Scientia Iranica A (2022) 29(6), 2868–2885

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
http://scientiairanica.sharif.edu

Chaos-embedded meta-heuristic algorithms for optimal design of truss structures

H. Yosefpoor[a] and A. Kaveh[b,*]

a. *Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*
b. *Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 1684613114, Iran.*

Received 22 January 2022; received in revised form 18 February 2022; accepted 8 May 2022

KEYWORDS
Chaotic map;
Cross-section optimization;
Truss structures;
Meta-heuristic algorithms.

Abstract. The success of chaos-embedded meta-heuristic algorithms mainly results from the satisfactory tradeoff between exploration and exploitation phases in meta-heuristics. Comparison of optimization results with algorithms in standard mode and embedded chaos indicates the significant improvement in the quality of the meta-heuristic algorithms, thus reducing the weight of truss structures. Four chaos meta-heuristic algorithms with logistic, Tenet, and Gaussian maps are considered to improve the results. Despite truss optimization being extremely nonlinear and non-convex and often having several local optima issues, the use of different chaotic maps allows avoiding local optima and achieving global optimum.

1. Introduction

Over the last three decades, attempts have been made to optimally design many different types of structures, given the limitation of available resources and the need to reduce the number of used materials. To this end, classical methods including numerical and mathematical methods have been proposed. However, each of these solutions is subject to its own limitations and difficulties. For example, in numerical methods, upon reaching the local optimum, the computational operation was stopped and no solution was provided to escape this local optimum. Furthermore, in analytical and mathematical methods, access to the objective function derivatives is inevitable, while most engineering problems cannot provide us with an explicit relation of the objective function. Today, with the complexity of issues and the increase in the number of decision variables, the lack of any response from classical methods has become evident. Therefore, to overcome these challenges, a particular category of optimization methods called meta-heuristic methods, which are based on decisions and principles of probabilistic and random search, has been developed. In these algorithms, the value of the objective function itself is used instead of derivatives and is of good efficiency in dealing with complex problems [1]. The source of inspiration in these algorithms can play an important role in their classification. Some of these sources of inspiration are the algorithms on the basis of the following: physical laws, swarm intelligence, evolution, behavior, environment, and social and human laws. For each of these groups, we have introduced the following optimization algorithms: Water Evaporation Optimization (WEO) [2], Charged System Search (CSS) [3], Colliding Bodies Optimization (CBO) [4,5], Vibrating Particles System (VPS) [6], Thermal Exchange Optimization (TEO) [7], Big Bang-Big Crunch (BB-BC) [8], Ray Optimization (RO) [9], and Harmony Search (HS) [10], which are inspired by physical laws. Cyclical Parthenogenesis Algorithms

*. *Corresponding author.*
E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

doi: 10.24200/sci.2022.59812.6441

(CPA) [11], Lion Pride Optimization (LPO) [12], Artificial Coronary Circulation System (ACCS) [13], Ant Colony Optimization (ACO) [14], Whale Optimization Algorithm (WOA) [15], Gray Wolf Optimization (GWO) [16], and Particle Swarm Optimization (PSO) [17,18] are inspired by swarming intelligence. Genetic Algorithms (GA) [19], Differential Evolution (DE) [20], and Evolutionary Strategy (ES) [21] are inspired by evolution. Shuffled Frog-Leaping Algorithm (SFLA) [22], Biogeography-Based Optimization (BBO) [23], Teaching-Learning-Based Optimization (TLBO) [24], and Imperialist Competitive Algorithm (ICA) [25] are also part of the memetic algorithms based on the environment and social and human laws, respectively. In each of these algorithms, some random numbers are considered, which we use in an alternative chaos system to fix them [26]. Mathematically, chaos represents the ability of a simple pattern and model; although this pattern has no sign of random phenomena, it can lead to the emergence of very disordered behaviors in the environment. Today, these dynamic systems have attracted the attention of many scientific communities and are observed in various fields such as engineering, medicine, biology, and economics [27]. The salient features of a chaotic system are as follows:

1. They are sensitive to initial conditions;
2. Their alternating rotation is dense;
3. They have quasi-random and non-periodic behavior [28].

The term butterfly effect is derived from a paper by Edward Lorenz. At the 99th Water Summit, he wrote an article entitled: "Can a butterfly flying in Brazil cause strong winds in Texas?" [27]. Experimental studies show that the use of chaotic signals instead of random signals has yielded very valuable results. In this research, chaotic maps in optimization algorithms including CPA, TLBO, BBO, and DE are investigated and in most cases, the optimization results for the algorithms are enhanced. In order to access a wide range of statistical data and increase the diversity of studies, in each modeling, three chaotic maps with three different scenarios have been considered.

Rapid convergence, stability, and robustness of responses are the main reasons for choosing these algorithms. Reducing the number of iterations and improving the values of standard deviation and coefficient of variation, compared to previous research, can confirm the superiority of these four algorithms.

2. Formulation of optimization problems

Optimization problems for trusses are generally defined in Eq. (1) such that all design limitations must be satisfied and the total weight of the structure must be as low as possible:

$$\begin{array}{ll} \text{Find} & A=\{A_1,A_2,...,A_n\} \\ \text{to minimize} & W(A)=\sum\limits_{i=1}^{n}\gamma_i A_i L_i \\ \text{subjected to} & g_j(A)\le 0; \quad j=1,2,...,m \\ & h_k(A)\le 0; \quad k=1,2,...,p \\ & \{A_L\}\le\{A\}\le\{A_U\} \end{array} \tag{1}$$

In this category of Eq. (1), A is the cross-section vector of the members, W the total weight of the structure, n the number of members of the structure, g_i and h_k are design constraints that can include stress, slender constraints, and nodal displacement. Moreover, A_L and A_U are the lower and upper bounds of the decision variables. However, the selected algorithms are used to optimize unbounded problems. In modeling, the penalty function method is used to convert the bound function into the unbound function. In this method, in case of no violation, the amount of the fine will be zero. Otherwise, in the event of any violation, the value of the penalty function is determined using Eqs. (2) to (6):

$$\sigma_i\le\sigma_{\max}\Rightarrow V_i=\max\left(0,\frac{\sigma_i}{\sigma_{\max}}-1\right); \quad i=1,2,\cdots,m, \tag{2}$$

$$\delta_j\le\delta_{\max}\Rightarrow V_j=\max\left(0,\frac{\delta_j}{\delta_{\max}}-1\right); \quad j=1,2,\cdots,n, \tag{3}$$

$$\lambda_k\le\lambda_{\max}\Rightarrow V_k=\max\left(0,\frac{\lambda_k}{\lambda_{\max}}-1\right); \quad k=1,2,\cdots,p, \tag{4}$$

$$F_{penalty}(A)=1+\eta\times\left\{\sum_{i=1}^{m}V_i+\sum_{j=1}^{n}V_j+\sum_{k=1}^{p}V_k\right\}, \tag{5}$$

$$\text{to minimize} \quad Mer(A)=W(A)\times F_{penalty}(A). \tag{6}$$

Eqs. (2), (3), and (4) are related to stress, displacement, and slenderness, respectively. The penalty function is presented in Eq. (5) and the objective function is formed after the penalty (merit function) in Eq. (6).

3. Meta-heuristic algorithms and chaotic map

In each meta-heuristic algorithm, a number of random parameters are selected during the sequential iteration process. Research shows that the type of random distribution function for selecting these parameters has a significant impact on the performance of the algorithms. These parameters play a major role in increasing the speed of convergence and creating a balance between the exploration and exploitation stages. Nowadays, chaotic dynamics series are a good alternative to these stochastic parameters. Chaos systems can provide convergence to global responses and have the potential to prevent them from falling into local optima trap. Although the series created by chaotic map appear to be similar to sentences with random distributions, there are major differences. To be specific, their values are deterministic, have nonlinear behavior, have a dynamic state, and not limited to a particular boundary and, also, the sentences of the series related to them are non-repetitive. By applying chaotic maps, the nonlinear and non-convex behaviors of the object function in truss optimization are easily controlled and adjusted by chaotic maps. Three different scenarios have been proposed to apply the chaotic maps and improve the optimization results. In the first scenario, the chaotic maps replace the random parameters related to the exploration part; for the second scenario, these functions replace the parameters related to the exploitation part, and finally, for the third scenario, the simultaneous application of chaotic maps in both exploration and exploitation parts is considered. By examining structural examples, this study determines the best chaotic map and the similar scenario for each of the algorithms. In a recent study that aimed to investigate the role of chaotic map in improving the results of algorithm optimization, four meta-heuristic algorithms with three chaotic maps are combined and the results are compared with the standard modes. Due to the wide range of studies and diversity, there is a close competition between the optimization results and the possibility of approaching the absolute optimal increases. In this research, the selected meta-heuristic algorithms are CPA, TLBO, BBO, and DE. Chaotic maps to be combined with these algorithms include logistic, Tent, and Gaussian chaotic maps. In the following, each of these algorithms is introduced.

3.1. Standard CPA

This algorithm was proposed by Kaveh and Zolghadr [11]. It is inspired by social behavior and wildlife reproduction of certain insects such as aphids, which can be reproduced with and without mating. In addition, like most meta-heuristic algorithms, it is population based. In the following, CPA algorithm is organized in the five following steps.

3.1.1. Basic steps of CPA

Step 1. Formation of the initial population: In the search space, the population of aphid is formed randomly. This number of aphid populations, denoted by nA, is formed in colonies with nC number and each with nM population. It is clear that $nM = nA/nC$ and nM are constant during optimization.

Step 2. Reproduction of aphids: To form the population of offspring in each colony, the number of $Fr \times nM$ offspring is formed without mating. The parents of these children are female and their selection is done randomly and from the best answers. In MATLAB coding, this selection and formation of the children population is as follows:

$$rf_i = round(1 + (Fr \times nM - 1) \cdot *\mathbf{rand}), \tag{7}$$

$$newCA = F + \alpha_1 \times \frac{randn}{NITs} \times (Ub - Lb). \tag{8}$$

Through Eq. (7), the index related to the female parent is determined, while through Eq. (8), the new children aphids related to the state without mating are placed in the new cell array. Now, it is time that offspring be formed by mating. The number of these offspring is $(1 - Fr) \times nM$. In this combination, each male parent M randomly selects a female parent F. The children of this generation, which were obtained by performing the act of mating state, according to Eq. (9) in the arrays new cells are placed:

$$newCA = M + \alpha_2 \times \mathbf{rand} \times (F - M). \tag{9}$$

Step 3. Flying of the best aphids and death of the worst aphids: After the formation of a new generation of offspring, the objective function is evaluated; there is a possibility that Pf (percentage of possible flight) one of the best winged aphids can be selected from Colony 1 and then, the worst aphids be replaced by them in Colony 2. To keep the colony population stable, removal of the worst aphids in Colony 2 is compared to the death of the aphids and replacement of the best flying ones. The probability associated with this step is based on Eq. (10):

$$Pf = \frac{(NITs - 1)}{(\max NITs - 1)}. \tag{10}$$

Step 4. Replacement of the best aphids: In each colony, the population of parents is compared with the children and from them, a number of nM to the best is selected to form the next generation.

Step 5. Checking of the termination conditions and repeating of the operation from Step 2, if necessary.

3.1.2. Chaos embedded Cyclical Parthenogenesis Algorithm (CCPA)

In this algorithm, two modes with and without mating are selected to form the offspring of the new generation. These two modes play the roles of exploration and exploitation. By substituting the chaotic map for random choices in these two steps, a variety of scenarios are proposed. In this research, the following scenarios have been investigated:

Scenario 1. Placement of the chaotic map in stages without mating stage: In this case, the first chaotic map **CHM1** in Eq. (7) replaces random selection, and the result is given in the form of Eq. (11):

$$rf_i = round\ (1 + (Fr \times nM) \cdot *\mathbf{CHM1}). \tag{11}$$

Scenario 2. Placement of the chaotic map in the mating stage: In this case, the second chaotic map **CHM2** in Eq. (9) replaces random selection, and the result is given in the form of Eq. (12):

$$newCA = M + \alpha_2 \times \mathbf{CHM2} \times (F - M). \tag{12}$$

Scenario 3. Placement of the chaotic map in both stages simultaneously: In this case, the two chaotic maps simultaneously in Eqs. (7) and (9) replace the random distribution.

3.2. Standard TLBO

The TLBO was introduced by Rao et al. in 2011 and it is based on the classroom learning process [24]. The algorithm consists of two stages: the effect of the teacher in the learning process and the students' influence on one another. Similar to other population-based algorithms, it starts with a series of random initial solutions, which are the same students, and the best and smartest of them in each period of repetition is known as the teacher. The academic level of students is enhanced in each repetition, phase in phase. The first phase is based on the transfer of knowledge from the teacher, also known as the teacher phase, and the second phase is the students' learning from each other or their interactions with one another, also known as the student phase. In the following, the TLBO algorithm is organized in four steps.

3.2.1. Basic steps in standard TLBO

Step 1. Introduce the parameters of the algorithm and form the initial student population: The basic parameters include the number of learners nL, the number of decision variables nV, and the maximum Number of Function Evaluations (NFEs), being the same parameter for the stopping criterion. Now, the initial population is randomly formed based on the search space and their estimation is performed in the objective function.

Step 2. Select the best of them as teacher T out of the introduced population. Then, the average position of students is calculated and based on the teacher phase, the improved academic level of students is determined using Eq. (13):

$$stepsize_i = T - TF_i \times MeanL \tag{13}$$

$$newL = L + \mathbf{rand}_{i,j} \times stepsize$$

$$i = 1, 2, ..., nL \text{ and } j = 1, 2, ..., nV. \tag{14}$$

In Eq. (13), the Teaching Factor (TF) is randomly selected as 1 or 2, indicating how successful the teacher has been in increasing the average academic level of students. The obtained values are evaluated; if they are better than the previous values, they are replaced.

Step 3. At this stage, the mutual effects of students on each other are considered. At this phase, each student exchanges information with another randomly selected student (except himself). It is possible to improve the information when the performance of the selected student is better; otherwise, it varies according to their position.

$$if \quad PFit_i < PFit_{rp} \Rightarrow stepsize_i = L_i - L_{rp}, \tag{15}$$

$$if \quad PFit_i > PFit_{rp} \Rightarrow stepsize_i = L_{rp} - L_i, \tag{16}$$

$$newL = L + \mathbf{rand_{i,j}} \times stepsize$$

$$i = 1, 2, \cdots, nL \quad \text{and} \quad j = 1, 2, \cdots, nV. \tag{17}$$

The obtained values are evaluated and if they are better than the previous values, they are replaced. The best of the populations are introduced in each stage.

Step 4. Check the termination conditions and repeat the operation from Step 2, if necessary.

3.2.2. Chaos-embedded Teaching-Learning-Based Optimization (CTLBO)

This algorithm consists of two important strategies, including the teacher's effect on the learning process and students' influence on one another. This plays a major role in the exploration and exploitation processes. By replacing the chaotic map in random selections in these steps, we will see a significant improvement in the performance of the algorithm. The proposed scenarios for this replacement are as follows:

Scenario 1. Placement of the chaotic map in the teacher effect stage of the learning process: In this case, the first chaotic function **CHM1** is replaced in Eq. (14) and the result is given in the form of Eq. (18):

$$newL = L + \mathbf{CHM1_{i,j}} \times stepsize. \tag{18}$$

Scenario 2. Placement of the chaotic map in the stage of students interacting with each other: In this case, the second chaotic map **CHM2** is replaced in Eq. (17) and the result is given in the form of Eq. (19):

$$newL = L + \mathbf{CHM2_{i,j}} \times stepsize. \tag{19}$$

Scenario 3. Placement of the chaotic map in both stages simultaneously: In this case, the two chaotic functions are simultaneously replaced in Eqs. (14) and (17).

3.3. Standard BBO

This algorithm was introduced in 2008 by Dan Simon [23]. The main inspiration of the algorithm is based on the distribution of plants, especially animals, in different geographical areas. In principle, animal and plant species try to use resources exclusively, but this will not be possible in most cases. Therefore, they need to have food, water, etc. In this context, an ecosystem forms a food chain by which other species can be fed. However, the tendency to monopolize has other consequences which, in turn, drive animals to more secluded places. Therefore, a settlement is known as a suitable place holding a smaller population. Therefore, animal species usually choose a more secluded place among several options. The higher concentration of animal population can be found in places featuring richer food resources. Therefore, the HIS habitat suitable index is introduced as a factor that helps determine the better location. In optimization problems, this coefficient is the same as the objective function of the problem. In minimization problems, the lower the coefficient value, the better. However, the habitats for which this coefficient is higher are, in fact, very large and because of the competitors, the species want to leave. Therefore, two very common terms of migration are considered: migration to habitat called 'Immigration', which expresses acceptance of immigration representing a percentage by λ, and migration from habitat called emigration, whose percentage state is expressed by μ. Habitat location is denoted by SIV indicating suitability index variable or decision variables in the selected space. In this level migration from habitats that have more population and their emigration coefficient μ is high towards habitats with high immigration coefficient λ takes place. Further, in order to add a variety in the search space, mutations are applied to the components of the variables with a certain probability. The basic steps of this algorithm in the standard mode are presented in the following.

3.3.1. Basic steps of BBO

Step 1. Formation of primary habitats: In the search space, a set of primary habitats is randomly formed and sorted based on the objective function. At this stage, we access the initial population. The number of this initial population is represented by npop.

Step 2. The values of the dimensionless coefficients λ and the migration μ are determined for the habitats.

Step 3. For each selected habitat location such as i, Steps 4 to 8 are repeated for the initial population.

Step 4. For each variable such as r in habitat i, Steps 5 to 8 are repeated for a number of array variables.

Step 5. With the probability λ_i in the variable X_{ir}, changes are made to the line with Steps 6 to 8.

Step 6. The origin of migration X_{jr} is determined using the values of μ vectors based on random selection in a discrete distribution (roulette wheel).

Step 7. The location of the new habitat is determined by migrating from X_{jr} to X_{ir} as follows:

$$if \quad \textbf{rand} \leq lambda(i) \Rightarrow$$

$$X_{ir}^{new} = X_{ir} + \alpha_k \cdot \times (X_{jr} - X_{ir}), \tag{20}$$

$$\alpha_k = 0.9. \tag{21}$$

In the original version, an alpha value of 0.9 was suggested.

Step 8. With a certain probability, jump changes with a specific random distribution (preferably 'normal distribution') are performed on the selected variable X_{ir}:

$$if \quad \textbf{rand} \leq pmutation \Rightarrow X_{ir}^{mut}$$

$$= X_{ir}^{new} + sigma \times randn, \tag{22}$$

$$sigma = 0.02 \times (Var\max - Var\min). \tag{23}$$

Sigma comprises a percentage of the decision space. In the original version, this value is 2%.

Step 9. Migrate the answers from the previous answers and after evaluating and sorting, select a number of npops from the best for the next step.

Step 10. Check the termination conditions and repeat the operation from Step 3, if necessary.

3.3.2. Chaos-embedded Biogeography-Based Optimization (CBBO)

In this algorithm, the location of the new habitat is based on two important strategies. In the first strategy, migration from X_{jr} to X_{ir} takes place for each location such as i and for each variable such as K in location i with a probability of λ corresponding to i. In the second strategy, random mutation changes are applied to the variables of the decision variable. These two play the same roles in the exploration and exploitation processes. Therefore, by replacing the chaotic map in random choices related to these steps, we will see a significant improvement in the performance of the algorithm. The proposed scenarios for this replacement are as follows:

Scenario 1. Placing the chaotic map in the first stage of the migration strategy of variables: In this case, the first chaotic map **CHM1** is replaced in Eq. (20), and the result is given in the form of Eq. (24):

$$if \quad \mathbf{CHM1} \leq lambda(i) \Rightarrow$$

$$X_{ir}^{new} = X_{ir} + \alpha_k \cdot \times (X_{jr} - X_{ir}). \tag{24}$$

Scenario 2. Placing the chaotic map in the second strategy phase of random mutation changes: In this case, the second chaotic map **CHM2** is applied to Eq. (22), and the result is given in the form of Eq. (25):

$$if \quad \mathbf{CHM2} \leq pmutation \Rightarrow$$

$$X_{ir}^{mut} = X_{ir}^{new} + sigma \times randn. \tag{25}$$

Scenario 3. Placing the chaotic map in both stages simultaneously: In this case, the two chaotic maps are replaced simultaneously in Eqs. (20) and (22).

3.4. Standard DE

This algorithm was proposed by Price and Storn [20]. DE algorithm is a random and population-based algorithm and is considered as an evolutionary one. In this algorithm, the method of generating new answers is unique. The main difference between DE algorithms and other evolutionary algorithms lies in the formation of the offspring population. Also, considering how to apply the mutation and select the step length of the mutation according to a specific possible distribution is of value. In contrast, in the DE algorithm, a temporary response is generated first with the mutation operator and then, a new response is generated using the crossover operator. In this algorithm, a known probabilistic distribution is not sampled for the mutation operation, but the distance and difference between the available responses are used for the mutation steps. In fact, the differences between members of the population contain very useful and valuable information about the objective function and the problem to be solved, the use of which can improve the search process and optimization action. In an optimization algorithm, the responses in the population often converge to a single response over time and after a series of iterations. Therefore, the distance between the population members becomes smaller. In addition, the size of the distance among the initial population members is significantly affected by their number. Therefore, the larger the number of population members, the smaller the distance between them, and vice versa. The distance between the members contains important information about their distribution. Accordingly, based on the distance between the members of the population, we can determine the length of the step and adjust the intensity of the mutation. Now, if this distance is greater, we should be able to reach the ideal point faster by choosing larger steps; conversely, for smaller distances, smaller steps facilitate searching for the ideal location with appropriate accuracy. Thus, in this algorithm, the step length and direction of the mutation are determined as follows: First, two members of the population are randomly selected. Then, the difference between the two selected members is calculated and introduced as the difference vector. Now, we consider the coefficient of the difference vector as the mutation vector. The use of difference vectors is characterized by a number of advantages, the major one being the use of information available to the members of the population, thus intelligently orienting the optimization and search process. According to the central limit theorem, as the population increases, the length of the jump steps will tend to a random quantity with a normal distribution. To perform the intersection of the proposed relations, a number of arrays are separately selected from both the initial and mutant populations, as shown in Figure 1.

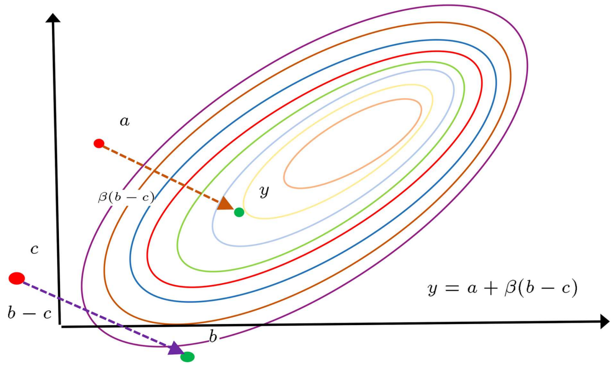

Figure 1. Mutation process in the differential evolution algorithm.

$$\begin{cases} X = \{x_1, x_2, ..., x_{nvar}\} \\ Y = \{y_1, y_2, ..., y_{nvar}\} \Rightarrow \quad Z = \{z_1, z_2, ..., z_{nvar}\} \\ \alpha = \beta \times (b - c) \end{cases} \tag{26}$$

$$Z_j = \begin{cases} y_i & \text{if} \quad j \in J \\ x_i & \text{if} \quad j \notin J \end{cases} \tag{27}$$

Moreover, to form the set J, first, we randomly select J_0 from the number 1 to *nvar*; then, we add the number J_0 to the set J, which is initially empty. Now, at all values up to 1 *nvar*, we repeat the following operations: We generate a random number like r_j that has a uniform distribution in the range of zeros and ones:

$$Z_j = \begin{cases} y_i & r \leq P_{cr} \text{ or } j \in j_0 \\ x_i & \text{otherwise} \end{cases} \tag{28}$$

If r is less than or equal to Pcr (intersection probability parameter), the number j is added to the set J. The basic steps in this algorithm in the standard mode are discussed as follows.

3.4.1. Basic steps in DE

Step 1. Define the algorithm parameters including number of *nvar* decision variables, top and bottom boundaries of decision variables, maximum number of iterations of the objective function, number of initial population npop, and upper and lower limits of *betamin* and *betamax* scale factor, which were 0.2 and 0.8, respectively, in the initial version. Finally, the probability of the percentage of *Pcr* crossover process are among the items that are determined at this stage.

Step 2. Form the initial population: Like most population-based algorithms, the initial population is formed in the search space randomly.

Step 3. Repeat the following steps until the termination conditions are met: The following steps are taken for each member of the population: A temporary response is generated using the jump operator. β is calculated as follows:

$$Beta = bet\_\mathrm{min} + \mathbf{rand}(\mathrm{var\ size})\cdot \times (beta\_\mathrm{max} - beta\_\mathrm{min}). \tag{29}$$

Using the intersection operator, a new response is created and evaluated based on the objective function. Now, if the new answer is better than the current answer, it will replace the current answer; otherwise, the current answer will be retained.

$$\text{if} \quad j = j_0 \quad \text{or} \quad \mathbf{rand} \leq Pcr$$

$$Z(j) = Y(j)$$

$$\text{Else}$$

$$Z(j) = X(j). \tag{30}$$

The best answer so far is obtained as output.

3.4.2. Chaos-embedded Differential Evolution (CDE)

This algorithm, like other evolutionary algorithms, includes two important stages of cross-over and mutation. From a general point of view, the main difference between the DE algorithms is the order of these two stages, in which the mutation takes place first followed by the cross-over. It also exploits the difference between responses as much as possible to achieve convergence and escape local optimum. It is clear that mutation and cross-over play the role of exploration and exploitation stages, respectively. Therefore, by replacing the chaotic functions by random choices related to these steps, we will see a significant improvement in the performance of the algorithm (Figure 2). The proposed scenarios for this replacement are as follows:

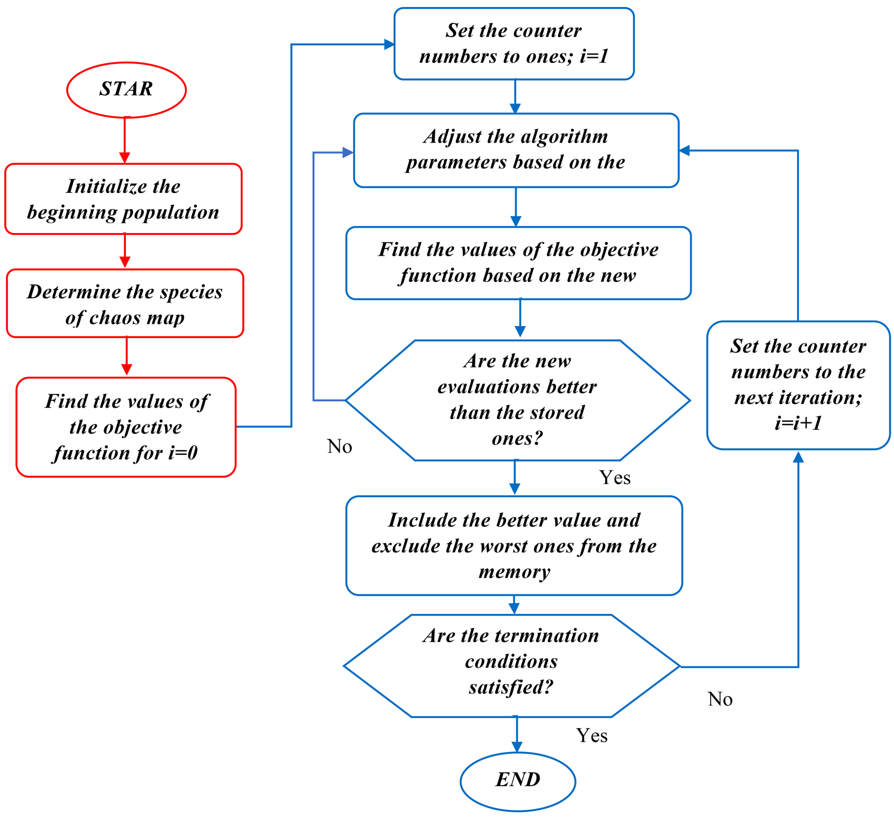

Figure 2. The flowchart of the chaos algorithm.

Scenario 1. Placing the chaotic map in the step of the mutation operator: In this case, the first turbulence function **CHM1** is substituted into Eq. (29), and the result is given in the form of Eq. (31):

$$Beta = bet\_\min + \mathbf{CHM1} \cdot \times (beta\_\max - beta\_\min). \tag{31}$$

Scenario 2. Placing the chaotic map in the cross-over operator stage: In this case, the second chaotic map **CHM2** is substituted into Eq. (30), and the result is given in the form of Eq. (32):

$$\text{if} \quad j = j_0 \quad \text{or} \quad \mathbf{CHM2} \leq Pcr$$

$$Z(j) = Y(j)$$

$$\text{Else}$$

$$Z(j) = X(j). \tag{32}$$

Scenario 3. Placing the chaotic map in both stages simultaneously: In this case, the two chaotic maps are substituted simultaneously into Eqs. (29) and (30).

4. Introduction of selected chaotic maps

Chaotic maps have shown no signs of random behavior, yet they cause very erratic behavior in the environment. The most important features of these maps are sensitivity to initial conditions and non-periodic and ergodic behavior [28]. The use of chaos sequences to evolve variables has significant advantages over other methods. In deterministic searches, compared to random search, higher speed and convergence to the general answer are achieved. In this research, Logistic, Tenet, and Gauss maps have been selected, as shown in Figure 3. In a Logistic map, it is very probable to converge from a local to global minimum. Therefore, this map is suitable for improving the exploration specifications of the algorithm. The Gaussian map is the most probable one in the local minimum range and is suitable for improving exploitation attribute. Finally, Tenet map simultaneously improves both conditions.

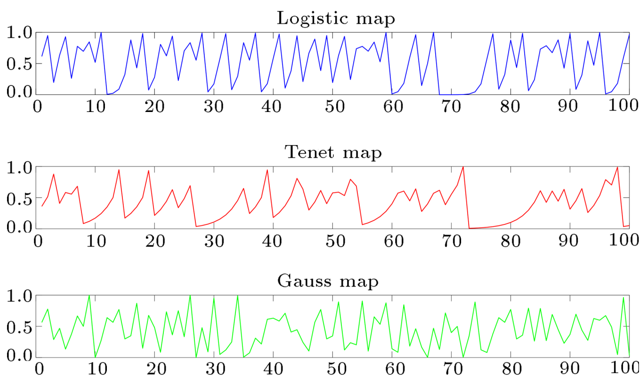

Figure 3. The chaotic value distribution during 100 iterations.

Therefore, the weakness of algorithms of any kind is overcome by selecting these maps.

4.1. Logistics map

This map exhibits nonlinear dynamic behaviors associated with biological populations. The sentences of chaotic sequences in the logistic map are obtained according to the following equations:

$$CHM_{k+1} = a \times CHM_k(1 - CHM_k), \tag{33}$$

$$CHM_{k+1} \in (0,1); CHM_0 \in (0,1);$$

$$CHM_0 \notin (0, 0.25, 0.5, 0.75, 1). \tag{34}$$

In the performed studies, $a = 4$ has been considered.

4.2. Tent map

This map is similar to logistics in some respects and depicts very specific effects of chaotic behavior. The chaotic sequence sentences in this function are expressed by the following equation:

$$CHM_{k+1} = \begin{cases} \frac{CHM_k}{0.7} & CHM_k < 0.7 \\ \frac{10}{3} \times CHM_k(1 - CHM_k) & CHM_k \geq 0.7 \end{cases} \tag{35}$$

4.3. Gauss map

The following equation shows the sentences of chaotic sequences in the Gauss function:

$$CHM_{k+1} = \begin{cases} 0 & CHM_k = 0 \\ \frac{1}{CHM_k} - \left[\frac{1}{CHM_k}\right] & CHM_k \neq 0 \end{cases} \tag{36}$$

5. Numerical examples of optimal truss design

In this research, the purpose of the optimal design of trusses is to find the optimal value for the cross sections, which has less weight and at the same time, the restrictions related to the permissible stress, the permissible deformation of the nodes and the slenderness of the members according to the regulation of compliance be made. Popular optimization examples have been selected in this regard and the efficiencies of algorithms, chaotic map, and proposed scenarios have been investigated.

5.1. A 47-bar power transmission tower

The topology and nodal numbering of the power transmission tower that consists of 47 members and 22 nodes are shown in Figure 4. The density of material is 0.3 lb/in^3 and the elasticity modulus for the members is 30.000 ksi. Both stress and buckling limits must be satisfied for all members. The allowable stress for tensile and compressive members is 20 ksi and 15 ksi, respectively. Euler buckling compressive

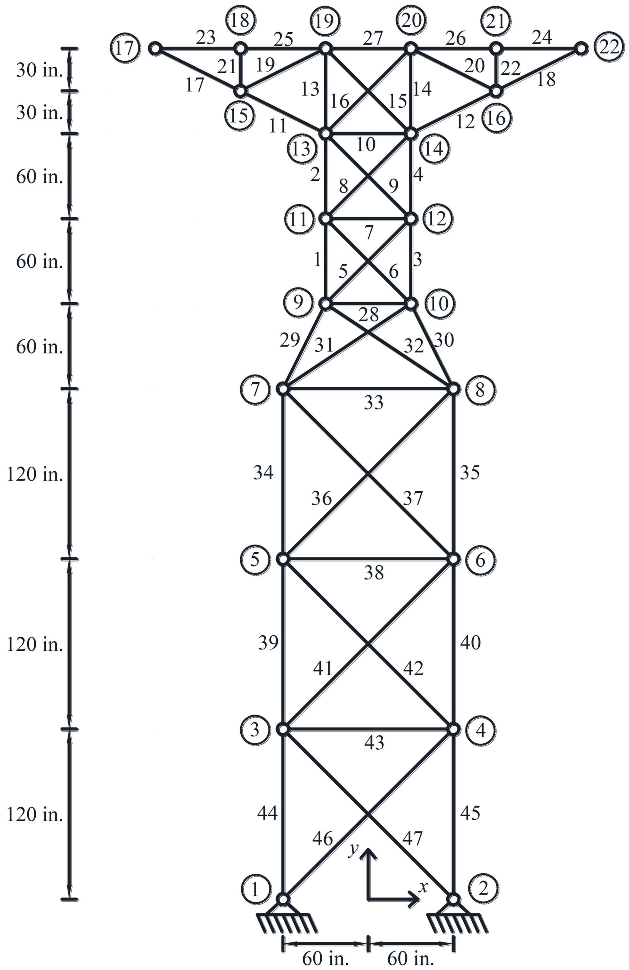

Figure 4. Schematic of the 47-bar power transmission tower.

stress for members is also calculated using the following equation:

$$\sigma_i^e = \frac{-KEA_i}{L_i^2} \qquad i = 1, 2, ..., 47, \tag{37}$$

where, K is a constant coefficient that must be selected according to the geometric type of the section, E the modulus of elasticity of the material, A_i the cross-sectional area of member, and L_i the length of the member. Here, we consider k to be 3.96. Structural nodes are affected by a combination of triple loading modes. The first group includes loads of 6 kP in the positive direction of the X-axis and 14 kP in the negative direction of the Y-axis, which are acting at Nodes 17 and 22. The second group consists of loads of 6 kg in the positive direction of the X-axis and 14 kg in the negative direction of the Y-axis, acting only at Node 17. The third group includes loads of 6 kP in the positive direction of the X-axis and 14 kP in the negative direction of the Y-axis, acting only at Node 22. The first group represents the oblique loads of both power transmission lines, and the second and third groups represent the state where one of the two lines breaks. Moreover, truss members are classified into 27 groups according to geometric symmetry. The cross-sectional areas of the members are chosen from the 64 discrete values of the AISC code.

In order to ensure the performance of the chaotic map and algorithms and to increase the accuracy and sensitivity of calculations, each of the modes has been performed 30 times independently, and the results related to the best response and the average value of responses are presented in Table 1. The graph of these results is also shown in Figure 5. The coefficient of variation of the responses, which indicates the stability and robustness of the responses, has been calculated and another criterion for the efficiency of chaotic maps and algorithms has been obtained. By examining the optimization results for different combinations of algorithms with the chaotic map and comparing them with those obtained in the standard mode, a significant improvement in reducing the weight of the 47-bar power transmission tower is achieved.

These results are given as follows: In the cyclical parthenogenesis algorithm, the logistic chaotic map in Scenario 1 weighing 2365.032 pounds yields an optimal response. In the TLBO, the Gaussian chaotic map in Scenario 2 weighing 2364.0782 pounds yields an optimal answer.

In the BBO, the logistic chaotic map in Scenario 3 weighing 2364.1827 pounds has an optimal answer. Finally, in the DE, the logistic chaotic map in Scenario 2 with a weight of 2362.6301 pounds yields an optimal answer. Table 2 compares the results of this study with a number of previous studies including HS [29], CBO [5], Enhanced Colliding Bodies Optimization (ECBO) [30], and VPS [6].

5.2. A 120-bar dome-shaped truss

The topology and nodal numbering of the 120-bar dome-shaped truss are shown in Figure 6. The material density is 0.288 lb/in^3 and the modulus of elasticity is 30.450 lb/in^2. The tensile and compressive stresses of permissible members are proposed in accordance with AISC ASD code. The maximum deformation for each node in all directions is set at 0.1969. All non-abutment nodes are affected by vertical loading, with an intensity of -13.49 kips at node one, -6.744 kips in Spheres 2 to 14, and 2.248 kips at other nodes. Moreover, truss members are classified into seven groups in terms of shape and geometric symmetry. The minimum cross-sectional area of the members, which is the lower limit of the decision variables, is 0.775 in^2, and the maximum cross-sectional area is 20 in^2. The allowable stress in tension and pressure is determined using the following equations:

$$\sigma_i = \begin{cases} 0.6F_y & \text{for} \quad \sigma_i \geq 0 \\ \sigma_i^- & \text{for} \quad \sigma_i < 0 \end{cases} \tag{38}$$

Table 1. Statistical results for the 47-bar power transmission tower.

Algorithms	Best	Mean	C.V (%)	Algorithms	Best	Mean	C.V (%)
CPA	2383.4062	2406.1394	0.90067	TLBO	2375.6054	2396.3259	0.81556
Logistic-1→ CCPA-21	**2365.032**	2391.9022	0.81021	Logistic-1→CTLBO-21	2374.8587	2406.2640	1.2542
Logistic-2→ CCPA-22	2370.4227	2409.0756	1.1493	Logistic-2→CTLBO-22	2370.5835	2394.6153	1.2491
Logistic-3→ CCPA-23	2372.1264	2393.5193	0.8169	Logistic-3→CTLBO-23	2367.1727	2403.5868	1.5985
Tent-1→ CCPA-31	2375.4972	2389.1271	0.5485	Tent-1→CTLBO-31	2366.5993	2388.2871	1.2165
Tent-2→ CCPA-32	2383.9673	2445.7064	2.3887	Tent-2→CTLBO-32	2379.4114	2395.0188	**0.59574**
Tent-3→ CCPA-33	2387.8112	2424.3097	1.3859	Tent-3→CTLBO-33	2385.4825	2410.8395	1.6433
Gauss-1→ CCPA-41	2376.3612	2391.5826	0.63873	Gauss-1→CTLBO-41	2373.3961	2412.7301	1.2681
Gauss-2→ CCPA-42	2373.6574	**2387.4008**	0.90321	Gauss-2→CTLBO-42	**2364.0782**	**2388.1237**	0.82143
Gauss-3→ CCPA-43	2370.8198	2383.3398	**0.30693**	Gauss-3→CTLBO-43	2378.9668	2405.1827	1.4351
Algorithms	**Best**	**Mean**	**C.V (%)**	**Algorithms**	**Best**	**Mean**	**C.V (%)**
BBO	2379.7486	2392.2093	0.36922	DE	2379.7022	2392.5358	0.59376
Logistic-1→CBBO-21	2374.9216	2385.5855	0.44635	Logistic-1→CDE-21	2373.5810	2392.3563	0.70631
Logistic-2→CBBO-22	2370.1165	2382.1141	0.6337	Logistic-2→ CDE-22	**2362.6301**	2389.5970	0.80487
Logistic-3→CBBO-23	**2364.1827**	2390.3094	1.0826	Logistic-3→ CDE-23	2369.7912	**2376.5127**	0.35929
Tent-1→CBB-31	2373.9801	2383.2691	0.38871	Tent-1→CDE-31	2380.8505	2396.9851	0.6832
Tent-2→CBBO-32	2371.5676	2390.3712	0.75472	Tent-2→CDE-32	2378.5494	2398.9571	0.69835
Tent-3→CBBO-33	2375.8513	2399.7306	0.68043	Tent-3→CDE-33	2365.3298	2376.9154	0.37785
Gauss-1→CBBO-41	2367.6249	**2380.3880**	**0.33034**	Gauss-1→ CDE-41	2373.4380	2385.7497	0.43924
Gauss-2→CBBO-42	2369.3647	2386.0523	0.60787	Gauss-2→CDE-42	2369.3244	2392.4670	0.66788
Gauss-3→CBBO-43	2368.2733	2391.4603	0.85606	Gauss-3→CDE-43	2383.1570	2390.0883	**0.24109**

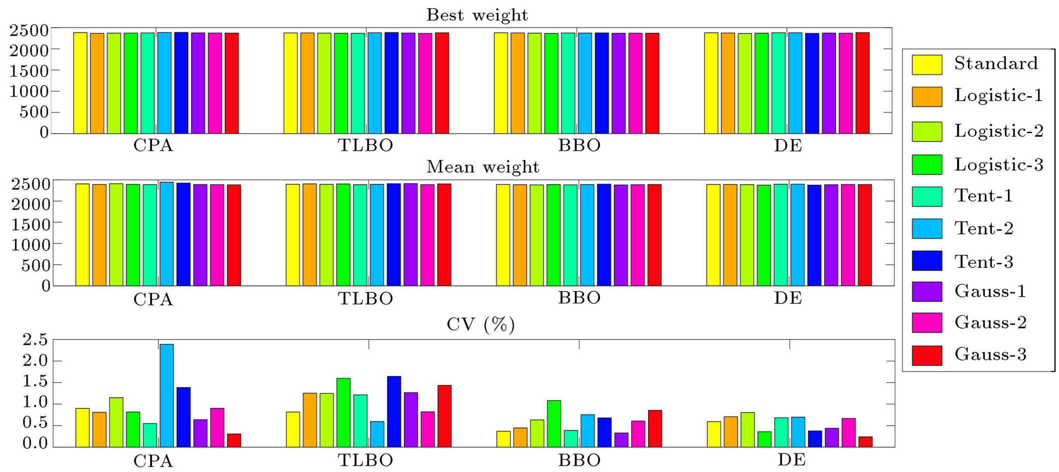

Figure 5. Optimization results in the standard mode and selection of the chaotic map for the 47-bar power transmission tower.

For compressive stresses, the following equations are used:

$$\sigma_i^- = \begin{cases} (1-\frac{\lambda^2}{2C^2})F_y/FS & \text{for} \quad \lambda < C \\ \frac{12\pi^2 E}{23\lambda^2} & \text{for} \quad \lambda \geq C \end{cases} \tag{39}$$

In this equation, we have:

$$FS = \left(\frac{5}{3} + \frac{3\lambda}{8C} - \frac{\lambda^3}{8C^3}\right); \qquad C = \sqrt{\left(\frac{2\pi^2 E}{F_y}\right)};$$

$$\lambda = \frac{kl}{r}; \quad r = aA^b; \quad a = 0.4993; \quad b = 0.6777. \tag{40}$$

The used components include modulus of elasticity E, the yield stress of steel F_y, the ratio of the limit slenderness C that separates the elastic from inelastic buckling region in comparison with the existing slandering λ. Moreover, k is the effective length factor of l and r is the radius of rotation of the limb. In order to ensure the viable performance of the chaotic map and algorithms and to increase the accuracy and sensitivity of calculations, each of the modes has been performed independently 30 times and the results related to the best response and the average value of responses are presented in Table 3. A graph of these results is shown in Figure 7.

Also, the coefficient of variation of the responses,

Table 2. Optimal design comparison for the 47-bar power transmission tower.

Number group	Element group	Lee et al. HS [29]	Kaveh & Mahdavi CBO [5]	Kaveh & Ilchi Ghazaan ECBO [30]	Kaveh & Ilchi Ghazaan VPS [6]	CCPA-21 present work	CTLBO-42 present work	CBBO-23 present work	CDE-22 present work
1	$A_1 - A_3$	3.840	3.840	3.840	3.840	3.8400	3.8400	3.8400	3.8700
2	$A_2 - A_4$	3.380	3.380	3.380	3.380	3.3800	3.3800	3.4700	3.4700
3	$A_5 - A_6$	0.766	0.785	0.766	0.766	0.7660	0.7660	0.7660	0.7660
4	A_7	0.141	0.196	0.111	0.111	0.1410	0.1110	0.1410	0.1110
5	$A_8 - A_9$	0.785	0.994	0.785	0.785	0.9940	0.9940	0.9940	0.9940
6	A_{10}	1.990	1.800	1.990	1.990	1.9900	1.9900	1.9900	1.9900
7	$A_{11} - A_{12}$	2.130	2.130	2.130	2.130	2.1300	2.1300	2.1300	2.3800
8	$A_{13} - A_{14}$	1.228	1.228	1.228	1.228	1.2280	1.2280	1.2280	1.2280
9	$A_{15} - A_{16}$	1.563	1.563	1.563	1.563	1.5630	1.5630	1.5630	1.5630
10	$A_{17} - A_{18}$	2.130	2.130	2.130	2.130	2.1300	2.1300	2.1300	2.1300
11	$A_{19} - A_{20}$	0.111	0.111	0.111	0.111	0.1110	0.1410	0.1110	0.1110
12	$A_{21} - A_{22}$	0.111	0.111	0.141	0.111	0.1110	0.1110	0.1960	0.1960
13	$A_{23} - A_{24}$	1.800	1.800	1.800	1.800	1.8000	1.8000	1.8000	1.8000
14	$A_{25} - A_{26}$	1.800	1.800	1.800	1.800	1.8000	1.8000	1.8000	1.8000
15	A_{27}	1.457	1.563	1.457	1.457	1.4570	1.4570	1.4570	1.4570
16	A_{28}	0.442	0.442	0.442	0.442	0.5630	0.4420	0.5630	0.6020
17	$A_{29} - A_{30}$	3.630	3.630	3.630	3.630	3.6300	3.5500	3.6300	3.6300
18	$A_{31} - A_{32}$	1.457	1.457	1.457	1.457	1.4570	1.5630	1.5630	1.4570
19	A_{33}	0.442	0.307	0.307	0.307	0.3910	0.9940	0.5630	0.3070
20	$A_{34} - A_{35}$	3.630	3.090	3.090	3.090	3.0900	2.9300	2.8800	3.0900
21	$A_{36} - A_{37}$	1.457	1.266	1.266	1.266	1.4570	1.5630	1.4570	1.2660
22	A_{38}	0.196	0.307	0.307	0.307	0.2500	0.1110	0.1410	0.3910
23	$A_{39} - A_{40}$	3.840	3.840	3.840	3.840	3.6300	3.3800	3.8400	3.6300
24	$A_{41} - A_{42}$	1.563	1.563	1.563	1.563	1.5630	1.4570	1.5630	1.5630
25	A_{43}	0.196	0.111	0.111	0.111	0.1410	0.1110	0.1110	0.1110
26	$A_{44} - A_{45}$	4.590	4.590	4.590	4.590	4.1800	4.1800	3.8400	4.1800
27	$A_{46} - A_{47}$	1.457	1.457	1.457	1.457	1.4570	1.5630	1.5630	1.4570
Best	Weight (lb)	2396.8	2386.0	2375.35	2374.81	2365.032	2364.0782	2364.1827	**2362.6301**
Mean	Weight (lb)	N/A	2462.76	2415.51	2415.07	2391.9022	**2388.1237**	2390.3094	2389.5970
Coefficient	Variation (CV)	N/A	44.79	41.01	35.65	0.81021	0.821430	1.08260	**0.80487**
NFE		45557	25000	N/A	N/A	15205	15205	15205	15205

Table 3. Statistical results for the 120-bar dome shaped truss.

Algorithms	Best	Mean	C.V (%)	Algorithms	Best	Mean	C.V (%)
CPA	33246.7800	33250.0071	0.0066764	TLBO	33249.0135	33249.3789	**0.00091253**
Logistic-1→CCPA-21	33247.2081	33250.8725	0.006972	Logistic-1→CTLBO-21	33248.2546	33248.2546	0.002845
Logistic-2→CCPA-22	33246.0188	33248.2271	.0052442	Logistic-2→CTLBO-22	33247.2225	33248.6311	0.0031598
Logistic-3→CCPA-23	**33242.4819**	**33247.6062**	0.010151	Logistic-3→CTLBO-23	33248.036	33248.8807	0.0017114
Tent-1→ CCPA-31	33248.899	33250.2159	**0.0031352**	Tent-1→CTLBO-31	33247.8734	33248.5277	0.0015347
Tent-2→CCPA-32	33246.7517	33293.1622	0.10643	Tent-2→CTLBO-32	33247.6774	33248.7601	0.0027666
Tent-3→CCPA-33	33243.1343	33328.6789	0.1123	Tent-3→CTLBO-33	33248.5594	33249.1730	0.0010844
Gauss-1→CCPA-41	33247.3949	33250.2586	0.0071388	Gauss-1→CTLBO-41	33245.7971	33248.0432	0.0045271
Gauss-2→CCPA-42	33244.9609	33249.1579	0.0086747	Gauss-2→CTLBO-42	**33245.5090**	**33247.9231**	0.0042095
Gauss-3→ CCPA-43	33249.6214	33252.5041	0.010215	Gauss-3→CTLBO-43	33246.001	33248.4871	0.0042349

Algorithms	Best	Mean	C.V (%)	Algorithms	Best	Mean	C.V (%)
BBO	33250.7708	33255.4069	0.014163	DE	33249.2681	33253.3395	0.0070276
Logistic-1→CBBO-21	33245.0895	33249.6987	**0.0006661**	Logistic-1→CDE-21	33245.5095	33249.3168	0.0078152
Logistic-2→CBBO-22	33241.6929	33251.1681	0.027271	Logistic-2→ CDE-22	33247.0278	33249.1613	0.0054182
Logistic-3→CBBO-23	33244.7138	33252.4491	0.019698	Logistic-3→ CDE-23	33244.6121	33248.4040	0.007266
Tent-1→CBBO-31	33241.3356	33246.4770	0.022279	Tent-1→ CDE-31	33242.6136	**33247.5176**	0.0097836
Tent-2→CBBO-32	33246.5544	33266.0958	0.059382	Tent-2→ CDE-32	33246.3652	33248.1966	0.0052935
Tent-3→CBBO-33	33249.0764	33271.8552	0.072742	Tent-3→ CDE-33	33247.6995	33248.5526	**0.002600**
Gauss-1→CBBO-41	33240.9881	33249.8182	0.021735	Gauss-1→ CDE-41	**33241.9660**	33247.9580	0.014293
Gauss-2→CBBO-42	33241.9045	**33246.2019**	0.085582	Gauss-2→ CDE-42	33242.8331	33247.6391	0.012065
Gauss-3→CBBO-43	**33240.5624**	33246.6531	0.020825	Gauss-3→ CDE-43	33246.5905	33248.4952	0.0058546

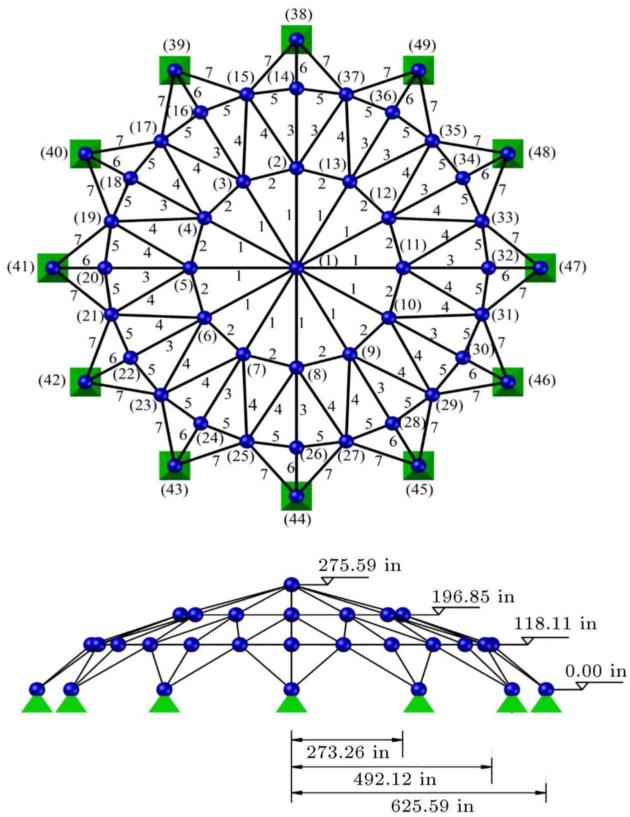

Figure 6. Schematic of the 120-bar dome shaped truss.

which indicates the stability and robustness of the responses, has been calculated and another criterion for the efficiency of the chaotic map and algorithms has been obtained. The minimum and maximum cross-sectional areas of all members are 0.775 in^2 and 20 in^2, respectively. By examining the optimization results for different combinations of chaotic map algorithms and comparing them with those in the standard mode, a significant improvement in reducing the weight of the 120-bar dome-shaped truss is achieved.

In the cyclical parthenogenesis algorithm, the logistic chaotic map in Scenario 3 weighing 33,242.4819 lb yields an optimal response. In the TLBO, the Gaussian chaotic map in Scenario 2 weighing 33245.5090 lb has an optimal response. In BBO, the Gaussian chaotic map in Scenario 3 with a weight of 33240.5624 lb has an optimal response. Finally, in the DE, Gaussian chaotic map in Scenario 1 with a weight of 33241.9660 lb has an optimal response. Table 4 compares the results of this study with previous findings including (HPSACO) [31], RO [9], and CBO [4].

5.3. A 200-bar planar truss structure

The topology and nodal numbering of the 200-bar planar truss structure are shown in Figure 8. The material density is 0.283 lb/in^3 and the module of elasticity is 30.000 ksi. The allowable stress for members 10 klb has been assumed. No deformation restrictions are intended for optimization. Truss loading is done in three independent groups. The first group is the lateral load of the structure, which includes a load of 1 kip along the positive axis of X_s and is applied to Nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, and 71. The second group covers the gravity load of the structure, which includes a load of 10 kip in the negative direction of the axis of the Y_s and is applied to Nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19,...,71, 72, 73, 74, and 75. The third group applies both loading groups together.

Truss members are classified into 29 groups. The minimum cross-section of the members, which is the lower limit of the decision variables, is considered as 0.1 in^2, while the maximum cross-section is 16 in^2. In order to ensure the performance of chaotic maps and algorithms and increase the accuracy and sensitivity of calculations, each state is independently executed 30 times and the results of the best response and the average amount of responses are presented statistically

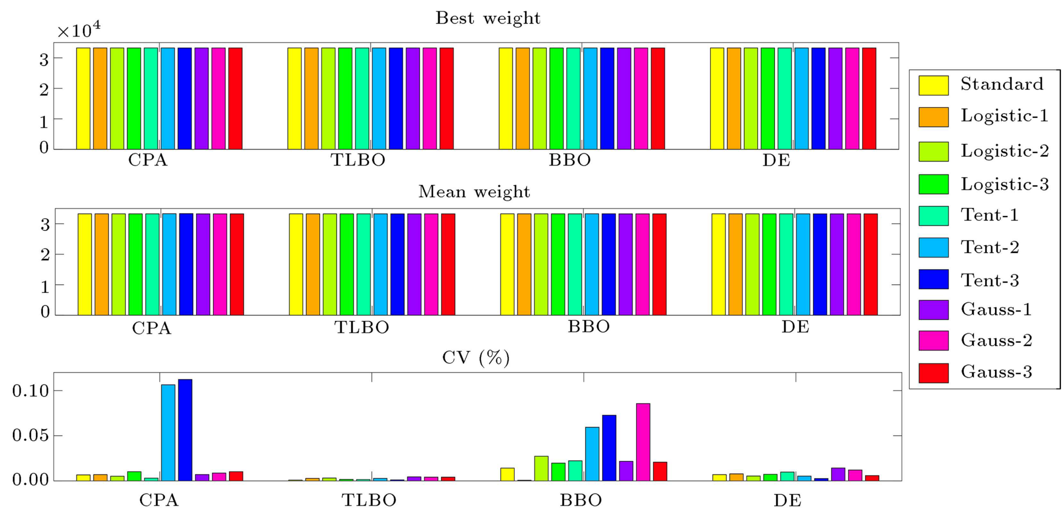

Figure 7. Optimization results in standard mode and chaotic maps for the 120-bar dome shaped truss.

Table 4. Optimal design comparison for the 120-bar dome shaped truss.

Number group	Kaveh & Talatahari HPSACO [31]	Kaveh & Khayatazad RO [9]	Kaveh & Mahdavi CBO [4]	CCPA-23 present work	CTLBO-42 present work	CBBO-43 present work	CDE-41 present work
1	3.095	3.030	3.0248	3.02412	3.02435	3.0248	3.02453
2	14.450	14.806	14.9543	14.7935	14.794	14.7029	14.8004
3	5.020	5.440	5.4607	5.06922	5.09075	5.09729	5.0796
4	3.352	3.124	3.1214	3.12896	3.13189	3.1266	3.12682
5	8.631	8.021	8.0552	8.48944	8.46675	8.48967	8.4641
6	3.432	3.614	3.3735	3.28278	3.2844	3.32243	3.30583
7	2.499	2.487	2.4899	2.49676	2.49633	2.49588	2.4968
Best weight (lb)	33248.9	33317.8	33286.3	33242.4819	33245.5090	**33240.5624**	33241.9660
Mean weight (lb)	N/A	N/A	33398.5	33247.6062	33247.9231	**33246.6531**	33247.9580
Coefficient variation (CV)	N/A	354.333	67.09	0.010151	**0.0042095**	0.020825	0.014293
NFE	N/A	N/A	N/A	20250	20250	20250	20250

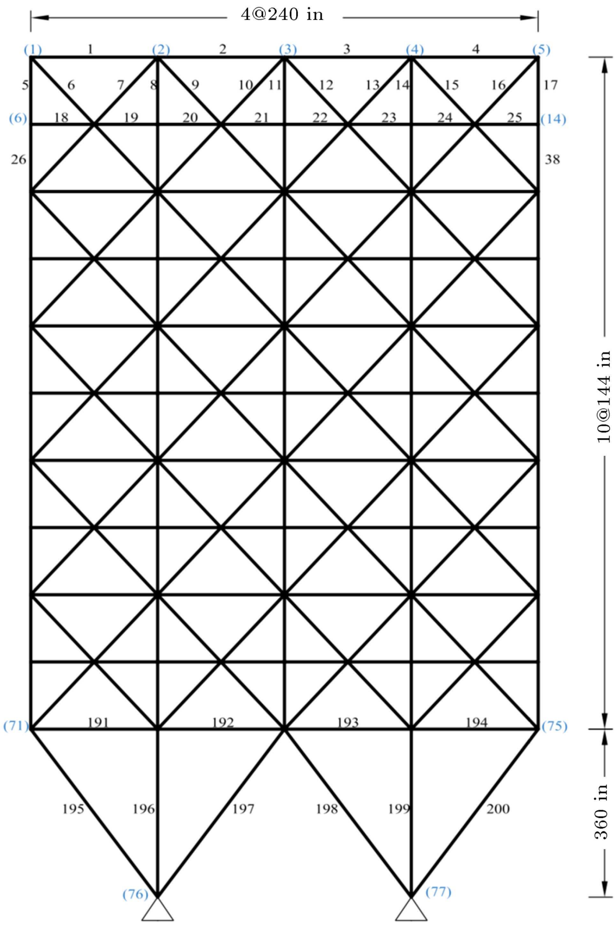

Figure 8. Schematic of a 200-bar planar truss structure.

in Tables 5 and shown in Figure 9. Moreover, the coefficient of changes in responses, which indicates the stability and robustness of the responses, has been calculated and another criterion for the efficiency of chaotic functions and algorithms has been achieved.

These results are given for each algorithm as follows. In the cyclical parthenogenesis algorithm, the Gaussian chaotic map in Scenario 2 weighing 25117.1081 lb has an optimal response. In the TLBO, the Gaussian chaotic map in Scenario 1 weighing 25128.7987 lb has an optimal response. In BBO, the logistic chaotic map in Scenario 2 with a weight of 25125.3835 lb has an optimal response. Finally, in the DE, the Gaussian chaotic map in Scenario 3 with a weight of 25102.878 lb has an optimal response. Table 6 compares the results of this study with a number of previous studies including PSO, PSOPC, HPSACO [31], and RO [32].

6. Discussion and concluding remarks

One of the most important features to increase the efficiency of meta-heuristic algorithms is to balance the exploration and exploitation stages. Chaotic maps have good potentials to establish this balance [33]. By replacing these maps in the exploration, exploitation, or both, different scenarios for optimization are obtained. In this research, upon the application of chaotic maps in several meta-heuristic algorithms, a significant improvement in truss weight optimization was achieved. Furthermore, in order to form a statistical population and determine the best weight, average weight, and coefficient of variation, each structural model was implemented with 30 independent replications. Other applications can be found in other meta-heuristics [34–36].

To provide the final results, the results of previous tables are normalized and then, combined. Eq. (41) is intended to summarize the information [29]. The results are presented in Table 7 to compare the performance of the algorithms.

$$NVal^{MV} = \frac{1}{S}\sum_{i=1}^{S}\left(\frac{Val^{M.V}}{\min\{Val_i\}} - 1\right). \qquad (41)$$

Table 5. Statistical results for the 200-bar planar truss structure.

Algorithms	Best	Mean	C.V (%)	Algorithms	Best	Mean	C.V (%)
CPA	25791.4552	26365.2532	1.9009	TLBO	26040.6007	27062.4480	5.5550
Logistic-1→ CCPA-21	25118.5905	25348.5381	0.85254	Logistic-1→CTLBO-21	25150.3705	25434.6628	1.2437
Logistic-2→ CCPA-22	25210.1143	25333.7911	0.40345	Logistic-2→CTLBO-22	25223.9130	25351.0160	0.47102
Logistic-3→ CCPA-23	25137.3614	25436.2941	1.2308	Logistic-3→CTLBO-23	25171.1421	25318.9719	0.78233
Tent-1→ CCPA-31	25202.3896	25481.4435	1.1080	Tent-1→CTLBO-31	25286.7338	25365.6319	1.6189
Tent-2→ CCPA-32	28181.7681	29695.0024	4.4680	Tent-2→CTLBO-32	25506.8124	25869.0872	1.6578
Tent-3→ CCPA-33	25495.7518	27538.2636	4.8412	Tent-3→CTLBO-33	25387.7182	**26223.6360**	2.1787
Gauss-1→ CCPA-41	25198.0061	25441.9059	0.81384	Gauss-1→CTLBO-41	**25128.7987**	25838.8707	2.4661
Gauss-2→ CCPA-42	**25117.1081**	**25250.8400**	**0.36203**	Gauss-2→CTLBO-42	25212.6524	25580.1194	1.4919
Gauss-3→ CCPA-43	25143.0994	25528.9189	1.1622	Gauss-3→CTLBO-43	25157.8303	25346.7924	**0.46121**
Algorithms	**Best**	**Mean**	**C.V (%)**	**Algorithms**	**Best**	**Mean**	**C.V (%)**
BBO	25574.0456	25760.4385	0.68062	DE	25528.1305	25667.6917	0.32072
Logistic-1→CBBO-21	25200.2392	25388.3653	0.65953	Logistic-1→CDE-21	25135.3133	25292.5288	0.44071
Logistic-2→CBBO-22	**25125.3835**	25373.1812	0.74767	Logistic-2→ CDE-22	25124.3969	25255.6444	0.37892
Logistic-3→CBBO-23	25174.1990	25519.1921	1.0559	Logistic-3→ CDE-23	25126.1243	**25181.3225**	**0.14413**
Tent-1→CBBO-31	25151.7825	25395.0075	0.65991	Tent-1→ CDE-31	25153.8718	25341.3289	0.46014
Tent-2→CBBO-32	25137.1304	**25242.048**	**0.47243**	Tent-2→ CDE-32	25137.1112	25207.5976	0.28446
Tent-3→CBBO-33	25296.7692	25548.1724	0.72931	Tent-3→ CDE-33	25205.8675	25238.5545	0.29325
Gauss-1→CBBO-41	25176.0167	25371.1231	0.55231	Gauss-1→ CDE-41	25120.6143	25336.1363	0.85912
Gauss-2→CBBO-42	25129.425	25365.0147	0.70812	Gauss-2→ CDE-42	25245.1079	25300.5538	0.38258
Gauss-3→CBBO-43	25272.9129	25379.0881	0.96872	Gauss-3→ CDE-43	**25102.878**	25225.2811	0.29596

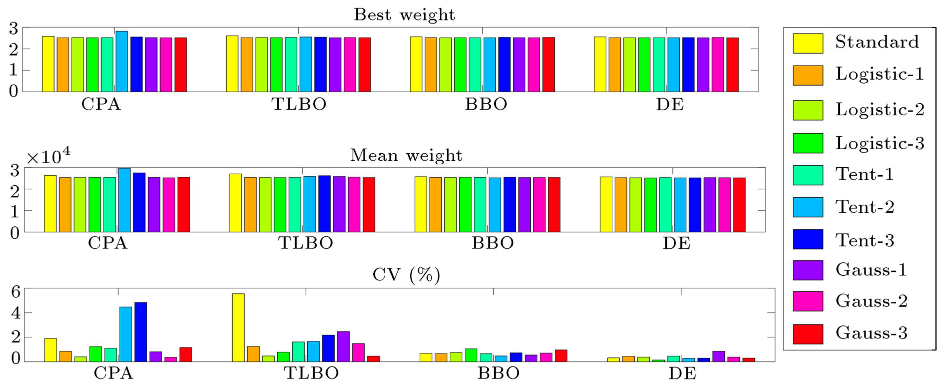

Figure 9. Optimization results in standard and chaotic maps for the 120-bar planar truss structure.

In this regard, Val^{MV} and $NVal^{MV}$ are the optimal values of the previous tables and the corresponding normalized values, respectively. S is also the number of structural examples examined and min $\{Val_i\}$ is the lowest value obtained in each study. Based on the results, the first three priorities of chaotic functions to improve the optimization results of algorithms are given in Table 8.

Some of the considerable results in this research are given as follows:

- In most cases, the combination of chaotic maps in meta-heuristic algorithms made a significant improvement. The main factor is the role of chaotic map in escaping local optimum and preventing premature convergence;

- In Scenarios 1 and 2, the chaotic maps are substituted in the exploration and exploitation steps. Accordingly, the following remarks conclude the paper:

Table 6. Optimal design comparison for the 200-bar planar truss structure.

Number group	Element group	Kaveh & Talatahari PSO [31]	Kaveh & Talatahari PSOPC [31]	Kaveh & Talatahari HPSACO [31]	Kaveh & Khayatazad RO [32]	CCPA-42 present work	CTLBO-41 present work	CBBO-22 present work	CDE-43 present work
1	1,2,3,4	0.8016	0.7590	0.1033	0.3820	0.10034	0.10087	0.10026	0.12402
2	5,8,11,14,17	2.4028	0.9032	0.9184	2.116	1.14968	0.96188	1.12759	1.0540
3	19,20,21,22,23,24	4.3407	1.1000	0.1202	0.102	0.10001	0.10657	0.15374	0.10278
4	18,25,56,63, 94,101,132, 139,170,177	5.6972	0.9952	0.1009	0.141	0.11166	0.24788	0.1000	0.10275
5	26,29,32,35,38	3.9538	2.1350	1.8664	3.662	2.03737	1.9440	2.14335	2.01135
6	6,7,9,10,12,13, 15,16,27,28,30, 31,33,34,36,37	0.5950	0.4193	0.2826	0.176	0.26422	0.26903	0.27695	0.27531
7	39,40,41,42	5.6080	1.0041	0.1000	0.121	0.16951	0.12954	0.12933	0.11122
8	43,46,49,52,55	9.1953	2.8052	2.9683	3.544	3.07121	3.04292	3.0312	3.11123
9	57,58,59,60,61,62	4.5128	1.0344	0.1000	0.108	0.1000	0.13428	0.16827	0.11182
10	64,67,70,73,76	4.6012	3.7842	3.9456	5.565	4.06498	4.0034	4.11759	4.07519
11	44,45,47,48, 50,51,53,54, 65,66,68,69, 71,72,74,75	0.5552	0.5269	0.3742	0.542	0.42596	0.39681	0.45812	0.41336
12	77,78,79,80	18.7510	0.4302	0.4501	0.138	0.10853	0.34913	0.1000	0.20185
13	81,84,87,90,93	5.9937	5.2683	4.96029	5.139	5.37705	5.21192	5.3859	5.3584
14	95,96,97,98,99,100	0.1000	0.9685	1.0738	0.101	0.10274	0.109244	0.24866	0.10221
15	102,105,108, 111,114	8.1561	6.0473	5.9785	8.742	6.3736	6.27338	6.50599	6.32858
16	82,83,85,86, 88,89,91,92, 103,104,106, 107,109,110, 112,113	0.2712	0.7825	0.78629	0.431	0.51652	0.60767	0.58281	0.57848
17	115,116,117,118	11.1520	0.5920	0.73743	0.998	0.20372	0.21674	0.1000	0.18244
18	119,122,125,128,131	7.1263	8.1858	7.3809	7.212	7.75462	7.6739	7.5961	7.85412
19	133,134,135, 136,137,138	4.4650	1.0362	0.66740	0.152	0.13473	0.57493	0.90077	0.11309
20	140,143,146,149,152	9.1643	9.2062	8.3000	8.452	8.74731	8.21321	8.7416	8.89872
21	120,121,123,124, 126,127,129,130, 141,142,144,145, 147,148,150,151	2.7617	1.4774	1.19672	0.835	0.72938	0.92768	1.06104	0.74381
22	153,154,155,156	0.5541	1.8336	1.000	0.413	0.95367	0.17437	0.1000	0.47438
23	157,160,163,166,169	16.1640	10.6110	10.8262	10.146	10.7196	10.6292	10.6708	10.7402
24	171,172,173, 174,175,176	0.4974	0.9851	0.1000	0.874	0.1000	0.10498	0.48027	0.10411
25	178,181,184,187,190	16.2250	12.5090	11.6976	11.384	11.5497	11.6133	11.8048	11.6915
26	158,159,161,162, 164,165,167,168, 179,180,182,183, 185,186,188,189	1.0042	1.9755	1.3880	1.197	1.2994	0.92038	1.20215	1.08102

Table 6. Optimal design comparison for the 200-bar planar truss structure (continued).

Number group	Element group	Kaveh & Talatahari PSO [31]	Kaveh & Talatahari PSOPC [31]	Kaveh & Talatahari HPSACO [31]	Kaveh & Khayatazad RO [32]	CCPA-42 present work	CTLBO-41 present work	CBBO-22 present work	CDE-43 present work
27	191,192,193,194	3.6098	4.5149	4.9523	5.747	5.9621	7.14378	5.63631	6.12934
28	195,197,198,200	8.3684	9.8000	8.8000	7.823	10.5188	10.9062	10.2859	10.7134
29	196,199	15.5620	14.5310	14.6645	13.655	12.9404	12.2776	12.2079	13.1072
Best	Weight (lb)	44081.4	28537.8	25156.5	25193.2	25117.1	25128.7	25125.3	**25102.8**
Mean	Weight (lb)	N/A	N/A	N/A	N/A	25250.8	25838.8	25373.1	**25225.2**
	Coefficient Variation(CV)	N/A	N/A	N/A	N/A	0.36203	2.4661	0.7476	**0.29596**
NFE		N/A	N/A	N/A	N/A	20250	20250	20250	20250

Table 7. Final normalized value with the participation of all the examples.

Algorithms	CPA			TLBO			BBO			DE		
	Best	Mean	C.V	Best	Mean	C.V	Best	Mean	C.V	Best	Mean	C.V
Standard	0.00898	0.0122	1.783	0.00818	0.0131	2.3049	0.00644	0.00582	**4.2698**	0.00561	0.00741	2.2468
Logistic-1	**0.00005**	0.00198	1.289	0.00164	0.00275	1.9435	0.00136	**0.00147**	7.5087	0.00105	**0.00192**	2.0741
Logistic-2	0.00109	0.00244	0.9423	**0.00118**	**0.00085**	3.1431	**0.00051**	0.00160	12.725	**0.00035**	0.00218	1.7486
Logistic-3	0.00163	0.00368	1.4656	0.00297	0.00346	2.3551	0.00084	0.00342	11.668	0.00125	0.00123	0.9907
Tent-1	0.00155	0.00335	0.7802	0.00189	0.00094	2.7472	0.00218	0.00205	7.1877	0.00169	0.00336	2.1604
Tent-2	0.02246	0.0355	10.4144	0.00375	0.00443	7.3197	0.00150	0.00152	17.722	0.00195	0.00325	1.3992
Tent-3	0.00509	0.0213	10.6561	0.00399	0.00869	4.9017	0.00300	0.00438	20.653	0.00292	0.00219	0.5181
Gauss -1	0.00154	0.00246	1.03705	0.00228	0.00746	3.3223	0.00161	0.00152	8.2875	0.00138	0.00281	2.3359
Gauss -2	0.00072	**0.00106**	0.9244	0.00311	0.00405	**1.8289**	0.00064	0.00158	27.0693	0.00165	0.00263	2.0904
Gauss -3	0.00259	0.00376	**0.8152**	0.00271	0.00344	3.1887	0.00216	0.00242	8.5902	0.00365	0.00293	**0.4474**

Table 8. Chaotic maps effective in improving the results of algorithms.

Algorithms	Chaotic map		
	First priority	Second priority	Third priority
CPA	**Logistic-1**	**Gauss-2**	**Gauss-3**
TLBO	**Logistic-2**	**Logistic-2**	**Gauss-2**
BBO	**Logistic-2**	**Logistic-1**	**Standard**
DE	**Logistic-2**	**Logistic-1**	**Gauss-3**

"The cyclical parthenogenesis algorithm enjoys viable exploitation and uses chaotic maps for the exploration stage. Other algorithms such as teaching-learning-based optimization, biogeography-based optimization, and differential evolution go against the previous case and use chaotic functions for the exploitation stage";

- With the chaotic maps at hand, determining the regulatory parameters of algorithms and sensitivity analysis is significantly reduced. In fact, selecting the starting sentence in chaotic maps replaces complex settings. It should be noted that in most cases, it is more difficult to find appropriate tuning parameters of each algorithm than self-optimization. Therefore, complex engineering problems can be solved by using chaotic maps without the need to find parameters;
- To investigate the stability and reliability of the answers, the coefficient of variation, which is the standard dimensionless deviation, was used;
- To select the initiating sentence in a series of chaotic maps, several initial iterations were performed before the main iterations and the appropriate initiating sentence was selected to improve the results by leaps and bounds;
- Among the chaotic maps studied, the logistic function in Scenarios 1 and 2 provided the best results;
- Chaotic maps are expected to significantly improve optimization problems based on frequency constraints.

References

1. Kaveh, A., *Advances in Meta-heuristic Algorithms for Optimal Design of Structures*, 3rd Edition, Springer,

Switzerland (2021). https://doi.org/10.1007/978-3-319-05549-7

2. Kaveh, A. and Bakhshpoori, T. "Water evaporation optimization: A novel physically inspired optimization algorithm", *Comput. Struct.*, **167**, pp. 69–85 (2016).
3. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: charged system search", *Acta Mech.*, **213**, pp. 267–289 (2010). https://doi.org/10.1007/s00707-009-0270-4
4. Kaveh, A. and Mahdavi, V.R. "Colliding bodies optimization: A novel meta-heuristic method", *Comput. Struct.*, **139**, pp. 18–27 (2014). https://doi.org/10.1016/j.compstruc.2014.04.005
5. Kaveh, A. and Mahdavi, V.R. "Colliding bodies optimization method for optimum discrete design of truss structures", *Comput. Struct.*, **139**, pp. 43–53 (2014).
6. Kaveh, A. and Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", *Scientia Iranica, Trans. A, Civil Eng. A.*, **24**(2), pp. 551–566 (2017). https://doi.org/10.24200/sci.2017.2417
7. Kaveh, A. and Dadras, A. "A novel meta-heuristic optimization algorithm: thermal exchange optimization", *Adv. Eng. Softw.*, **110**, pp. 69–84 (2017). https://doi.org/10.1016/j.advengsoft.2017.03.014
8. Erol O.K. and Eksin, I. "New optimization method: Big Bang-Big Crunch", *Adv. Eng. Softw.* , **37**, pp. 106–111 (2006).
9. Kaveh, A, and Khayatazad M. "A novel meta-heuristic method: Ray optimization", *Comput. Struct.*, **112–113**, pp. 283–294 (2012). https://doi.org/10.1016/j.compstruc.2012.09.003
10. Lee, K.S., and Geem, Z.W. "A new structural optimization method based on the harmony search algorithm", *Comput. Struct.*, **82**, pp. 781–798 (2004).
11. Kaveh, A. and Zolghadr, A. "Cyclical parthenogenesis algorithm: a new meta-heuristic algorithm", *Adv. Eng. Softw.*, **18**(5), pp. 673–701 (2017).
12. Kaveh, A. and Mahjoubi, S. "Lion pride optimization algorithm: a meta-heuristic method for global optimization problems", *Scientia Iranica, Trans. A: Civil Eng.*, **25**(6), pp. 3113–3132 (2018).
13. Kaveh, A. and Kooshkebaghi, M. "Artificial coronary circulation system; A new bio-inspired meta-heuristic algorithm", *Scientia Iranica, Trans A: Civil Eng.*, **26**(5), pp. 2731–2747 (2019).
14. Dorigo, M., Maniezzo, V., and Colorni, A. "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **26**(1), pp. 29–41 (Feb. 1996). DOI: 10.1109/3477.484436
15. Mirjalili, S. and Lewis, A. "The whale optimization algorithm", *Adv. Eng. Softw.* **95**, pp. 51–67 (2016). https://doi.org/10.1016/j.advengsoft.2016.01.008
16. Mirjalili, S., Mirjalili, S.M., and Lewis, A. "Grey wolf optimizer", *Adv. Eng. Softw.*, **69**, pp. 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
17. Kennedy, J. and Eberhart, R. "Particle swarm optimization", In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942–1948 (1995).
18. Kaveh, A. and Zolghadr, A. "Democratic PSO for truss layout and size optimization with frequency constraints", *Comput. Struct*, **130**, pp. 10–21 (2014). https://doi.org/10.1016/j.compstruc.2013.09.002
19. Holland, J.H. "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, USA (1975).
20. Price, K.V. and Storn, R.M., *Differential Evolution: A Practical Approach to Global Optimization*, Springer (2005).
21. Schwefel, H.-P. "Evolutions strategy and numerical optimization", Ph.D. Thesis. Technical University Berlin (1975).
22. Eusuff, M., Lansey, K., and Pasha, F. "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization", *Engineering Optimization*, **38**(2), pp. 129–154 (2006). https://doi.org/10.1080/03052150500384759
23. Simon D "Biogeography-based optimization", *IEEE Trans. Evol. Comput.*, **12**(6), pp. 702–713 (2008). DOI: 10.1109/TEVC.2008.919004
24. Rao, R.V., Savsani, V.J., and Vakharia, D.P. "Teaching-learning-based optimization: a novel method for constrained mechanical design optimiz-ation problems", *Comput. Aided Des.*, **43**(3), pp. 303–315 (2011).
25. Talatahari, S., Kaveh, A., and Sheikholeslami, R. "Chaotic imperialist competitive algorithm for optimum design of truss structures", *Struct. Multidiscip. Optim.*, **46**(3), pp. 355–67 (2012).
26. Kaveh, A. and Vazirinia, Y. "Chaotic vibrating particles system for resource-constrained project scheduling problem", *Scientia Iranica, Trans. A: Civil Eng.*, **27**(4), pp. 1826–1842 (2020).
27. Peitgen, H.-O., Jörgens, H., and Saupe, D. "Chaos and Fractals", *New Frontiers of Science.*, Springer, Berlin (2006).
28. Kaveh, A., Sheikholeslami, R., Talatahari, S., et al. "Chaotic swarming of particles: a new method for size optimization of truss structures", *Adv. Eng. Softw.* **67**, pp. 136–47 (2014).
29. Lee, K.S., Han, S.W., and Geem, Z.W. "Discrete size and discrete-continuous configuration optimization methods for truss structures using the harmony search algorithm", *Int. J. Optim. Civil Eng.*, **1**, pp. 107–26 (2011).
30. Kaveh, A. and Ilchi Ghazaan, M. "Enhanced colliding bodies optimization for design problems with continuous and discrete variables", *Adv. Eng. Softw.*, **77**, pp. 66–75 (2014). https://doi.org/10.1016/j.advengsoft.2014.08.003
31. Kaveh, A. and Talatahari, S. "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", *Comput. Struct.*, **87**, pp. 267–283 (2009).

32. Kaveh, A. and Khayatazad, M. "Ray optimization for size and shape optimization of truss structures", *Comput. Struct.*, **117**, pp. 82–94 (2013).

33. Kaveh, A. and Javadi, S.M. "Chaos-based firefly for optimization of cyclically large-size braced steel domes with multiple frequency constraints", *Comput. Struct.*, **214**, pp. 28–39 (2019). https://doi.org/10.1016/j.compstruc.2019.01.006

34. Kaveh, A., Mottaghi, L., and Izadifard, R.A. "An integrated method for sustainable performance-based optimal seismic design of RC frames with non-prismatic beams", *Scientia Iranica, Trans. A: Civil Eng.*, **28**(5), pp. 2596–2612 (2021). http://scientiairanica.sharif.edu

35. Kaveh, A. and Rahami, H. "Analysis, design and optimization of structures using force method and genetic algorithm", *Int. J. Numer. Methods Eng.*, **65**(10), pp. 1570–1584 (2006). DOI: 10.1002/nme.1506

36. Fan, W., Chen, Y., Li, J., et al. "Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications", *Structures*, **33**, pp. 3954–3963 (2021). DOI: 10.1016/j.istruc.2021.06.110

Biographies

Hosain Yosefpoor is a PhD student at Tehran University of Science and Research. He also received his BSc and MSc degrees in Civil Engineering from Iran University of Science and Technology, Tehran, Iran. His research interests include artificial intelligence applications in structural design.

Ali Kaveh graduated from University of Tabriz in 1969 and continued his studies on structures at Imperial College of Science and Technology at London University and then, received his MSc, DIC, and PhD degrees in 1970 and 1974, respectively. He then joined Iran University of Science and Technology. Professor Kaveh is the author of 730 papers published in international journals and 170 papers presented at national and international conferences. He has authored 23 books in Persian and 14 books in English published by Wiley, Research Studies Press, American Mechanical Society, and Springer.