 https://doi.org/10.3311/PPci.40467

Periodica Polytechnica Civil Engineering, 69(3), pp. 884–903, 2025

Chaos-based Swarm Intelligence Algorithms for Optimal Design of Truss Structures

Ali Kaveh[1*], Hosein Yousefpoor[2]

[1] School of Civil Engineering, Iran University of Science and Technology, Narmak, 1684613114 Tehran, Iran
[2] Department of Civil Engineering, Maragheh Branch, Islamic Azad University, 5519747591 Maragheh, Iran
[*] Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 17 March 2025, Accepted: 27 May 2025, Published online: 06 June 2025

Abstract

The incorporation of chaos functions into metaheuristic algorithms leads to significant progress in the results of optimal design of truss structures. Chaos functions, by forming chaotic mutations, create the necessary conditions to create a balance between exploration and exploitation. With this balance, the algorithm is saved from premature convergence and, by forming chaotic series, a jump from local optima to global optima is achieved. In this research, chaos functions are formed in the basic steps of three meta-heuristic swarm intelligence algorithms and three new chaos algorithms. These algorithms include the Chaotic Grey Wolf Optimizer (CGWO), the Chaotic Crow Search Algorithm (CCSA), and the Chaotic Cyclical Parthenogenesis Algorithm (CCPA). To improve the optimization results, three different scenarios are examined and the chaotic results are compared with the standard case. In these scenarios, chaos series replace the exploration, exploitation, or both stages simultaneously.

Keywords

exploration, exploitation, meta-heuristic algorithms, premature convergence, local optima, global optima

1 Introduction

The advantages and superiority of truss structures have led to the use of this group of structures in various fields of engineering. In industrial buildings, aircraft hangars, and sheds, choosing a truss system is a unique option for covering large openings. In power transmission towers and telecommunication towers, the truss system will be required in large numbers and is the only system for which no other alternative can be found. In pedestrian overpasses, the truss structure of the bridge is associated with a large number of members. All of this emphasizes the importance of optimizing and constructing this group of structures in a light and economical manner for structural designers. Therefore, a large part of scientific research and optimization activities in recent decades have been dedicated to this topic. Designers of this group of structures need to consider other components such as cost and weight of the structure, in addition to engineering components such as stress, deformation, slenderness, and buckling, and the economic aspects of the structure need to be analyzed in the form of mathematical models. Traditional gradient-based methods and objective function derivative formation are the first solutions used by designers for optimization. In choosing the number of profiles, we will encounter discrete optimization, where the gradient-based method will not be able to reach optimal solutions. Numerical methods are the second proposed method for optimizing structures. In this method, the selected point for starting the iterative operation is of great importance, and if the selected starting point is not appropriate, the results of the calculations will stop when reaching the local optimum and, by falling into the trap of local optima, will undergo premature convergence and maturity, and there will be no way to jump from these local optima. The number of decision variables will have an increasing effect on the optimization time, and as the number of decision variables increases, the time for mathematical calculations increases tremendously. To get rid of these problems and benefit from high computational speed, researchers in the field of artificial intelligence have made great efforts and succeeded in presenting meta-heuristic algorithms. These algorithms start with an initial population and at each stage, the next generations are improved by taking inspiration from natural or physical phenomena. The reason for modeling natural

Cite this article as: Kaveh, A., Yousefpoor, H. "Chaos-based Swarm Intelligence Algorithms for Optimal Design of Truss Structures", Periodica Polytechnica Civil Engineering, 69(3), pp. 884–903, 2025. https://doi.org/10.3311/PPci.40467

phenomena is that if there is a need for a capability, nature has done it best [1]. The *first source* of inspiration is the pattern of genetic evolution of living things over millions of years since their beginnings. According to the theory of evolution, the world we live in is constantly changing. Therefore, any creature that intends to survive in such an environment must be able to adapt to the surrounding conditions. Although dinosaurs were powerful in size, they left the competition scene due to lack of adaptation, and instead, animals such as polar bears and arctic foxes adapted to the harsh conditions prevailing in the Arctic and have maintained their presence. These algorithms are based on Darwin and Lamarck's theory of evolution of living organisms. Four operators play a major role in evolution. These operators include natural selection, crossover, mutation, and symbiosis. In each selection step, the population of offspring is selected and merged using a roulette wheel. The best traits of the generations are passed on genetically to the offspring in future generations. Examples of these algorithms include Evolutionary Strategy (ES) [2], Differential Evolution (DE) [3], and Genetic Algorithm (GA) [4]. The *second inspiration* is to take advantage of the group behavior and swarm intelligence of animals in searching for and accessing food and other needs. By exchanging information in their surrounding search space, animals can demonstrate a cooperative optimization pattern that none of them could perform alone. Group members move through the search space and, by taking specific measures, gather at an optimal point, such as a food source. This group behavior and planning is evident in birds, fish, ants, and bees. The main components in this inspiration include population, cooperation, communication, information exchange, information flow and self-ordering. Some of these algorithms include Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6], Artificial Bee Colony (ABC) [7], Gray Wolf Optimizer (GWO) [8], Crow Search Algorithm (CSA) [9] and Cyclical Parthenogenesis Algorithm (CPA) [10]. The *third group* of metaheuristic algorithms have chosen physical laws as their source of inspiration. Some of these laws include: Coulomb's law, Faraday's law, Gauss's law, the principle of energy and momentum. The ability to access classical and regular relationships is a characteristic of this group. Some of the physically inspired metaheuristic algorithms are: Harmony Search (HS) [11], Water Evaporation Optimization (WEO) [12], Big Bang-Big Crunch (BB-BC) [13], Thermal Exchange Optimization (TEO) [14], Tug-of-War Optimization (TWO) [15], charged System Search (CSS) [16], Colliding Bodies Optimization (CBO) [17], and Vibrating Particles System (VPS) [18]. The inspiration is not limited to the groups introduced, and the inspiration and introduction of new metaheuristic algorithms continues without interruption and without any limitations. Biogeography-Based Optimization (BBO) [19], Imperialist Competitive Algorithm (ICA) [20], Teaching-Learning Based Optimization (TLBO) [21], and Invasive Weed Optimization (IWO) [22] are examples of metaheuristic algorithms with different sources of inspiration, each of which has produced significant improvements in optimization. Despite extensive efforts in introducing new metaheuristic algorithms in the field of optimization, in most of them the computational process encounters pitfalls that bring the operation to a halt. In this case, the algorithm gets trapped in local optima and suffers from premature convergence or premature maturity. The research conducted in this paper shows that by utilizing chaos functions and creating chaotic mutations, we can move from local optima to global optima and improve the optimization conditions. From a mathematical perspective, chaos refers to the ability of a structure and model to not show any signs of random events in practice, but to cause chaotic reactions in the environment. Important features of a chaotic system include: it is highly sensitive to initial conditions, its cyclical recurrence pattern is dense and compact, and in its chaotic and disordered behavior, one can see a specific order being extracted that is not repetitive. The potential inherent in chaos functions to create a balance between exploration and exploitation can be investigated and examined with different scenarios [23]. Some chaos functions converge to global minima in chaotic series formation with a very high probability from local minima, and these functions are suitable for improving the search conditions of algorithms. However, another group of chaos functions drive the search space towards local minima with a very high probability and are suitable for improving the extraction conditions. In a recent study, chaos functions have been embedded in three metaheuristic algorithms and the results have been compared with the standard case. The metaheuristic algorithms have been selected from the swarm intelligence group, and by forming chaotic algorithms, independent responses in the space of decision variables are significantly improved.

2 Formulation of structural optimization problems

For each optimization problem, three main parts are considered. These three parts include the objective function,

decision variables, and design constraints. For the design of truss structures, the objective function is of the minimization type and the weight of the structure must be able to satisfy the design constraints with the lowest possible weight. These constraints, which are often determined based on the rules of the regulations, include the allowable stress limit of the members, the displacement of the nodes and the slenderness of the members. The decision variable for estimating the weight of the structure is the cross-sectional area of the members, which must be within the feasible range, and for its selection, the lower and upper limits of the decision variables are observed. The general form for these problems is proposed according to Eq. (1):

$$\begin{aligned} &\text{Find} && A=\{A_1, A_2, A_3, ..., A_n\} \\ &\text{to Minimize} && W(A)=\sum_{i=1}^{n}\gamma_i \times A_i \times L_i \\ &\text{Subjected to} && g_j(A)\le 0; \quad j=1,2,3,...,m \\ & && h_k(A)\le 0; \quad k=1,2,3,...,p \\ & && \{A_L\}\le\{A\}\le\{A_U\} \end{aligned} \tag{1}$$

According to the provided equation, A is the cross-sectional area of the members, W is the total weight of the structure, n is the number of members of the structure, g_i and h_i are design constraints. These constraints can include stress, member slender and nodal displacement. Also, A_l and A_u are the upper and lower bounds of the decision variables. The initial form of meta-heuristic algorithms for optimization of unconstrained problems is presented. For this reason, the penalty function method with Lagrange coefficients is used in the modeling to convert the bounded function into unbounded one. In this method, if there is no violation and the answers satisfy the restrictions, the amount of the penalty will be zero. But if there is a violation of the design constraints, its value is calculated according from Eqs. (2)–(6) and included in the penalized objective function:

$$\sigma_i \le \sigma^{\max} \Rightarrow V_i = \max\left(0, \frac{\sigma_i}{\sigma^{\max}} - 1\right); \; i=1,2,3,...,n, \tag{2}$$

$$\delta_j \le \delta^{\max} \Rightarrow V_j = \max\left(\frac{\delta_j}{\delta^{\max}} - 1\right); \; j=1,2,3,...,n, \tag{3}$$

$$\lambda_k \le \lambda^{\max} \Rightarrow V_k = \max\left(0, \frac{\lambda_k}{\lambda^{\max}} - 1\right); \; k=1,2,3,...,p, \tag{4}$$

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

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

Eqs. (2)–(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 Introduction of Selected Chaos Map

In metaheuristic algorithms, the balance between the exploration and exploitation stages is not optimally maintained, so the convergence rate towards the global optimum first slows down and then stops. In such cases, the algorithm is trapped in local optima. Instead, chaotic functions have the potential to create chaotic jumps and take the algorithm out of the traps associated with local optima. These functions have their own characteristics. Some of these characteristics are: They do not exhibit any traces of random behavior, but they provide access to global optimal positions by creating irregular jump behaviors. In addition to these, they exhibit other characteristics, including: high sensitivity to initial conditions, non-periodic and deterministic dynamic behaviors and ergodic, non-reversibility of their mathematical criterion. These functions have sudden jump conditions and have sufficient talent to save metaheuristic algorithms from the trap of local optimum and transition in the vicinity of global optimum. In a classification, chaos functions can be introduced into two groups. The mathematical model of some chaos functions is such that they converge from local minima to global minima with a very high probability. This group can be useful for improving algorithms that have problems in the exploration phase. In another group of chaos functions, the mathematical relationships are such that the decision space is located in the local optima with a very high probability. These functions are suitable for algorithms that have weaknesses in the exploitation phase [23]. To familiarize yourself with the distribution form of chaos functions, the distribution diagram of several chaos functions for a repetition interval of 100 sentences is presented in Fig. 1. These functions have achieved significant results in combination with metaheuristic algorithms and improved optimization conditions. In order to form chaotic metaheuristic algorithms, the probability distribution functions related to the search

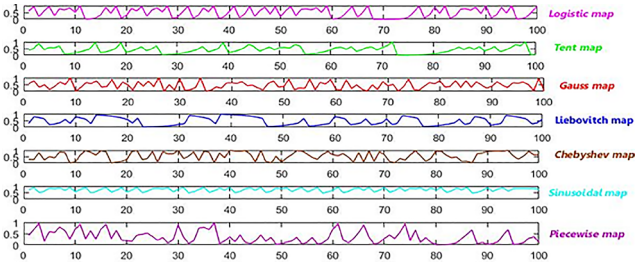

Fig. 1 Distribution of numerical values in 100 iterations for a number of well-known chaos functions

and extraction stage must first be identified in the standard mode of the algorithm and then chaotic functions must be replaced in different scenarios. The probability distribution function of the algorithm itself in the standard mode is in most cases uniformly distributed. However, in some cases, Levy distribution, Cauchy distribution, and normal distribution have also been used. In the first scenario, the chaos function replaces the probability distribution function of the exploration stage. In the second scenario, the chaos function replaces the probability distribution function of the exploitation stage. Finally, in the third scenario, the chaos function will replace both simultaneously. Now, by comparing the optimization results with the standard case, the degree of improvement in the results will be introduced. The chaos algorithm consists of four main steps. Flowchart of chaos algorithm for interfacing among steps presented in Fig. 2.

3.1 Logistics map

This map appears in nonlinear dynamic behaviors related to biological populations [24]. The statements of chaotic sequences in the logistic function are obtained according to Eqs. (7) and (8):

$$CHM_{k+1} = a \times CHM_k \left(1 - CHM_k\right), \tag{7}$$

$$\begin{aligned} &CHM_{k+1} \in (0,1), \ \ CHM_k \in (0,1), \\ &CHM_0 \notin (0, 0.25, 0.50, 0.75, 1). \end{aligned} \tag{8}$$

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

3.2 Gauss map

Using this function in nonlinear dynamic behaviors has good results [25]. The statements of chaotic sequences in the Gaussian function are obtained according to Eq. (9):

$$CHM_{k+1} = \begin{cases} 0 & CHM_k = 0 \\ \dfrac{1}{CHM_k} - \left[\dfrac{1}{CHM_k}\right] & CHM_k \neq 0 \end{cases}. \tag{9}$$

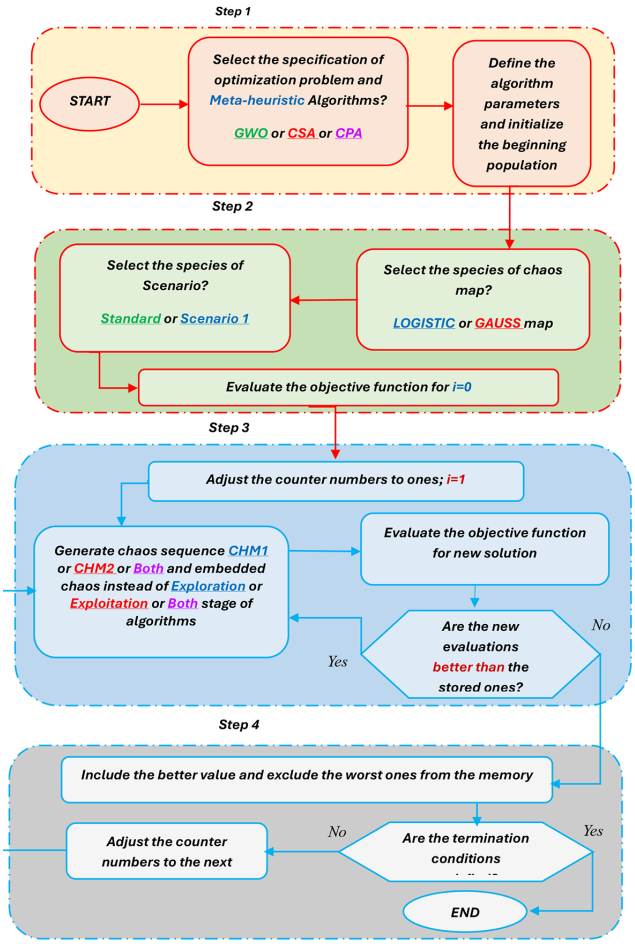

Fig. 2 Flowchart of chaos algorithm for interfacing among steps

4 Meta- heuristic algorithms and applying chaos functions

For metaheuristic algorithms, in the standard case, the necessary arrangements for convergence towards global optima are foreseen in two stages of exploration and exploitation. According to these stages, first, parts of the search space that have a superior strategy are selected and then their neighborhood is examined with as much precision as possible. In swarm intelligence algorithms, these two stages are somehow considered in the inspiration stages. For the Gray Wolf Optimizer (GWO) algorithm, surrounding the prey and then attacking the prey are considered the search and extraction stages in the algorithm, respectively. In the Crow Search Algorithm (CSA), when crow i is chasing crow j, the exploration mode is applied, and if crow j is checking the food without chasing, it plays the role of exploitation mode. Also, in the Cyclical Parthenogenesis Algorithm (CPA), reproduction with mating and without mating play the roles of exploration and exploitation, respectively. In these algorithms, at each stage of selecting the search space, we will need

to create diversity and variety in order to have a comprehensive coverage of the scope of the investigation [26]. In the standard mode, the algorithms use random distribution functions to achieve these goals. These distribution functions vary according to the perspectives of each algorithm and can include Uniform, Normal, Logistic, and Levy distributions [27]. Although the selection of these parameters is done randomly, they play a decisive role in the efficiency of the algorithm. By analyzing the sensitivity of metaheuristic algorithms, these key factors can significantly improve their performance in the following cases:

- Increasing the speed of algorithm convergence in reaching global optima.
- Jumping out of the trap of local optima and moving to the position of global optima.
- Creating a proper balance between exploration and exploitation.

Chaotic series derived from chaotic functions have unique properties that can improve the weaknesses of metaheuristic algorithms. While the structure of these series resembles stochastic processes, their values are deterministic, nonlinear, dynamic, and non-recursive. These series are non-convergent towards a specific limit, and there is no inverse for their generating chaotic functions [28]. With these properties and characteristics, their incorporation into metaheuristic algorithms with different scenarios can produce chaotic mutations and save the algorithm from the trap of local optima. In the *first scenario*, chaotic mutations are introduced in the exploration part of the algorithm. And the speed of operation for convergence increases. The *second scenario* is specific to algorithms where the exploitation part of the algorithm requires a chaotic distribution and the functions suitable for this stage have the largest distribution in local optima. For the *third scenario*, the chaotic function simultaneously replaces both the exploration and exploitation parts. Which scenario brings about desirable results for a specific algorithm is possible by modeling chaotic algorithms. In fact, the nonlinear and non-convex behavior of the objective functions in structural optimization has created these conditions for metaheuristic algorithms [29]. In this research, first, selected algorithms from the swarm intelligence group are examined in the standard mode, and then the results of embedding a number of chaos functions with three scenarios are presented, and finally, by creating a competition between the best chaos mode and the standard mode, the degree of improvement in optimization results is introduced.

4.1 Standard Grey Wolf Optimizer (GWO)

The Gray Wolf Optimizer (GWO) algorithm is inspired by the way the Gray Wolf hunts for prey. This algorithm was presented in 2014 by Mirjalili et al. [8]. This algorithm is in the category of swarm intelligence algorithms and is population-based. Gray wolves belong to the Canidae family and are considered apex predators. They often prefer to live in packs, each of which has an average of 5 to 12 wolves. The social hierarchy in wolves is very specific and strict. The leaders of each pack are a male and a female, who are called the alpha pack and are not necessarily the strongest members of the pack. The next group is the beta pack, which is under the command of the alpha. The lowest rank is the omega, which has the role of protector, who is the last wolf allowed to eat food. If a wolf does not belong to these three packs, he/she belongs to the delta pack, which includes elders, guardians, observers and hunters. One of the interesting social behaviors in gray wolves is their group hunting. In this hunt, the following three stages are considered.

- Track, chase and approach prey.
- Attempt to surround and harass prey until prey stops moving.
- Attack towards prey.

4.1.1 Basic steps in Grey Wolf Optimizer

Step 1 Determine Initial Values: The Gray Wolf Optimization Algorithm, like other population-based algorithms, begins by introducing initial proposed responses. These responses are selected within the decision space of the variables.

Step 2 Evaluation and Group Formation: By evaluating the initial responses, wolves belonging to the Alpha, Beta, and Delta groups are selected, which include the best positions of the first, second, and third solutions, respectively. The other solutions form the Omega group.

Step 3 Surrounding the Prey: The new position of each member of the omega group to surround the prey is determined according to the following mathematical Eq. (10):

$$\begin{aligned}
&\vec{D} = \left|\vec{C} \times \vec{X}_p(t) - \vec{X}(t)\right| \\
&\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \times \vec{D} \\
&\vec{X}_p : \text{position vector of prey} \\
&\vec{X} : \text{position vector of grey wolf from } "\omega" \text{ level} \\
&\vec{A} \text{ and } \vec{C} : \text{are coefficient vectors}
\end{aligned} \quad . \quad (10)$$

To determine the coefficient vectors, we proceed as follow Eqs. (11) and (12):

$$\vec{A} = 2\vec{a} \times \vec{r}_1 - \vec{a}, \tag{11}$$

$$\vec{C} = 2 \times \vec{r}_2. \tag{12}$$

In these relations, the components of vector *a* are selected in a linear decreasing manner from 2 to 0 during the iteration. Also, r_1 and r_2 are random vectors in the range 0 and 1.

Step 4 Prey Hunting: In the search space, we have no knowledge of the position of the prey vector, so we consider the position of the prey vector to be the average of the positions of α, β, and δ, and the hunting equations for each of the wolves in group ω are formed as follow Eq. (13):

$$\begin{aligned}
&\vec{X}_1(t) = \vec{X}_\alpha(t) - \vec{A}_1 \times \vec{D}_\alpha(t), \ \vec{D}_\alpha(t) = \left|\vec{C}_1 \times \vec{X}_\alpha(t) - \vec{X}(t)\right| \\
&\vec{X}_2(t) = \vec{X}_\beta(t) - \vec{A}_2 \times \vec{D}_\beta(t), \ \vec{D}_\beta(t) = \left|\vec{C}_2 \times \vec{X}_\beta(t) - \vec{X}(t)\right| \\
&\vec{X}_3(t) = \vec{X}_\delta(t) - \vec{A}_3 \times \vec{D}_\delta(t), \ \vec{D}_\delta(t) = \left|\vec{C}_3 \times \vec{X}_\delta(t) - \vec{X}(t)\right|
\end{aligned} \tag{13}$$

The new position for the desired wolf is obtained from the following Eq. (14):

$$\vec{X}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3}. \tag{14}$$

Step 5 The termination conditions are checked and if necessary, repeat the siege and baiting steps.

4.1.2 Chaos-Embedded Grey Wolf Optimizer (CGWO)

In order to determine the new location for the wolves of group ω, two exploration and exploitation strategies are considered in Eqs. (11) and (12). By replacing the chaos functions in the random selections related to these steps, we will have a significant improvement in the performance of the algorithm. The proposed scenarios for performing this replacement are as follows:

Scenario 1 Replacing the chaos function in the exploration phase: In this case, the first chaos function CHM_1 in Eq. (11). replaces the random selection term. The results of applying the chaos function will be as follow Eq. (15):

$$\vec{A} = 2\vec{a} \times C\vec{H}M_1 - \vec{a}. \tag{15}$$

Scenario 2 Replacing the chaos function in the exploration phase: In this case, the second chaos function CHM_2 is replaced in Eq. (12). to determine the values of the vector *C*. For this purpose, we have Eq. (16):

$$\vec{C} = 2 \times C\vec{H}M_2. \tag{16}$$

Scenario 3 Chaos function replacement in both stages: In this case, the chaos function is applied simultaneously in both stages and to replace random selections.

4.2 Standard Crow Search Algorithm (CSA)

This algorithm was presented in 2016 by Askarzadeh et al. [9] based on the behavior of crows. Crows are considered to be among the most intelligent birds. They can remember faces, give warnings when face with unfriendly behavior, and in most cases remember the hiding places of their food for months afterwards. They also track and remember where other birds hide their food and steal it at the right time, but if their hideout is stolen, they take extra precautions, such as moving hiding places. The social behavior of crows is very similar to the optimization process. Based on this behavior, crows hide their excess food in certain locations and retrieve the stored food when needed. Crows are greedy birds and imitate each other to better access food sources. Finding a hidden food source by a crow is not an easy task because if a crow realizes that another is following it, it will try to deceive that crow by going to other locations. From an optimization perspective, crows are searchers and evaluate each location of the examined space with the quality of the food source as it is. In presenting the Crow Search Algorithm (CSA), the following principles are considered.

- Crows live in groups.
- Crows maintain their roosting positions.
- They chase each other to steal from their roosting positions.
- Crows protect their roosting positions from thieves.

4.2.1 Basic steps in Crow Search Algorithm (CSA)

Step 1 Determine the algorithm parameters including: crow population, search space dimensions, lower and upper bounds for decision variables, maximum number of iterations, flight duration, and probability of wakefulness.

Step 2 Determining the initial position and memory of the crows: Within the search space, scattered points from the interval related to the decision variables are assigned as the initial positions of the crows. The initial position assignment is done randomly. The information matrix related to the crows' positions is as Eq. (17):

$$[\text{Crows}] = \begin{bmatrix} x_1^1 x_2^1 \ldots x_d^1 \\ x_1^2 x_2^2 \ldots x_d^2 \\ . \; . \; . \; . \\ . \; . \; . \; . \\ x_1^N x_2^N \ldots x_d^N \end{bmatrix}. \tag{17}$$

The memory matrix for each crow is formed according to Eq. (18). Since in the first iteration the crows still have no experience, it is assumed that they are hiding their food in the initial positions:

$$[\text{Memory}] = \begin{bmatrix} m_1^1 m_2^1 \ldots m_d^1 \\ m_1^2 m_2^2 \ldots m_d^2 \\ . \; . \; . \; . \\ . \; . \; . \; . \\ m_1^N m_2^N \ldots m_d^N \end{bmatrix}. \tag{18}$$

Step 3 Evaluate the objective function values: The food quality of each crow is calculated by entering its position components into the objective function.

Step 4 Determining the new position of each crow: In this step, crow *i* randomly selects one of the crows in the group, such as *j*, and follows it to discover the location of the food hidden by the crow. Therefore, the new position of the crow is obtained by Eqs. (14)–(17). as follow Eqs. (19)–(21):

$$\text{if} \quad \text{rand}_j \geq AP^{j,iter}, \tag{19}$$

$$x^{i,iter+1} = x^{i,iter} + \text{rand}_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter} \right), \tag{20}$$

$$\begin{aligned} &\text{else} \\ &x^{i,iter+1} = \text{a random position of search space.} \end{aligned} \tag{21}$$

In these equations, AP is the crow's alertness coefficient and fl represents the flight length. And the values rand_i and rand_j are random values in the range 0 and 1 with uniform distribution.

Step 5 Checking the new crow positions: The new positions are compared with the lower and upper bounds of the decision variables. If these positions are within the acceptable range, the previous positions are replaced with these positions and the results are updated, but otherwise the previous positions are not changed.

Step 6 The objective function value is evaluated for the new position of each crow.

Step 7 The crows' memory is updated according to Eq. (22):

$$m^{i,iter+1} = \begin{cases} m^{i,iter+1} & f\left(x^{i,iter+1}\right) \text{ is better than } f\left(x^{i,iter}\right) \\ m^{i,iter} & \text{otherwise} \end{cases}. \tag{22}$$

According to this equation, if the value of the objective function for each crow improves in the new position, the crow's previous memory value is replaced with the new position. Otherwise, the previous position is maintained.

Step 8 the termination conditions are checked and if necessary, the steps from the fourth step are repeated. Otherwise, the operation is terminated.

4.2.2 Chaos-Embedded Crow Search Algorithm (CCSA)

To determine the new location of the crows, two exploration and exploitation strategies are considered in Eqs. (19) and (20). By replacing the chaos functions in the random selections related to these steps, we will have a significant improvement in the performance of the algorithm. The proposed scenarios for performing this replacement are as follows:

Scenario 1 Replacing the chaos function in the exploration phase: In this case, the first chaos function CHM_1 in Eq. (19). replaces the random selection term. The results of applying the chaos function will be as Eq. (23):

$$\text{if } CHM_1 \geq AP^{j,iter}. \tag{23}$$

Scenario 2 Chaos Function Substitution in the Exploitation Phase: In this case, the second chaos function CHM_2 is substituted into Eq. (20) to determine the random values rand_i. By performing this substitution, Eq. (24) is obtained as follow Eq. (24):

$$x^{i,iter+1} = x^{i,iter} CHM_2 \times fl^{i,iter} \times \left(m^{i,iter} - x^{i,iter} \right). \tag{24}$$

Scenario 3 Chaos Function Substitution in Both Stages: In this case, the chaos function is applied simultaneously in both stages and to replace the random choices of Eqs. (19) and (20).

4.3 Standard Cyclical Parthenogenesis Algorithm (CPA)

This algorithm considers the fundamental aspects of aphid life. Two types of reproduction with mating and without mating allow aphids to exploit favorable conditions and rapidly grow their population. The proposed algorithm, like most algorithms, is population-based and was proposed in 2017 by Kaveh and Zolghadr [10]. In reproduction with mating, two different solutions share

information, while in conditions without mating, a new solution is formed using only the information of a female parent. Using this inspiration in metaheuristic algorithms creates the best conditions for creating a balance between exploration and exploitation and in most cases prevents the operation from stopping at local optima and premature convergence. In the presented algorithm, first for the with mating stage, the initial responses are spread across wide areas of the search space, and then for the without mating stage, the selected responses are analyzed more carefully in the neighborhood.

4.3.1 Basic steps in Cyclical Parthenogenesis Algorithm

Step 1 Formation of parent aphids: In the interval of decision variables, the initial aphids are formed and considered as the parent population. Their formation is random and occurs according to the following Eq. (25):

$$x_{ij}^{0} = x_{j,\min} + \text{rand}\left(x_{j,\max} - x_{j,\min}\right) \quad j = 1,2,...,n_A. \tag{25}$$

In this relation, X_{ij}^{0} specifies the j_{th} component of the i_{th} aphid population, $X_{j,\max}$ and $X_{j,\min}$ indicate the upper and lower bounds of the decision variables, respectively. The total number of aphids is equal to n_A, which is placed in n_C colonies, each with a population of n_M. Accordingly, it is clear that the population size of each colony is obtained by dividing the total population by the number of colonies, and n_M is also constant during the optimization operation.

Step 2 Procreation of the offspring population: To form offspring aphids in each colony, $F_r \times n_M$ number of offspring are formed without mating. The parents of these aphids are female and their selection is done randomly and from the best answers. The Matlab coding for this stage and the formation of the offspring population is according to the Eqs. (26) and (27):

$$rf_i = \text{round}\left(1 + \left(Fr \times nM - 1\right).\times \mathbf{rand}\right), \tag{26}$$

$$x_{ij}^{k+1} = F_j^k + \alpha_1 \times \frac{\text{randn}}{NITs} \times \left(x_{j,\max} - x_{j,\min}\right). \quad j = 1,2,...,n_A \tag{27}$$

In Eq. (26), the index related to the female parent is introduced, and in Eq. (27), new offspring related to the without mating are formed in the new cell array. Next, it is the turn to form fertilized aphids. The number of these offspring $(1–F_r)_{xnM}$, in which each male parent M and a female parent F are randomly selected, and according to Eq. (28), new offspring related to the with mating state are placed in the new cell array:

$$x_{ij}^{k+1} = M_j^k + \alpha_2 \times \text{rand} \times \left(F_j^k - M_j^k\right) \quad j = 1,2,...,n_A. \tag{28}$$

Step 3 Flight of the best aphid and death of the worst aphid: After the formation of a new generation of offspring, the objective function is evaluated and with the probability P_f one of the best winged aphids is selected from colony 1 and, by reproducing, it replaces the worst aphid in colony 2. In order to keep the colony population constant, the removal of the worst aphid from colony 2 is likened to the death of the aphid and the replacement of the best aphid with flight. The probability related to this step is based on Eq. (29):

$$pf = \frac{NITs - 1}{\max NITs - 1}. \tag{29}$$

Step 4 Replacing the best aphids: In each colony, the population of parents is compared with the offspring, and from among them, n_M of the best ones is selected to form the next generation.

Step 5 The termination conditions are checked and if necessary, the operation is repeated from step 2.

In this regard, bestT is the best team so far and *NITs* is a repeat number.

4.3.2 Chaos-Embedded Cyclical Parthenogenesis Algorithm (CCPA)

The two main stages for the birth and formation of the offspring population in aphids are reproduction with mating and without mating. The balance between these stages solves the problem of premature convergence in the algorithm. Reproduction with mating plays the role of exploration and without mating plays the role of exploitation. By using chaos functions in the random selections of these two stages, significant chaos mutations are obtained and the algorithm is saved from the trap of local optima. The replacement of chaos functions with several scenarios is done as follows.

Scenario 1 Replacement of chaos function in the without mating reproduction stage: In this case, the first chaos function ***CHM*1** introduced in Eq. (26) is applied in the standard case, and chaotic selection replaces random selection. This replacement is shown in Eq. (30):

$$rf_i = \text{round}\left(1 + \left(Fr \times nM - 1\right).\times \boldsymbol{CHM1}\right). \tag{30}$$

Scenario 2 Replacing the chaos function in the reproductive stage with mating: In this case, the second chaos function ***CHM*2** introduced in Eq. (28) is applied in the standard case, and chaotic selection replaces random selection. The results will be in accordance with Eq. (31):

$$x_{ij}^{k+1} = M_j^k + \alpha_2 \times \boldsymbol{CHM2} \times \left(F_j^k - M_j^k\right) \quad j = 1,2,...,n_A. \tag{31}$$

Scenario 3 Replacing the chaos function in both stages simultaneously: In this case, both chaos functions are applied simultaneously in Eqs. (26) and (28), and chaotic choices replace random choices.

5 Numerical Examples of Optimal Truss Design

In order to compare the presented metaheuristic algorithms in standard and chaotic modes, we consider well-known examples of truss structures. To determine the optimal design in truss structures, the main goal is to design the smallest cross-sectional area of each member that can satisfy the design constraints. The design constraints are determined based on the steel structure code and include providing the allowable deformation at the nodes, observing the allowable stress and the slenderness range of the members [30]. Two steps are considered for modeling each algorithm for the optimal design. In the first stage, the selected metaheuristic algorithm competes in the standard mode with 6 chaotic states with different functions and scenarios. In the next stage, the competition is carried out between the selected chaotic states themselves, and finally, with a wide range of investigations and a wide variety of possible solutions from the space of decision variables, a quasi-optimal state is selected [31]. In this research, we will have a significant challenge and competition to move towards the global optimum, and the best metaheuristic algorithm, the best chaotic function, and the best scenario will win the competition.

5.1 A 120-bar Dome Shaped Truss

Fig. 3 shows a 120-bar dome-shaped truss. This truss dome is a well-known example, which is often used by researchers to investigate the effectiveness of metaheuristic algorithms. The numbering of nodes and members, as well as the geometric structure of the structure, is presented in the figure. The specific weight of the materials used in the construction of the structure is 0.288 lb/in^3 and the modulus of elasticity of the structural members is 30,450 klb/in^2. The AISC ASD code has been selected to determine the allowable stress of the tensile and compressive members. Based on the code relations, these stresses must be within the range introduced in Eqs. (32)–(34). To apply the deformation limit of the nodes, the maximum deformation of each node in all directions is introduced as 0.1969 in. Also, the loading of the structure is applied to all non-supported nodes. The intensity of the applied loads

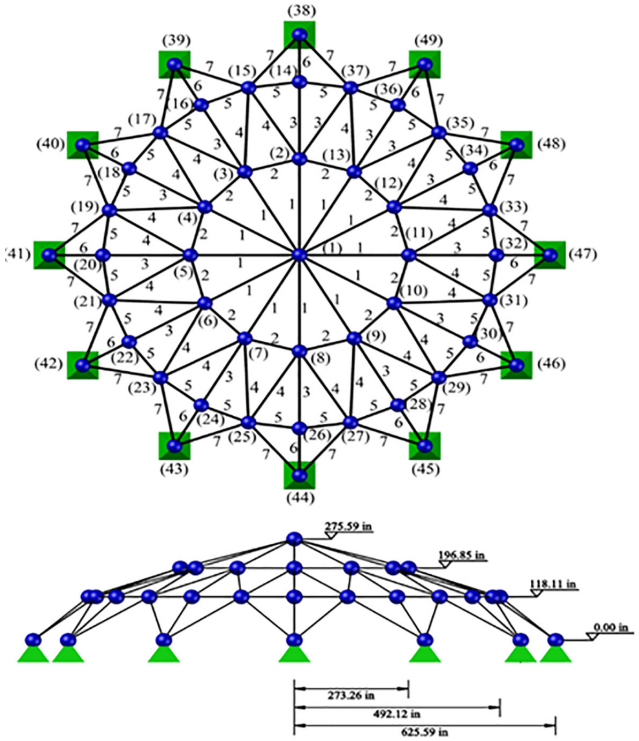

Fig. 3 Schematic of a 120-bar dome shaped truss

is –13.49 kips at node 1, –6.744 kips at nodes 2 to 14, and –2.248 kips at the remaining nodes, respectively. The geometric symmetry of the structure with respect to the longitudinal and transverse axes has been used to classify the dome members, and all members have been considered in seven groups according to the proposed figure. In the optimization operation, the range of decision variables is 0.775 in^2 and 20 in^2. The allowable stress in tension and pressure is determined based on Eq. (32):

$$\sigma_i = \begin{cases} 0.6F_y & \text{for} \quad \sigma_i \geq 0 \\ \sigma_i^- & \text{for} \quad \sigma_i < 0 \end{cases}. \tag{32}$$

For compressive stresses, we have the following Eq. (33):

$$\sigma_i^- = \begin{cases} \left(1 - \dfrac{\lambda^2}{2C^2}\right) F_y / FS & \text{for} \quad \lambda < C \\ \dfrac{12\pi^2 E}{23\lambda^2} & \text{for} \quad \lambda \geq C \end{cases}. \tag{33}$$

In this equation we have the Eq. (34):

$$FS = \left(\frac{5}{3} + \frac{3\lambda}{8C} - \frac{\lambda^3}{8C^3}\right); \; C = \sqrt{\left(\frac{2\pi^2 E}{F_y}\right)}; \\ \lambda = \frac{kl}{r}; \; r = aA^b; \; a = 0.4993; \; b = 0.6777. \tag{34}$$

The parameters used include E the modulus of elasticity, F_y the yield stress of the steel, C the boundary slenderness ratio that separates the elastic and inelastic buckling regions compared to the existing slenderness λ. Also, k is the effective length factor of the member and r is the radius of gyration of the member. To facilitate the application of code constraints, four cases are considered. These cases are:

- Case 1: We consider stress constraints without deformation constraints.
- Case 2: Both stress and deformation constraints are considered along the X and Y axes.
- Case 3: Deformation constraints are considered only along the Z axis and stress constraints are not considered.
- Case 4: All constraints are considered simultaneously.

Also, the range of decision variables for modes 1 and 2 is proposed to be between 0.775 and 5 in^2, and for modes 3 and 4 between 0.775 and 20 in^2. To form statistical information, each of the standard and chaotic metaheuristic algorithms was run 30 times independently, and the results related to the best response, the average of the responses, and the coefficient of variation were determined. The coefficient of variation of the responses can be useful in analyzing the statistical results. This coefficient is a good criterion for expressing the stability of the responses, the sensitivity of the responses, and the correct selection of the specific parameters of each algorithm. The statistical results of all optimization models are presented in Table 1. Also, for quick access to the optimization results and comparison of the competition of the seven groups in the standard and chaotic modes, a bar chart is displayed in Fig. 4. The convergence history of the optimization operation for 140 iterations indicates the good speed of the chaotic modes in moving towards the optimal responses. The diagram of this history is drawn in Fig. 5.

For the final comparison and competition between chaotic modes, Table 2 is presented. In this table, the abbreviation "NFE" represents the Number of Evaluations of the objective Function. By examining this table, a significant improvement in the optimization results has been achieved. These results for each of the algorithms are:

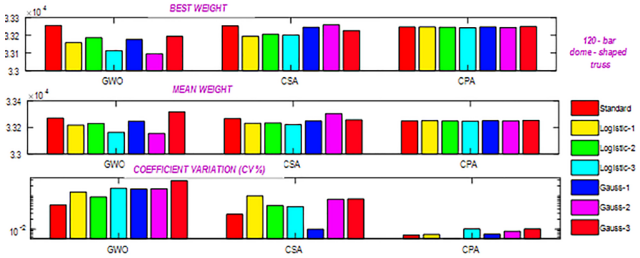

Fig. 4 Optimization results in standard mode and selection of chaos map for the 120-bar dome shaped truss

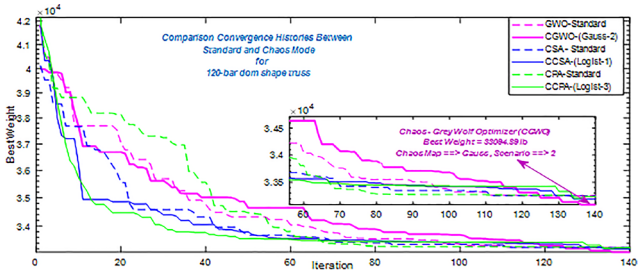

Fig. 5 The convergence histories for the 120-bar dome shaped truss

Table 1 Statistical results for the 120-bar dome shaped truss

Algorithms	Statistical Information	GWO Standard	CGWO-21 Logist-1	CGWO-22 Logist-2	CGWO-23 Logist-3	CGWO-31 Gauss-1	CGWO-32 Gauss-2	CGWO-33 Gauss-3
1-GWO	Best	33255.701	33158.269	33186.433	33112.897	33176.406	33094.890	33193.978
	Mean	33269.944	33218.549	33230.189	33162.690	33246.495	33153.968	33317.504
	C.V (%)	0.053949	0.13238	0.09323	0.16841	0.16167	0.16487	0.28685
Algorithms	Statistical Information	CSA Standard	CCSA-21 Logist-1	CCSA-22 Logist-2	CCSA-23 Logist-3	CCSA-31 Gauss-1	CCSA-32 Gauss-2	CCSA-33 Gauss-3
2-CSA	Best	33253.664	33194.593	33205.740	33200.959	33245.059	33259.449	33226.744
	Mean	33267.026	33230.911	33233.960	33221.073	33248.359	33304.662	33257.065
	C.V (%)	0.028281	0.10013	0.0516	0.04785	0.009988	0.07990	0.080873
Algorithms	Statistical Information	CPA Standard	CCPA-21 Logist-1	CCPA-22 Logist-2	CCPA-23 Logist-3	CCPA-31 Gauss-1	CCPA-32 Gauss-2	CCPA-33 Gauss-3
3-CPA	Best	33246.780	33247.208	33246.018	33242.481	33247.394	33244.960	33249.621
	Mean	33250.007	33250.872	33248.227	33247.606	33250.258	33249.157	33252.504
	C.V (%)	0.00667	0.00697	0.00524	0.01015	0.00713	0.00867	0.01021

Table 2 Optimal design comparison for the 120-bar dome shaped truss

Number group	GWO Standard	CGWO Gaus-2	CSA Standard	CCSA Logis-1	CPA Standard	CCPA Logis-3
1	3.02619	3.02432	3.02503	3.02805	3.02512	3.02412
2	14.5694	14.755	14.6564	13.776	14.591	14.7935
3	5.1921	5.19525	5.06365	5.44826	5.17932	5.06922
4	3.14234	2.94748	3.1377	3.01136	3.13214	3.12896
5	8.45703	8.55476	8.49103	8.81836	8.47334	8.48944
6	3.33359	3.32646	3.39259	3.36933	3.32423	3.28278
7	2.49437	2.49697	2.49522	2.4968	2.49491	2.49676
Best Weight (lb)	33255.701	33094.890	33253.664	33194.593	33246.780	33242.481
Mean Weight (lb)	33269.944	33153.968	33267.026	33230.911	33250.007	33247.606
Coefficient Var (CV)	0.053949	0.16487	0.028281	0.10013	0.00667	0.01015
NFE	11280	11280	11280	11280	11280	11280

- In the Gray Wolf optimizer (GWO), the Gaussian chaos function with scenario 2 has reduced the total weight of the structure to 33094.890 lb.
- In the Crow Search Algorithm (CSA), the logistic chaos function with scenario 1 has reduced the total weight of the structure to 33194.593 lb.
- In the Cyclical Parthenogenesis Algorithm (CPA), the logistic chaos function with scenario 3 has reduced the total weight of the structure to 33242.481 lb.

By comparing chaotic algorithms, the Chaotic Gray Wolf Optimizer (CGWO) resulted in the lowest weight for the structure at 33094.890 lb.

5.2 A 200-bar planar truss structure

Fig. 6 shows the 200-bar planar truss. This truss structure is one of the well-known examples, which is often used to investigate the effectiveness of metaheuristic algorithms. The numbering of nodes and members, as well as the geometric structure of the structure, is presented in the figure. The specific weight of the materials used in the construction of the structure is 0.288 lb/in^3 and the modulus of elasticity of the members is suggested to be 30,000 klb/in^2. The allowable stress for the members is determined to be 10 klb/in^2 and there is no deformation limit for the nodes. The loads applied to the truss are examined in three groups. These groups are independent. The *first group* is the lateral load of the structure, which includes a load of 1 klb in the positive direction of the X-axis and is applied at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, and 71. The *second group* includes the gravity load of the structure, which includes a load of 10 klb in the negative direction of the Y-axis and is applied at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, and . . . . 71, 72, 73, 74, and 75. The *third group* applies both loading groups together. To facilitate optimal design, the truss members are classified into 29 groups. For the optimization operation, the decision variable range is 0.1 in^2 lower bound and 16 in^2 upper bound.

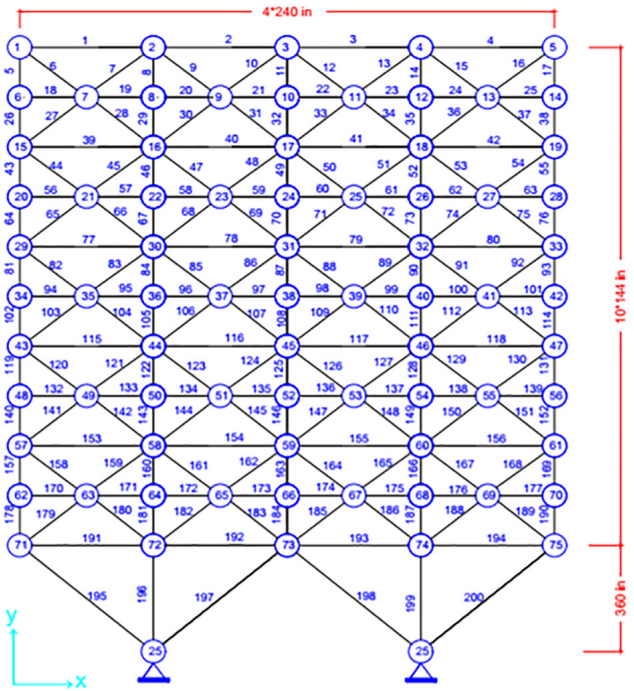

Fig. 6 Schematic of a 200-bar planar truss structure

To access statistical information, each of the standard and chaotic metaheuristic algorithms was run 30 times independently and the results related to the best response, average responses and coefficient of variation are presented. The coefficient of variation of responses can be useful in analyzing statistical results. This coefficient is a good criterion for expressing the stability of responses, sensitivity of responses and correct selection of specific parameters of each algorithm.

Statistical information of all optimization models is presented in Table 3. Also, for quick access to the optimization results and comparison of the competition of seven groups in standard and chaotic modes, a bar chart is displayed in Fig. 7. The convergence history of the optimization operation for 150 iterations indicates the good speed of chaotic modes in moving towards optimal responses. The diagram of this history is drawn in Fig. 8.

For the final comparison and competition between chaotic modes, Table 4 is presented. By examining this table, a significant improvement in the optimization results has been achieved. These results for each of the algorithms are:

- In the Gray Wolf Optimizer (GWO), Gaussian chaos function with scenario 1, the total weight of the structure has been reduced to 24,659.21 pounds.
- In the Crow Search Algorithm (CSA), Gaussian chaos function with scenario 2, the total weight of the structure has been reduced to 24,806.36 pounds.
- In the Cyclical Parthenogenesis Algorithm (CPA), Gaussian chaos function with scenario 2, the total weight of the structure has been reduced to 25117.10 pounds.

By comparing chaotic algorithms, the Chaotic Gray Wolf Optimizer (CGWO) resulted in the lowest weight for the structure at 24659.21 pounds.

5.3 A 582-bar Tower Truss Structure

The 582-bar tower truss is a well-known example, which is often used by researchers to investigate the effectiveness of metaheuristic algorithms. The dimensional specifications and member numbering for this structure are shown in Fig. 9. Weight optimization of this structure with continuous variables was first proposed by Kaveh et al. In the optimization relations, the modulus of elasticity of the structural members is considered to be 203893.6 MPa. The steel recommended for the design of the structure with a yield stress of 253.1 MPa. To apply the deformation limit of the nodes, the maximum deformation of each node in all directions has been determined to be 8 cm. The limit for slenderness of the members in tension parts is maximum 300 and for compression parts is maximum 200. The allowable stress of the tensile and compressive members is selected according to

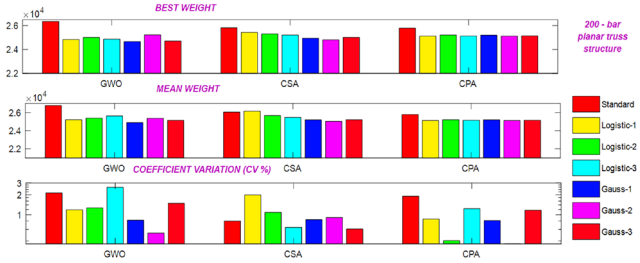

Fig. 7 Optimization results in standard and the chaos map for the 200-bar planar truss structure

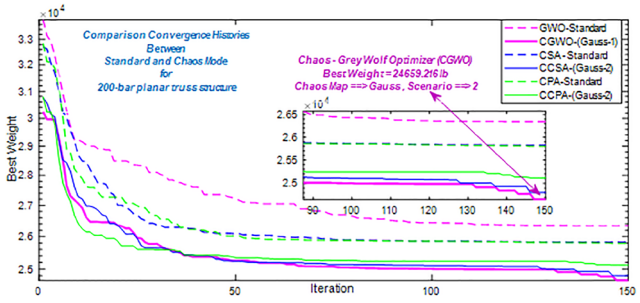

Fig. 8 The convergence histories for the 200- bar planar truss structure

Algorithms	Statistical Information	GWO Standard	CGWO-21 Logist-1	CGWO-22 Logist-2	CGWO-23 Logist-3	CGWO-31 Gauss-1	CGWO-32 Gauss-2	CGWO-33 Gauss-3
1-GWO	Best	26343.007	24837.033	25020.260	24867.351	24659.216	25226.472	24716.553
	Mean	26777.799	25203.961	25386.605	25634.988	24891.138	25362.004	25129.958
	C.V (%)	2.1277	1.1833	1.2670	4.0809	0.82486	0.52692	1.4791
Algorithms	Statistical Information	CSA Standard	CCSA-21 Logist-1	CCSA-22 Logist-2	CCSA-23 Logist-3	CCSA-31 Gauss-1	CCSA-32 Gauss-2	CCSA-33 Gauss-3
2-CSA	Best	25827.658	25428.709	25307.501	25217.796	24938.073	24806.363	25021.539
	Mean	26050.933	26148.966	25660.188	25460.188	25193.651	25028.433	25208.448
	C.V (%)	0.79915	1.9823	1.0764	0.64267	0.84013	0.90168	0.60548
Algorithms	Statistical Information	CPA Standard	CCPA-21 Logist-1	CCPA-22 Logist-2	CCPA-23 Logist-3	CCPA-31 Gauss-1	CCPA-32 Gauss-2	CCPA-33 Gauss-3
3-CPA	Best	25791.455	25118.590	25210.114	25137.361	25198.006	25117.108	25143.099
	Mean	26365.253	25348.538	25333.791	25436.294	25441.905	25250.84	25528.918
	C.V (%)	1.9009	0.85254	0.40345	1.2308	0.81384	0.36203	1.1622

Table 3 Statistical results for the 200- bar planar truss structure

Table 4 Optimal design comparison for 200-- bar planar truss structure

Number group	Element group	GWO Standard	CGWO-31 Gauss-1	CSA Standard	CCSA-32 Gauss-2	CPA Standard	CCPA-32 Gauss-2
1	1,2,3,4	0.309047	0.102001	0.1	0.1	0.100045	0.100339
2	5,8,11,14,17	1.50311	0.967523	1.49672	1.49672	0.957302	1.14968
3	19,20,21,22,23,24	0.100175	0.175671	0.100769	0.100769	0.1	0.100008
4	18,25,56,63,94,101,132,139,170,177	0.116486	0.105626	0.1	0.1	0.1	0.111659
5	26,29,32,35,38	2.19921	1.94261	2.05603	2.05603	1.9507	2.03737
6	6,7,9,10,12,13,15,16,27,28,30,31,33,34, 36,37	0.391609	0.288587	0.275154	0.275154	0.292773	0.264223
7	39,40,41,42	0.243918	0.164868	0.150466	0.150466	0.331111	0.169507
8	43,46,49,52,55	3.22107	3.03419	3.49775	3.49775	3.07893	3.07121
9	57,58,59,60,61,62	0.187243	0.120591	0.129516	0.129516	0.1	0.1
10	64,67,70,73,76	5.37341	4.03326	4.42528	4.42528	4.07896	4.06498
11	44,45,47,48,50,51,53,54,65,66,68,69,71, 72,74,75	0.454666	0.424282	0.449637	0.449637	0.503	0.425966
12	77,78,79,80	0.161817	0.14102	0.1	0.1	0.1	0.108528
13	81,84,87,90,93	5.56815	5.26571	5.77794	5.77794	5.49553	5.37705
14	95,96,97,98,99,100	0.160035	0.10858	0.15415	0.15415	0.114396	0.102744
15	102,105,108,111,114	6.49948	6.26106	6.3296	6.3296	6.49354	6.37358
16	82,83,85,86,88,89,91,92,103,104,106,107, 109,110,112,113	0.610523	0.520442	0.627061	0.627061	0.661921	0.516515
17	115,116,117,118	0.130033	0.733951	0.139984	0.139984	0.108615	0.203728
18	119,122,125,128,131	8.07418	7.67938	7.84931	7.84931	8.02395	7.75462
19	133,134,135,136,137,138	0.108138	0.111756	0.107096	0.107096	0.214248	0.134733
20	140,143,146,149,152	9.07201	8.72157	9.02625	9.02625	9.00663	8.74731
21	120,121,123,124,126,127,129,130,141,142, 144,145,147,148,150,151	0.724611	0.972569	0.723246	0.723246	0.751787	0.729385
22	153,154,155,156	0.938049	0.308911	0.830863	0.830863	0.778415	0.953673
23	157,160,163,166,169	11.2392	10.8471	10.8209	10.8209	11.1788	10.7196
24	171,172,173,174,175,176	0.201731	0.168948	0.232744	0.232744	0.12962	0.1
25	178,181,184,187,190	12.2678	11.5303	12.1174	12.1174	12.1315	11.5497
26	158,159,161,162,164,165,167,168,179,180, 182,183,185,186,188,189	1.42721	1.0688	1.43102	1.43102	1.29886	1.29949
27	191,192,193,194	5.47571	5.53726	5.62783	5.62783	5.82254	5.96218
28	195,197,198,200	9.94375	10.5145	10.3616	10.3616	10.2569	10.5188
29	196,199	14.7038	11.6449	13.691	13.691	14.4343	12.9404
	Best Weight (lb)	26343.01	24659.21	25827.65	24806.36	25791.45	25117.10
	Mean Weight (lb)	26777.79	24891.13	26050.93	25028.43	26365.25	25250.84
	Coefficient Var (CV)	2.1277	0.82486	0.79915	0.90168	1.9009	0.36203
	NFE	14345	14345	14345	14345	14345	14345

the AISC ASD proposed relationships, which are presented in Eqs. (32)–(34). It should be noted that the components used include E the modulus of elasticity, F_y the yield stress of the steel, C the boundary slenderness ratio that separates the elastic and inelastic buckling regions compared to the existing slenderness λ. Also, k is the effective length factor of the members and r is the radius of gyration of the members. The effective length factor is considered to be 1 for all members. The loads acting on the tower are considered in a group that includes lateral loads of 5 kN in both the X and Y axes and gravity loads of –30 kN in the Z axis. This loading is applied to all free nodes of the tower.

To facilitate optimization, all members of the tower have been classified into types. Based on the geometric

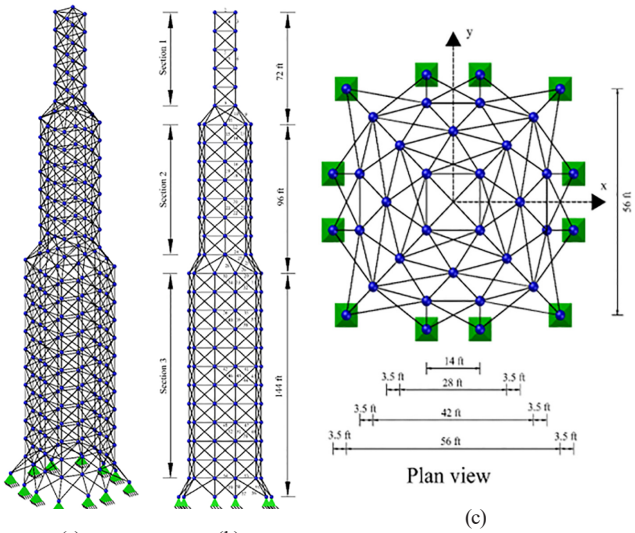

Fig. 9 Schematic of a 582-bar tower truss structure (a) 3D view, (b) side view, (c) top view

symmetry of the structure with respect to the longitudinal and transverse axes as well as the floors, the number of type groups is 32 groups, which are applied in the structural model as shown in the figure. Another information item for optimization models is the introduction of the decision variable range. For this structure, the lower limit of the decision variables is 20 cm^2 and the upper limit is 1000 cm^2. To access statistical information, each of the standard and chaotic metaheuristic algorithms was run 30 times independently and the results related to the best response, the average of the responses, and the coefficient of variation were determined.

The coefficient of variation of responses can be useful in analyzing statistical results. This coefficient is a good criterion for expressing the stability of responses, the sensitivity of responses, and the correct selection of specific parameters for each algorithm. The statistical results of all optimization models are presented in Table 5. Also, for quick access to the optimization results and comparison of the competition of seven groups in standard and chaotic modes, a bar chart is displayed in Fig. 10. The convergence history of the optimization operation for 180 iterations indicates the good speed of chaotic modes in moving towards optimal responses. The diagram of this history is drawn in Fig. 11. For the final comparison and competition between chaotic modes, Table 6 is presented. By examining this table, a significant improvement in the optimization results has been achieved. These results for each of the algorithms are:

- In the Gray Wolf Optimization (GWO), the Gaussian chaos function with scenario 2 has reduced the total volume of the structure to 15.5347 m^3.
- In the Crow Search Algorithm (CSA), the logistic chaos function with scenario 1 has reduced the total volume of the structure to 15.6507 m^3.
- In the Cyclical Parthenogenesis Algorithm (CPA), the Gaussian chaos function with scenario 2 has reduced the total volume of the structure to 15.4171 m^3.

By comparing chaotic algorithms, the Chaotic Cyclical Parthenogenesis Algorithm (CCPA), has resulted in the lowest volume for the structure, which is 15.4171 m^3.

6 Discussion of the algorithms

In a recent study, three well-known metaheuristic algorithms from the group of swarm intelligence selection and the reasons for stagnation in reaching global optima have been examined. In selection algorithms, in cases where the models under study have a large number of variables, the balance between the two basic stages of exploration and exploitation is interrupted and the algorithm stops at local optima. We can interpret this situation as premature convergence, and in the terminology of some algorithm designers, they say that the curse of dimensionality has affected the algorithm. Although in some algorithms, they have tried to escape the trap of local optima by designing a jump stage and move towards global optima, research shows that the jump stage of the algorithms itself is not productive in the standard mode and a fundamental solution must be considered. In this regard, chaos functions have a good potential to establish a balance between exploration and exploitation [23]. Also, in order to have the same conditions, the initial population in Examples 1, 2, and 3 is set to 80, 95, and 100, respectively. The mutation rate is 10% at the beginning of the optimization, which gradually decreases to 3% at the end. Chaos functions in each of the algorithms improve the weakness of imbalance in a way. In metaheuristic algorithms, functions with different probability distributions are used for each of the search and extraction stages to create diversity in the search space. The type of probability distribution varies depending on the algorithm, the most famous of which are normal, Cauchy, and Levy distributions. However, none of these functions have the power of chaotic mutation, and we will need to replace these chaotic functions at different positions in the algorithm to improve the optimization results. Therefore, three different scenarios are formed, and ultimately, in one of them, the greatest balance between the exploration and exploitation stages is achieved. The chaos functions

Table 5 Statistical results for the 582-bar tower truss structure

Algorithms	Statistical Information	GWO Standard	CGWO-21 Logist-1	CGWO-22 Logist-2	CGWO-23 Logist-3	CGWO-31 Gauss-1	CGWO-32 Gauss-2	CGWO-33 Gauss-3
1-GWO	Best	19.85572	19.92493	21.9985	20.23321	16.57130	15.5347	16.89907
	Mean	20.5127	20.18630	22.3221	21.1175	16.6794	15.5881	17.00230
	C.V (%)	4.85650	1.46700	1.81628	4.96328	0.59908	0.29972	0.57361
Algorithms	Statistical Information	CSA Standard	CCSA-21 Logist-1	CCSA-22 Logist-2	CCSA-23 Logist-3	CCSA-31 Gauss-1	CCSA-32 Gauss-2	CCSA-33 Gauss-3
2-CSA	Best	18.19611	15.6507	18.7133	19.08721	16.57129	19.8431	19.86427
	Mean	18.80231	15.7227	19.3194	19.74361	17.2139	20.5212	20.0212
	C.V (%)	3.54559	0.42112	2.6706	3.83921	3.75855	3.96613	0.91786
Algorithms	Statistical Information	CPA Standard	CCPA-21 Logist-1	CCPA-22 Logist-2	CCPA-23 Logist-3	CCPA-31 Gauss-1	CCPA-32 Gauss-2	CCPA-33 Gauss-3
3-CPA	Best	20.71633	18.5672	19.3742	19.1187	17.7283	15.4171	16.9693
	Mean	21.52711	19.2832	19.8221	20.0463	18.9598	15.4439	17.0124
	C.V (%)	4.55080	5.29270	3.31095	6.58697	6.7978	0.19811	0.31860

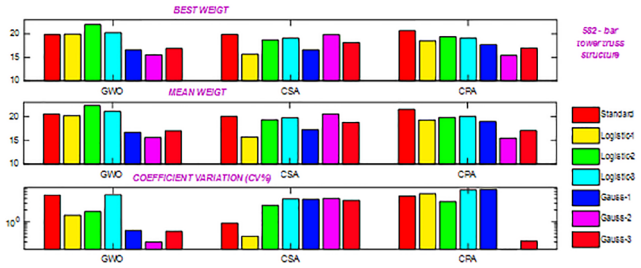

Fig. 10 Optimization results in standard and the chaos map for the 582 bar tower truss structure

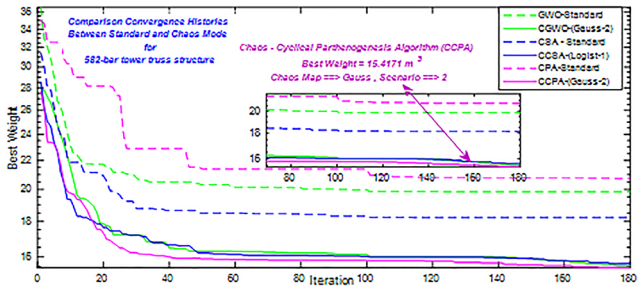

Fig. 11 The convergence histories for the 582-bar tower truss structure

themselves also have different conditions. In a number of chaos functions, the search space converges with a very high probability from local minima to global minima, in which case these functions are suitable for improving the search conditions, while in another group of chaos functions, the search space converges with a very high probability to local minima and are suitable for improving the extraction conditions. The result is that by selecting at least two chaos functions and each with three scenarios, the statistical space required by the recent study has been formed [32]. Since the results are based on discrete structural examples, we will need to contribute the results of all the stages to combine and summarize the information. Using Eq. (35), a general conclusion regarding the success rate of each chaos mode with comprehensive participation will be possible:

$$\mathrm{Val}_{\mathrm{com}}^{MV}=\frac{1}{S}\sum_{i=1}^{S}\left(\frac{\mathrm{Val}^{MV}}{\mathrm{Val}_{i,\min}}-1\right). \tag{35}$$

In the introduced relationship, an attempt is made to apply the results of all models and extract the success percentage for each of the states, including the standard state and six chaotic states. By combining the results of all examples comparing the efficiency of standard and chaotic modes for each of the metaheuristic algorithms, a significant challenge arises in introducing the most optimal scenario. For this purpose, Val^{Mv} and $\mathrm{Val}_{\mathrm{com}}^{Mv}$ are the optimal values of the previous tables and the corresponding combined values, respectively [27, 28]. Also, S is the number of structural examples examined and $\mathrm{Val}_{i,\min}$ is the lowest value obtained in each study. To access the normalized results and select the optimal design for each of the meta-heuristic algorithms, the values obtained from the combination of problems in another relation are applied Eq. (36). In this regard, in order to facilitate the interpretation of normalized results, inverse values have been considered, so those that have a high percentage will correspond to the optimal design:

$$\mathrm{Val}_{\mathrm{norm}}^{MV}=\frac{\dfrac{1}{\mathrm{Val}_{\mathrm{com}}^{MV}}}{\displaystyle\sum_{j=1}^{nopt}\frac{1}{\mathrm{Val}_{j,\mathrm{com}}^{MV}}}\times 100. \tag{36}$$

Table 6 Optimal design comparison for 582-bar tower truss structure

Element Group	GWO Standard	CGWO-32 Gauss-2	CSA Standard	CCSA-21 Logist-1	CPA Standard	CCPA-32 Gauss-2
1	23.03938	20.06355	21.10793	20	21.32077	20.06593
2	246.9272	155.1647	147.3718	157.8389	99.95783	151.2329
3	33.67345	30.90637	33.03011	30.86969	33.91471	30.9833
4	77.87588	118.3799	149.0555	139.313	95.56768	125.4084
5	30.07553	26.87356	29.02754	26.83133	29.62998	26.76897
6	25.01488	20.10464	25.69651	20.07462	26.42537	20.00163
7	113.5223	109.1716	81.53227	106.5454	187.6107	93.69028
8	27.16429	25.00219	28.77883	25.11106	24.89734	25.27872
9	21.08023	20	23.46077	20	22.46645	20.09486
10	79.50981	83.55701	131.8152	71.92074	221.2876	100.4542
11	24.00528	23.03243	26.75922	22.66337	23.74318	21.78523
12	125.3873	152.4565	311.1666	157.4075	232.559	140.844
13	151.0339	145.3192	156.5978	148.0464	101.9919	146.1685
14	87.59071	98.08288	79.64381	84.6313	160.5204	89.47754
15	201.1891	138.5492	98.96328	155.4782	111.5661	147.5739
16	37.23665	31.0552	52.7292	31.3835	36.79648	31.4191
17	108.717	127.0848	107.9573	129.4626	117.859	122.7136
18	39.0356	23.68318	35.2376	23.91704	27.63773	23.90438
19	32.65908	20	20.04085	20.0153	27.16355	20.20777
20	150.1616	83.19053	110.176	70.09786	261.5027	79.76006
21	24.58382	21.47422	31.75628	21.72332	21.88616	21.60413
22	20.94666	20	22.37363	20.03305	28.1286	20.02239
23	388.0318	35.52802	63.40127	49.02267	44.67642	52.14127
24	21.03091	20.18354	24.04908	20	25.10025	20.10298
25	22.56864	20.10975	22.5029	20	22.92893	20.01235
26	41.55715	20.07573	38.13768	20.41595	39.51534	21.17999
27	21.76659	20.01834	26.79099	20	23.53016	20.17599
28	20.61138	20.13832	27.77936	20.64422	30.60028	20.00967
29	333.5885	28.7142	20	20.62733	413.3706	20.06573
30	23.40145	20.30114	32.5149	20.04698	27.54583	20.00205
31	24.98237	20.52048	482.6035	20.00546	24.39115	20.13808
32	28.0665	20.0411	21.36959	20	312.7653	20.203
Best (m^3)	19.85572	15.5347	18.19611	15.6507	20.71633	15.4171
Mean (m^3)	20.5127	15.5881	18.80231	15.7227	21.52711	15.4439
CV (%)	4.85650	0.29972	3.54559	0.42112	4.55080	0.19811
NFE	18100	18100	18100	18100	18100	18100

In this comparison, *nopt* is the number of cases examined for optimization in each example, which in this study is 7 cases. Table 7 presents the normalized results with the participation of all structural examples. Also, in order to quickly access the final results of the best weight, the best mean and the best coefficient of variation, pie charts are a suitable option. Whit the formation of these charts, the following results have been obtained.

6.1 Results of optimal design for best weight

By determining the success rate of each chaotic state, we can determine the most useful chaotic function and scenario in each metaheuristic algorithm to improve the optimization results. In cases where the success rate is the highest, the greatest balance between search and extraction has been achieved. Based on the final results for the recent research, the best weight in the Chaotic

Table 7 Final normalized value with the participation of all examples

Category	Algorithms	Standard	Logistic 21	Logistic 22	Logistic 23	Gauss 31	Gauss 32	Gauss 33
Best Weight	GWO	3.5383	4.2607	2.8673	3.9911	17.9645	54.0337	13.3444
	CSA	6.3914	52.3753	6.0771	5.5595	20.015	4.8698	4.7119
	CPA	0.0201	0.0364	0.0286	0.0309	0.0485	99.7626	0.073
Mean Weight	GWO	2.9197	3.7284	2.5405	2.9982	15.8507	60.9989	10.9636
	CSA	8.3063	43.8857	7.7747	7.2449	19.3395	6.4274	7.0216
	CPA	0.0106	0.0184	0.0162	0.0153	0.0198	99.8783	0.0413
CV (%)	GWO	4.2308	11.704	10.7301	3.1596	21.673	37.5367	10.9658
	CSA	15.4164	13.0585	14.345	12.3276	17.7497	9.2757	17.8271
	CPA	1.8992	1.8364	3.1793	1.414	1.4409	76.872	13.3582

Gray Wolf Optimizer (CGWO) and Chaotic Cyclical Parthenogenesis Algorithm (CCPA) jointly belongs to the Gaussian chaos function with the second scenario, and for the Chaotic Crow Search Algorithm (CCSA), it belongs to the logistic chaos function with the first scenario. Also, the highest success rate is achieved in the Chaotic Cyclical Parthenogenesis Algorithm (CCPA). The distribution of percentages between chaotic states to determine the best weight is shown in Fig. 12.

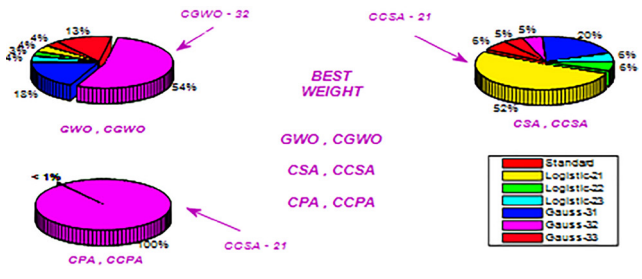

Fig. 12 The final results of the optimal design to determine the best weight

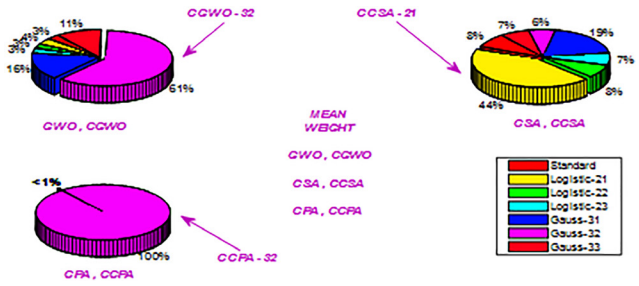

Fig. 13 The final results of the optimal design to determine the best mean

6.2 Results of optimal design for best mean

Given that each optimization model has been repeated 30 times, therefore, obtaining the highest percentage of success between the average of the repeated states is also considered. Based on the final results for the recent research, the best average weight in the Chaotic Gray Wolf Optimizer (CGWO) and Chaotic Cyclical Parthenogenesis Algorithm (CCPA) jointly belongs to the Gaussian chaos function with the second scenario, and for the Chaotic Crow Search Algorithm (CCSA), it belongs to the logistic chaos function with the first scenario. Also, the highest percentage of success has been achieved in the Chaotic Cyclical Parthenogenesis Algorithm (CCPA). The distribution of percentages between the chaotic states to determine the best average weight is shown in Fig. 13.

The correspondence between the results of the best weight and the best average weight shows the stability of the responses and the correct selection of the parameters of the algorithms.

6.3 Results of optimal design for best coefficient of variation

In cases where the results of the iterations have similar conditions, the stability of the chaotic algorithm converges towards the ideal and stable state. In these conditions, the success rate related to the coefficient of variation plays a fundamental role. Based on the final results for the recent research, the best coefficient of variation in the Chaotic Gray Wolf Optimizer (CGWO) and Chaotic Cyclical Parthenogenesis Algorithm (CCPA) jointly belongs to the Gaussian chaos function with the second scenario, and for the Chaotic Crow Search Algorithm (CCSA), it belongs to the Gaussian chaos function with the third scenario. Also, the highest success rate has been achieved in the Chaotic Cyclical Parthenogenesis Algorithm (CCPA). The distribution of percentages between chaotic modes to determine the best coefficient of variation is shown in Fig. 14.

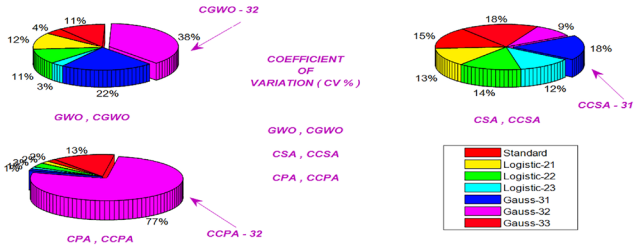

Fig. 14 The final results of the optimal design to determine the best coefficient of variation

7 Conclusion remarks

In the recent paper, three popular algorithms inspired by swarm intelligence are selected and chaos maps are embedded in them. The results of this embedding have led to significant improvements in structural optimization. Some of the conclusions are below:

- In most cases, the combination of chaos functions in metaheuristic algorithms has produced significant improvements compared to the standard case. The main factor may be the effect of chaos functions in escaping local optima and preventing premature convergence. In cases where geometric optimization is performed, analysis in the elasto-plastic limit can significantly improve the optimization results [33].
- For geometric optimization of members at the plastic limit, the connection method affects the optimization process. This issue should be further considered in future research [34, 35].
- With the development of new inspirations in metaheuristic algorithms of swarm intelligence and access to the mysterious behaviors of animals in their daily activities, researchers' efforts to apply them to optimization problems are increasing. The success of these algorithms in flat, linear, and low-variable problems is commendable. However, with the increase in the number of variables and nonlinear behaviors, convergence threatens the maturity of these algorithms.
- To improve the optimization conditions, we can review two basic stages of the algorithm. These two stages that affect the convergence speed of metaheuristic algorithms include the exploration and exploitation stages.
- In the standard design of algorithms, probability distribution functions are used to create diversity and variety to access unknown parts of the search space. Algorithms that use the normal bell distribution have a wide range and produce larger search steps. Therefore, in these algorithms, the exploration conditions are well met, but they have difficulty in exploitation. In algorithms that recommend the Cauchy distribution, the search space is examined with a limited and smaller range. In this group, the exploitation conditions are well met, but the balance between these two stages will not be met. After the unsuccessful experience of the two aforementioned distributions, the Levy distribution was proposed, but the desired results were not achieved.
- Chaotic functions offer great potential to the algorithm to save it from the trap of local optima. In the early stages, we will need long steps to master unknown areas of the search space, so those chaos functions that converge with a high probability from local optima to global optima save the algorithm from the risk of premature maturity. In the later stages, when the algorithm has somewhat understood the positions of global optima, we will need smaller and more cautious steps, and this situation results in another group of chaos functions in which they are likely to be within the range of local optima.
- For a greater challenge, chaos functions from both groups should be examined with three scenarios and by comparing their success rates, the most successful case in this competition should be determined. This approach provides a significant improvement for problems that have more variables and elements.
- According to the results of Table 7, the success rate of all three algorithms in chaotic states is impressive and has a good improvement. Among these three algorithms, the greatest improvement belongs to the Chaotic Cyclical Parthenogenesis Algorithm (CCPA), which has achieved the best weight, the best average weight, and the best coefficient of variation for all three components.
- To select the starting sentence in the series of chaos functions, performing several initial iterations before the main iterations and selecting the appropriate starting sentence improves the results in a leapfrog manner.
- Among the chaos functions, the Gaussian function with scenario 2 has provided the greatest improvement in the optimization results.

References

[1] Kaveh, A. "Advances in metaheuristic algorithms for optimal design of structures", Springer, Cham, 2021. ISBN 9783030593926 https://doi.org/10.1007/978-3-030-59392-6

[2] Schwefel, H.-P. "Evolutions strategy and numerical optimization", Level (PhD), Technical University Berlin, 1975. https://doi.org/10.1007/978-3-0348-5927-1_5

[3] Price, K. V., Storn, R. M., Lampinen, J. A. "Differential evolution: A practical approach to global optimization", Springer, 2005. ISBN 978-3-540-20950-8 https://doi.org/10.1007/3-540-31306-0

[4] Holland, J. H. "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence", University of Michigan Press, Ann Arbor, MI, USA, 1975.

[5] Kennedy, J., Eberhart, R. "Particle swarm optimization", In: Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948. ISBN 0-7803-2768-3 https://doi.org/10.1109/ICNN.1995.488968

[6] Dorigo, M., Maniezzo, V., 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, 1996. https://doi.org/10.1109/3477.484436

[7] Karaboga, D. "An idea based on honey bee swarm for numerical optimization", Erciyes University, Kayseri, Türkiye, (Rep. TR06), 2005.

[8] Mirjalili, S., Mirjalili, S. M., Lewis, A. "Grey wolf optimizer", Advances in Engineering Software, 69, pp. 46–61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007

[9] Askarzadeh, A. "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm", Computers and Structures, 169, pp. 1–12, 2016. http://doi.org/10.1016/j.compstruc.2016.03.001

[10] Kaveh, A., Zolghadr, A. "Cyclical parthenogenesis algorithm: A new meta-heuristic algorithm", Asian Journal of Civil Engineering, 18(5), pp. 673–702, 2017.

[11] Alatas, B. "Chaotic harmony search algorithms", Applied Mathematics and Computation, 216(9), pp. 2687–2699, 2010. https://doi.org/10.1016/j.amc.2010.03.114

[12] Kaveh, A., Bakhshpoori, T. "Water evaporation optimization: a novel physically inspired optimization algorithm", Computers and Structures, 167, pp. 69–85, 2016. https://doi.org/10.1016/j.compstruc.2016.01.008

[13] Erol, O. K., Eksin, I. (2006) "A new optimization method: Big bang–big crunch", Advances in Engineering Software, 37(2), pp. 106–111, 2006. https://doi.org/10.1016/j.advengsoft.2005.04.005

[14] Kaveh, A., Dadras, A. "A novel meta-heuristic optimization algorithm: thermal exchange optimization", Advances in Engineering Software, 110, pp. 69–84, 2017. https://doi.org/10.1016/j.advengsoft.2017.03.014

[15] Kaveh, A., Zolghadr, A. "A novel meta-heuristic algorithm: tug of war optimization", International Journal of Optimization in Civil Engineering, 6(4), pp. 469–492, 2016.

[16] Kaveh, A., Talatahari, S. "A novel heuristic optimization method: charged system search", Acta Mechanica, 213, pp. 267–289, 2010. https://doi.org/10.1007/s00707-009-0270-4

[17] Kaveh, A., Mahdavi, V. R. "Colliding bodies optimization: A novel meta-heuristic method", Computers and Structures, 139, pp.18–27, 2014. https://doi.org/10.1016/j.compstruc.2014.04.005

[18] Kaveh, A., Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", Scientia Iranica, Transaction A: Civil Engineering, 24(2), pp. 551–566, 2017. https://doi.org/10.24200/sci.2017.2417

[19] Simon, D. "Biogeography-based optimization", IEEE Transactions on Evolutionary Computation, 12(6), pp. 702–713, 2008. https://doi.org/10.1109/TEVC.2008.919004

[20] Talatahari, S., Kaveh, A., Sheikholeslami, R. "Chaotic imperialist competitive algorithm for optimum design of truss structures", Structural and Multidisciplinary Optimization, 46(3), pp. 355–367, 2012. https://doi.org/10.1007/s00158-011-0754-4

[21] Rao, R. V., Savsani, V. J., Vakharia, D. P. "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", Computer-Aided Design, 43(3), pp. 303–315, 2011. https://doi.org/10.1016/j.cad.2010.12.015

[22] Mehrabian, A. R., Lucas, C. "A novel numerical optimization algorithm inspired from weed colonization", Ecological Informatics, 1(4), pp. 355–366, 2006. https://doi.org/10.1016/j.ecoinf.2006.07.003

[23] Kaveh, A., Yousefpoor, H. "Chaotic meta-heuristic algorithms for optimal design of structures", Springer, Cham, 2024. ISBN 978-3-031-48917-4 https://doi.org/10.1007/978-3-031-48918-1

[24] Ott, E. "Chaos in dynamical systems", Cambridge University Press, 2002. ISBN 9780511803260 https://doi.org/10.1017/CBO9780511803260

[25] Bucolo, M., Caponetto, R., Fortuna, L., Frasca, M., Rizzo, A. "Does chaos work better than noise?", IEEE Circuits and Systems Magazine, 2(3), pp. 4–19, 2002. https://doi.org/10.1109/MCAS.2002.1167624

[26] Yousefpoor, H., Kaveh, A. "Chaos-embedded meta-heuristic algorithms for optimal design of truss structures", Scientia Iranica, Transactions A: Civil Engineering, 29(6), pp. 2868–2885, 2022. https://doi.org/10.24200/sci.2022.59812.6441

[27] Kaveh, A., Yousefpoor, H. "Chaotically enhanced meta-heuristic algorithms for optimal design of truss structures with frequency constraints", Periodica Polytechnica Civil Engineering, 66(3), pp. 900–921, 2022. https://doi.org/10.3311/PPci.20220

[28] Kaveh, A., Zarfam. P., Aziminejad, A., Yosefpoor, H. "Comparison of four chaotic metaheuristic algorithms for optimal design of largescale truss structures", Iranian Journal of Science and Technology, Transactions of Civil Engineering, 46, pp. 4067–4091, 2022. https://doi.org/10.1007/s40996-022-00908-8

[29] Peitgen, H.-O., Jürgens, H., Saupe, D. "Chaos and fractals: New frontiers of science", Springer, 2004. ISBN 978-0-387-20229-7

[30] Kaveh, A., Yousefpour, H. "Comparison of three chaotic meta-heuristic algorithms for the optimal design of truss structures with frequency constraints", Periodica Polytechnica Civil Engineering, 67(4), pp. 1130–1151, 2023.
https://doi.org/10.3311/PPci.22594

[31] Kaveh, A., Yousefpoor, H. "Chaotic optimization of trusses with frequency constraints with three metaheuristic algorithms", Iranian Journal of Science and Technology, Transactions of Civil Engineering, 48, pp. 271–293, 2024.
https://doi.org/10.1007/s40996-023-01223-6

[32] Kaveh, A., Yosefpoor, H. "Competition of three chaotic meta-heuristic algorithms with physical inspiration for optimal design of truss structures", Periodica Polytechnica Civil Engineering, 68(4), pp. 1211–1228, 2024.
https://doi.org/10.3311/PPci.36853

[33] Movahedi Rad, M., Habashneh, M., Lógó, J. "Elasto-plastic limit analysis of reliability based geometrically nonlinear bi-directional evolutionary topology optimization", Structures, 34, pp. 1720–1733, 2021.
https://doi.org/10.1016/j.istruc.2021.08.105

[34] Grubits, P., Cucuzza, R., Habashneh, M., Domaneschi, M., Aela, P., Movahedi Rad, M. "Structural topology optimization for plastic-limit behavior of I-beams, considering various beam-column connections", Mechanics Based Design of Structures and Machines, 53(4), pp. 2719–2743, 2024.
https://doi.org/10.1080/15397734.2024.2412757

[35] Cucuzza, R., Aloisio, A., Movahedi Rad, M., Domaneschi, M. "Constructability-based design approach for steel structures: From truss beams to real-world inspired industrial buildings", Automation in Construction, 166, 105630, 2024.
https://doi.org/10.1016/j.autcon.2024.105630