36 | https://doi.org/10.3311/PPci.21147
Creative Commons Attribution

Periodica Polytechnica Civil Engineering, 67(1), pp. 36–56, 2023

Optimal Design of Large-scale Dome Truss Structures with Multiple Frequency Constraints Using Success-history Based Adaptive Differential Evolution Algorithm

Ali Kaveh[1*], Kiarash Biabani Hamedani[1], Bamdad Biabani Hamedani[2]

[1] School of Civil Engineering, Iran University of Science and Technology (IUST), Narmak, Tehran, PO Box 16846-13114, Iran
[2] Department of Industrial Engineering, Faculty of Engineering, Islamic Azad University, Karaj, PO Box 3149968111, Iran
* Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 09 September 2022, Accepted: 23 September 2022, Published online: 28 September 2022

Abstract

The success-history based adaptive differential evolution (SHADE) algorithm is an efficient modified version of the differential evolution (DE) algorithm, and it has been successfully applied to solve some real-world optimization problems. However, to the best of our knowledge, it has been rarely applied in the field of structural optimization. The optimal design of structures with frequency constraints is well known as a highly nonlinear and non-convex optimization problem with many local optima. In this paper, the SHADE algorithm is examined in the context of size optimization of large-scale truss structures with multiple frequency constraints. In SHADE, a historical memory of successful control parameter settings is used to guide the generation of new control parameters. In order to demonstrate the effectiveness and efficiency of SHADE, three truss optimization problems with multiple frequency constraints are presented. The three examples considered in this paper include a 600-bar single-layer dome-shaped truss, a 1180-bar single-layer dome-shaped truss, and a 1410-bar double-layer dome-shaped truss. The results obtained by the SHADE algorithm are presented and compared with the best-known results reported in the literature. Numerical results indicate the effectiveness and superior performance of SHADE over other algorithms in terms of solution accuracy and robustness. It is worth mentioning that in all the three cases considered, the optimal designs obtained by SHADE are the best ones reported in the literature so far. However, SHADE often requires fewer structural analyses than those required by the other algorithms.

Keywords

structural optimization, frequency constraints, truss structures, metaheuristic algorithms, success-history based adaptive differential evolution

1 Introduction

The natural frequencies of a structure are the most essential characteristics to determine the dynamic behavior of the structure [1]. In design practice, situations are frequently encountered where the design requirements of a structure or structural component include lower limits on one or more of the natural frequencies [2]. For example, in the structural design of satellites, it is necessary to impose certain frequency requirements on the design process to avoid dynamic coupling between two satellite subsystem frequencies or between low-frequency modes of the launcher and the satellite structure [3]. In such situations, the optimal design of structures with frequency constraints is of great practical importance because it provides an effective way to control and manipulate the dynamic characteristics of a structure in a variety of ways, and thus improve its dynamic performance [4].

Over the last few decades, and many researchers have given considerable attention to develop efficient optimization algorithms to solve structural optimization problems with frequency constraints. Pioneering works in this field were carried out using gradient-based optimization techniques [5–7]. However, it has been recognized for many years that frequency constraints are highly nonlinear, non-convex, and implicit with respect to the design variables, and this makes dynamic sensitivity analysis difficult or even impossible [8, 9]. Thus, in structural optimization problems with multiple frequency constraints, it is very difficult to obtain an initial feasible design, which satisfies the multiple frequency constraints simultaneously [9]. It is well known however that the performance of gradient-based optimization techniques heavily depends on the selection of initial search point [10]. Moreover, these techniques require

Cite this article as: **Cite this article as:** Kaveh, A., Hamedani, K. B., Hamedani, B. B. "Optimal Design of Large-scale Dome Truss Structures with Multiple Frequency Constraints Using Success-history Based Adaptive Differential Evolution Algorithm", Periodica Polytechnica Civil Engineering, 67(1), pp. 36–56, 2023. https://doi.org/10.3311/PPci.21147

complex and time-consuming dynamic sensitivity analysis and may easily get trapped in local optima [9]. For these reasons, gradient-based optimization techniques may face serious difficulties in solving this class of structural optimization problems. To overcome these difficulties, metaheuristic algorithm could be served as appropriate alternatives. In recent decades, thanks to substantial advances in the computational capability of computers, numerous metaheuristic algorithms have been introduced to solve a wide range of optimization problems. The distinctive feature of these algorithms is that they do not need to gradient information, which makes them easy to implement. Over the past three decades, various metaheuristic algorithms have been used successfully to solve structural optimization problems with frequency constraints. In this regard, Wei et al. [9] proposed the niche hybrid genetic algorithm (NHGA) for size and shape optimization of truss structures with frequency constraints. Subsequently, Wei et al. [11] improved the computational cost of the NHGA by using the advantages of parallel computing and introduced the niche hybrid parallel genetic algorithm (NHPGA) to solve truss shape and size optimization problems with frequency constraints. Kaveh and Zolghadr [12] proposed a hybridization of the charged system search (CSS) and the big bang-big crunch (BB-BC) algorithms and applied it to the optimization of truss structures with multiple frequency constraints. Kaveh and Ilchi Ghazaan [13] utilized the enhanced colliding bodies optimization (ECBO) algorithm and its cascade version (ECBO-Cascade) for size optimization of large-scale dome structures with multiple natural frequency constraints. Kaveh [14] adopted democratic particle swarm optimization (DPSO) for the optimization of large-scale dome trusses under multiple frequency constraints. Kaveh and Ilchi Ghazaan [15] introduced a hybrid metaheuristic algorithm, called MDVC-UVPS, by combining vibrating particles system (VPS) algorithm, multi-design variable configuration (Multi-DVC) cascade optimization, and upper bound strategy (UBS), and applied it to large-scale dome optimization problems with multiple frequency constraints. Many other similar studies exist in the literature that utilized various metaheuristic algorithms to solve truss optimization problems with frequency constraints [16–23].

Differential evolution (DE) is a simple yet powerful population-based evolutionary algorithm proposed by Storn and Price [24]. Since its development in 1997, DE has been proven to be very competitive particularly in solving complex numerical optimization problems [25], and has been successfully applied to a wide range of optimization problems arising in different fields of applied science and engineering [26]. However, it has been well recognized that the search performance of DE depends strongly on the setting of its control parameters, including the population size N, the scaling factor F, and the crossover rate CR [27, 28]. Furthermore, it is well known that the optimal setting of these parameters is usually a problem-specific task, and thus is usually determined by trial and error or by sensitivity analysis [29]. Motivated by this, numerous efforts have been focused on the development of adaptive or self-adaptive mechanisms to dynamically adjust the control parameters throughout the evolutionary process [28, 30–33]. Among them, adaptive differential evolution with optional external archive (JADE) [31] is a well-known and effective variant of DE, in which a control parameter adaptation mechanism is employed to automatically adjust the control parameters F and CR to appropriate values. JADE also adopts a greedy mutation strategy called "DE/current-to-$p$best/1" with optional external archive, which exploits the information from both the best solution and other good solutions. Success-history based adaptive differential evolution (SHADE) [28] is an improved version of JADE, and it uses a success history-based parameter adaptation scheme. In SHADE, unlike JADE which generates new control parameter values based on the mean values for F and CR that have been successful in previous generations, a historical record of successful control parameter settings is employed to guide the generation of future control parameter values [34]. The main distinctive feature of the SHADE algorithm is that it has the ability to automatically adjust the control parameters F and CR. SHADE was ranked the fourth best algorithm among the 21 algorithms participated in the 2013 IEEE congress on evolutionary computation (IEEE CEC 2013) [35], and has been successfully applied to some real-world optimization problems, including optimal power follow (OPF) [36], protein structure prediction (PSP) [37], etc. However, it has been rarely applied in the field of structural optimization [23, 38]. From the results in [23], it seems that better results could have been achieved by SHADE if an appropriate parameter setting had been considered for the problem under study.

The main purpose of this study is to investigate the efficiency and effectiveness of the SHADE algorithm in dealing with large-scale truss optimization problems with multiple frequency constraints. To evaluate the performance of SHADE, three benchmark examples are presented and discussed. These examples include a 600-bar single-layer dome truss with 25 design variables, a 1180-bar single-layer

dome truss with 59 design variables, and a 1410-bar double-layer dome truss with 47 design variables. The results obtained by SHADE are compared with the best-known solutions from the literature. It should be noted that since SHADE is an improved variant of the traditional differential evolution (DE), in order to allow a fair comparison of the results with previously published references, improved and hybrid metaheuristic algorithms are often considered in the comparison. The experimental results revealed that SHADE provides superior results compared to the results reported in previous studies.

The rest of this paper is organized as follows: After the Introduction section, Section 2 presents the formulation of truss optimization problems with frequency constraints. Section 3 describes the SHADE algorithm. In Section 4, three benchmark truss sizing optimization problems with frequency constraints are investigated. Finally, Section 5 provides the concluding remarks.

2 Formulation of truss optimization with frequency constraints

In a truss sizing optimization problem with multiple frequency constraints, the objective of optimization is generally to minimize the total weight of the structure while satisfying some constraints on the natural frequencies of the structure. Both the shape and topology of the structure are pre-specified and remain unchanged during the optimization process. Hence, the cross-sectional areas of the members are considered as the only design variables of the problem. Each design variable is confined within its permissible range. The mathematical formulation of this problem can be expresses as follows [23]:

$$\text{Find}: X = \left[x_1, x_2, \ldots, x_D\right], \tag{1}$$

$$\text{to minimize}: W\left(X\right) = \sum_{n=1}^{m} \rho_n A_n L_n, \tag{2}$$

$$\text{subject to}: \begin{cases} \omega_j \geq \omega_j^* & \text{for some natural frequencies } j \\ \omega_k \leq \omega_k^* & \text{for some natural frequencies } k, \\ x_k^l \leq x_k \leq x_k^u, & k = 1,2,\ldots,D \end{cases} \tag{3}$$

where $\boldsymbol{X}$ is the vector of design variables representing member cross-sectional areas; D is the number of design variables, i.e., the number of member groups; x_k is the n-th design variable, i.e., the cross-sectional area of the k-th member group; $W(X)$ denotes the objective function which is the total weight of the truss; m is the number of truss members; ρ_n, A_n, and L_n are the material density, cross sectional area, and length of the n-th truss member, respectively; ω_j is the j-th natural frequency of the structure and ω_j^* is its corresponding lower bound; ω_k is the k-th natural frequency of the structure and ω_k^* is its corresponding upper bound; and x_k^l and x_k^u are the lower and upper bounds of the k-th design variable, respectively.

Since the optimization problem formulated above is a constrained one, it is necessary to transform it to an unconstrained optimization problem. The popular penalty function method is adopted in this research for this purpose. In the penalty function method, a penalty term is incorporated in the objective function for any violation of the constraints [39]. In this way, the original constrained optimization problem is transformed into an unconstrained one. The penalty function used here is a dynamic multiplicative penalty function called the Kaveh-Zolghadr technique, which is formulated as follows [40]:

$$f_{penalty}\left(X\right) = \left(1 + \varepsilon_1 . v\right)^{\varepsilon_2}, \; v = \sum_{i=1}^{c} v_i, \tag{4}$$

where $f_{penalty}(X)$ is the penalty function; v denotes the sum of violations of the constraints; c is the number of constraints; and v_i is the relative violation of the i-th constraint. If the i-th constraint is satisfied, then the value of v_i will set to zero, otherwise it is calculated based on the severity of violation. This can be expressed mathematically by the following equation:

$$v_i = \begin{cases} \left|1 - \dfrac{\omega_i}{\omega_i^*}\right| & \textit{if the } i-\textit{th frequency constraint is violated} \\ 0 & \textit{otherwise} \end{cases}. \tag{5}$$

In Eq. (4), ε_1 and ε_2 are the parameters of the penalty function, and they need to be tuned to adjust the severity of the penalty being applied to infeasible solutions. The appropriate setting of these parameters is a challenging task and requires many preliminary trials, because they may affect the exploration and exploitation tendencies of the optimizer [41]. Indeed, if the parameters ε_1 and ε_2 are too small, too much effort is devoted to searching infeasible regions of the search space while feasible regions are not explored effectively and the algorithm may even fail to converge to a feasible solution. On the other hand, if these parameters are too large, infeasible regions of the search space may not be explored enough and thus premature convergence may occur [42]. In this research, the parameter ε_1 is set as a constant value, whereas the parameter ε_2 is set to increase linearly with the number of iterations. It follows

that at different stages of the search process, different penalty values are imposed on the same constraint violation values. Indeed, at the beginning of the optimization process, low penalty values are assigned to infeasible solutions. As a result, even highly infeasible solutions are allowed to enter the population of early generations, which means that the search agents are allowed to move freely through the entire search space, which means that more exploration is provided. However, as the number of iterations increases, so does the penalty values. As a result, during the final stages of the search process, feasible solutions are preferred over infeasible ones, and infeasible solutions with smaller violations are preferred over infeasible solutions with larger violations. In this way, the search process is directed towards feasible regions of the search space, meaning that more exploitation is provided [43].

Using the penalty function method, the above constrained optimization problem can be rewritten as the following unconstrained optimization problem:

$$\text{Find}: X = \left[x_1, x_2, \ldots, x_D\right], \tag{6}$$

$$\text{to minimize}: P(X) = W(X) \times f_{penalty}(X), \tag{7}$$

$$\text{subject to}: x_k^l \le x_k \le x_k^u, \; k = 1,2,\ldots,D, \tag{8}$$

where $P(X)$ is the penalized objective function.

To determine the natural frequencies and associated mode shapes of an undamped structure, the following algebraic equation known as the matrix eigenvalue problem must be solved [44]:

$$k\phi_n = \omega_n^2 m\phi_n, \; n = 1,2,\ldots,N, \tag{9}$$

where k and m denote the stiffness and mass matrices of the structure, respectively; ω_m is the n-th natural frequency of the structure; ϕ_n is the n-th mode shape of the structure; and N is the number of degrees of freedom of the structure.

3 Success-history based adaptive differential evolution (SHADE) algorithm

As mentioned earlier, SHADE is an improved version of JADE [31], and it employs a historical memory of successful control parameter settings to guide the generation of the next control parameter values [28]. In this research, we use SHADE 1.1[1], the latest version of the SHADE algorithm [33]. We first describe the steps involved in SHADE in Sections 3.1 to 3.6. Then, in Section 3.7., the SHADE algorithm is characterized in terms of exploration and exploitation of the search space. In addition, the assumptions and limitations of the applied SHADE algorithm are also discussed.

3.1 Initialization

Similar to many other population-based metaheuristic algorithms, SHADE starts with a population of randomly generated individuals. Then, by following a loop of evolutionary operations such as generation and selection, the population is updated. This process of is repeated until the termination criteria are satisfied. The initial population is represented as follows:

$$x_{j,i}^{(0)} = Lb_j + rand[0,1] * \left(Ub_j - Lb_j\right), \quad i = 1,2,\ldots,N \text{ and } j = 1,2,\ldots,D, \tag{10}$$

where superscript (0) represents initialization; $x_{j,i}^{(0)}$ denotes the j-th design variable of the i-th individual of the initial population; N is the population size; D is the number of design variables; Lb_j and Ub_j are the lower and upper bounds of the j-th design variable, respectively; and $rand[0,1]$ is a uniformly distributed random number between zero and one.

3.2 History-based parameter adaptation

As mentioned before, SHADE employs a history-based parameter adaption scheme to guide the selection of future control parameter values [28]. As shown in Table 1, SHADE maintains a historical memory containing H entries for both the DE control parameters CR and F. The parameter $F \in [0,1]$ is a scaling factor to control the magnitude of the differential mutation operator and $CR \in [0,1]$ is the crossover rate [33]. At the beginning of the search process, all M_{CR} and M_F values are initialized to 0.5, i.e., $M_{CR,i} = M_{F,i} = 0.5 (i = 1, 2, \ldots, H)$. At each generation of the SHADE algorithm, for each individual x_i, the control parameters CR_i and F_i are calculated by the following equations:

$$CR_i = \begin{cases} 0 & if \; M_{CR,r_i} = \perp \\ randn_i\left(M_{CR,r_i}, 0.1\right) & otherwise \end{cases}, \tag{11}$$

$$F_i = randc_i\left(M_{F,r_i}, 0.1\right), \tag{12}$$

Table 1 The historical memory used by the SHADE algorithm [33]

Index	1	2	…	H–1	H
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	…	$M_{CR,H-1}$	$M_{CR,H}$
M_F	$M_{F,1}$	$M_{F,2}$	…	$M_{F,H-1}$	$M_{F,H}$

[1] The source code of the SHADE algorithm is available at https://ryojitanabe.github.io/code/SHADE1.1.1_CEC2014_Matlab_Octave.zip

where *randn* and *randc* represent the Gaussian and Cauchy distributions, respectively; and r_i is an integer randomly selected from 1 to H. In case a value for CR_i outside the interval [0, 1] is generated, it is replaced by the limit value to which it is closer. If the value of F_i is found to be greater than 1, then it is truncated to 1, and if $F_i < 0$, Eq. (12) is executed repeatedly to generate a valid value [31]. In Eq. (11), when the M_{CR,r_i} value becomes equal to the termination value $\perp$, CR_i is set to zero.

3.3 The DE/current-to-*p*best/1 mutation strategy

SHADE uses a mutation strategy called "DE/current-to-*p*best/1", which was first proposed by Zhang and Sanderson [31] for JADE. In DE/current-to-*p*best/1, the mutation vector is generated by the following equation:

$$v_{i,G} = x_{i,G} + F_i.\left(x_{pbest,G} - x_{i,G}\right) + F_i.\left(x_{r_1,G} - x_{r_2,G}\right). \quad (13)$$

In the above equation, G is the index of the current generation; $x_{i,G}$ is the i-th individual of the generation G; F_i is the F parameter used by individual x_i; and $v_{i,G}$ is the mutant vector corresponding to $x_{i,G}$. The individual $x_{pbest,G}$ is randomly chosen from the top $p \times N (p \in [0,1])$ individuals in the G-th generation. The individuals $x_{r_1,G}$ and $x_{r_2,G}$ are randomly chosen from the individuals in the G-th generation such that the indices r_1 and r_2 differ from each other as well as i. The greediness of the DE/current-to-*p*best/1 mutation strategy is controlled by the parameter p, which balances the trade-off between exploration and exploitation [33]. In SHADE, each individual x_i has an associated p_i, which is given by the following equation [28]:

$$p_i = rand\left[p_{min}, 0.2\right], \quad (14)$$

where p_{min} is set in such a way that at least two individuals are nominated to represent the individual $x_{pbest,G}$, i.e., $p_{min} = 2/N$ [28]. As can be seen from Eq. (14), the maximum value for p_i is 0.2, as suggested by Zhang and Sanderson [31].

After the mutation strategy is applied, each element of the mutant vector that falls outside the search space is re-initialized according to the following equation proposed in [31]:

$$v_{j,i,G} = \begin{cases} \left(x_{j,i,G} + Lb_j\right)/2 & \text{if } v_{j,i,G} < Lb_j \\ \left(x_{j,i,G} + Ub_j\right)/2 & \text{if } v_{j,i,G} > Ub_j \end{cases}. \quad (15)$$

After the mutant vector $\boldsymbol{v}_{i,G}$ is generated, it is combined with the parent vector $\boldsymbol{x}_{i,G}$ in order to generate the trial vector $\boldsymbol{u}_{i,G}$. In the SHADE algorithm, the binomial crossover operator, which is the most common crossover operator used in DE, is used and implemented as follows [33]:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } rand\left[0,1\right) \le CR_i \text{ or } j = j_{rand} \\ x_{j,i,G} & \text{otherwise} \end{cases}, \quad (16)$$

where $rand[0,1)$ is a uniformly distributed random number from [0,1); and j_{rand} is a design variable index which is selected uniformly and randomly from the interval [1, D].

3.4 Selection

After all the trial vectors have been generated, the selection operation is performed to determine that either the trial vectors or the parent ones are chosen for the next generation. In SHADE, similar to the standard DE, each trial vector $\boldsymbol{u}_{i,G}$ is compared against its corresponding parent vector $\boldsymbol{x}_{i,G}$, and the one with a better fitness value is survived to the next generation. This can be expressed as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if } P\left(u_{i,G}\right) \le P\left(x_{i,G}\right) \\ x_{i,G} & \text{otherwise} \end{cases}, \quad (17)$$

where $P(u_{i,G})$ and $P(x_{i,G})$ are the penalized objective function values associated with the i-th trial vector and parent vector, respectively.

3.5 External archive

In order to maintain the population diversity, SHADE uses an external archive. Parent vectors which were replaced by their corresponding trial vectors, and therefore were not able to survive in the next generation, are preserved in the archive. While executing the mutation operation, $x_{r_2,G}$ in Eq. (13) is selected from $P \cup A$, the union of the population P and the archive A. In this study, the size of the archive is set to the same as the size of the population, i.e., $|A| = |P|$, as in [28]. Whenever the size of the archive exceeds $|A|$, some elements are randomly removed from the archive to provide space for the newly inserted elements.

3.6 Historical memory update

In each generation, the CR_i and F_i values that have been successfully used to generate a trial vector $\boldsymbol{u}_{i,G}$ better than the corresponding parent vector $\boldsymbol{x}_{i,G}$ are recorded as S_{CR} and S_F, respectively, and at the end of the generation, the memory contents are updated using the algorithm shown in Algorithm 1 [33]. In this algorithm, the index $k (1 \le k \le H)$ specifies the position in the memory to be updated. At the beginning of the search process, k is initialized to 1, and it is increased by one every time a new element is inserted into the memory. If the index k gets larger than the memory size, i.e., $k > H$, then it is reset to one. It is worth noting

that if all the individuals in a generation fail to generate a trial vector better than the corresponding parent vector, i.e., $S_{CR} = S_F = 0$, then the memory is not updated.

In Algorithm 1, the term $mean_{WL}$ denote the weighted Lehmer mean, which is calculated using the following equation [33]:

$$mean_{WL}\left(S\right) = \frac{\sum_{k=1}^{|S|} \omega_k . S_k^2}{\sum_{k=1}^{|S|} \omega_k . S_k}, \tag{18}$$

$$\omega_k = \frac{\Delta P_k}{\sum_{l=1}^{|S_{CR}|} \Delta P_l}, \tag{19}$$

$$\Delta P_k = \left| P\left(u_{k,G}\right) - P\left(x_{k,G}\right) \right|, \tag{20}$$

where ΔP_k is the amount of fitness improvement, which is used in order to influence the parameter adaption. Note that the term S in Eq. (18) refers to either S_{CR} or S_F.

Finally, the pseudocode of the SHADE algorithm is shown in Algorithm 2.

3.7 Exploration and exploitation characteristics of the SHADE algorithm

A close examination of Eq. (13) reveals that it is not possible to clearly distinguish between exploration and exploitation phases of the SHADE algorithm. This is because of the fact that the DE/current-to-*p*best/1 mutation strategy utilizes not only the information of the best solution and other good solutions in the current population (see the second term of Eq. (13)), but also the information of recently explored inferior solutions (see the third term of Eq. (13)). Indeed, on one hand, any of the top $p \times N (p \in [0,1])$ individuals of the current population can be randomly selected to represent $x_{pbest,G}$ in Eq. (13). It is mentioned again that the parameter p controls the balance between exploration and exploitation. In fact, if the value of p is small, only a few of the top individuals are nominated to represent $x_{pbest,G}$, and this allows for better exploitation of the search space (i.e., the algorithm behaves more greedily). In contrast, if the value of p is large, many of the top individuals are nominated to represent $x_{pbest,G}$, and thus wider exploration of the search space is encouraged (i.e., the algorithm behaves more randomly). The aforementioned feature of the SHADE algorithm makes it particularly well suited for solving multimodal optimization problems where more greedy mutation strategies such as DE/current-to-best/1

Algorithm 1 The memory update algorithm used by SHADE [33]

if $S_{CR} \neq 0$ **and** $S_F \neq 0$ **then**
  if $M_{CR,k,G} = \perp$ **or** $\max(S_{CR}) = 0$
    $M_{CR,k,G+1} = \perp$;
  else
    $M_{CR,k,G+1} = mean_{WL}(S_{CR})$;
  end
  $M_{F,k,G+1} = mean_{WL}(S_F)$;
  $k = k + 1$;
  if $k > H$ **then**
    $k = 1$;
  end if
else
$M_{CR,k,G+1} = M_{CR,k,G}$;
$M_{F,k,G+1} = M_{F,k,G}$;
end if

Algorithm 2 Pseudocode of the SHADE algorithm

Initialization phase
$G = 1$;
Archive $A = 0$;
Index counter $k = 1$;
Intialize population $P_G = (x_{i,G}, x2,_G, \ldots, x_{N,G})$ randomly;
Set all values M_{CR} anf M_F to 0.5;
Main loop
while The termination criteria are not satisfied **do**
  $S_{CR} = 0$, $S_F = 0$;
  for $i = 1$ **to** N **do**
    r_i = Select from [1,H] randomly;
    if $M_{CR,r_i} = \perp$ **then**
      $CR_{i,G} = 0$;
    else
      $CR_{i,G} = randn_i(M_{CR,r_i}, 0.1)$;
    end if
    $F_{i,G} = randc_i(M_{F,r_i}, 0.1)$;
    $p_{i,G} = rand[p_{min}, 0.2]$;
    Generate trial vector $\boldsymbol{u}_{i,G}$ according to DE/current-to-*p*best/1;
  end for
  for $i = 1$ **to** N **do**
    if $P(\boldsymbol{u}_{i,G}) \leq P(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
    else
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
    end if
    if $P(\boldsymbol{u}_{i,G}) \leq P(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G} \rightarrow A$;
      $CR_{i,G} \rightarrow S_{CR}, F_{i,G} \rightarrow S_F$;
    end if
  end for
  If necessary, remove randomly selected individuals from the archive A such that the archive size never exceeds $|A|$;
  Update the memory contents using the algorithm shown in Algorithm 1;
  $G = G + 1$;
end **while**

often leads to premature convergence. On the other hand, in order to improve the population diversity and enhance the exploration ability, the recently explored inferior solutions are also incorporated into the DE/current-to-*p*best/1 mutation strategy, and their difference from the current population is considered as a promising progress direction towards the global optimum. Therefore, in spite of the greedy characteristics of SHADE, the information provided by recently explored inferior solutions is utilized to diversify the population in order to prevent the algorithm from premature convergence.

Like other metaheuristic algorithm, SHADE does not make any assumptions about the problem being optimized and can therefore be applied to a wide class of optimization problems. However, according to the no-free-lunch theorem, it is not possible to have a general-purpose universal metaheuristic that can outperform all other metaheuristics for every conceivable optimization problem. So, like other population-based metaheuristic algorithms, SHADE may require a high computational cost to obtain the global optimal solution of an optimization problem, or even it may not reach it at all. In addition, SHADE cannot be directly appliedto discrete optimization problems and requires some adjustments.

4 Numerical examples

In this section, in order to demonstrate the efficiency and effectiveness of the SHADE algorithm in solving structural optimization problems with frequency constraints, the algorithm is applied to three large-scale truss optimization problems with frequency constraints. The problems include a 600-bar single-layer dome-shaped truss with 25 design variables, a 1180-bar single-layer dome-shaped truss with 59 design variables, and a 1410-bar double-layer dome-shaped truss with 47 design variables. Table 2 lists the material properties, cross-sectional area bounds, and frequency constraints of the above-mentioned problems. The results obtained by the SHADE algorithm are presented and compared with the best-known results reported in the literature. In order to allow a fair comparison of the results with previously published references, improved and hybrid metaheuristic algorithms are often considered for the comparison purpose. Due to the stochastic nature of metaheuristic algorithms, 20 independent successful runs are implemented for each problem. Note that different randomly generated initial populations are considered in each run. The best weight, the worst weight, mean weight, and standard deviation of the optimized weights obtained by SHADE over the runs are provided in the results tables. The design variables corresponding to the best design, the number of objective function evaluations required to reach the best design, and the constraint violation for the best design are also reported. The Friedman rank test is also carried out to rank the algorithms on the basis of the best weight, mean weight, and standard deviation of the optimized weights. Some sensitivity analyses are performed on a number of important parameters of the SHADE algorithm. Based on the results of sensitivity analysis, which will be presented in Section 4.1.1, both the population size N and the memory size H are set to 50 in all experiments. For all experiments, the maximum number of objective function evaluations (*MaxNFEs*) is set to 20000 as the termination criterion of the search process. It should be noted that since it is known that a number of algorithms in the literature do not satisfy the frequency constraints, we therefore consider only those algorithms which have provided fully feasible optimal designs with no constraint violation. For this purpose, by means of the finite element method (FEM) code used in this research for free vibration analysis of truss structures, we check the optimal designs reported in the literature for possible constraint violation. It is pointed out that all the codes are implemented in the Matlab environment and computations are carried out by a PC with Windows 10, Intel(R) Core (TM) i5-7200U CPU 2.50 GHz 2.71 GHz, and 8.00 GB RAM.

4.1 The 600-bar single-layer dome

The first design example considered is the sizing optimization of the 600-bar dome-shaped truss structure shown in Fig. 1(a)–(b). The dome is a rotationally periodic structure with 24 identical substructures, each of which has 9 nodes and 25 elements (see Fig. 1(c)). Consequently, the

Table 2 Material properties, cross-sectional area bounds, and frequency constraints of the optimization problems

Property	600-bar truss	1180-bar truss	1410-bar truss
Elasticity modulus (N/m^2)	2×10^{11}	2×10^{11}	2×10^{11}
Material density (kg/m^3)	7850	7850	7850
Cross-sectional area bounds (m^2)	$0.0001 \le A_i \le 0.01$	$0.0001 \le A_i \le 0.01$	$0.0001 \le A_i \le 0.01$
Frequency constraints (Hz)	$\omega_1 \ge 5, \omega_3 \ge 7$	$\omega_1 \ge 7, \omega_3 \ge 9$	$\omega_1 \ge 7, \omega_3 \ge 9$

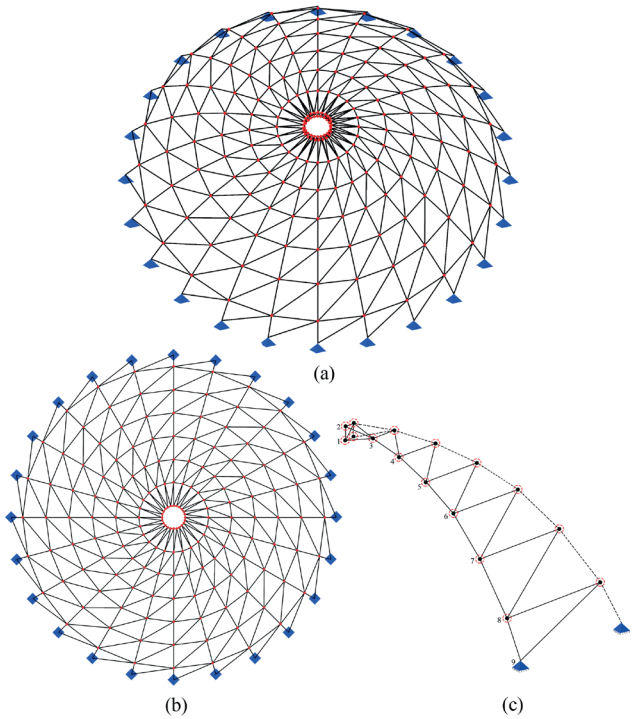

Fig. 1 Schematic of the 600-bar single-layer dome: (a) perspective view; (b) top view; (c) substructure

whole structure has a total of 216 nodes and 600 elements, and the angle between any two adjacent substructures is 15 degrees. Table 3 shows the Cartesian coordinates of the nodes of the first substructure. The only design variables of the problem are the 25 cross-sectional areas of the 25 elements of the substructures. Therefore, this is a sizing optimization problem with 25 design variables. A non-structural mass of 100 kg is added to each free node of the dome. As can be seen from Table 2, frequency constraints are imposed on the first and third natural frequencies of the structure.

4.1.1 Sensitivity analysis of the population size and the memory size

Before the optimization process starts, sensitivity analyses are carried out to study the influence of population size and memory size on the performance of the SHADE algorithm. The 600-bar dome-shaped truss problem is considered for this purpose. In order to determine the best value of population size, it is varied from 40 to 80 with a step size of 10. The maximum number of objective function evaluations is set to 20000 in all the experiments except for N = 60($MaxNFEs$ = 20040), N = 70($MaxNFEs$ = 20020), and N = 90($MaxNFEs$ = 20070). Note that in all the experiments, the memory size is set to be equal to the population size, i.e., $H = N$. Table 4 summarizes the results of sensitivity analysis of population size of the SHADE algorithm over the 20 runs of experiments. As is evident from the table, with a population size of N = 50, the SHADE algorithm achieved the best performance in terms of best weight, mean weight, worst weight, and standard deviation. The same procedure is followed to determine the best value of memory size. For this purpose, different memory sizes (e.g., 10, 20, 30, 40, 50, 100, 150, 200, and 1000) are considered for the SHADE algorithm. For all the experiments, the maximum number of objective function evaluations is set to 20000 and the population size is set to 50. The results of sensitivity analysis of memory size of the SHADE algorithm are presented in Table 5. From this table it is observed that the SHADE algorithm has the best

Table 3 Nodal coordinates (m) of the sub-structure of the 600-bar single-layer dome

Node number	Coordinates (x, y, z)
1	(1.0, 0.0, 7.0)
2	(1.0, 0.0, 7.5)
3	(3.0, 0.0, 7.25)
4	(5.0, 0.0, 6.75)
5	(7.0, 0.0, 6.0)
6	(9.0, 0.0, 5.0)
7	(11.0, 0.0, 3.5)
8	(13.0, 0.0, 1.5)
9	(14.0, 0.0, 0.0)

Table 4 Sensitivity analysis of the population size of the SHADE algorithm (600-bar dome-shaped truss)

Statistical results	Population size						
	N = 40	**N = 50**	N = 60	N = 70	N = 80	N = 90	N = 100
Best weight (kg)	6057.58	**6057.42**	6057.60	6057.99	6060.90	6062.58	6068.42
NFEs	19920	**19850**	19860	19110	19920	19710	19900
Mean weight (kg)	6060.36	**6058.02**	6058.18	6059.78	6063.40	6068.40	6075.95
Worst weight (kg)	6073.67	**6058.93**	6059.59	6062.16	6066.44	6071.10	6081.27
Standard deviation (kg)	3.50	**0.39**	0.46	0.93	1.45	1.84	3.00
MaxNFEs	20000	**20000**	20040	20020	20000	20070	20000
Number of runs	20	**20**	20	20	20	20	20

performance in all statistical measures when the memory size H is equal to 50. Therefore, in all optimization problems, the population size N and the memory size H are set to 50 and 50, respectively. It is worth mentioning that all optimized weights reported in Tables 4 and 5 for sensitivity analysis correspond to fully feasible designs with no constraint violation.

4.1.2 Comparison with other metaheuristic optimization algorithms

This problem was previously investigated by many researchers using various metaheuristic algorithms such as ECBO and ECBO-Cascade [13], DPSO [14], MDVC-UVPS [15], chaotic firefly algorithms based on Logistic map (CLFA) and Gaussian map (CGFA) [16], enhanced forensic-based investigation (EFBI) [17], chaotic water strider algorithm (chaotic WSA) [18], set-theoretical-based Jaya algorithm (ST-JA) [19], parameter free Jaya algorithm (PFJA) [20], a new variant of the colliding bodies optimization (CBO) algorithm called MWQI-CBO [21], Rao algorithms (Rao-1, Rao-2, and Rao-3) [22], improved slime mould algorithm (ISMA) [23], etc. Table 6 compares the results obtained by the SHADE algorithm with previous results reported in the literature. The connectivity information of the nodes of the first substructure is also shown in Table 6. It is evident from the table that SHADE has better performance than all the other compared algorithms in terms of accuracy and robustness. In fact, the SHADE algorithm showed the best results with respect to best weight, mean weight, worst weight, and standard deviation. The best weight obtained by SHADE is 6057.42 kg, which to the best of our knowledge is the best result reported in the literature so far (1,07 kg lower than that of CGFA (6058.49 kg)). The mean weight of the SHADE algorithm is 6058.02 kg, which is even lower than the best weights obtained by other reported algorithms. The worst weight obtained by the SHADE algorithm is 6058.93 kg, which is only slightly higher than the best weight of CGFA (6058.49 kg), but much lower than the best weights obtained by other compared algorithms. It can also be seen from Table 6 that the standard deviation of the SHADE algorithm, which is 0.39 kg, is significantly lower than those of the others, demonstrating the high robustness and reliability of the SHADE algorithm. In terms of computational cost, the maximum number of objective function evaluations of the SHADE algorithm is 20000, which is more than those of DPSO (9000), CGFA (10000), EFBI (12000), and ST-JA (12000), but less than or equal to those of the others. It is however obvious from the results that SHADE performs much better than DPSO, CGFA, EFBI, and ST-JA in terms of best, mean, and worst weights as well as standard deviation of optimized weights. The SHADE algorithm requires only 13250 objective function evaluations to find a feasible optimum design corresponding to a structural weight of 6058.36 kg, which is lower than the best weights obtained by other reported algorithms. Table 7 presents the results of the Friedman rank test. As mentioned before, the SHADE algorithm ranked first with respect to all indicators, namely best weight, mean weight, and standard deviation. Table 8 presents the first five natural frequencies evaluated at the best designs of SHADE and other compared algorithms. The first and third natural frequencies derived from the FEM code are also reported (please see rows 1* and 3* in Table 8). It is worth noting that, according to the results reported in the literature [13–21, 23], none of the frequency constraints are violated. However, the results derived from the FEM code show that the best designs of CGFA and chaotic WSA only slightly violate the frequency constraints (less than 0.0004% and less than 0.0005%, respectively), which may probably be due to the loss of accuracy caused by the rounding of the design variable values reported in Table 6. Therefore, the best designs obtained by CGFA and chaotic WSA are considered as feasible solutions, and thus

Table 5 Sensitivity analysis of the memory size of the SHADE algorithm (600-bar dome-shaped truss)

Statistical results	Memory size								
	H = 10 (0.2*N)	H = 20 (0.4*N)	H = 30 (0.6*N)	H = 40 (0.8*N)	**H = 50 (1.0*N)**	H = 100 (2.0*N)	H = 150 (3.0*N)	H = 200 (4.0*N)	H = 1000 (20.0*N)
Best weight (kg)	6058.51	6057.91	6057.43	6057.69	**6057.42**	6057.63	6057.93	6058.57	6059.66
NFEs	19900	19900	20000	19950	**19850**	19950	20000	19900	17900
Mean weight (kg)	6061.02	6059.28	6058.74	6058.76	**6058.02**	6058.41	6059.06	6059.79	6061.60
Worst weight (kg)	6065.05	6062.74	6060.55	6061.26	**6058.93**	6059.95	6061.68	6064.11	6063.93
Standard deviation (kg)	1.67	1.11	0.64	0.88	**0.39**	0.57	0.75	1.21	1.28
MaxNFEs	20000	20000	20000	20000	**20000**	20000	20000	20000	20000
Number of runs	20	20	20	20	**20**	20	20	20	20

Table 6 Comparison of optimal cross-sectional areas (cm^2) for the 600-bar dome-shaped truss

Element number (element nodes)	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	CGFA [16]	EFBI [17]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	MWQI-CBO [21]	ISMA [23]	This study SHADE
1 (1-2)	1.0299	1.365	1.2575	1.3190	1.0999	1.4829	1.3964	1.1867	1.1414	1.1035	**1.3072**
2 (1-3)	1.3664	1.391	1.3466	1.3826	1.4922	1.2619	1.5177	1.2967	1.1930	1.5801	**1.4158**
3 (1-10)	5.1095	5.686	4.9738	4.9379	6.0744	4.9784	5.5370	4.5771	4.9972	6.2180	**5.0912**
4 (1-11)	1.3011	1.511	1.4025	1.3222	1.6234	1.4155	1.2549	1.3356	1.3359	1.0522	**1.3762**
5 (2-3)	17.0572	17.711	17.3802	17.1285	17.4918	17.5189	16.7759	18.3157	16.4705	17.1566	**17.1326**
6 (2-11)	34.0764	36.266	37.9742	37.4657	37.2118	36.8573	36.8528	38.5097	40.9204	36.5568	**36.7970**
7 (3-4)	13.0985	13.263	13.0306	12.7071	12.7873	13.0251	12.8198	13.5917	12.5481	12.8425	**12.7967**
8 (3-11)	15.5882	16.919	15.9209	15.4252	14.8239	15.0761	15.4141	16.8824	16.8270	15.3463	**15.3146**
9 (3-12)	12.6889	13.333	11.9419	11.3642	12.1764	11.6297	12.0638	13.8766	12.3559	11.9044	**11.4263**
10 (4-5)	10.3314	9.534	9.1643	9.3343	9.0163	9.5607	9.3500	9.5286	9.8049	9.4559	**9.3386**
11 (4-12)	8.5313	9.884	8.4332	8.3872	8.5044	8.2689	8.2980	9.4218	8.8128	8.1976	**8.3567**
12 (4-13)	9.8308	9.547	9.2375	9.1101	8.9951	8.8515	8.8205	9.7643	8.9853	9.0644	**9.1674**
13 (5-6)	7.0101	7.866	7.2213	7.1472	7.0357	7.0387	7.4253	7.2431	7.4324	7.6937	**7.1952**
14 (5-13)	5.2917	5.529	5.2142	5.1701	5.0993	5.2711	5.1621	5.3913	4.4777	5.1748	**5.1822**
15 (5-14)	6.2750	7.007	6.7961	6.6239	6.1918	6.5632	6.6351	6.7468	6.7637	6.7264	**6.6314**
16 (6-7)	5.4305	5.462	5.2078	5.2427	4.9514	5.1025	4.9351	5.1493	5.3079	4.8059	**5.1532**
17 (6-14)	3.6414	3.853	3.4586	3.5213	3.9186	3.4304	3.5639	3.8342	3.7870	3.6390	**3.5759**
18 (6-15)	7.2827	7.432	7.6407	7.6096	7.6312	7.7083	8.0435	8.0665	7.5167	7.7180	**7.6149**
19 (7-8)	4.4912	4.261	4.3690	4.2877	4.4271	4.3958	4.2061	4.2800	4.3198	4.0911	**4.3305**
20 (7-15)	1.9275	2.253	2.1237	2.1684	2.3280	2.0435	2.3310	2.2509	1.9381	2.1339	**2.1479**
21 (7-16)	4.6958	4.337	4.5774	4.6704	4.8534	4.4764	4.4953	4.5372	4.8992	4.4482	**4.6249**
22 (8-9)	3.3595	4.028	3.4564	3.5380	3.9632	3.659	3.4287	3.5615	3.2783	3.4785	**3.5261**
23 (8-16)	1.7067	1.954	1.7920	1.8252	1.8527	1.9727	1.8660	1.7744	1.8130	1.8191	**1.8462**
24 (8-17)	4.8372	4.709	4.8264	4.8110	4.7818	4.8843	4.9318	4.6445	4.8722	4.8903	**4.7961**
25 (9-17)	2.0253	1.410	1.7601	1.6589	1.4354	1.6167	1.5022	1.6141	1.9181	1.7001	**1.6374**
Best weight (kg)	6140.51	6344.55	6115.10	6058.49	6076.35	6064.04	6065.811	6333.251	6147.96	6068.34	**6057.42**
											6058.36[1]
Mean weight (kg)	6175.33	6674.71	6119.95	6076.67	6098.52	6081.23	6072.734	6380.31	6215.29	6083.93	**6058.02**
Worst weight (kg)	N/A	N/A	N/A	N/A	6113.56	N/A	6084.749	N/A	N/A	6095.41	**6058.93**
Standard deviation (kg)	34.08	473.21	16.23	22.42	11.95	8.29	6.182	47.396	51.42	7.36	**0.39**
NFEs	17300	N/A	17513	N/A	N/A	N/A	N/A	8580	16560	20000	**19850**
MaxNFEs	20000	9000	30000	10000	12000	30000	12000	25000	20000	20000	**20000**
CV^2 (%)	**0**	**0**	**0**	**0.0004**	**0**	**0.0005**	**0**	**0**	**0**	**0**	**0**
Number of runs	5	10	30	20	10	20	10	20	20	25	**20**

[1] The SHADE algorithm has achieved a feasible design corresponding to a structural weight of 6058.36 kg after 13250 objective function evaluations.
[2] Constraint violation

these algorithms are included in the comparison. An examination of the results presented in Table 8 shows that the natural frequencies calculated by the FEM code are very close to those reported in the literature, and the very small differences are probably due to the rounding of the design variable values, as discussed above. Fig. 2 shows the convergence histories of the best and mean results obtained by the SHADE algorithm for the 600-bar truss. From the figure it can be seen that the two curves lie very close to each other, confirming the high robustness of the SHADE algorithm. Fig. 3 shows the optimized weights obtained by SHADE for the 600-bar truss. The figure indicates that in 10 out of 20 runs, the SHADE algorithm has converged to solutions with weights below the mean weight.

Table 7 The Friedman rank test results for the 600-bar dome-shaped truss

	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	CGFA [16]	EFBI [17]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	MWQI-CBO [21]	ISMA [23]	This study SHADE
Friedman rank of best weight	8	11	7	2	6	3	4	10	9	5	**1**
Friedman rank of mean weight	8	11	7	3	6	4	2	10	9	5	**1**
Friedman rank of standard deviation	8	11	6	7	5	4	2	9	10	3	**1**

Table 8 Natural frequencies (Hz) of the 600-bar dome-shaped truss evaluated at optimal designs

Frequency number	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	CGFA [16]	EFBI [17]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	MWQI-CBO [21]	ISMA [23]	This study SHADE
1	5.001	5.000	5.000	5.000	5.0001	5.0005	5.0002	5.0011	5.004	5.0003	**5.0000**
1*[1]	**5.0031**	**5.0089**	**5.0023**	**5.0000**[2]	**5.0001**	**5.0005**	**5.0002**	**5.0100**	**5.0064**	**5.0003**	**5.0000**
2	5.001	5.000	5.000	5.000	5.0001	N/A	5.0002	5.0011	5.004	5.0003	**5.0000**
3	7.001	7.000	7.000	7.000	7.0000	7.0000	7.0002	7.0000	7.000	7.0002	**7.0000**
3*	**7.0070**	**7.0213**	**7.0047**	**7.0000**	**7.0000**	**7.0000**[3]	**7.0002**	**7.0210**	**7.0056**	**7.0002**	**7.0000**
4	7.001	7.000	7.000	7.000	7.0000	N/A	7.0006	7.0000	7.000	7.0002	**7.0000**
5	7.002	7.000	7.000	7.001	7.0024	N/A	7.0006	7.0000	7.001	7.0003	**7.0000**

[1] The results derived from our finite element method (FEM) code the first and third natural frequencies
[2] The more precise value is 4.999980 Hz.
[3] The more precise value is 6.999966 Hz.

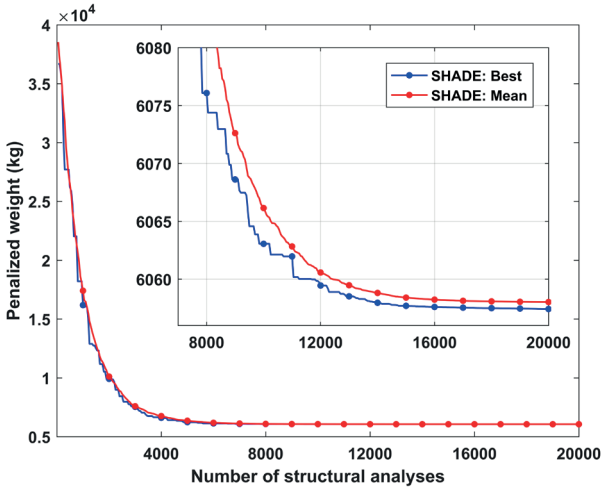

Fig. 2 The best and mean convergence histories of the SHADE algorithm for the 600-bar dome-shaped truss

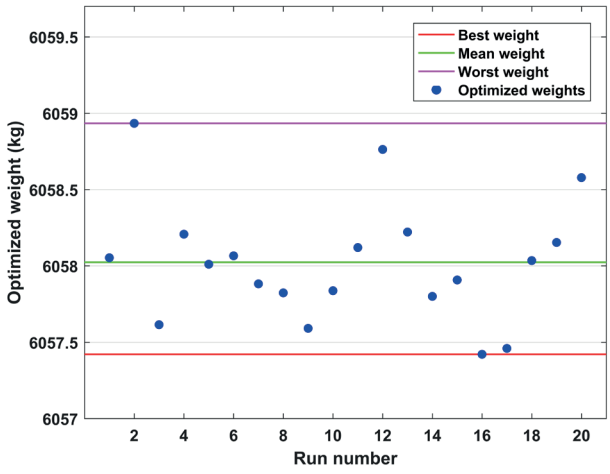

Fig. 3 Diversity of the optimized weights obtained by the SHADE algorithm for the 600-bar dome-shaped truss

4.2 The 1180-bar single-layer dome

The 1180-bar single-layer dome-shaped truss structure shown in Fig. 4(a–b) is considered as the second design example. The dome is also a rotationally periodic structure, and it consists of 20 identical substructures, each of which spans an angle of 18 degrees. As shown in Fig. 4(c), each substructure has 20 nodes and 59 elements, resulting in a total of 400 nodes and 1180 elements. The nodal Cartesian coordinates of the first substructure are shown in Table 9. The connectivity information of the nodes of the first substructure is given also in Table 10. The 59 cross-sectional areas of the 59 elements of the substructures are the only design variables of the problem. Hence, this is a sizing optimization problem with 59 design variables. A non-structural mass of 100 kg is attached to all free nodes of the dome. Similar to the previous design example, frequency constraints are imposed on the first and third natural frequencies of the structure.

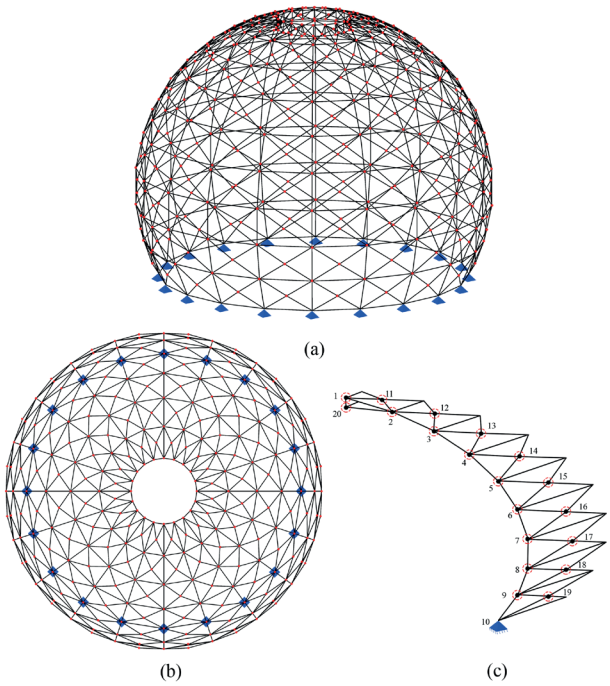

Fig. 4 Schematic of the 1180-bar single-layer dome: (a) perspective view; (b) top view; (c) substructure

This structure was previously optimized using several metaheuristic algorithms such as ECBO-Cascade [13], DPSO [14], MDVC-UVPS [15], chaotic WSA [18], PFJA [20], Rao-1 and Rao-2 [22], ISMA [23], etc. Table 10 presents a comparison of the optimal results obtained by the SHADE algorithm with those reported in the literature. It is observed from the table that SHADE performs much better than all the other compared algorithms in terms of accuracy and efficiency. Indeed, the SHADE algorithm produced the best results in terms of best, mean, and worst weights. The best design obtained by the SHADE algorithm has a structural weight of 37321.72 kg, which to our best knowledge is the best result reported in the literature so far (45.75 kg lower than that of the ISMA (37367.47 kg)). The mean weight of the SHADE algorithm is 37343.26 kg, which is even lower than the best weights obtained by other reported algorithms, indicating the high accuracy and robustness of the SHADE algorithm. The standard deviation obtained by SHADE is 19.25 kg, which is only higher than that of the Rao-2 (8.58 kg), but much lower than those of other compared algorithms. In terms of computational cost, the maximum number of objective function evaluations of the SHADE algorithm is 20000, which is less than or equal to those of the others. It is noted that the SHADE algorithm requires only 15800 objective function evaluations to achieve a feasible design corresponding to a structural weight of 37362.17 kg, which is better than the best weights obtained by other algorithms in Table 10. The results of the Friedman rank test are presented in Table 11. It is seen from the table that the SHADE algorithm ranked first in terms of best and mean weights. Table 12 lists the first five natural frequencies corresponding to the optimal designs obtained by SHADE and other algorithms in the literature. The first and third natural frequencies calculated by the FEM code used in this research are also reported (please see rows 1* and 3* in Table 12). It is clear from the table that none of the frequency constraints are violated, indicating the feasibility of the optimal designs reported in Table 10. Therefore, all the constraint violations (CV values in Table 10) are set to zero. It can be observed from Table 12 that the values derived by the FEM code for the first and third natural frequencies are very close to those reported in the literature, and the very small differences are probably due to the rounding of the design variable values. Fig. 5 compares the convergence histories of the best and mean results obtained by the SHADE algorithm for the 1180-bar truss. It can be seen that the curves are very close together, which demonstrates high stability and reliability of the SHADE algorithm. The optimized weights obtained by SHADE for the 1180-bar truss are shown in Fig. 6. From the figure it can be seen that 12 out of 20 optimized weights achieved by the SHADE algorithm fell below the mean weight.

Table 9 Nodal coordinates (m) of the sub-structure of the 1180-bar single-layer dome

Node number	Coordinates (x, y, z)	Node number	Coordinates (x, y, z)
1	(3.1181, 0.0, 14.6723)	11	(4.5788, 0.7252, 14.2657)
2	(6.1013, 0.0, 13.7031)	12	(7.4077, 1.1733, 12.9904)
3	(8.8166, 0.0, 12.1354)	13	(9.9130, 1.5701, 11.1476)
4	(11.1476, 0.0, 10.0365)	14	(11.9860, 1.8984, 8.8165)
5	(12.9904, 0.0, 7.5000)	15	(13.5344, 2.1436, 6.1013)
6	(14.2657, 0.0, 4.6358)	16	(14.4917, 2.2953, 3.1180)
7	(14.9179, 0.0, 1.5676)	17	(14.8153, 2.3465, 0.0)
8	(14.9179, 0.0, -1.5677)	18	(14.9179, 2.2953, -3.1181)
9	(14.2656, 0.0, -4.6359)	19	(13.5343, 2.1436, -6.1014)
10	(12.9903, 0.0, -7.5001)	20	(3.1181, 0.0, 13.7031)

4.3 The 1410-bar double-layer dome

In the last design example, we consider the size optimization of the 1410-bar double-layer dome-shaped truss shown in Fig. 7(a)–(b). The dome is also a rotationally periodic structure consisting of 30 identical substructures. Each substructure, which spans an angle of 12 degrees, has 13

Table 10 Comparison of optimal cross-sectional areas (cm^2) for the 1180-bar dome-shaped truss

Element number (element nodes)	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
1 (1-2)	8.0110	7.926	7.3691	6.9078	7.952	7.2170	7.8959	7.3290	**7.1228**
2 (1-11)	8.7028	10.426	9.3399	10.7524	10.466	9.6995	9.7301	8.9843	**9.6201**
3 (1-20)	3.1616	2.115	2.7203	2.9439	2.089	2.4203	3.0395	2.2198	**2.3299**
4 (1-21)	13.6820	14.287	13.2822	13.487	14.219	13.7398	14.0549	14.9049	**13.7900**
5 (1-40)	3.2865	3.846	3.6758	3.3147	3.944	3.0521	3.3696	3.5063	**3.5864**
6 (2-3)	6.0397	5.921	6.1391	6.9318	5.979	6.1827	6.4880	5.6930	**6.0257**
7 (2-11)	8.4370	7.955	7.0964	7.6325	7.775	7.2240	6.5686	7.3918	**7.1295**
8 (2-12)	6.4122	6.697	6.0208	6.2343	6.351	6.7305	6.4686	6.0614	**6.6226**
9 (2-20)	2.6346	1.889	2.1225	1.3899	1.896	2.3985	1.9171	1.8952	**1.9530**
10 (2-22)	11.7440	11.881	12.3488	12.9919	11.908	12.1406	9.5624	12.2717	**11.7304**
11 (3-4)	7.9272	7.121	6.8578	6.9162	7.241	6.9400	6.9007	7.2064	**7.2836**
12 (3-12)	5.4548	6.080	5.7773	5.119	5.647	5.5447	4.7037	5.5662	**6.0315**
13 (3-13)	6.7221	6.599	6.9931	8.7795	6.700	7.1585	6.6932	6.6235	**6.6959**
14 (3-23)	8.1544	7.772	7.3355	6.684	7.799	7.2911	7.0856	7.7710	**7.0130**
15 (4-5)	9.7560	9.358	10.5464	9.317	9.198	9.1486	8.9411	9.3547	**9.1492**
16 (4-13)	6.5905	6.213	6.9589	6.483	6.282	6.7518	6.1621	5.3972	**6.3961**
17 (4-14)	7.0392	8.200	8.0977	8.2833	7.695	8.0406	7.9640	7.7431	**8.2877**
18 (4-24)	6.9219	7.799	7.7738	8.0703	7.520	7.5381	7.9441	6.8348	**7.5500**
19 (5-6)	11.6919	11.752	12.4614	12.7141	11.840	12.1508	11.0882	12.3818	**12.7054**
20 (5-14)	9.8890	7.494	7.8154	7.0934	7.230	8.2227	8.6185	7.9853	**8.3432**
21 (5-15)	9.3316	9.696	10.2039	10.069	10.211	9.4734	9.4246	9.8527	**10.6372**
22 (5-25)	9.1093	9.177	8.9262	9.7217	9.252	8.6688	9.0507	10.4516	**9.0214**
23 (6-7)	18.1212	17.326	16.5275	17.2315	17.222	17.2540	16.8125	14.9770	**17.4130**
24 (6-15)	10.6725	11.797	9.0166	9.7761	11.417	10.7425	10.1184	12.3260	**10.3270**
25 (6-16)	13.5340	14.002	13.8204	13.2779	14.196	13.4192	13.1714	13.1421	**13.6138**
26 (6-26)	12.0248	11.562	11.4021	11.5212	11.639	11.7981	11.7831	11.3121	**11.4680**
27 (7-8)	23.1245	23.981	24.2631	21.2086	24.065	24.1043	22.7860	23.3776	**23.5722**
28 (7-16)	15.2630	12.996	14.5494	13.0618	13.377	13.6533	14.7236	13.1092	**14.0620**
29 (7-17)	18.3075	16.591	17.7753	17.9725	16.469	18.1446	18.0196	18.8585	**18.1928**
30 (7-27)	15.2361	15.910	15.4594	14.0147	16.057	15.1399	15.9128	15.6792	**15.0966**
31 (8-9)	40.0749	34.642	34.1372	33.5273	34.125	33.6250	36.5751	32.4636	**34.3266**
32 (8-17)	18.4775	19.860	19.1254	20.1075	18.866	18.8972	17.4997	18.8287	**18.3050**
33 (8-18)	26.0689	25.079	24.1954	23.098	24.600	24.7076	24.8820	22.7241	**23.9768**
34 (8-28)	21.2213	18.965	21.5899	22.0597	21.103	21.2846	22.2113	20.0712	**21.2256**
35 (9-10)	46.3724	47.514	49.4717	49.187	47.696	49.5489	49.6996	48.8253	**48.3061**
36 (9-18)	23.6689	28.133	26.2915	26.835	27.760	24.9924	24.7757	29.6377	**25.0328**
37 (9-19)	35.0703	33.023	33.7558	31.4569	33.518	33.2066	32.2627	33.7600	**33.3295**
38 (9-29)	27.9369	32.263	29.7608	30.2512	31.773	31.2906	30.2725	32.0940	**30.6787**
39 (10-19)	34.2912	33.401	34.0489	34.4764	33.592	36.3060	34.5930	36.5235	**36.1348**
40 (10-30)	1.0726	1.344	1.0024	1.1023	1.000	1.0023	1.0130	1.0010	**1.0031**
41 (11-21)	8.5106	9.327	9.0344	9.9842	9.455	9.4299	10.1462	9.7921	**9.4517**
42 (11-22)	6.8664	7.202	7.5316	7.5443	7.189	6.7476	6.7192	7.0223	**7.4488**
43 (12-22)	5.8229	6.792	6.3726	7.6993	6.767	6.7758	6.8349	6.7364	**6.5251**
44 (12-23)	5.3986	6.228	5.7643	6.0238	6.322	5.8630	6.2105	5.4003	**5.7713**
45 (13-23)	8.0669	6.601	6.7270	6.4087	6.720	6.7186	6.3702	6.8698	**6.7446**

Continuation of Table 10

Element number (element nodes)	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
46 (13-24)	6.9797	6.584	6.7021	6.4428	6.425	5.3396	5.8349	6.3982	**6.3670**
47 (14-24)	7.2735	8.320	7.8082	8.4235	8.451	8.0751	7.8920	8.0176	**7.8546**
48 (14-25)	9.1827	8.844	8.1225	8.7143	8.176	8.2697	7.7339	8.0822	**8.1084**
49 (15-25)	10.6227	11.254	10.1777	9.8677	10.069	11.1665	10.3879	10.4368	**9.9441**
50 (15-26)	11.5740	12.162	10.1825	11.4715	12.219	10.4236	10.3624	10.6422	**10.1114**
51 (16-26)	15.5194	13.854	13.4590	15.248	13.257	12.8953	12.7819	13.2563	**13.7876**
52 (16-27)	14.1342	13.844	13.9788	12.9199	13.782	13.9238	13.4706	14.3907	**13.5021**
53 (17-27)	17.1612	17.536	18.1070	19.5895	17.573	18.5162	19.3227	18.5010	**18.2259**
54 (17-28)	19.0798	20.551	19.2212	20.1524	19.909	18.9697	20.1512	18.2627	**18.9636**
55 (18-28)	23.4414	24.072	23.4359	24.519	24.019	23.7699	26.2932	24.5053	**24.1903**
56 (18-29)	26.5726	27.287	27.6479	25.838	27.701	25.4931	27.4623	27.0310	**25.0437**
57 (19-29)	33.4104	32.965	33.6805	36.4546	32.918	32.4964	34.8006	34.6221	**33.0168**
58 (19-30)	37.1198	36.940	35.7035	37.7461	37.001	35.7653	35.6905	35.3352	**37.3380**
59 (20-40)	4.7593	3.837	4.7617	3.7146	3.864	4.6791	5.3624	4.7698	**4.4459**
Best weight (kg)	37770.71	37779.81	37451.77	37642.38	37695.59	37367.47	37520.42	37508.67	**37321.72**
									37362.17[1]
Mean weight (kg)	37885.15	38294.45	37545.53	37795.53	37755.05	37432.92	37557.4	37526.99	**37343.26**
Worst weight (kg)	N/A	N/A	N/A	N/A	N/A	37555.37	N/A	N/A	**37405.11**
Standard deviation (kg)	133.84	550.5	64.85	165.32	58.025	45.13	22.66	8.58	**19.25**
NFEs	19180	N/A	19391	N/A	18650	16320	44790	44280	**20000**
MaxNFEs	20000	50000	30000	30000	25000	20000	45000	45000	**20000**
CV (%)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Number of runs	5	10	30	20	20	25	10	10	**20**

[1] The SHADE algorithm has achieved a feasible design corresponding to a structural weight of 37362.17 kg after 15800 objective function evaluations.

Table 11 The Friedman rank test results for the 1180-bar dome-shaped truss

	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
Friedman rank of best weight	8	9	3	6	7	2	5	4	**1**
Friedman rank of mean weight	8	9	4	7	6	2	5	3	**1**
Friedman rank of standard deviation	7	9	6	8	5	4	3	1	**2**

Table 12 Natural frequencies (Hz) of the 1180-bar dome-shaped truss evaluated at optimal designs

Frequency number	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
1	7.000	7.000	7.000	7.0001	7.0000	7.0000	7.0007	7.0005	**7.0000**
1*	**7.0023**	**7.0088**	**7.0019**	**7.0001**	**7.0084**	**7.0000**	**7.0007**	**7.0005**	**7.0000**
2	7.001	7.000	7.001	N/A	7.0000	7.0000	7.0007	7.0005	**7.0000**
3	9.002	9.000	9.000	9.0049	9.0024	9.0000	9.0008	9.0027	**9.0000**
3*	**9.0245**	**9.0472**	**9.0124**	**9.0049**	**9.0541**	**9.0000**	**9.0008**	**9.0027**	**9.0000**
4	9.002	9.000	9.000	N/A	9.0024	9.0000	9.0008	9.0027	**9.0000**
5	9.062	9.005	9.005	N/A	9.0129	9.0033	9.0174	9.0273	**9.0007**

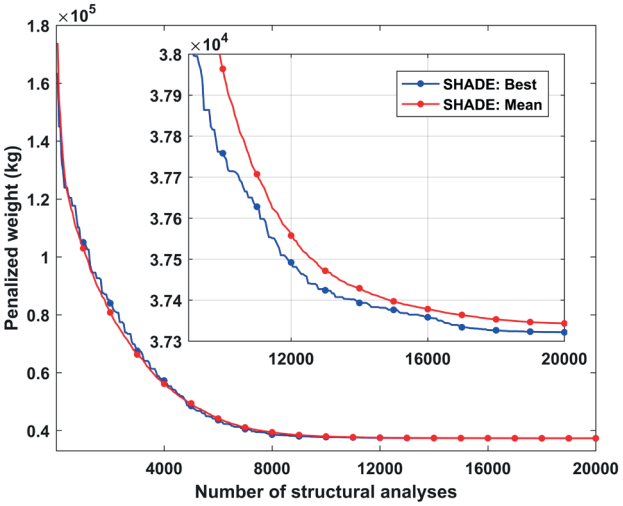

Fig. 5 The best and mean convergence histories of the SHADE algorithm for the 1180-bar dome-shaped truss

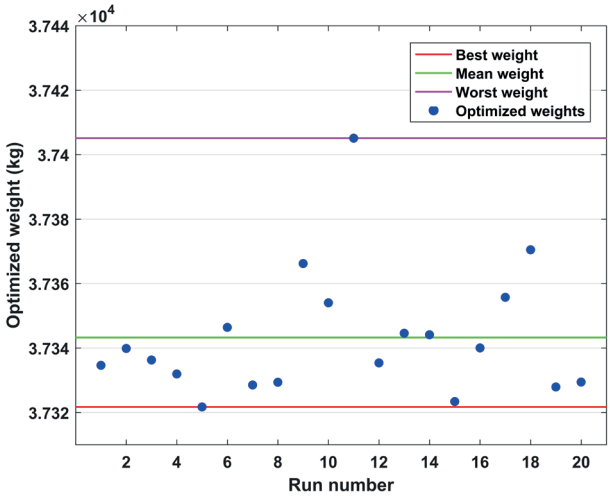

Fig. 6 Diversity of the optimized weights obtained by the SHADE algorithm for the 1180-bar dome-shaped truss

nodes and 47 elements (please see Fig. 7(c)). Therefore, the entire structure has a total of 390 nodes and 1410 elements. Table 13 lists the nodal Cartesian coordinates of the first substructure. The connectivity information of the nodes of the first substructure is also given in Table 14. The cross sectional area of each element of the substructure is considered as a sizing design variable. However, both the shape and topology of the structure are kept unchanged during the optimization process. Therefore, this is a sizing optimization problem with 47 design variables. Non-structural masses of 100 kg are attached to all free nodes. As shown in Table 2, frequency constraints are imposed on the first and third natural frequencies of the structure.

The 1410-bar dome optimization problem was previously solved using different metaheuristic algorithms such as ECBO-Cascade [13], DPSO [14], MDVC-UVPS [15],

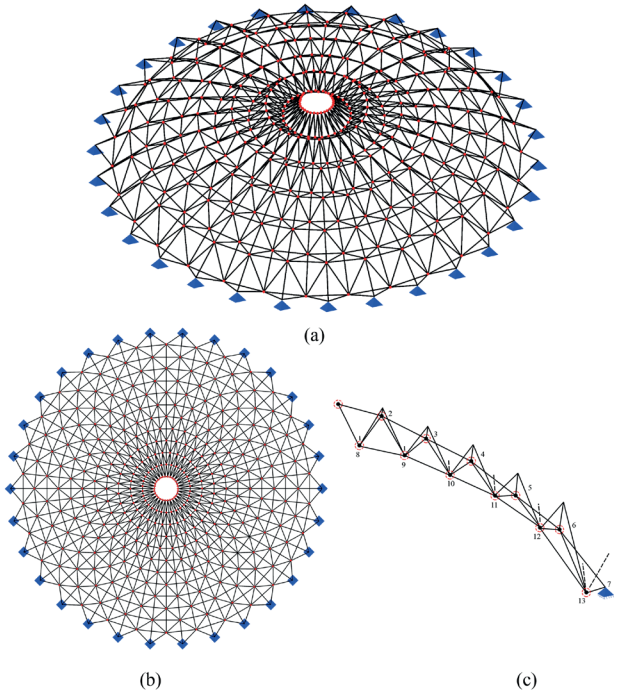

Fig. 7 Schematic of the 1410-bar double-layer dome: (a) perspective view; (b) top view; (c) substructure

Table 13 Nodal coordinates (m) of the sub-structure of the 1410-bar double-layer dome

Node number	Coordinates (x, y, z)	Node number	Coordinates (x, y, z)
1	(1.0, 0.0, 4.0)	8	(1.989, 0.209, 3.0)
2	(3.0, 0.0, 3.75)	9	(3.978, 0.418, 2.75)
3	(5.0, 0.0, 3.25)	10	(5.967, 0.627, 2.25)
4	(7.0, 0.0, 2.75)	11	(7.956, 0.836, 1.75)
5	(9.0, 0.0, 2.0)	12	(9.945, 1.0453, 1.0)
6	(11.0, 0.0, 1.25)	13	(11.934, 1.2543, -0.5)
7	(13.0, 0.0, 0.0)		

chaotic WSA [18], ST-JA [19], PFJA [20], Rao-1 and Rao-2 [22], ISMA [23], etc. A comparison of the optimal results obtained by the SHADE algorithms with those given by other algorithms is presented in Table 14. We can see from the results of the table that in terms of best, average, and worst weights, SHADE produces better results than other compared algorithms. The best weight of the SHADE algorithm is 10236.73 kg, which to the best of our knowledge, is the best feasible solution reported in the literature so far for this problem (44.33 kg lower than that of the Rao-2 (10281.06 kg)). The mean and worst weights obtained by SHADE are also even lower than the best weights obtained by other algorithms in Table 14. The standard deviation values obtained by the Rao-1 and Rao-2 algorithms (4.12 kg and 3.06 kg, respectively)

Table 14 Comparison of optimal cross-sectional areas (cm^2) for the 1410-bar dome-shaped truss

Element number (element nodes)	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
1 (1-2)	7.9969	7.209	5.8499	6.3476	5.3860	6.1902	7.5409	7.4709	5.8948	**5.8073**
2 (1-8)	6.1723	5.006	4.5115	5.188	4.4361	4.4036	4.2469	6.3452	4.9416	**4.4575**
3 (1-14)	35.5011	38.446	19.4823	24.4074	27.7395	31.2253	39.4526	36.3515	29.1048	**29.2417**
4 (2-3)	10.2510	9.438	8.8480	8.5241	8.1780	8.4715	9.1605	10.4516	9.7312	**9.2327**
5 (2-8)	5.3727	4.313	5.0084	5.5439	6.1189	4.8590	5.6874	6.0640	6.9672	**5.0947**
6 (2-9)	1.3488	1.494	1.3568	1.202	1.5996	1.5759	1.6950	1.6703	1.4920	**1.5153**
7 (2-15)	11.4427	8.455	17.4331	14.6949	18.7495	12.9451	10.0121	13.4066	19.5519	**16.7115**
8 (3-4)	9.7157	9.488	9.1098	9.3726	9.2413	9.3263	9.5577	8.7980	9.2523	**9.2952**
9 (3-9)	1.3005	3.480	2.8712	1.462	2.6310	3.2716	3.1140	1.6167	1.7498	**2.5666**
10 (3-10)	2.5046	3.495	3.5473	2.5768	2.2419	3.2878	3.3181	2.0968	2.1261	**2.8494**
11 (3-16)	10.7849	16.037	12.3768	10.722	7.9773	12.6719	11.3804	5.7884	6.0843	**9.7496**
12 (4-5)	10.1954	9.796	10.1099	8.7231	10.1147	10.0979	9.1226	9.7369	9.8695	**9.4270**
13 (4-10)	2.2300	2.413	2.5797	1.9054	2.1849	2.5803	2.0173	2.2057	1.9067	**2.4404**
14 (4-11)	5.1186	5.681	5.8381	3.8895	6.4570	5.3769	5.1731	4.8995	4.2620	**5.0692**
15 (4-17)	14.0053	15.806	13.6402	12.8913	18.2837	16.0581	17.2736	17.6401	16.6403	**16.5099**
16 (5-6)	8.9713	8.078	9.9096	8.05	8.2311	8.6789	8.9213	8.3681	8.4249	**8.4388**
17 (5-11)	4.0756	3.931	3.6543	2.9941	3.1540	3.3199	3.6048	3.5504	2.7837	**3.5592**
18 (5-12)	5.9211	6.099	6.1529	7.2349	5.8960	6.4966	7.0700	6.3018	7.1246	**6.2774**
19 (5-18)	10.6915	10.771	11.2448	15.3852	19.8678	10.8804	11.4002	11.9611	13.3816	**11.9678**
20 (6-7)	10.6220	13.775	13.1071	13.8992	13.5511	14.0056	12.9837	12.7131	12.9834	**13.1426**
21 (6-12)	4.5064	4.231	5.2361	5.6867	5.5768	5.0843	5.7632	4.8717	5.3181	**5.5297**
22 (6-13)	8.4086	6.995	7.0691	7.8515	6.7898	6.9952	7.5395	7.3640	6.7909	**7.1816**
23 (6-19)	5.8405	1.837	2.0015	1.0011	1.0175	1.0270	1.0251	1.1361	1.0845	**1.0710**
24 (7-13)	5.0342	4.397	4.7178	4.327	4.1502	4.3788	4.1529	4.8399	4.4445	**4.7317**
25 (8-9)	3.8932	2.115	2.6101	3.5281	2.7295	2.1951	2.8327	3.3318	2.9908	**2.5033**
26 (8-14)	6.1647	4.923	4.5434	4.5177	4.1142	4.2562	5.0185	5.4117	4.7755	**4.3824**
27 (8-15)	6.8990	4.047	4.6174	6.8448	5.8905	4.6605	4.9278	5.6457	6.6551	**5.2803**
28 (8-21)	11.6387	5.906	9.6758	12.9102	11.5388	8.8694	9.3110	11.8737	14.1558	**11.3890**
29 (9-10)	3.8343	3.392	3.6296	3.8706	4.3093	3.2333	3.0206	4.1043	4.3985	**3.5157**
30 (9-15)	1.4772	1.902	1.4891	1.0192	1.8975	1.7611	1.1071	1.0000	1.9432	**1.5929**
31 (9-16)	1.3075	4.381	3.4020	1.2962	2.5412	3.2831	1.7734	1.3190	1.6670	**2.3454**
32 (9-22)	4.4876	8.442	6.2153	2.5497	4.6417	7.1936	4.8328	2.4216	3.8517	**4.9127**
33 (10-11)	6.0196	5.011	5.9308	5.6478	5.4994	4.9840	5.0213	5.0380	5.6797	**5.1727**
34 (10-16)	2.6729	3.577	3.2334	2.775	2.4481	3.6672	2.5054	2.0298	2.2237	**2.6676**
35 (10-17)	1.6342	2.805	2.7173	2.1062	2.0894	2.4062	2.3223	2.7563	2.1280	**2.3925**
36 (10-23)	1.8410	2.024	1.3932	2.5833	1.7796	2.1576	3.2361	4.7604	1.0056	**2.8330**
37 (11-12)	6.8841	6.709	6.5660	7.3146	7.9676	7.1043	6.7371	6.3381	6.9507	**7.3064**
38 (11-17)	4.1393	5.054	4.8170	3.7673	4.9463	5.2070	5.0727	4.0629	4.9632	**5.0684**
39 (11-18)	3.3264	3.259	3.2626	2.9003	3.3697	3.6853	3.1729	3.4340	3.0043	**3.5115**
40 (11-24)	1.0000	1.063	1.0165	1	1.0136	1.0007	1.0001	1.0016	1.0043	**1.0005**
41 (12-13)	6.9373	5.934	7.2529	7.0355	6.9306	6.6302	6.1117	6.8108	6.5501	**6.7922**
42 (12-18)	4.4568	7.057	5.9226	6.9735	6.4813	6.6773	6.2896	6.7405	6.4851	**6.2743**
43 (12-19)	4.6758	5.745	5.3115	5.5549	5.3985	5.2167	5.3444	5.5967	5.4731	**5.1402**
44 (12-25)	1.0084	1.185	1.0010	1.0001	1.0063	1.0016	1.0001	1.0080	1.0002	**1.0024**

Continuation of Table 14

Element number (element nodes)	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	**This study**
										SHADE
45 (13-19)	7.5103	7.274	7.7499	8.5706	7.8888	8.1289	7.8981	7.7374	7.5534	**7.4893**
46 (13-20)	5.2449	4.798	4.7836	5.6116	4.3830	4.5151	5.1166	4.9869	4.7768	**4.6857**
47 (13-26)	1.0550	1.515	1.0035	1.0127	1.0086	1.0010	1.0052	1.0000	1.0222	**1.0049**
Best weight (kg)	10504.20	10453.84	10345.12	10318.99	10283.94	10326.296	10309.41	10322.21	10281.06	**10236.73**
										10276.75[1]
Mean weight (kg)	10590.67	11100.57	10393.83	10521.67	10379.632	10399.828	10556.67	10331.59	10289.22	**10255.52**
Worst weight (kg)	N/A	N/A	N/A	N/A	10491.617	N/A	10825.00	N/A	N/A	**10275.46**
Standard deviation (kg)	52.51	334.20	39.15	122.1458	57.586	75.441	130.92	4.12	3.06	**11.15**
NFEs	19460	N/A	17750	N/A	19620	16900	20000	37050	44850	**20000**
MaxNFEs	20000	50000	20000	30000	20000	25000	20000	45000	45000	**20000**
CV (%)	**0**	**0**	**0**	**0.0034**	**0**	**0**	**0**	**0**	**0**	**0**
Number of runs	5	10	30	20	10	20	25	10	10	**20**

[1] The SHADE algorithm has achieved a feasible design corresponding to a structural weight of 10276.75 kg after 12050 objective function evaluations.

are however lower than that of the SHADE algorithm (11.15 kg). In terms of computational cost, the maximum number of objective function evaluations of the SHADE algorithm is 20000, which is less than or equal to those of other algorithms. The results show that the SHADE algorithm requires only 12050 objective function evaluations to find a feasible design corresponding to a structural weight of 10276.75 kg, which is lower than the best weights of other reported algorithms. Table 15 presents the results of the Freidman rank test. As mentioned above, it can be seen that the SHADE algorithm ranked first in terms of best and mean weights. In terms of standard deviation, the SHADE algorithm however ranked third after Rao-1 and Rao-2. The first five natural frequencies evaluated at the best designs obtained by SHADE and other compared algorithms are listed in Table 16. The first and third natural frequencies obtained by the FEM code used in the present study are also presented (please see rows 1* and 3* in Table 16). From the results reported in the literature [13–15, 18–20, 22, 23], it is observed that none of the frequency constraints are violated. However, the results derived from the FEM code indicate that the best design obtained by chaotic WSA only slightly violates the frequency constraints (less than 0.0035%), which may probably be due to the loss of accuracy caused by the rounding of the values of the design variables reported in Table 14. Therefore, the best design gained by chaotic WSA is considered as a feasible solution, and thus this algorithm is included in the comparison. As is clear from the results in Table 16, the natural frequencies obtained by the FEM code are very close to those reported in the literature, and the very small differences are probably due to the rounding of the values of the design variables, as noted above. Note that in the case of this example, CLFA and CGFA are excluded from the comparison because the best designs obtained by these algorithms are highly infeasible, as also noted by Dede et al. [22]. The convergence histories of the best and mean results obtained by the SHADE algorithm for the 1410-bar truss are depicted in Fig. 8. It can be seen from the figure that the curves are very close together, demonstrating

Table 15 The Friedman rank test results for the 1410-bar dome-shaped truss

	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study
										SHADE
Friedman rank of best weight	10	9	8	5	3	7	4	6	2	1
Friedman rank of mean weight	9	10	5	7	4	6	8	3	2	1
Friedman rank of standard deviation	5	10	4	8	6	7	9	2	1	3

Table 16 Natural frequencies (Hz) of the 1410-bar dome-shaped truss evaluated at optimal designs

Frequency number	ECBO-Cascade [13]	DPSO [14]	MDVC-UVPS [15]	chaotic WSA [18]	ST-JA [19]	PFJA [20]	ISMA [23]	Rao-1 [22]	Rao-2 [22]	This study SHADE
1	7.002	7.001	7.000	7.0000	7.0002	7.0009	7.0001	7.0040	7.0006	**7.0001**
1*	**7.0048**	**7.0127**	**7.0027**	**6.9998**[1]	**7.0002**	**7.0125**	**7.0001**	**7.0040**	**7.0006**	**7.0001**
2	7.003	7.001	7.001	N/A	7.0002	7.0009	7.0001	7.0040	7.0006	**7.0001**
3	9.001	9.003	9.000	9.0021	9.0006	9.0001	9.0002	9.0013	9.0001	**9.0000**
3*	**9.0047**	**9.0105**	**9.0021**	**9.0021**	**9.0006**	**9.0083**	**9.0002**	**9.0013**	**9.0001**	**9.0000**
4	9.001	9.005	9.000	N/A	9.0021	9.0002	9.0002	9.0023	9.0021	**9.0003**
5	9.003	9.005	9.000	N/A	9.0021	9.0002	9.0005	9.0030	9.0021	**9.0003**

[1] The more precise value is 6.999762 Hz.

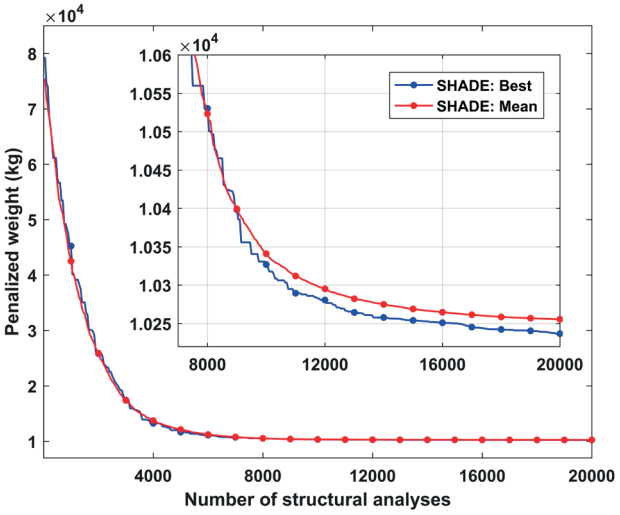

Fig. 8 The best and mean convergence histories of the SHADE algorithm for the 1410-bar dome-shaped truss

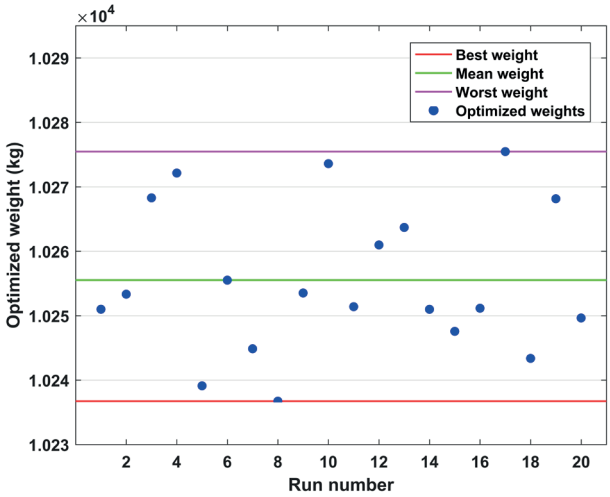

Fig. 9 Diversity of the optimized weights obtained by the SHADE algorithm for the 1410-bar dome-shaped truss

the high robustness of the SHADE algorithm. Fig. 9 shows the optimized weights obtained by the SHADE algorithm for the 1410-bar truss. The figure indicates that in 13 out of 20 runs, the optimized weights obtained by the SHADE algorithm are below the mean weight.

5 Conclusions

The success-history based adaptive differential evolution (SHADE) algorithm is an advanced variant of differential evolution (DE). The distinctive feature of the SHADE algorithm is the use of a historical memory of successful control parameter settings to guide the generation of the next control parameter values. SHADE has shown very promising performance in optimization of benchmark mathematical functions. However, it has been less used for real-world optimization problems. Owing to the highly nonlinear and non-convex constraints involved, structural optimization with frequency constraints is considered to be one of the most challenging fields in structural optimization. In the present work, in order to tackle this class of optimization problems, the SHADE algorithm is applied to solve the size optimization problem of large-scale dome truss structures with multiple frequency constraints. The design examples include a 600-bar single-layer dome truss, a 1180-bar single-layer dome truss, and a 1410-bar double-layer dome truss. The results obtained by the SHADE algorithm are presented and compared with the best-known results reported in the literature. Based on the preceding results and discussion, some concluding remarks can be drawn as follows:

1. The SHADE algorithm seems to have the most promising performance in terms of solution accuracy and computational efficiency.
2. In all the three cases considered, the optimal designs obtained by the SHADE algorithm are the best ones reported in the literature so far.

3. Numerical results reveal that in all of the test cases, the mean weight gained by the SHADE algorithm is lower than the best weights achieved by all the compared algorithms, which demonstrates the effectiveness and robustness of the SHADE algorithm in solving frequency-constrained structural optimization problems.
4. The number of structural analyses required by SHADE to reach a feasible design better than the optimal designs found by the compared algorithms is generally far less than those required by these algorithms to reach the optimal designs, which shows the computational efficiency of the SHADE algorithm.
5. From the free vibration analysis results, it is observed that the optimal designs obtained by the SHADE algorithm are fully feasible, indicating that the algorithm could exactly satisfy the frequency constraints.

We believe that the main reason for the excellent performance of the SHADE algorithm in this research is mainly due to some features of the DE/current-to-*p*best/1 mutation strategy, which make the algorithm particularly suitable for large-scale optimization problems with many design variables. For example, in SHADE, instead of using only the information of the single best solution, the information of a set of most promising solutions are utilized with the aim to balance the greediness of the mutation and the diversity of the population. This feature helps the algorithm to prevent premature convergence in high-dimensional search spaces.

Declaration of competing interest

The authors declare that they have no conflicts of interest associated with this publication and no financial support has been received for this work.

References

[1] Yuan, X., Peng, Z., Dong, S., Zhao, B. "A New Tensegrity Module - "Torus", Advances in Structural Engineering, 11(3), pp. 243–251, 2008.
https://doi.org/10.1260%2F136943308785082616

[2] Turner, M. J. "Design of minimum mass structures with specified natural frequencies", AIAA Journal, 5(3), pp. 406–412, 1967.
https://doi.org/10.2514/3.3994

[3] Kodiyalam, S., Graichen, C. M., Connell, I. J., Finnigan, P. M. "Object-oriented, optimization-based design of satellite structures", Journal of Spacecraft and Rockets, 31(2), pp. 312–318, 1994.
https://doi.org/10.2514/3.26439

[4] Grandhi, R. "Structural optimization with frequency constraints - A review", AIAA Journal, 31(12), pp. 2296–2303, 1993.
https://doi.org/10.2514/3.11928

[5] Grandhi, R.V., Venkayya, V. B. "Structural optimization with frequency constraints", AIAA Journal, 26(7), pp. 858–866, 1988.
https://doi.org/10.2514/3.9979

[6] Kiusalaas, J., Shaw, R. C. J. "An algorithm for optimal structural design with frequency constraints", International Journal of Numerical Methods in Engineering, 13(2), pp. 283–295, 1978.
https://doi.org/10.1002/nme.1620130206

[7] Levy, R., Chai, K. "Implementation of natural frequency analysis and optimality criterion design", Computers & Structures, 10(1-2), pp. 277–282, 1979.
https://doi.org/10.1016/0045-7949(79)90096-8

[8] Moosavian, H., Mesbahi, P., Moosavian, N., Daliri, H. "Optimal design of truss structures with frequency constraints: a comparative study of DE, IDE, LSHADE, and CMAES algorithms", Engineering with Computers, 2021.
https://doi.org/10.1007/s00366-021-01534-0

[9] Wei, L., Zhao, M., Wu, G., Meng, G. "Truss optimization on shape and sizing with frequency constraints based on genetic algorithm", Computational Mechanics, 35, pp. 361–368, 2005.
https://doi.org/10.1007/s00466-004-0623-8

[10] Bateni, S. M., Mortazavi-Naeini, M., Ataie-Ashtiani, B., Jeng, D. S., Khanbilvardi, R. "Evaluation of methods for estimating aquifer hydraulic parameters", Applied Soft Computing, 28, pp. 541–549, 2015.
https://doi.org/10.1016/j.asoc.2014.12.022

[11] Wei, L., Tang, T., Xie, X., Shen, W. "Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm", Structural and Multidisciplinary Optimization, 43, pp. 665–682, 2011.
https://doi.org/10.1007/s00158-010-0600-0

[12] Kaveh, A., Zolghadr, A. "Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability", Computers & Structures, 102–103, pp. 14–27, 2012.
https://doi.org/10.1016/j.compstruc.2012.03.016

[13] Kaveh, A., Ilchi Ghazaan, M. "Optimal design of dome truss structures with dynamic frequency constraints", Structural and Multidisciplinary Optimization, 53, pp. 605–621, 2016.
https://doi.org/10.1007/s00158-015-1357-2

[14] Kaveh, A. "Advances in metaheuristic algorithms for optimal design of structures", Springer, 2021. ISBN 978-3-319-35062-2
https://doi.org/10.1007/978-3-319-05549-7

[15] Kaveh, A., Ilchi Ghazaan, M. "A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures", Engineering Optimization, 50(2), pp. 235–252, 2018.
https://doi.org/10.1080/0305215X.2017.1313250

[16] Kaveh, A., Javadi, S. M. "Chaos-based firefly algorithms for optimization of cyclically large-size braced steel domes with multiple frequency constraints", Computers & Structures, 214, pp. 28–39, 2019.
https://doi.org/10.1016/j.compstruc.2019.01.006

[17] Kaveh, A., Biabani Hamedani, K., Kamalinejad, M."An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures", Computers & Structures, 256, 106643, 2021.
https://doi.org/10.1016/j.compstruc.2021.106643

[18] Kaveh, A., Amirsoleimani, P., Dadras Eslamlou, A., Rahmani, P. "Frequency-constrained optimization of large-scale dome-shaped trusses using chaotic water strider algorithm", Structures, 32, pp. 1604–1618, 2021.
https://doi.org/10.1016/j.istruc.2021.03.033

[19] Kaveh, A., Biabani Hamedani, K., Joudaki, A., Kamalinejad, M. "Optimal analysis for optimal design of cyclic symmetric structures subject to frequency constraints", Structures, 33, pp. 3122–3136, 2021.
https://doi.org/10.1016/j.istruc.2021.06.054

[20] Degertekin, S. O., Bayar, G. Y., Lamberti, L. "Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints", Computers & Structures, 245, 106461, 2021.
https://doi.org/10.1016/j.compstruc.2020.106461

[21] Kaveh, A., Ilchi Ghazaan, M., Saadatmand, F. "Colliding bodies optimization with Morlet wavelet mutation and quadratic interpolation for global optimization problems", Engineering with Computers, 38, pp. 2743–2767, 2022.
https://doi.org/10.1007/s00366-020-01236-z

[22] Dede, T., Atmaca, B., Grzywinski, M., Rao, R. V. "Optimal design of dome structures with recently developed algorithm: Rao series", Structures, 42, pp. 65–79, 2022.
https://doi.org/10.1016/j.istruc.2022.06.010

[23] Kaveh, A., Hamedani, K. B., Kamalinejad, M. "Improved slime mould algorithm with elitist strategy and its application to structural optimization with natural frequency constraints", Computers & Structures, 264, 106760, 2022.
https://doi.org/10.1016/j.compstruc.2022.106760

[24] Storn, R., Price, K. "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, 11, pp. 341–359, 1997.
https://doi.org/10.1023/A:1008202821328

[25] Wu, G., Shen, X., Li, H., Chen, H., Lin, A., Suganthan, P. N. "Ensemble of differential evolution variants", Information Sciences, 423, pp. 172–186, 2018.
https://doi.org/10.1016/j.ins.2017.09.053

[26] Bilal, Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A. "Differential Evolution: A review of more than two decades of research", Engineering Applications of Artificial Intelligence, 90, 103479, 2020.
https://doi.org/10.1016/j.engappai.2020.103479

[27] Das, S., Suganthan, P. N. "Differential evolution: A survey of the state-of-the-art", IEEE Transactions on Evolutionary Computation, 15(1), pp. 4–31, 2010.
https://doi.org/10.1109/TEVC.2010.2059031

[28] Tanabe, R., Fukunaga, A. "Success-history based parameter adaptation for differential evolution", In: 2013 IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, Mexico, 2013, pp. 71–78. ISBN:978-1-4799-0453-2
https://doi.org/10.1109/CEC.2013.6557555

[29] Yuan, X., Zhang, Y., Wang, L., Yuan, Y. "An enhanced differential evolution algorithm for daily optimal hydro generation scheduling", Computers & Mathematics with Applications, 55(11), pp. 2458–2468, 2008.
https://doi.org/10.1016/j.camwa.2007.08.040

[30] Qin, A. K., Huang, V. L., Suganthan, P. N. "Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE Transactions on Evolutionary Computation, 13(2), pp. 398–417, 2008.
https://doi.org/10.1109/TEVC.2008.927706

[31] Zhang, J., Sanderson, A. C. "JADE: adaptive differential evolution with optional external archive", IEEE Transactions on Evolutionary Computation, 13(5), pp. 945–958, 2009.
https://doi.org/10.1109/TEVC.2009.2014613

[32] Mallipeddi, R., Suganthan, P. N., Pan, Q. K., Tasgetiren, M. F. "Differential evolution algorithm with ensemble of parameters and mutation strategies", Applied Soft Computing, 11(2), pp. 1679–1696, 2011.
https://doi.org/10.1016/j.asoc.2010.04.024

[33] Tanabe, R., Fukunaga, A. S. "Improving the search performance of SHADE using linear population size reduction", In: 2014 IEEE Congress on Evolutionary Computation (CEC 2014), Beijing, China, 2014, pp. 1658–1665. ISBN:978-1-4799-1488-3
https://doi.org/10.1109/CEC.2014.6900380

[34] Tanabe, R., Fukunaga, A. "Evaluating the performance of SHADE on CEC 2013 benchmark problems", In: 2013 IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, Mexico, 2013, pp. 1952–1959. ISBN:978-1-4799-0453-2
https://doi.org/10.1109/CEC.2013.6557798

[35] Ergezer, M., Simon, D. "Probabilistic properties of fitness-based quasi-reflection in evolutionary algorithms", Computers & Operations Research, 63, pp. 114–124, 2015.
https://doi.org/10.1016/j.cor.2015.03.013

[36] Biswas, P. P., Suganthan, P. N., Amaratunga, G. A. J. "Optimal power flow solutions using algorithm success history based adaptive differential evolution with linear population reduction", In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018), Miyazaki, Japan, 2018, pp. 249–254. ISBN:978-1-5386-6650-0
https://doi.org/10.1016/j.enconman.2017.06.071

[37] Narloch, P. H., Dorn, M. "Evaluating the Success-History Based Adaptive Differential Evolution in the Protein Structure Prediction Problem", In: Applications of Evolutionary Computation 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Seville, Spain, 2021, pp. 194–209. ISBN 978-3-030-72698-0
https://doi.org/10.1007/978-3-030-72699-7_13

[38] Pholdee, N., Bureerat, S. "A comparative study of eighteen self-adaptive metaheuristic algorithms for truss sizing optimisation", KSCE Journal of Civil Engineering, 22, pp. 2982–2993, 2018.
https://doi.org/10.1007/s12205-017-0095-y

[39] Onnen, C., Babuška, R., Kaymak, U., Sousa, J. M., Verbruggen, H. B., Isermann, R. "Genetic algorithms for optimization in predictive control", Control Engineering Practice, 5(10), pp. 1363–1372, 1997.
https://doi.org/10.1016/S0967-0661(97)00133-0

[40] Kaveh, A., Zolghadr, A. "Democratic PSO for truss layout and size optimization with frequency constraints", Computers & Structures, 130, pp. 10–21, 2014.
https://doi.org/10.1016/j.compstruc.2013.09.002

[41] Kaveh, A., Hamedani, K. B. "Improved arithmetic optimization algorithm and its application to discrete structural optimization", Structures, 35, pp. 748–764, 2022.
https://doi.org/10.1016/j.istruc.2021.11.012

[42] Jordehi, A. R. "A review on constraint handling strategies in particle swarm optimisation", Neural Computing and Applications, 26, pp. 1265–1275, 2015.
https://doi.org/10.1007/s00521-014-1808-5

[43] Kaveh, A., Biabani Hamedani, K. "Discrete Structural Optimization with Set-Theoretical Jaya Algorithm", Iranian Journal of Science and Technology, Transactions of Civil Engineering, 2022.
https://doi.org/10.1007/s40996-022-00868-z

[44] Chopra, A. K. "Dynamics of structures", Pearson Education India, 2007. ISBN 8131713296