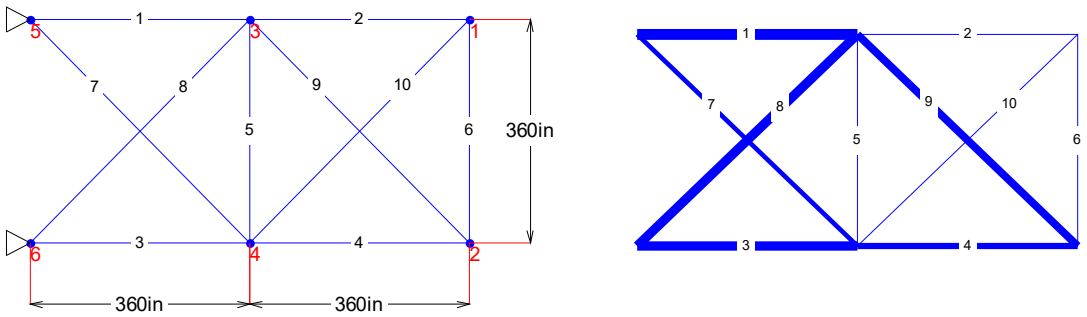
5. Numerical examples

Four benchmark examples of truss structures with 10, 52, 72, and 200 bars are employed in this article to investigate the performance of ADDE. The 10-bar, 52-bar and 200-bar trusses are planar structures, whereas the 72-bar truss is a space structure. The main input data of the problems are given in Table 1. The ADDE is used to solve each problem with 20 independent runs. In all cases, the population size is set to NP=25, which appears reasonable. The maximum iterations performed, T_{max}, are 100 for the 10-bar truss, 150 for the 52-bar and 72-bar trusses, and 300 for the 200-bar truss. Thus, the number of function evaluations is 2500 for the 10-bar truss, 3750 for the 52-bar and 72-bar trusses and 7500 for the 200-bar truss. The results obtained by the proposed algorithm are compared with those of some other modern metaheuristics, like MBA, TLBO, CBO, ECBO, WCA, IMBA, ESASS and ADS, reported recently in the literature. These methods are chosen among various metaheuristics for comparison considering their high quality of optimal solution and computational efficiency. In addition, ADDE is also compared with three other adaptive DE variants, which are aeDE, SHADE (Tanabe and Fukunaga 2013) and JADE. The results of aeDE are taken from Ref. (Ho-Huu *et al.* 2016), whereas the results of SHADE and JADE are obtained by solving each problem with 20 random runs. The control parameters adopted for JADE are population size NP=40, ratio of top-rank solution p=0.2 and learning rate c=0.1, while those for SHADE are population size NP=25, ratio of top-rank solution p=0.1 and memory size H=10.

Furthermore, to better understand the performance of ADDE, the influence of each modification introduced in ADDE algorithm is investigated. However, these investigations are presented only for the first example of the 10 bar-truss problem due to space limitation. The results and discussions are explained in the followings.

5.1 The 10-bar planar truss

The truss layout is depicted in Fig. 1(a). The structure is subjected to downward vertical loads of 100 kips at node 2 and node 4. The design variables are the bar element cross-section areas, which can be chosen from the list: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5 (in^2). The material properties and constraints are given in Table 1. Fig. 1(b) shows the relative virtual effect of element cross-sectional areas of the optimum truss structure obtained by ADDE.

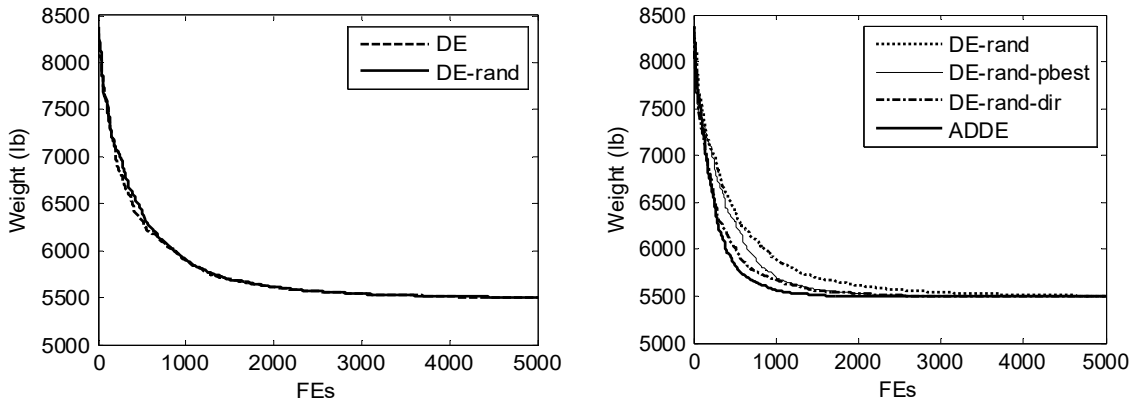


(a) Layout of the 10-bar truss (b) The optimized geometry by ADDE

Fig. 1 The 10-bar truss structure

First, the effectiveness of random scaling factor is examined in this example. For this purpose, the conventional DE with scaling factor of 0.8 and crossover rate of 0.9 and the DE with random scaling factor (DE-rand) are used. The mutation operator 'DE/current-to-rand/1' (Eq. (7)) is adopted as the base algorithm. The problem is solved by each algorithm 20 times with the same number of iteration T_{max}=200. The average convergence histories of minimum weight for the two algorithms are plotted in Fig. 2(a). It is observed that two curves are almost similar, which indicates the effectiveness of the random scaling factor. Thus, the random scaling factor is utilized in further investigation of the other modification.

Second, the influences of the adaptive 'current-to-pbest' mutation and the directional variation rule on the performance of DE are investigated. Four different algorithms, including the DE with random scaling factor, the DE with random scaling factor and the adaptive mutation (DE-rand-pbest), DE with random scaling factor and the directional rule (DE-rand-dir) and DE with all three proposed strategies, i.e., ADDE, are used to perform this problem. Fig. 2(b) shows the average convergence histories of the algorithms. Clearly, DE-rand-pbest and DE-rand-dir converge faster than DE-rand, and ADDE gives the best convergence velocity. It is also seen that, DE-rand-dir is

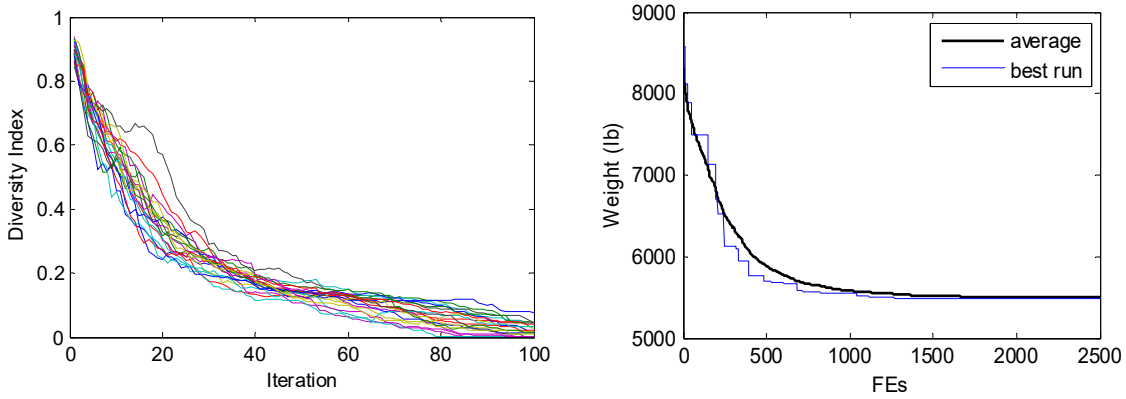


(a) Convergence of DE and DE-rand (b) Convergence of different strategies

Fig. 2 Optimization history for 10-bar truss by different algorithms

Table 2 Optimization results by different strategies for the 10-bar truss

Algorithm	Best weight (lb)	Average weight (lb)	Worst weight (lb)	Standard deviation (lb)
DE-rand	5490.738	5504.521	5549.214	19.052
DE-rand-pbest	5490.738	5497.037	5569.510	19.933
DE-rand-dir	5490.738	5495.867	5531.036	11.267
ADDE	5490.738	5491.872	5513.423	5.073



(a) Diversity index recorded in 20 runs (b) Convergence of minimum weight

Fig. 3 Optimization history for 10-bar truss by ADDE

faster than DE-rand-pbest at sooner generation (about early 40 generations in this example). This is due to the fact that at sooner generations, the p value is large and the mutation 'current-to-pbest' performs similar to 'current-to-rand'.

Table 2 presents the statistical results obtained by these algorithms after 200 generations (i.e., 5000 function evaluations. It is seen that all algorithms provide the same optimal weight of 5490.738 lb, while ADDE give the best results with respects to average weight, worst weight and standard deviation of weight.

Next, the optimization results obtained by this work are compared with those acquired by other algorithms recently reported in the literature, including MBA (Sadollah *et al*. 2012), TLBO (Camp and Farshchin 2014), ADS (Hasançebi and Azad 2015) and aeDE (Ho-Huu *et al.* 2016). Table 3 lists the statistical results, including the best optimal solution, the best weight and its required function evaluations (FE), the average weight, the worst weight, and the standard deviation. It is seen that the best weight of 5490.738 lb obtained by ADDE, SHADE and JADE is the same as that of TLBO, ADS and aeDE and lighter than that of MBA. The average weight from ADDE is 5495.187 lb, which is better than the results from other methods, while the deviation produced by ADDE is smaller than that of TLBO, ADS and aeDE. Furthermore, ADDE is also computationally efficient. In the best run, ADDE obtains the optimal design after 1247 function calls, while the number of function evaluations for MBA, TLBO and aeDE are 3600, 5183 and 2380, respectively. The ADS uses fewest analyses (1000). However, as can be seen from Table 3, ADS gives the most unstable results with highest average weight and deviation. Comparing with JADE, ADDE is slightly better with respects to average weight and worst weight. It is noted that on average, ADDE

Table 3 Comparison on optimal designs of 10-bar truss

Size of members (in^2)	MBA	TLBO	ADS	aeDE	This study		
					SHADE	JADE	ADDE
1	30.00	33.5	33.5	33.5	33.5	33.500	33.500
2	1.62	1.62	1.62	1.62	1.62	1.620	1.620
3	22.90	22.9	22.9	22.9	22.9	22.900	22.900
4	16.90	14.2	14.2	14.2	14.2	14.200	14.200
5	1.62	1.62	1.62	1.62	1.62	1.620	1.620
6	1.62	1.62	1.62	1.62	1.62	1.620	1.620
7	7.97	7.97	7.97	7.97	7.97	7.970	7.970
8	22.90	22.9	22.9	22.9	22.9	22.900	22.900
9	22.90	22	22	22	22	22.000	22.000
10	1.62	1.62	1.62	1.62	1.62	1.620	1.620
Best weight (lb)	5507.758	5490.74	5490.74	5490.738	5490.738	5490.738	5490.738
FE	3600	5183	1000	2380	2069	1973	1247
Average weight (lb)	5527.296	5503.21	5539.97	5502.623	5499.003	5495.504	5495.187
Worst weight (lb)	5536.965	-	5591.43	5549.204	5518.098	5543.319	5534.742
Standard deviation	11.38	20.33	35.86	20.780	9.985	12.008	12.452

obtained these results with 2500 function evaluations, while JADE used 4000 function evaluations. ADDE is also better than SHADE in terms of average weight and computational efficiency, however, SHADE gives smallest worst weight and smallest standard deviation of weight.

The convergence of the minimum weight by ADDE is shown in Fig. 3(b), while the histories of the diversity index recorded in 20 runs are plotted in Fig. 3(a). Fig. 3(b) shows that the best run converges rather fast comparing with the average convergence of all runs. From Fig. 3(a) it is seen that high values of diversity index are provided in the early stages of the optimization process, which allow the algorithm to explore the search space adequately. As the evolution continues, the individuals confine to more promising regions of the search space in order to perform local search and diversity index values gradually decrease. At the end of optimization process, a certain amount of diversity is still maintained in most runs, which shows that the optimization is still in progress.

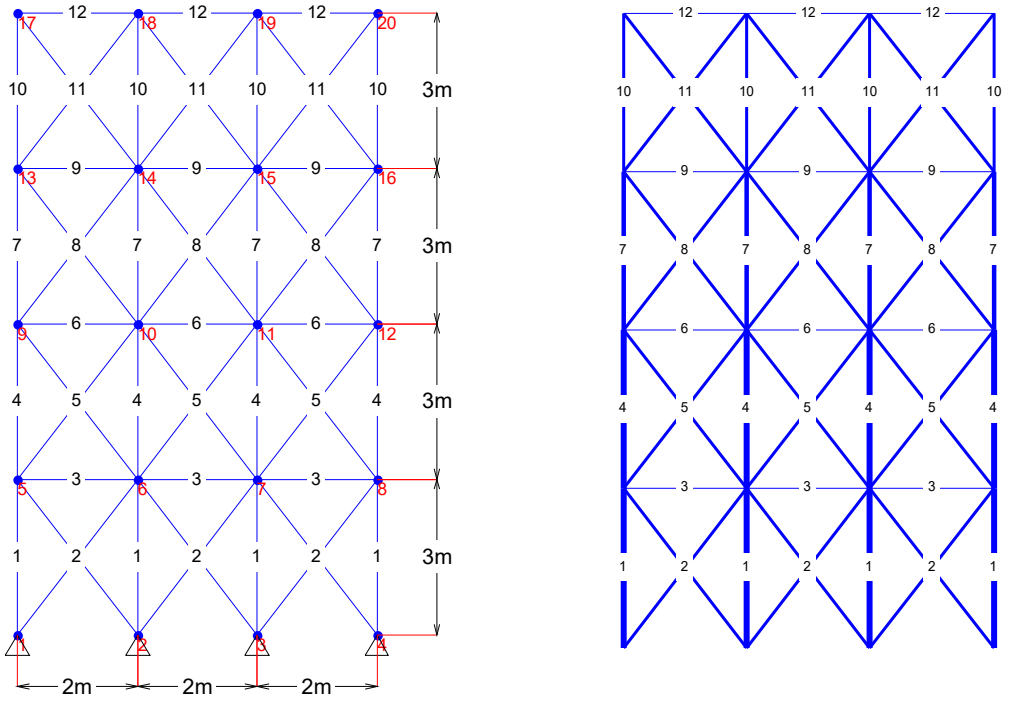
5.2 The 52-bar planar truss

The second example is a 52-bar planar truss structure illustrated in Fig. 4(a). The cross-section area of the truss elements are categorized in 12 groups as shown in Fig. 4(a), and their values are chosen from discrete values listed in Table 4. The material properties and constraints are given in Table 1. The loading condition includes vertical loads of 200 kN and horizontal loads of 100 kN applied at nodes 17, 18, 19 and 20.

This problem has been studied by many authors using different optimization techniques. Table 5 lists the results from ADDE, SHADE and JADE together with those given by some other methods, including CBO (Kaveh and Mandavi 2014), WCA and IMBA (Sadollah *et al*. 2015), and aeDE (Ho-Huu *et al.* 2016). The statistical results highlight the best optimal solution, the best weight and its required function evaluations, the average weight, the worst weight and the standard

Table 4 Available cross-section areas of the AISC code

No.	in^2	mm^2	No.	in^2	mm^2	No.	in^2	mm^2	No.	in^2	mm^2
1	0.111	71.613	17	1.563	1008.385	33	3.840	2477.414	49	11.500	7419.340
2	0.141	90.968	18	1.620	1045.159	34	3.870	2496.769	50	13.500	8709.660
3	0.196	126.451	19	1.80	1161.288	35	3.880	2503.221	51	13.900	8967.724
4	0.25	161.29	20	1.990	1283.868	36	4.180	2696.769	52	14.200	9161.272
5	0.307	198.064	21	2.130	1374.191	37	4.220	2722.575	53	15.500	9999.980
6	0.391	252.258	22	2.380	1535.481	38	4.490	2896.768	54	16.000	10322.560
7	0.442	285.161	23	2.620	1690.319	39	4.590	2961.284	55	16.900	10903.204
8	0.563	363.225	24	2.630	1696.771	40	4.800	3096.768	56	18.800	12129.008
9	0.602	388.386	25	2.880	1858.061	41	4.970	3206.445	57	19.900	12838.684
10	0.766	494.193	26	2.930	1890.319	42	5.120	3303.219	58	22.000	14193.520
11	0.785	506.451	27	3.090	1993.544	43	5.740	3703.218	59	22.900	14774.164
12	0.994	641.289	28	3.130	729.031	44	7.220	4658.055	60	24.500	15806.420
13	1.000	645.16	29	3.380	2180.641	45	7.970	5141.925	61	26.500	17096.740
14	1.228	792.256	30	3.470	2238.705	46	8.530	5503.215	62	28.000	18064.480
15	1.266	816.773	31	3.550	2290.318	47	9.300	5999.988	63	30.000	19354.800
16	1.457	939.998	32	3.630	2341.931	48	10.850	6999.986	64	33.500	21612.860



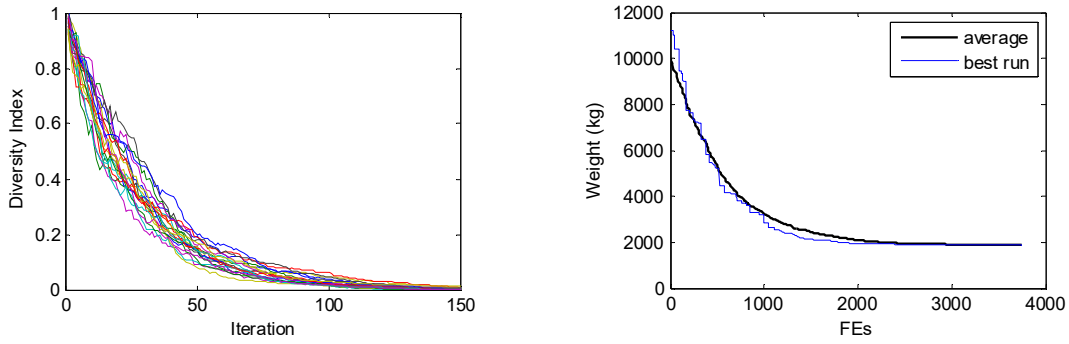
(a) Layout of the 52-bar truss (b) The optimized geometry by ADDE

Fig. 4 The 52-bar truss structure

deviation. It is found that ADDE, SHADE and JADE obtain the same optimum weight of 1902.605 kg as that given by WBA, IMBA and aeDE. Although the best weight given by CBO is 1899.35 kg, its best solution results in a maximum stress of 180.0873 MPa, which violates the constraint. With respects to the average weight and standard deviation, IMBA ranks first, ADDE stands second and SHADE stands third. The average weight and deviation obtained by ADDE are 1903.407 kg and 1.952 kg, which are much closed to the results of IMBA. However, ADDE is more efficient than IMBA in terms of computational cost. In the best run ADDE needs only 2819 function evaluations to reach the optimal solution while this number for IMBA is 4750. On the

Table 5 Comparison on optimal designs of 52-bar truss

Size of grouped members (mm^2)	CBO	WCA	IMBA	aeDE	This study		
					SHADE	JADE	ADDE
1	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
2	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
3	388.386	494.193	494.193	494.193	494.193	494.193	494.193
4	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
5	939.998	940.000	940.000	939.998	939.998	939.998	939.998
6	506.451	494.193	494.193	494.193	494.193	494.193	494.193
7	2238.705	2283.705	2283.705	2238.705	2238.705	2238.705	2238.705
8	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
9	506.451	494.193	494.193	494.193	494.193	494.193	494.193
10	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868
11	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
12	506.451	494.193	494.193	494.193	494.193	494.193	494.193
Best weight (kg)	1899.35	1902.605	1902.605	1902.605	1902.605	1902.605	1902.605
FE	3840	7100	4750	3720	3656	5343	2819
Average weight (kg)	1963.12	1909.856	1903.076	1906.735	1905.489	1913.098	1903.407
Worst weight (kg)	2262.8	1912.646	1904.83	1925.714	1929.918	1972.016	1911.268
Standard deviation	106.01	7.09	1.13	6.679	6.232	18.594	1.952



(a) Diversity index recorded in 20 runs (b) Convergence of minimum weight

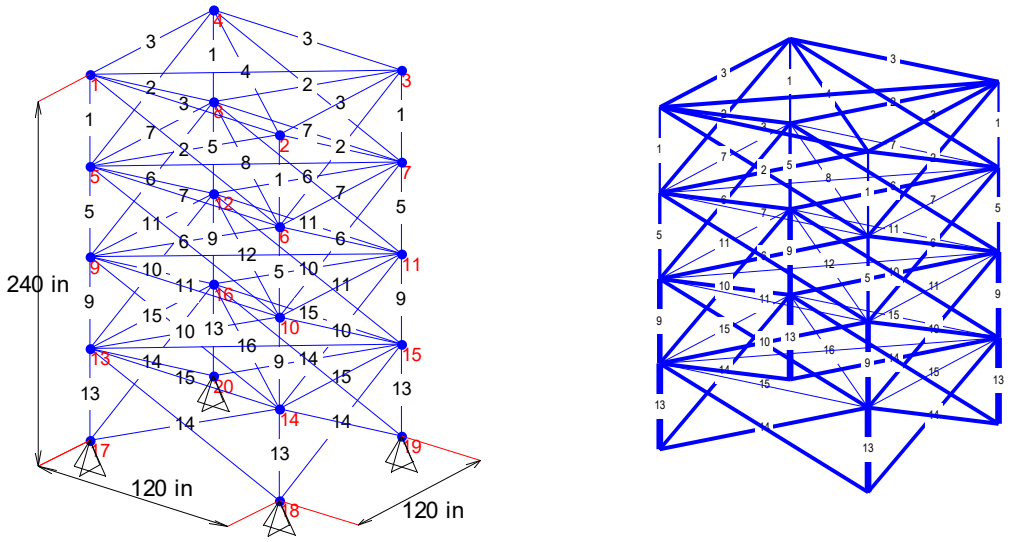
Fig. 5 Optimization history for 52-bar truss by ADDE

other hand, the results of JADE are not as good as ADDE and the others, except CBO. From this example, it is confirmed that ADDE is very competitive to the other considered algorithms.

Fig. 5(b) plots the convergence histories of the minimum weight found by ADDE against the number of function evaluations. It is observed that the convergence curve of the best run is quite similar with the average convergence curve. The change in diversity index is illustrated in Fig. 5(a). In all runs, the diversity index gradually decreases with similar trends to small values at the end of optimization process, which indicates that the ADDE has good exploration/exploitation balance and stability. The best optimized geometry of the structure is illustrated in Fig. 4(b).

5.3 The 72-bar space truss

The third example considered in this study is the 72-bar space truss depicted in Fig. 6(a). The truss elements are categorized in sixteen member groups considering the structural symmetry (Fig. 6(a)). The cross-section areas are chosen from the list of allowable values in Table 4. The structure is subjected to two loading cases, which are given in Table 6. The material properties and constraints are summarized in Table 1.



(a) Layout of the 72-bar space truss (b) The optimized geometry by ADDE

Fig. 6 The 72-bar truss structure

Table 6 Loading conditions for 72-bar truss

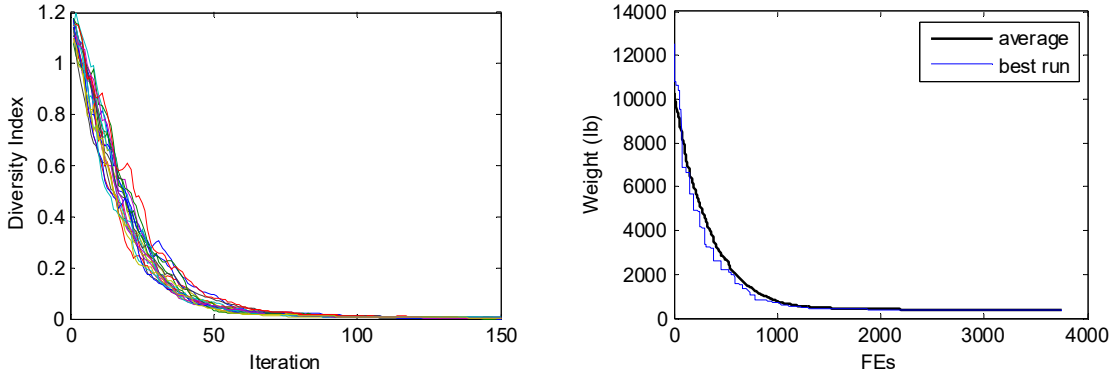
Load case	Node	F_x (kips)	F_y (kips)	F_z (kips)
1	1	5	5	-5
2	1	0	0	-5
	2	0	0	-5
	3	0	0	-5
	4	0	0	-5

This truss sizing optimization is solved by ADDE, SHADE and JADE and compared with literature, including ECBO (Kaveh and Ghazaan 2015) , WCA and IMBA (Sadollah *et al.* 2015), and aeDE (Ho-Huu *et al.* 2016). The results are shown in Table 7. It is noted that the member group order and node sequence in this study are different from that in the other studies. It is seen that the ADDE, SHADE and JADE give the same optimal weight of 389.334 lb as compared to that reported by the other studies. As for average weight and worst weight, IMBA ranks first and ADDE stands second, while ADDE provides smallest standard deviation of 0.632 lb. It is also emphasized that ADDE is more computationally efficient than the other algorithms. ADDE uses 2515 analyses in the best run to obtain the best weight, which is about 60.5%, 40.2%, 54.7% and 14.8% of those required by aeDE, IMBA, WCA and ECBO, respectively. In this example, JADE shows worst performance, whereas the results of SHADE are quite similar to those of aeDE.

Fig. 7(b) plots the history of the minimum weight found by ADDE, which shows very fast convergence rate. It is also seen that the best run curve and the average curve are much closed. It explains for the small standard deviation of weight provided by ADDE in this problem. The diversity indices in 20 runs are plotted in Fig. 7(a). These curves show rapid decrement of the population diversity, which indicates great exploration and exploitation capacities of ADDE in this problem. The optimized geometry of the structure is shown in Fig. 6(b).

Table 7 Comparison on optimal designs of 72-bar truss

Size of grouped members (in^2)	ECBO	WCA	IMBA	aeDE	This study		
					SHADE	JADE	ADDE
1	0.196	0.196	0.196	0.196	0.196	0.196	0.196
2	0.563	0.563	0.563	0.563	0.563	0.563	0.563
3	0.391	0.391	0.391	0.391	0.391	0.391	0.391
4	0.563	0.563	0.563	0.563	0.563	0.563	0.563
5	0.563	0.563	0.563	0.563	0.563	0.563	0.563
6	0.563	0.563	0.563	0.563	0.442	0.563	0.563
7	0.111	0.111	0.111	0.111	0.111	0.111	0.111
8	0.111	0.111	0.111	0.111	0.111	0.111	0.111
9	1.228	1.228	1.228	1.228	1.228	1.228	1.228
10	0.442	0.563	0.563	0.442	0.563	0.563	0.563
11	0.111	0.111	0.111	0.111	0.111	0.111	0.111
12	0.111	0.111	0.111	0.111	0.111	0.111	0.111
13	1.990	1.990	1.990	1.990	1.990	1.990	1.990
14	0.563	0.442	0.442	0.563	0.563	0.442	0.442
15	0.111	0.111	0.111	0.111	0.111	0.111	0.111
16	0.111	0.111	0.111	0.111	0.111	0.111	0.111
Best weight (lb)	389.33	389.334	389.334	389.334	389.334	389.334	389.334
FE	17010	4600	6250	4160	3684	5579	2515
Average weight (lb)	391.59	389.941	389.823	390.913	390.956	392.098	389.891
Worst weight (lb)	–	393.778	389.457	393.325	394.323	411.703	391.326
Standard deviation	–	1.43	0.84	1.161	1.381	4.836	0.632



(a) Diversity index recorded in 20 runs (b) Convergence of minimum weight

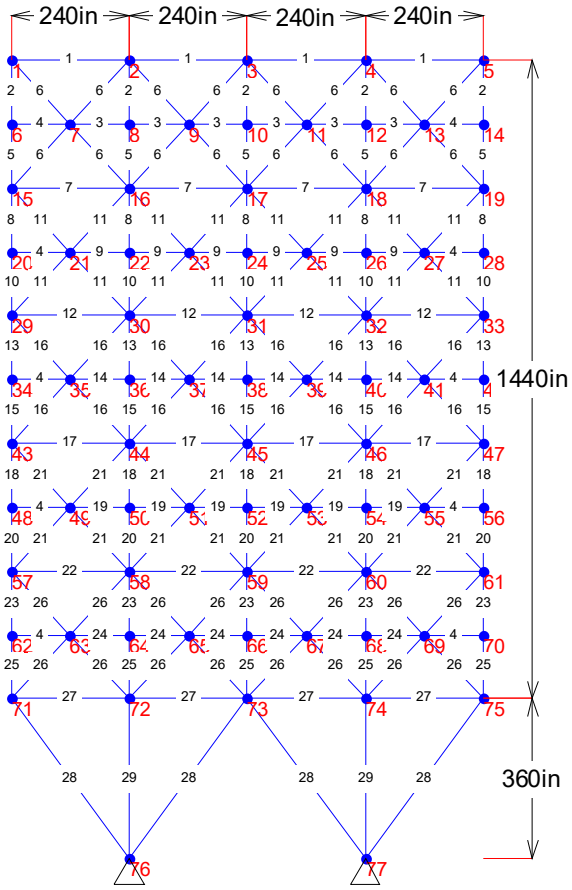
Fig. 7 Optimization history for 72-bar truss by ADDE

5.4 The 200-bar planar truss

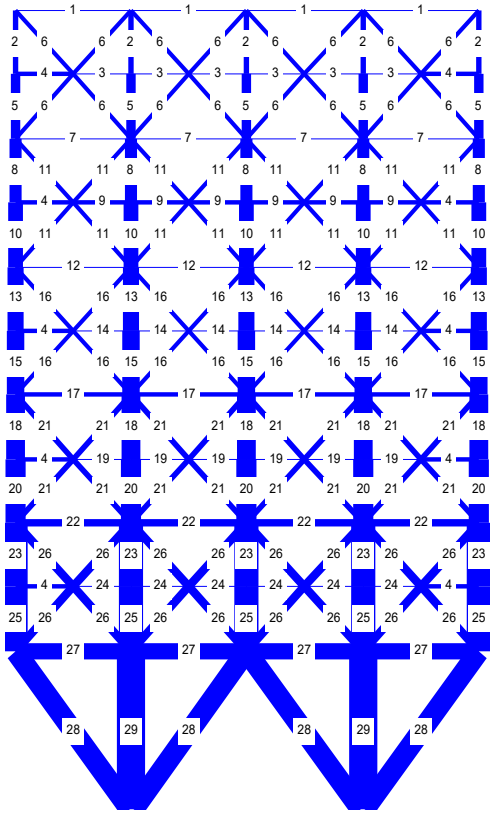
The last example investigated in this study is the benchmark 200-bar planar truss structure shown in Fig. 8(a). This structure is considered as a large-scale, size optimization problem in some recent studies (Azad and Hasançebi 2014; Hasançebi and Azad 2015; Ho-Huu *et al.* 2016). The design variables include all bar element cross-sectional areas which are categorized into 29 groups by considering geometrical symmetry as marked in Fig. 8(a). The values of cross-section area (in^2) are selected from the list: 0.1, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18, 23.68, 28.08, 33.7 (in^2). The structure is subjected to three loading conditions: (1) 1.0 kip acting in the positive *x*-direction at nodes 1, 6, 15 20, 29, 34, 43, 48, 57, 62 and 71; (2) 10.0 kips acting in the negative y-direction at nodes 1-6, 8, 10, 12, 14, 16-20, 22, 24, 26,28-34, 36, 38, 40, 42-48, 50, 52, 54, 56-62, 64, 66, 68, and 70-75; and (3) conditions 1 and 2 acting together. The material properties and constraints are given in Table 1.

The optimization results obtained by this work are listed in Table 8 in comparison with those from the other studies. The best weight obtained by ADDE is 26960.152 lb, which is lighter than that given by ESASS (28075.488 lb), ADS (27190.49 lb), aeDE (27858.500 lb), SHADE (27831.174 lb) and JADE (27991.297). Moreover, only the best solutions found by ADDE, SHADE and JADE satisfy the stress constraint, while the best solutions of the other methods violate the constraint at some degree. According to the results given, the maximum stresses in truss elements are recalculated and they are 10.088 ksi for ESASS, 10.0347 ksi for ADS, 10.0719 ksi for aeDE, and 10 ksi for ADDE. The ADDE gives the best results regarding the average weight, worst weight and standard deviation, while JADE provides the worst results. The computational effort of ADDE is about 50% and 44.5% lower than that of ESASS and aeDE, respectively, although it is higher than that of ADS. Fig. 8(b) shows the relative virtual effect of element cross-sectional areas of the optimized geometry obtained using ADDE.

The diversity indices and convergence histories of minimum weight provided by ADDE are illustrated in Fig. 9. Similar trends of the diversity indices in 20 runs are observed here. The trends show gradual decrement of diversity index, which indicates that the algorithm performs well global exploration and local exploitation. Some amount of diversity maintained at the end of the optimization implies that the search is still in progress.



(a) Layout of the 200-bar truss



(b) The optimized geometry by ADDE

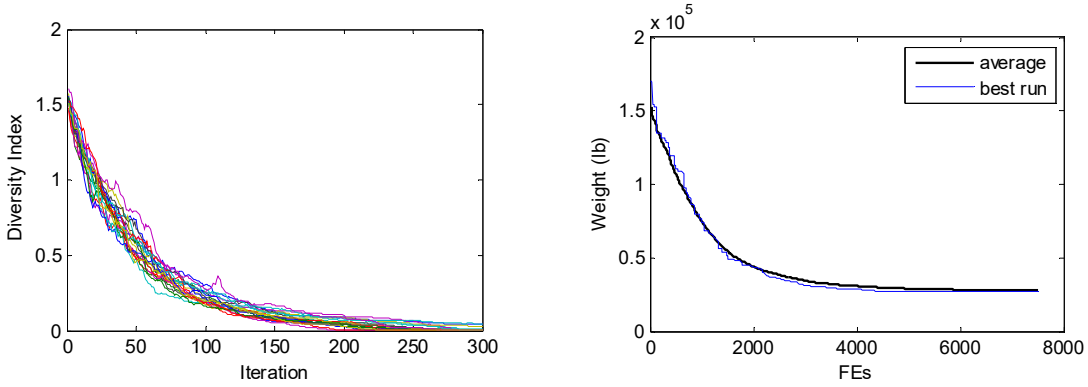
Fig. 8 The 200-bar truss structure

Table 8 Comparison on optimal designs of 200-bar truss

Size of grouped Members (in^2)	ESASS	ADS	aeDE	This study		
				SHADE	JADE	ADDE
1	0.1	0.1	0.1	0.100	0.100	0.100
2	0.954	0.954	0.954	0.954	0.954	0.954
3	0.1	0.347	0.347	0.100	0.100	0.100
4	0.1	0.1	0.1	0.347	0.100	0.100
5	2.142	2.142	2.142	2.142	2.142	2.142
6	0.347	0.347	0.347	0.347	0.347	0.347
7	0.1	0.1	0.1	0.100	0.100	0.347
8	3.131	3.131	3.131	3.813	3.131	3.131
9	0.1	0.1	0.347	0.100	0.347	0.100
10	4.805	4.805	4.805	4.805	4.805	4.805

Table 8 Continued

Size of grouped Members (in^2)	ESASS	ADS	aeDE	This study		
				SHADE	JADE	ADDE
11	0.347	0.44	0.539	0.347	0.347	0.539
12	0.1	0.1	0.347	0.100	0.100	0.100
13	5.952	5.952	5.952	5.952	5.952	5.952
14	0.1	0.1	0.1	0.100	0.539	0.100
15	6.572	6.572	6.572	6.572	6.572	6.572
16	0.44	0.539	0.954	0.440	0.954	0.440
17	0.539	0.1	0.44	0.539	0.100	0.539
18	7.192	8.525	8.525	8.525	8.525	8.525
19	0.44	0.539	0.1	0.539	0.347	0.347
20	8.525	9.3	9.3	8.525	9.300	9.300
21	0.954	0.954	0.954	0.954	0.954	0.954
22	1.174	0.1	1.081	0.539	2.800	0.100
23	10.85	10.85	13.33	10.850	13.330	13.330
24	0.44	0.954	0.539	0.100	0.100	0.100
25	10.85	13.33	14.29	13.330	14.290	13.330
26	1.764	1.333	2.142	0.954	2.142	0.954
27	8.525	7.192	3.813	6.572	3.813	5.952
28	13.33	10.85	8.525	13.330	8.525	10.850
29	13.33	14.29	17.17	14.290	17.170	14.290
Best weight (lb)	28075.488	27190.49	27858.500	27831.174	27991.297	26960.152
Const. violation	Yes	Yes	Yes	No	No	No
FE	11156	5000	12325	4546	8464	6189
Average weight (lb)	-	28146.1	28425.871	28760.941	29727.876	27969.510
Worst weight (lb)	-	29667.76	29415.000	30050.680	32233.551	28901.109
Standard deviation	-	786.6	481.590	702.478	1323.370	422.130



(a) Diversity index recorded in 20 runs (b) Convergence of minimum weight

Fig. 9 Optimization history for 200-bar truss

Table 9 Average function evaluations used by ADDE for 10-bar, 37-bar, 72-bar and 200-bar trusses

Problem	T_{max}	NP	Fitness evaluations	Constraint evaluations (Structural analyses)	% skipped
10-bar truss	100	25	2500	1557	37.72
52-bar truss	150	25	3750	2966	20.91
72-bar truss	150	25	3750	2577	31.28
200-bar truss	300	25	7500	5512	26.51

In summary, the experiment results and comparison demonstrate that the proposed ADDE algorithm is competitive to some state-of-the-art metaheuristic algorithms for truss optimization with discrete variables, especially in terms of computational efficiency. It also showed in general better performance than that of three advanced adaptive DE variants.

It is worth to note that in ADDE the number of constraint evaluations is different from and lower than the number of fitness evaluations. It is due to the fact that in the algorithm, the constraints and the objective function are treated separately and one can avoid constraint evaluations, which involve structural analysis, when a trial solution can be judged by it fitness (objective function value). In other words, if the compared solution (the target solution) is feasible and the fitness of the trial solution is not better than that of it, the trial solution will be skipped without constraint evaluation. In this way, more structural analyses can be saved. The actual number of structural analyses and the percentage of skipped analyses on average are listed in Table 9. However, for the sake of comparison purpose, only the number of fitness evaluations is mentioned in the previous discussion.

6. Conclusions

In this article, the adaptive directional differential evolution, ADDE, is presented for solving truss sizing optimization with discrete variables. It is demonstrated that the proposed adaptation approach based on population diversity is capable to well balance between global exploration and local exploitation as well as locate the promising solutions. More specifically, in the sooner generations exploring the domain containing the global optimum is ensured, and in the later generations exploitation is strengthened for accelerating the convergence speed. Computational efficiency and simplicity are amongst the remarkable features of the proposed algorithm. Numerical results show that ADDE in most cases provides optimal solutions as good as or better than the similar results from some state-of-the-art metaheuristics, including one recently developed improved DE variant. The benefit of ADDE is that it often uses fewer structural analyses than those required by the other methods. This indicates that ADDE is a promising optimizer for complex structural optimization problems like truss structures with discrete variables.

Acknowledgments

This work was supported by National University of Civil Engineering, Vietnam (NUCE) under grant research number 98-2015/KHXD. This support is gratefully acknowledged by the author.

References

Azad, S.K. and Hasançebi, O. (2014), "An elitist self-adaptive step-size search for structural design optimization", *Appl. Soft Comput.*, **19**, 226-235.

Azad, S.K. and Hasançebi, O. (2015), "Discrete sizing optimization of steel trusses under multiple displacement constraints and load cases using guided stochastic search technique", *Struct. Multidisciplin. Optimiz.*, **52**(2), 383-404.

Azad, S.K., Hasançebi, O. and Saka, M.P. (2014), "Guided stochastic search technique for discrete sizing optimization of steel trusses: A design-driven heuristic approach", *Comput. Struct.*, **134**, 62-74.

Azad, S.K., Hasançebi, O., Azad, S.K. and Erol, O.K. (2013), "Upper bound strategy in optimum design of truss structures: A big bang-big crunch algorithm based application", *Adv. Struct. Eng.*, **16**(6), 1035-1046.

Bennage, W.A. and Dhingra, A.K. (1995a), "Optimization of truss topology using tabu search", *Int. J. Numer. Meth. Eng.*, **38**(23), 4035-4052.

Bennage, W.A. and Dhingra, A.K. (1995b), "Single and multiobjective structural optimization in discrete-continuous variables using simulated annealing", *Int. J. Numer. Meth. Eng.*, **38**(16), 2753-2773.

Bland, J.A. (2001), "Optimal structural design by ant colony optimization", *Eng. Optimiz.*, **33**(4), 425-443.

Bureerat, S. and Pholdee, N. (2015), "Optimal truss sizing using an adaptive differential evolution algorithm", *J. Comput. Civ. Eng.*, **30**(2), 04015019.

Camp, C.V. (2007), "Design of space trusses using Big Bang-Big Crunch optimization", *J. Struct. Eng.*, **133**(7), 999-1008.

Camp, C.V. and Bichon, B.J. (2004), "Design of space trusses using ant colony optimization", *J. Struct. Eng.*, **130**(5), 741-751.

Camp, C.V. and Farshchin, M. (2014), "Design of space trusses using modified teaching-learning based optimization", *Eng. Struct.*, **62**, 87-97.

Das, S., Abraham, A., Chakraborty, U.K. and Konar, A. (2009), "Differential evolution using a neighborhood-based mutation operator", *Evolutionary Computation, IEEE Transactions on*, **13**(3), 526-553.

Das, S. and Suganthan, P.N. (2011), "Differential evolution: a survey of the state-of-the-art", *Evolutionary Computation, IEEE Transactions on*, **15**(1), 4-31.

Deb, K. (2000), "An efficient constraint handling method for genetic algorithms", *Comput. Meth. Appl. Mech. Eng.*, **186**(2), 311-338.

Elsayed, S.M., Sarker, R.A. and Essam, D.L. (2011), "Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems", *Evolutionary Computation (CEC), 2011 IEEE Congress on*, IEEE.

Gong, W., Cai, Z., Ling, C.X. and Li, H. (2011), "Enhanced differential evolution with adaptive strategies for numerical optimization", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **41**(2), 397-413.

Hajela, P. and Lin, C.Y. (1992), "Genetic search strategies in multicriterion optimal design", *Struct. Optimiz.*, **4**(2), 99-107.

Hasançebi, O. and Azad, S.K. (2014), "Discrete size optimization of steel trusses using a refined big bang-big crunch algorithm", *Eng. Optimiz.*, **46**(1), 61-83.

Hasançebi, O. and Azad, S.K. (2015), "Adaptive dimensional search: a new metaheuristic algorithm for discrete truss sizing optimization", *Comput. Struct.*, **154**, 1-16.

Ho-Huu, V., Nguyen-Thoi, T., Vo-Duy, T. and Nguyen-Trang, T. (2016), "An adaptive elitist differential evolution for optimization of truss structures with discrete design variables", *Comput. Struct.*, **165**, 59-75.

Kaveh, A. and Ahmadi, B. (2014), "Sizing, geometry and topology optimization of trusses using force method and supervised charged system search", *Struct. Eng. Mech.*, **50**(3), 365-382.

Kaveh, A. and Ghazaan, M.I. (2014), "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", *Adv. Eng. Soft.*, **77**, 66-75.

Kaveh, A. and Ghazaan, M.I. (2015), "A comparative study of CBO and ECBO for optimal design of

skeletal structures", *Comput. Struct.*, **153**, 137-147.
Kaveh, A. and Mahdavi, V.R. (2014), "Colliding bodies optimization method for optimum discrete design of truss structures", *Comput. Struct.*, **139**, 43-53.
Kaveh, A. and Zolghadr, A. (2012), "Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability", *Comput. Struct.*, **102**, 14-27.
Krempser, E., Bernardino, H., Barbosa, H. and Lemonge, A. (2012), "Differential evolution assisted by surrogate models for structural optimization problems", *Proceedings of the international conference on computational structures technology (CST). Civil-Comp Press*.
Kushida, J.I., Hara, A. and Takahama, T. (2015), "Rank-based differential evolution with multiple mutation strategies for large scale global optimization", *Evolutionary Computation (CEC), 2015 IEEE Congress on*, IEEE.
Lampinen, J. and Zelinka, I. (1999), "Mixed integer-discrete-continuous optimization by differential evolution", *Proceedings of the 5th International Conference on Soft Computing*.
Lee, K.S., Geem, Z.W., Lee, S.H. and Bae, K.W. (2005), "The harmony search heuristic algorithm for discrete structural optimization", *Eng. Optimiz.*, **37**(7), 663-684.
Li, L.J., Huang, Z.B. and Liu, F. (2009), "A heuristic particle swarm optimization method for truss structures with discrete variables", *Comput. Struct.*, **87**(7), 435-443.
Mallipeddi, R., Suganthan, P.N., Pan, Q.K. and Tasgetiren, M.F. (2011), "Differential evolution algorithm with ensemble of parameters and mutation strategies", *Appl. Soft Comput.*, **11**(2), 1679-1696.
Pholdee, N., Bureerat, S., Park, W.W., Kim, D.K., Im, Y.T., Kwon, H.C. and Chun, M.S. (2015), "Optimization of flatness of strip during coiling process based on evolutionary algorithms", *Int. J. Precision Eng. Manufact.*, **16**(7), 1493-1499.
Qin, A.K. and Suganthan, P.N. (2005), "Self-adaptive differential evolution algorithm for numerical optimization", *2005 IEEE congress on evolutionary computation*, IEEE.
Rahnamayan, S., Tizhoosh, H.R. and Salama, M. (2008), "Opposition-based differential evolution", *Evolutionary Computation, IEEE Transactions on*, **12**(1), 64-79.
Rajeev, S. and Krishnamoorthy, C.S. (1992), "Discrete optimization of structures using genetic algorithms", *J. Struct. Eng.*, **118**(5), 1233-1250.
Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2012), "Mine blast algorithm for optimization of truss structures with discrete variables", *Comput. Struct.*, **102**, 49-63.
Sadollah, A., Eskandar, H., Bahreininejad, A. and Kim, J.H. (2015), "Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures", *Comput. Struct.*, **149**, 1-16.
Stolpe, M. (2015), "Truss optimization with discrete design variables: a critical review", *Struct. Multidisciplin. Optimiz.*, **53**(2), 349-374.
Storn, R. and Price, K. (1997), "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces", *J. Glob. Optimiz.*, **11**(4), 341-359.
Takahama, T. and Sakai, S. (2012), "Differential evolution with dynamic strategy and parameter selection by detecting landscape modality", *Evolutionary Computation (CEC), 2012 IEEE Congress on*, IEEE.
Tanabe, R. and Fukunaga, A. (2013), "Success-history based parameter adaptation for differential evolution", *2013 IEEE Congress on Evolutionary Computation*, IEEE.
Wang, Y., Cai, Z. and Zhang, Q. (2011), "Differential evolution with composite trial vector generation strategies and control parameters", *Evolutionary Computation, IEEE Transactions on*, **15**(1), 55-66.
Wang, Z., Tang, H. and Li, P. (2009), "Optimum design of truss structures based on differential evolution strategy", *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, IEEE.
Wu, G., Mallipeddi, R., Suganthan, P.N., Wang, R. and Chen, H. (2016), "Differential evolution with multi-population based ensemble of mutation strategies", *Inform. Sci.*, **329**, 329-345.
Xiang, W.L., Meng, X.L., An, M.Q., Li, Y.Z. and Gao, M.X. (2015), "An Enhanced Differential Evolution Algorithm Based on Multiple Mutation Strategies", *Computational intelligence and neuroscience*, *2015*.
Yang, Y. and Yao, M. (2014), "Differential evolution with M-fitness method", *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, IEEE.

Zamuda, A., Brest, J. and Mezura-Montes, E. (2013), "Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization", *Evolutionary Computation (CEC), 2013 IEEE Congress on*, IEEE.

Zhang, J. and Sanderson, A.C. (2009), "JADE: adaptive differential evolution with optional external archive", *Evolutionary Computation, IEEE Transactions on*, **13**(5), 945-958.

CC