Computers and Structures 242 (2021) 106353

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Chaotic coyote algorithm applied to truss optimization problems

Juliano Pierezan [a], Leandro dos Santos Coelho [a,b,*], Viviana Cocco Mariani [a,c], Emerson Hochsteiner de Vasconcelos Segundo [c], Doddy Prayogo [d,e]

[a] Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba, PR, Brazil
[b] Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil
[c] Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil
[d] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei, Republic of China
[e] Department of Civil Engineering, Petra Christian University, Surabaya, Indonesia

ARTICLE INFO

Article history:
Received 11 February 2020
Accepted 2 August 2020

Keywords:
Structural optimization
Discrete truss structures
Coyote optimization algorithm
Metaheuristic algorithms
Chaotic sequences

ABSTRACT

The optimization of truss structures is a complex computing problem with many local minima, while metaheuristics are naturally suited to deal with multimodal problems without the need of gradient information. The Coyote Optimization Algorithm (COA) is a population-based nature-inspired metaheuristic of the swarm intelligence field for global optimization that considers the social relations of the coyote proposed to single-objective optimization. Unlike most widespread algorithms, its population is subdivided in packs and the internal social influences are designed. The COA requires a few control hyperparameters including the number of packs, the population size, and the number maximum of generations. In this paper, a modified COA (MCOA) approach based on chaotic sequences generated by Tinkerbell map to scatter and association probabilities tuning and an adaptive procedure of updating parameters related to social condition is proposed. It is then validated by four benchmark problems of structures optimization including planar 52-bar truss, spatial 72-bar truss, 120-bar dome truss and planar 200 bar-truss with discrete design variables and focus in minimization of the structure weight under the required constraints. Simulation results collected in the mentioned problems demonstrate that the proposed MCOA presented competitive solutions when compared with other state-of-the-art metaheuristic algorithms in terms of results quality.

1. Introduction

Structural optimization is a subject that has gained the attention of researchers because of its direct and large applicability to the design of structures. Besides, the truss design is the most classical benchmark in structural optimization [1–21]. Discrete optimization of truss structures is a hard computing problem with many local minima. Metaheuristic algorithms are naturally suited for discrete optimization problems as they do not require gradient information. Truss optimization may subject to static and dynamic constraints. Static constraints include structural kinematic stability, maximum allowable stress in truss members, maximum allowable deflection in the truss nodes and critical buckling load. However, dynamic constraints impose limits on the natural frequency of the desired truss to avoid the destructive resonance phenomenon. Taking both static and dynamic constraints into account the search space becomes non-convex and may subterfuge the solver to trap in a local optimum.

Implementation is more challenging in the case of discrete optimization problems such as several truss designs that usually entail a complex design space with multiple local optima. Also, several types of research have verified that the classical optimization methods based on gradient information involving the calculation of first and/or second derivatives are not efficient enough or always efficient in dealing with many larger-scale real-world multimodal, non-continuous, and non-differentiable problems [9]. To increase the efficiency and accuracy of the optimization methods and to overcome the computational shortcomings of conventional optimization methods linked to structural design, researchers have encouraged to rely on metaheuristic optimization algorithms. The term *metaheuristic* describes higher-level heuristics that are proposed for the solution to a wide range of optimization problems.

Shih and Lee [10] applied the modified double-cuts approach for large-scale fuzzy optimization in 25-bar and 72-bar truss design problems. The proposed approach was better than the

* Corresponding author.
E-mail addresses: leandro.coelho@pucpr.br (L. dos Santos Coelho), viviana.mariani@pucpr.br (V. Cocco Mariani), prayogo@petra.ac.id (D. Prayogo).

https://doi.org/10.1016/j.compstruc.2020.106353
0045-7949/

single-cut approach and easy programming. Several truss structures with discrete variables were solved by a hybrid particle swarm optimizer (PSO) with harmony search (HS) scheme, called (HPSO) by Li et al. [11]. The HPSO approach was validated and compared with the PSO and the PSO with the passive congregation being able to accelerate the convergence rate effectively. Degertekin and Hayalioglu [12] applied a teaching–learning-based optimization (TLBO) for optimization of truss structures. It obtained results as good as or better than the other metaheuristic algorithms in terms of both the optimum solutions and the convergence capability in most cases. A different approach named Ray optimization was applied to minimize the size, shape and weight of truss structures [13]. An adaptive dimensional search (ADS) was proposed for discrete truss sizing optimization problems [14]. The robustness of the ADS was investigated and verified using two benchmark examples as well as three real-world problems. When ADS was compared with other metaheuristic techniques indicates that it is capable of locating improved solutions using much lesser computational effort.

The mine blast algorithm (MBA), improved MBA (IMBA), and the water cycle algorithm (WCA) were applied for weight minimization of truss structures including discrete sizing variables, offering a good degree of competitiveness with other state-of-the-art metaheuristics [15]. A novel adaptive hybrid evolutionary firefly algorithm (AHEFA) was applied for shape and size optimization of truss structures under multiple frequency constraints. Accordingly, the convergence rate is significantly improved with high solution accuracy [16]. The authors Assimi and Jamali [17] utilized the hybrid genetic programming algorithm for optimum connectivity table among the truss nodes, and optimal cross-sectional areas subject to design constraints were considered both types of continuous and discrete design variables.

Degertekin and his collaborators [18] modified the Jaya Algorithm (JA) improving convergence speed and reducing the number of structural analyses required in the optimization process. Six classical weight minimization problems of truss structures including sizing, layout and large-scale optimization problems with up to 204 design variables were solved. Discrete sizing/layout variables and simplified topology optimization were considered. A novel Jaya Algorithm (JA) was proposed by Degertekin et al. [19] for discrete optimization, denoted as discrete advanced JA (DAJA), and applied for truss structures under stress and displacement constraints. Results collected in seven benchmark problems demonstrated the superiority of DAJA over other state-of-the-art metaheuristic algorithms. The multi-objective colliding bodies optimization (MOCBO) algorithm was proposed by Kaveh and Mahdavi [20] and used to solve two truss structural bi-objective functions. The accuracy and efficiency of the optimization algorithm were compared with literature with promising performance.

An extension of the basic truss layout optimization using various materials was considered in [21]. A novel improved version of the particle swarm optimization algorithm (GEMPSO) was developed for solving six benchmark problems and three truss structures with two multi-material layouts, showing that the appropriate use of expensive stronger materials can reduce the overall cost of structures. The Electromagnetism-like Firefly Algorithm (EFA) was developed by Le et al. [1] and applied for discrete structural optimization. The improved performance of the EFA in comparison with other optimizers was demonstrated by six optimization problems related to truss structures. An adaptive elitist differential evolution (aeDE) was used by Ho-Huu et al. [2] for the optimization of truss structures with discrete design variables. That technique helps preserve the balance between global and local searching abilities in the differential evolution (DE). Numerical results reveal that aeDE was more efficient than the DE and other methods in terms of the quality of solution and convergence rate.

Metaheuristics such as evolutionary algorithms and swarm intelligence paradigms have been designed for tackling many problems in various fields as competitive alternative solvers because they do not require gradient information, easy implementation process, and bypass the local optima problem [22–28]. Swarm-based algorithms try to mimic the social behavior of nature creatures who swarm, herds, schools or flocks for foraging, migration and enemy skipping. The swarm-based algorithms comprised of environment and agents. These agents interact with each other in the environment and converge to a common solution for a problem using an environmental mechanism, interaction mechanism, and activities of agents. In terms of recently proposed swarm intelligence paradigms, the Coyote Optimization Algorithm (COA) [29] is a promising and competitive stochastic population-based approach for global optimization tasks. It considers the principles of coyote social relations. The Cultural Coyote Optimization Algorithm (CCOA) was proposed and validated by Pierezan et al. [30] under a set of benchmark functions from the Institute of Electrical and Electronics Engineers (IEEE) Congress on Evolutionary Computation (CEC) 2017 and gas turbine problem. The results showed that the CCOA outperforms state-of-the-art metaheuristics.

Maintaining a good balance between the convergence and the diversity is particularly crucial for the performance of a metaheuristic algorithm. In this context, the key capabilities of metaheuristic algorithms such as COA to be able to find reasonable solutions are exploration and exploitation.

Exploration and exploitation are fundamental concepts of any search algorithm. The exploration may be described as the ability of the algorithm to investigate the different promising regions in a given search space whereas exploitation ensures the searching of optimal solutions around the promising regions, a kind of local search. It is important for a metaheuristic algorithm maintaining an appropriate balance between the exploration and exploitation behaviors to be competitive in terms of robustness and performance. However, it is difficult to balance between these phases due to its stochastic nature. A latent viewpoint interprets exploration and exploitation as a global search and local search, respectively. Pure exploration degrades the precision of the search process but increases its capacity to find new potential solutions. On the other hand, pure exploitation allows refining existent solutions but adversely driving the process to locally optimal solutions.

On the other hand, in recent years, growing interests in chaos theory and its features have stimulated the studies of chaos applied in optimization algorithms design. Chaos is a kind of a feature of a nonlinear dynamic system which exhibits bounded unstable dynamic behavior, ergodic, non-period behavior depended on initial condition and control parameters. Due to the benefit of a few properties as stochasticity and ergodicity of chaos, the idea of using chaotic sequences instead of random sequences has been noticed in several fields, one of these fields is the optimization theory [31–40]. In this paper, a modified COA (MCOA) approach based on chaotic sequences generated by Tinkerbell map [41–43] is proposed to improve the exploration behavior. Additionally, an adaptive procedure of updating parameters related to social condition is adopted to update the exploitation behavior based on the historical record of success in the search. The proposed MCOA is validated by truss optimization problems with discrete design variables and focus in minimization of the structure weight under the required constraints.

The rest of this paper is organized as follows. Section 2 presents a description of the classical COA and the proposed COA variant using Tinkerbell chaotic map. After, details of the truss optimization problems are shown in Section 3. Finishing, the numerical results and conclusive remarks are given in Sections 4 and 5, respectively.

2. Description of the optimization methods

In this section, the fundamentals of the COA are presented. After, the proposed modified COA using Tinkerbell chaotic map is detailed.

2.1. Coyote optimization algorithm (COA)

Unlike most widespread algorithms, the population in the COA is subdivided in packs and the internal social influences are designed [29]. The COA requires only a few control hyperparameters including the number of packs, the population size, and the number maximum of generations (the stopping criterion). The source code of the COA for single-objective optimization in MATLAB©, R and Python are given in https://github.com/jkpir/COA.

Step 1: Coyotes population initialization. The population (X) is composed of N_p packs with N_c coyotes each and the initialization (iteration $t = 0$) inside the search space defined by the interval $[lb, ub]^D$occurs as follows:

$$soc_{c,j}^{p,t} = lb_j + r_j \cdot (ub_j - lb_j), \tag{1}$$

where $c = [1, 2, \cdots, N_c]$, $p = [1, 2, \cdots, N_p]$, $j = [1, 2, \cdots, D]$, D is the dimension of the optimization problem and r_j is a random number inside $[0, 1]$ generated by a uniform probability distribution. The packs are randomly grouped by the same distribution and the initial coyotes ages ($age_c^{p,0}$) are all equal to 0.

Step 2: Coyote's adaptation. The coyote adaptation is evaluated due to the objective function. It is a consequence of its social condition, which means:

$$fit_c^{p,t} = f(soc_c^{p,t}). \tag{2}$$

While the stopping criterion is not reached, repeat Steps 3 to 10.

For each p^{th} pack, repeat Steps 3 to 8.

Step 3: Alpha coyote definition. In nature, the alpha coyote is the one that presents the best social condition. In the COA, it means the best (i.e. the smallest or the highest) objective function cost, or:

$$alpha^{p,t} = \left\{ soc_c^{p,t} | arg_{c=\{1,2,\cdots,N_c\}} min_{f\left(soc_c^{p,t}\right)} \right\}. \tag{3}$$

Step 4: The social tendency. The social behavior of coyotes is naturally influenced by the alpha and the other coyotes of the pack. In COA, this phenomenon is represented by the cultural tendency of the pack ($ct^{p,t}$), which is the median of the coyote's social conditions:

$$ct_j^{p,t} = median\left(soc_{c,j}^{p,t}\right) \forall c \in \{1, 2, \cdots, N_c\} \tag{4}$$

for $j = [1, 2, \cdots, D]$.

For each c^{th} coyote of the p^{th} pack, repeat Steps 5 to 7.

Step 5: Social condition update. The social condition is updated according to the influence of the alpha coyote (δ_a) and the social tendency (δ_t), generated from two random coyotes of the pack (cr_1 and cr_2), which means:

$$new\_soc_c^{p,t} = soc_c^{p,t} + r_1 \cdot \delta_t + r_2 \cdot \delta_a \tag{5}$$

where

$$\delta_t = \mathrm{ct}^{p,t} - \mathrm{soc}_{\mathrm{cr}_1}^{p,t} \tag{6}$$

$$\delta_a = \mathrm{alpha}^{p,t} - \mathrm{soc}_{\mathrm{cr}_2}^{p,t} \tag{7}$$

and r_1 and r_2 are, respectively, the weights of the pack and the alpha influence, both random numbers inside the range [0,1] generated with uniform distribution of probability.

Step 6: New social condition evaluation. The objective function cost is calculated considering the new social condition.

$$new\_fit_c^{p,t} = f(new\_soc_c^{p,t}), \tag{8}$$

Step 7: Adaptation. The coyotes choose the social condition that best fits the environment to keep it to the next iteration, which means the best (i.e. the smallest or the highest) objective function cost, such that:

$$soc_c^{p,t+1} = \begin{cases} new\_soc_c^{p,t}, & \text{if } new\_fit < fit_c^{p,t} \\ soc_c^{p,t}, & \text{otherwise} \end{cases} \tag{9}$$

Step 8: Birth and death. First, a pup is generated with age equals to 0 and considering the scatter probability (P_s) and the association probability (P_a), such that:

$$pup_j^{p,t} = \begin{cases} \mathrm{soc}_{\mathrm{k}_1,j}^{\mathrm{p,t}}, & \text{if } rnd_j < P_a \text{ or } j = j_1 \\ \mathrm{soc}_{\mathrm{k}_2,j}^{\mathrm{p,t}}, & \text{if } rnd_j \geq P_s + P_a \text{ or } j = j_2 \\ R_j, & \text{otherwise} \end{cases} \tag{10}$$

where these probabilities are calculated as follows:

$$P_s = \frac{1}{D}, \tag{11}$$

$$P_a = \frac{(1 - P_s)}{2}, \tag{12}$$

k_1 and k_2 are the two selected coyotes from the p^{th} pack, j_1 and j_2 are two random dimensions of the problem, R_j is a random number inside the decision variable bound of the j^{th} dimension and rnd_j is a random number inside [0,1] generated with uniform probability distribution. After that, the pup social condition is evaluated and the death rule is applied, according to the following algorithm:

Step	Description
1	Compute the group of worst adapted coyotes than the pup (ω)
2	Compute the number of coyotes inside ω (φ)
	If $\varphi = 1$
3	The pup survives and the only coyote in ω dies
	Else if $\varphi > 1$
4	The pup survives and the oldest coyote in ω dies
	Else
5	The pup dies.

Step 9: Transition between packs. Along with the coyote's life, it can evict from a pack and go to another one. In COA, two random coyotes from different packs change their positions with probability P_e, such that:

$$P_e = 0.005 \cdot N_c^2 \tag{13}$$

Step 10: Ages update. The coyotes age is updated every iteration, which means:

$$age_c^{p,t+1} = age_c^{p,t} + 1 \tag{14}$$

Step 11: Solution selection. The best-adapted coyote among all packs is selected as the solution of the optimization problem.

2.2. Modified coyote optimization algorithm (MCOA)

An important issue that needs to be addressed is the value of control parameters of COA (see details in [29,30,44]). The control parameters manage the balance between exploitation (using the existing material in the population to best effect) and exploration (searching for better coyotes). The control parameters related to the scatter and association probabilities are key factors affecting the COA's convergence.

Choosing suitable control parameter values in COA as in another evolutionary and swarm-based approaches is frequently a problem-dependent task. Suitable control parameters are different for different function problems including structural optimization (see examples in [1–21,45]). The difficulty in the use of COA arises in that the choice of these is mainly based on empirical evidence and practical experience.

As mentioned in the Introduction section, the design of optimization techniques using chaotic sequences [31–40] has received a great deal of attention in the literature. Optimization approaches using chaotic sequences are generally based on ergodicity, stochastic properties and irregularity of chaotic signals. The use of chaotic sequences in COA can be helpful to escape more easily from local minima than can be done through the traditional COA.

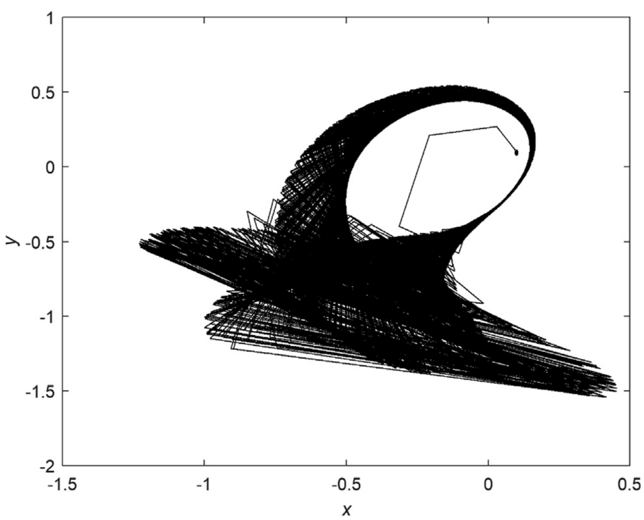

Fig. 1. Phase plots of Tinkerbell map using 3000 iterations.

In the MCOA, Tinkerbell chaotic map-generating values are adopted. The Tinkerbell map [41–43] is an example of a strange attractor, where the two-dimensional quadratic map of its map is given by

$$x_{t+1} = x_t^2 - y_t^2 + a \cdot x_t + b \cdot y_t \tag{15}$$

$$y_{t+1} = 2x_t y_t + c \cdot x_t + d \cdot y_t \tag{16}$$

where a, b, c, d are non-zero parameters and t is the iteration. For parameter values (adopted to the COA design with different initial conditions) $a = 0.9$, $b = -0.6013$, $c = 2.0$, and $d = 0.5$, we get the chaotic attractor of this map as shown in Fig. 1. For generating the Fig. 1, the initial conditions are $x_0 = 0.1$ and $y_0 = 0.1$.

Considering a population with N_p packs with N_c coyotes each, a summary of the MCOA steps is shown in Fig. 2. In step 11, it is adopted a normalized Tinkerbell chaotic map-generating values in the range [0, 1] to scatter and association probabilities tuning.

In this case, in a preprocessing phase of Tinkerbell map data to utilization in MCOA, T values are generated using the equations (15) and (16). After the values of x_{t+1} (equation (17)) are normalized using a linear scaling function to scatter and association probabilities tuning. The linear scaling function makes use of the maximum and minimum values of x_{t+1}. The linear scaling function in the range [0.025, 0.075] transforms a variable x_{t+1} into x^*_{t+1} in the following way:

$$x^*_{t+1} = \frac{x_{t+1} - min(x)}{max(x) - min(x)} \tag{17}$$

where $x = (x_1, \ldots, x_T)$, T is number of iterations, min(x) and max(x) are the minimum and maximum values of x_{t+1}, respectively.

In the original COA, r_1 and r_2 are random numbers inside the range [0,1] generated with uniform distribution of probability. In terms of exploitation behavior, an adaptive procedure of updating parameters r_1 and r_2 in equation (5) related to social condition is employed in the MCOA based on adaptive differential evolution called JADE. Details about the JADE can be found in [46].

Step	***Description***
1	Define the control parameters
2	Initialize N_p packs with N_c coyotes each
3	Evaluate the coyote's adaptation
	While stopping criterion is not achieved **do**
	For each p pack **do**
4	Define the alpha coyote of the pack
5	Compute the pack's social tendency
	For each c coyote **do**
6	Update the coyote's social condition
7	Evaluate the new social condition
8	Coyote's adaptation (objective function calculus)
	End for
9	Birth and death inside the pack
	End for
10	Select the population to the next iteration
11	Packs reorganization according to the scatter and association probabilities using Tinkerbell chaotic maps
12	Update parameters related to social condition in an adaptive form
13	Update the coyotes' ages
	End while
14	Return the best solution

Fig. 2. Steps of the MCOA.

The adopted procedure in MCOA uses optional external archive and an adaptive parameter method. The external archive not only can provide the progress direction but also can improve the diversity of the swarm. The adaptive parameter method lets the r_1 and r_2 be generated following Gaussian distributions truncated to [0,1]. Their expected mean values are adaptively updated using the successful values of r_1 and r_2. In general terms, the parameters r_1 and r_2 for each coyote are updated based on their historical record of success.

At each iteration, a r_1 and r_2 are generated for each coyote according to Gaussian distributions with location parameter μ_{r1} and μ_{r2} with a standard deviation of 0.1. The parameters μ_{r1} and μ_{r2} are initialized to 0.5 and then updated at the end of each generation according to the following equations:

$$\mu_{r1} = (1 - c_r) * \mu_{r1} + c_r * mean_A(S_{r1}) \tag{18}$$

$$\mu_{r2} = (1 - c_r) * \mu_{r2} + c_r * mean_A(S_{r2}) \tag{19}$$

where c_r is a constant $\in [0, 1]$. S_{r1} and S_{r2} are the set of all successful r_1 and r_2 in the current iteration, and $mean_A(.)$ is the arithmetic mean. In this paper, the adopted c_r equals to 0.05.

3. Benchmarks of truss optimization

In this section, four benchmarks of truss optimization problems including planar 52-bar truss, 72-bar space truss structure, 120-bar dome truss, and planar 200-bar truss structure are described. All adopted benchmarks are constrained optimization problems.

When the problems contain constraints, the feasible region is reduced, leading to many difficulties in solving them. As a rule, in constraint handling methods based on penalty functions [47,48], a penalty term is added to the objective function penalizing the function values outside the feasible region.

In this paper, the constraint-handling strategy utilized by COA and MCOA approaches relies on a simple transformation of the original cost function of the optimization problem. Transforming constrained optimization problem into an unconstrained problem is the core idea of the penalty function whose formula is given by:

$$\varphi(X) = f(X) + p(X) \tag{20}$$

$$p(X) = \gamma * nvc \sum_{i=1}^{m} max(0, g_i(X))^2 \tag{21}$$

where *max* is the maximum value, $i = 1,\ldots,m$ inequality constraints, $f(X)$ is the objective function (minimization of the structure weight), $\varphi(X)$ is the extended objective function, $p(X)$ is the penalty value defined by the inequality constraints $g_i(X)$, *nvc* is the number of violated constraints, and γ is a positive constant called penalty factor. The penalty value is added to the fitness function because low values are preferred as expected in a minimization problem. In this paper, γ equals to 10^{20} was adopted.

3.1. A planar 52-bar truss structure

This section carries out the optimization problem for a 52-bar planar truss structure shown in Fig. 3. The mass density E and the modulus of elasticity q of the constitutive material are respectively 207 GPa and 7860 kg/m^3. The tension and compression stress are subjected not to higher than a magnitude of 180 MPa. In this problem, the horizontal and vertical loads that are applied to the nodes from 17 to 20 are set to P_x = 100 kN and P_y = 200 kN, correspondingly. All bars of this planar 52-bar truss are gathered into 12 groups: (1) A_1–A_4, (2) A_5–A_{10}, (3) A_{11}–A_{13}, (4) A_{14}–A_{17}, (5) A_{18}–A_{23}, (6) A_{24}–A_{26}, (7) A_{27}–A_{30}, (8) A_{31}–A_{36}, (9) A_{37}–A_{39}, (10) A_{40}–A_{43}, (11) A_{44}–A_{49}, and (12) A_{50}–A_{52}; wherein, each group contains the bars having a same value of the cross-sectional area. Therefore, there are 12 design variables in this optimal design problem. Discrete values of cross-sectional areas can be selected from Table 1.

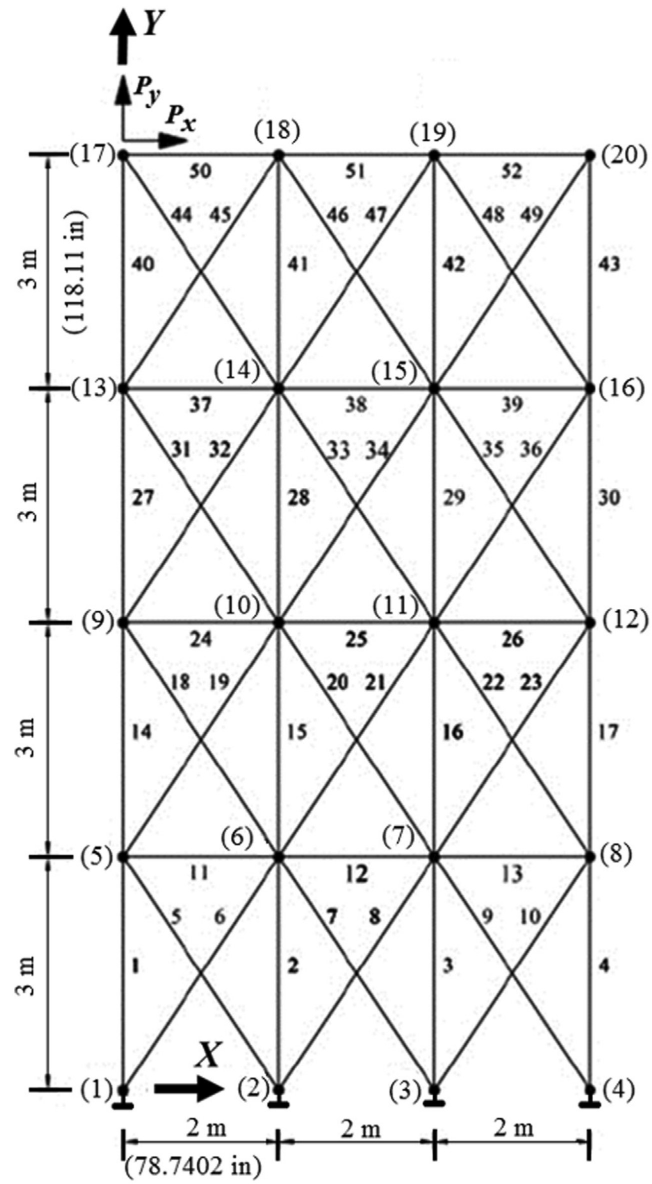

Fig. 3. A 52-bar space truss structure. Source: [15].

This problem has been investigated using several optimization algorithms such as genetic algorithm (GA) and modified GA by Wu and Chow [45], HS by Lee et al. [49], heuristic particle swarm optimization (HPSO) by Li et al. [11], MBA, WCA and IMBA by Sadollah et al. [15,50], colliding bodies optimization (CBO) by Kaveh and Mahdavi [51], TLBO by Dede [52], hybrid harmony search algorithm (HHS) by Cheng et al. [53], DE and aeDE by Ho-Huu et al. [2].

3.2. A spatial 72-bar space truss structure

The second example executes the optimization problem for a 72-bar space truss structure as shown in Fig. 4. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10^4 ksi. The stress limitations of the members are ±25,000 psi. All nodal displacements must be smaller than ±0.25 in. There are 72 truss elements

Table 1
List of the cross-sectional areas from the AISC (American Institute of Steel Construction) design code.

Number	in^2	mm^2	Number	in^2	mm^2
1	0.111	71.613	33	3.840	2477.414
2	0.141	90.968	34	3.870	2496.769
3	0.196	126.451	35	3.880	2503.221
4	0.250	161.290	36	4.180	2696.769
5	0.307	198.064	37	4.220	2722.575
6	0.391	252.258	38	4.490	2896.768
7	0.442	285.161	39	4.590	2961.284
8	0.563	363.225	40	4.800	3096.768
9	0.602	388.386	41	4.970	3206.445
10	0.766	494.193	42	5.120	3303.219
11	0.785	506.451	43	5.740	3703.218
12	0.994	641.289	44	7.220	4658.055
13	1.000	645.160	45	7.970	5141.925
14	1.228	792.256	46	8.530	5503.215
15	1.266	816.773	47	9.300	5999.988
16	1.457	939.998	48	10.850	6999.986
17	1.563	1008.385	49	11.500	7419.340
18	1.620	1045.159	50	13.500	8709.660
19	1.800	1161.288	51	13.900	8967.724
20	1.990	1283.868	52	14.200	9161.272
21	2.130	1374.191	53	15.500	9999.980
22	2.380	1535.481	54	16.000	10,322.560
23	2.620	1690.319	55	16.900	10,903.204
24	2.630	1696.771	56	18.800	12,129.008
25	2.880	1858.061	57	19.900	12,823.684
26	2.930	1890.319	58	22.000	14,193.520
27	3.090	1993.544	59	22.900	14,774.164
28	3.130	2019.351	60	24.500	15,806.420
29	3.380	2180.641	61	26.500	17,096.740
30	3.470	2238.705	62	28.000	18,064.480
31	3.550	2290.318	63	30.000	19,354.800
32	3.630	2341.931	64	33.500	21,612.860

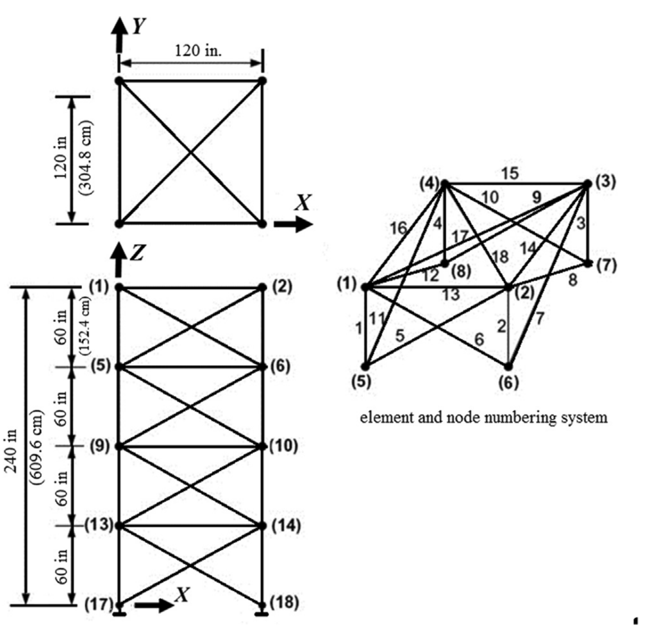

Fig. 4. A 72-bar space truss structure. Source: [56].

which are divided into 16 groups: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}–A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66}, (15) A_{67}–A_{70}, and (16) A_{71}–A_{72}. This problem was previously examined by Li et al. [11], Sadollah et al. [15,50], Wu and Chow [45], Lee et al. [49], Kaveh and Mahdavi [51], Kaveh and Talatahari [54], and Kaveh and Ghazaan [55].

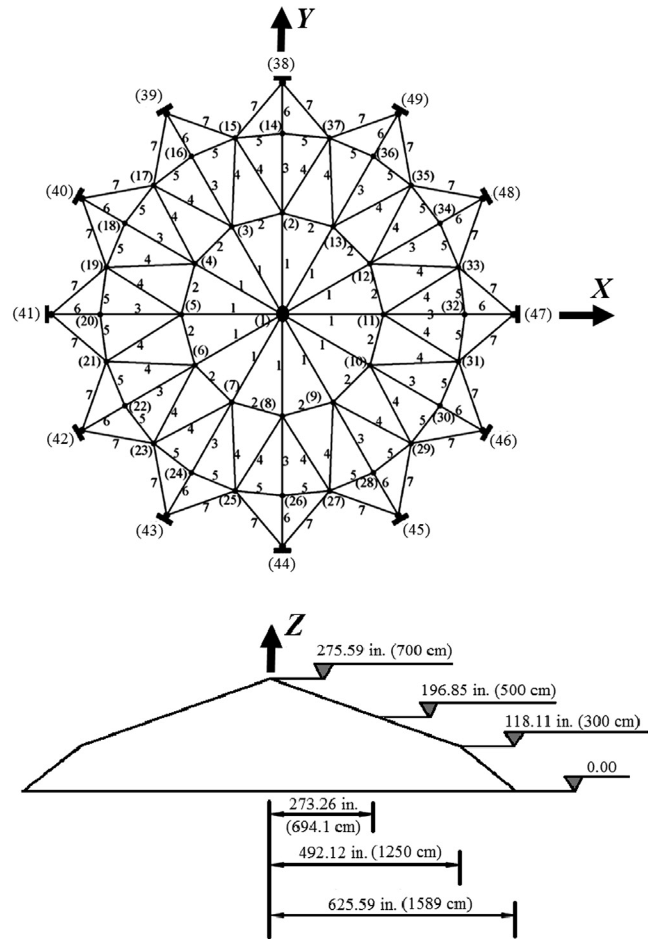

Fig. 5. A 120-bar dome structure. Source: [50].

3.3. A 120-bar dome structure

A 120-bar dome truss, shown in Fig. 5, is considered the third case study. The problem has been studied as a benchmark optimization problem with static constraints. The symmetry of the structure about the *X*-axis and *Y*-axis is considered to group the 120 members into seven independent size variables. A constant lumped mass is attached as 3000 kg (6613,868 lb) at node 1, 500 kg (1102.31 lb) at nodes 2 to 13, and 100 kg (220.462 lb) at the rest of the free nodes. The elements are clustered into 7 groups by considering symmetry about the *Z*-axis.

This problem was previously solved by Kaveh and Mahdavi [51] using CBO, Kaveh and Zolghadr [58] using democratic particle swarm optimization (DPSO), Kaveh and Zolghadr [59] based on hybridization of the Charged System Search and the Big Bang-Big Crunch algorithms (CSS-BBBC), and Tejani et al. [60] using adaptive symbiotic organisms search (SOS), among others.

3.4. A planar 200-bar truss structure

The fourth case study considered for size optimization is the 200-bar plane truss structure. A constant lumped mass of 100 kg is attached at each of the upper nodes (nodes 1–5), whereas all elements are grouped into 29 groups corresponding to 29 design variables by considering geometrical symmetry, as shown in Fig. 6 and Table 2. All member are made of steel: the material density and modulus of elasticity are 0.283 lb/in^3 (7933.410 kg/m^3) and 30,000 ksi (206,000 MPa), respectively. This truss is subjected to

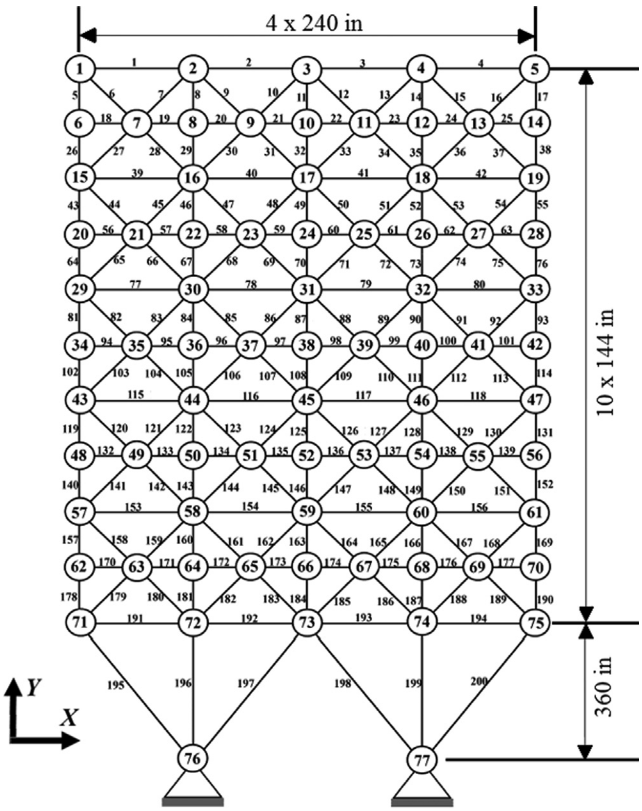

Fig. 6. A 120-bar dome structure. Source: [61].

constraints only on stress limitations of ± 10 ksi. There are three loading conditions: (i) 1.0 kip acting in the positive X-direction at nodes 1, 6, 15, 20, 29, 43, 48, 57, 62, and 71; (ii) 10 kips acting in the negative Y-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24,..., 71, 72, 73, 74, and 75; and (iii) conditions (i) and (ii) acting together. the minimum cross-sectional area of all is 0.1 in^2 (0.6452 cm^2) and the maximum cross-sectional area is 20 in^2 (129.03 cm^2).

This problem was previously solved by Kaveh and Talatahari [61] using a hybrid scheme based on particle swarm optimization, ant colony and harmony search (HPSACO), Lamberti [62] using a simulated annealing approach (SA), Degertekin et al. [63] using a harmony search approach (SAHS) and Degertekin and Hayalioglu [12] proposed a TLBO method. Furthermore, recently Kim and Byun [64] presented a diversity-enhanced cyclic neighborhood network topology particle swarm optimizer (CNNT-PSO).

4. Numerical results

In this section, four benchmarks of structure optimization problems with discrete design variables are solved by the COA and MCOA which were described in Section 2. The MCOA was implemented in MATLAB©, adopting the parameters: number of independent runs is 50 times to provide statistically meaningful results, N_p equal to 10 packs, N_c set to 5 coyotes, and stopping criterion of 8000 objective function evaluations, except to the planar 200-bar truss structure with 30,000 objective function evaluations. It is important to mention that the candidate solutions are rounded for the nearest integer (discrete variables) to objective function evaluation.

In this paper, the structural optimization problem minimizes the truss weight by finding the optimal nodal positions and optimal elemental cross-sectional areas such that it satisfies multiple natural frequency constraints. Therefore, the objective function is formulated for the structural weight by neglecting the weight of lumped masses, where nodal coordinates and the element cross-sectional areas are the design variables.

Simulation results in Tables 3–6 show that the MCOA obtained competitive results to planar 52-bar truss (D = 12), spatial 72-bar truss (D = 16), 120-bar dome truss (D = 7) and planar 200 bar-truss (D = 29) optimization problems, respectively. The best results in Tables 3–6 are in bold.

In terms of the MCOA for the planar 52-truss case study reported in Table 3, MCOA found an optimum weight of 1902.605 lb after 5392 structural analyses. It suitably agrees with the results obtained by other researches. It outperforms the best result presented in [11] in terms of the best (minimum) of the weights. Furthermore, MCOA presents the same best results than the results mentioned in [1,2,50], but it presents superior results in terms of the mean and worst weight than the results in [1,2,11,50]. However, the IMBA [15] presented superior results considering the mean weight when compared with MCOA.

For the spatial 72-truss case study, the best results in Table 4 for the weights of MCOA outperform the best results presented in [1,2,50,51,54] in terms of the mean of the weights. MCOA achieved

Table 2
Design variables for the planar 200-bar truss structure.

Element number (design variable)	Member in the group	Element number (design variable)	Member in the group
1	1, 2, 3, 4	16	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
2	5, 8, 11, 14, 17	17	115, 116, 117, 118
3	19, 20, 21, 22, 23, 24	18	119, 122, 125, 128, 131
4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177	19	133, 134, 135, 136, 137, 138
5	26, 29, 32, 35, 38	20	140, 143, 146, 149, 152
6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34, 36, 37	21	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151
7	39, 40, 41, 42	22	153, 154, 155, 156
8	43, 46, 49, 52, 55	23	157, 160, 163, 166, 169
9	57, 58, 59, 60, 61, 62	24	171, 172, 173, 174, 175, 176
10	64, 67, 70, 73, 76	25	178, 181, 184, 187, 190
11	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75	26	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
12	77, 78, 79, 80	27	191, 192, 193, 194
13	81, 84, 87, 90, 93	28	195, 197, 198, 200
14	95, 96, 97, 98, 99, 100	29	196, 199
15	102, 105, 108, 111, 114		

Table 3
Optimization results for the planar 52-bar dome structure case study.

Design variable	Le et al. [1] EFA	Ho-Huu et al. [2] aeDE	Li et al. [11] PSO	Sadollah et al. [50] Mine	Sadollah et al. [15] WCA	Sadollah et al. [15] IMBA	COA	MCOA
A_1–A_4	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055
A_5–A_{10}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288
A_{11}–A_{13}	494.193	494.193	363.225	494.193	494.193	494.193	494.193	494.193
A_{14}–A_{17}	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219
A_{18}–A_{23}	939.998	939.998	940.000	939.998	940.000	940.000	939.998	939.998
A_{24}–A_{26}	494.193	494.193	494.193	494.193	494.193	494.193	494.193	494.193
A_{27}–A_{30}	2238.705	2238.705	2238.705	2238.705	2283.705	2283.705	2238.705	2238.705
A_{31}–A_{36}	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385
A_{37}–A_{39}	494.193	494.193	388.386	494.193	494.193	494.193	494.193	494.193
A_{40}–A_{43}	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868
A_{44}–A_{49}	1161.288	1161.288	1161.228	1161.288	1161.288	1161.288	1161.288	1161.288
A_{50}–A_{52}	494.193	494.193	792.256	494.193	494.193	494.193	494.193	494.193
Best weight (lb)	**1902.605**	**1902.605**	1905.490	**1902.605**	**1902.605**	**1902.605**	**1902.605**	**1902.605**
Worst weight (lb)	1910.942	1925.714	–	1912.646	1912.646	**1904.830**	1931.551	1908.923
Mean weight (lb)	1904.775	1906.735	–	1906.076	1909.856	**1903.076**	1909.172	1903.928
Standard deviation (lb)	3.045	6.679	–	4.090	7.090	1.130	8.129	2.913
Number of structural analysis	2894	3402	–	5450	7100	4750	7050	5390

Table 4
Optimization results for the spatial 72-bar truss case study.

Design variable	Le et al. [1] EFA	Ho-Huu et al. [2] aeDE	Kaveh and Talatahari [54]DHPSACO	Sadollah et al. [50] Mine	Sadollah et al. [15] WCA	Sadollah et al. [15] IMBA	Kaveh and Mahdavi [51] CBO	COA	MCOA
A_1–A_4	1.990	1.990	1.800	0.196	1.990	1.990	1.620	1.990	1.990
A_5–A_{12}	0.563	0.563	0.442	0.563	0.442	0.442	0.563	0.563	0.563
A_{13}–A_{16}	0.111	0.111	0.141	0.442	0.111	0.111	0.111	0.111	0.111
A_{17}–A_{18}	0.111	0.111	0.111	0.602	0.111	0.111	0.111	0.111	0.111
A_{19}–A_{22}	1.228	1.228	1.228	0.442	1.266	1.266	1.457	1.228	1.228
A_{23}–A_{30}	0.442	0.442	0.563	0.442	0.563	0.563	0.442	0.442	0.442
A_{31}–A_{34}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
A_{35}–A_{36}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
A_{37}–A_{40}	0.563	0.563	0.563	1.266	0.422	0.422	0.602	0.563	0.563
A_{41}–A_{48}	0.563	0.563	0.563	0.563	0.422	0.422	0.563	0.563	0.563
A_{49}–A_{52}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111
A_{53}–A_{54}	0.111	0.111	0.250	0.111	0.111	0.111	0.111	0.111	0.111
A_{55}–A_{58}	0.196	0.196	0.196	1.800	0.196	0.196	0.196	0.196	0.196
A_{59}–A_{66}	0.563	0.563	0.563	0.602	0.563	0.563	0.602	0.563	0.563
A_{67}–A_{70}	0.391	0.391	0.442	0.111	0.442	0.442	0.391	0.391	0.391
A_{71}–A_{72}	0.563	0.563	0.563	0.111	0.602	0.602	0.563	0.563	0.563
Best weight (lb)	**389.334**	**389.334**	390.380	390.730	**389.334**	**389.334**	391.070	**389.334**	**389.334**
Worst weight (lb)	393.325	393.826	–	399.490	393.778	**389.457**	495.970	393.965	392.158
Mean weight (lb)	390.913	391.376	–	395.432	389.941	**389.823**	403.710	393.618	390.162
Standard deviation (lb)	1.161	1.376	–	3.040	1.430	0.840	24.800	1.561	1.018
Number of structural analysis	3123	4101	–	11,600	4600	6250	6000	6800	5750

Table 5
Optimization results for the spatial 120-bar dome case study.

Design variable	Kaveh and Mahdavi [51] CBO	Kaveh and Zolghadr [58] DPSO	Kaveh and Zolghadr [59] CSS-BBBC	Tejani et al. [60] SOS	COA	MCOA
1	19.6917	19.607	17.448	19.5715	19.4994	19.4994
2	41.1421	4.1290	49.076	39.8327	40.3890	40.3890
3	11.1550	11.136	12.365	10.5879	10.6073	10.6073
4	21.3207	21.025	21.979	21.2194	21,1126	21,1126
5	9.8330	10.060	11.190	10.0571	9.8420	9.8420
6	12.8520	12.758	12.590	11.8322	11.7715	11.7715
7	15.1602	15.414	13.585	14.7503	14.8384	14.8384
Best weight (kg)	8889.1303	8890.48	9046.34	8710.33	**8707.2432**	**8707.2432**
Worst weight (kg)	–	–	–		8737.1328	**8734.4957**
Mean weight (kg)	8891.2540	8895.99	–		8720.5461	**8713.4877**
Standard deviation (kg)	1.7926	89.38	–		6.9123	9.1185
Number of structural analysis	6000	6000	4000	4000	5600	5250

Table 6
Optimization results for the planar 200-bar truss structure.

Design variable	Kaveh and Talatahari [54] HPSACO	Lamberti [62] SA	Degertekin [63] SAHS	Degertekin and Hayalioglu [12] TLBO	Kim and Byun [64] CNNT-PSO	COA	MCOA
1	0.1033	0.1468	0.154	0.146	0.1482	0.1441	0.1390
2	0.9184	0.9400	0.941	0.941	0.9405	0.9395	0.9355
3	0.1202	0.1000	0.100	0.100	0.1000	0.1000	0.1000
4	0.1009	0.1000	0.100	0.101	0.1000	0.1000	0.1000
5	1.8664	1.9400	1.942	1.941	1.9408	1.9395	1.9355
6	0.2826	0.2962	0.301	0.296	0.2975	0.2947	0.2909
7	0.1000	0.1000	0.100	0.100	0.1000	0.1000	0.1000
8	2.9683	3.1042	3.108	3.121	3.1067	3.1027	3.0816
9	0.1000	0.1000	0.100	0.100	0.1000	0.1000	0.1000
10	3.9456	4.1042	4.106	4.173	4.1067	4.1027	4.0816
11	0.3742	0.4034	0.409	0.401	0.4057	0.3987	0.3967
12	0.4501	0.1912	0.191	0.181	0.1897	0.1831	0.2959
13	4.96029	5.4284	5.428	5.423	5.4343	5.3821	5.3854
14	1.0738	0.1000	0.100	0.100	0.1000	0.1000	0.1000
15	5.9785	6.4284	6.427	6.422	6.4340	6.3821	6.3853
16	0.78629	0.5734	0.581	0.571	0.5745	0.5720	0.6332
17	0.73743	0.1327	0.151	0.156	0.1366	0.3389	0.1842
18	7.3809	7.9717	7.973	7.958	7.9803	7.9871	8.0396
19	0.66740	0.1000	0.100	0.100	0.1000	0.1000	0.1000
20	8.3000	8.9717	8.974	8.958	8.9802	8.9871	9.0395
21	1.19672	0.7049	0.719	0.720	0.71089	0.8188	0.7460
22	1.0000	0.4196	0.422	0.478	0.4659	0.1435	0.1306
23	10.8262	10.8636	10.892	10.897	10.9110	10.9723	10.9114
24	0.1000	0.1000	0.100	0.100	0.1000	0.1000	0.1000
25	11.6976	11.8606	11.887	11.897	11.9112	11.9722	11.9114
26	1.3880	1.0339	1.040	1.080	1.0712	0.8947	0.8627
27	4.9523	6.6818	6.646	6.462	6.5030	6.7474	6.9169
28	8.8000	10.8113	10.804	10.799	10.7210	10.8536	10.9674
29	14.6645	13.8404	13.870	13.922	13.9310	13.7759	13.6742
Best weight (lb)	**25,156.5**	25,445.63	25,491.9	25,488.15	25,453.0957	25,453.11	25,450.18
Worst weight (lb)	–	–	25,799.3	25,563.05	**25,466.0958**	25,666.43	25,557.53
Mean weight (lb)	–	–	25,610.2	25,533.14	**25,459.1089**	25,545.51	25,522.07
Standard deviation (lb)	–	–	141.85	27.44	3.1544	52.74	47.62
Number of structural analysis			19,670	28,059	1,500,000	29,750	27,720

the best weight as 389.334 lb after a minimum of 5750 structural analyses. In this case, the IMBA [15] found the best results in terms of the mean weight of all tested optimizers.

Table 5 provides the design variables and compares the results obtained by using the COA and MCOA and those of the other researches for the spatial 120-bar dome case study. It can be observed that the MCOA was the most efficient optimizer in terms of optimized weight and robustness when compared with the proposed approaches in [51,58–60] in terms of mean and best objective function values. All optimization runs of COA and MCOA were successful and converged to a feasible design.

According to Table 6, the result obtained by the MCOA is meaningfully lighter than those of the SAHS [63] and TLBO [12] approaches. It can be observed that MCOA is competitive with the other optimizers such as [12,62,63] in finding optimum designs considered in this study. Furthermore, the mean weight is close to the best weight and a small standard deviation on optimized weight is observed. The results of the COA and MCOA satisfied all design constraints with feasible solutions. The COA and MCOA obtained results without violation of the constraints. Kim and Byun [64] commented on the feasibility of the solutions presented in the literature for the planar 200-bar truss structure.

5. Concluding remarks and direction of future research

In this paper, a modified version of the Coyote Optimization Algorithm (COA) denoted MCOA was proposed to solve four structures optimization problems. This version uses the Tinkerbell chaotic map-generating to define the scatter and association probabilities and an adaptive procedure of updating parameters related to social condition.

The results showed that the MCOA is competitive with recent results from literature in terms of the best and mean (objective function values to 50 runs) measures as presented in Tables 3–6. Comparing the MCOA with the original COA, the worst measures were smaller in both cases, which means that the robustness of the algorithm was improved.

In future works, further improvements on the MCOA using ensemble strategies will be a pursuit to other optimization classes and multiobjective approaches in structural optimization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank National Council of Scientific and Technologic Development of Brazil - CNPq (Grants: 150501/2017-0-PDJ, 405101/2016-3-Univ, 404659/2016-0-Univ, 204910/2017-0-PDE and 204893/2017-8-PDE, 307958/2019-1-PQ, 307966/2019-4-PQ) and Fundação Araucária (PRONEX-FA/CNPq 042/2018) for its financial support of this work. Furthermore, the authors wish to thank the Editor and anonymous referees for their constructive comments and recommendations, which have significantly improved the presentation of this paper.

References

[1] Le DT, Bui D-K, Ngo TD, Nguyen Q-H, Nguyen-Xuan H. A novel hybrid methods combining electromagnetism-like mechanism and firefly algorithms for constrained design optimization of discrete truss structures. Comput Struct 2019;212:20–42.

[2] Ho-Huu V, Nguyen-Thoi T, Vo-Duy T, Nuyen-Tang T. An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. Comput Struct 2016;165:59–75.

[3] Talatahari S, Kaveh A, Sheikholeslami R. An efficient charged system search using chaos for global optimization problems. Int J Optim Civil Eng 2011;2:305–25.

[4] Sheikholeslami R, Kaveh A. A survey of chaos embedded meta-heuristic algorithms. Int J Optim Civil Eng 2013;4:617–33.

[5] Kaveh A, Javadi SM. Chaos-based firefly algorithms for optimization of cyclically large-size braced steel domes with multiple frequency constraints. Comput Struct 2019;214:28–39.

[6] Kaveh A, Moghanni RM, Javadi SM. Optimum design of large steel skeletal structures using chaotic firefly optimization algorithm based on the Gaussian map. Struct Multidisp Optim 2019:879–94.

[7] Kaveh A. Advances in metaheuristic algorithms for optimal design of structures. 2nd ed. Switzerland: Springer International Publishing; 2017.

[8] Kaveh A. Applications of metaheuristic optimization algorithms in civil engineering. Switzerland: Springer; 2017.

[9] Wu G, Pedrycz W, Suganthan PN, Mallipeddi R. A variable strategy for evolutionary algorithms handling equality constraints. Appl Soft Comput 2015;37:774–86.

[10] Shih CJ, Lee HW. Modified double-cuts approach in 25-bar and 72-bar fuzzy truss optimization. Comput Struct 2006;84:2100–4.

[11] Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 2009;87(7–8):435–43.

[12] Degertekin SO, Hayalioglu MS. Sizing truss structures using teaching-learning-based optimization. Comput Struct 2013;119:177–88.

[13] Kaveh A, Khayatazad M. Ray optimization for size and shape optimization of truss structures. Comput Struct 2013;117:82–94.

[14] Hasançebi O, Azad SK. Adaptive dimensional search: a new metaheuristic algorithm for discrete truss sizing optimization. Comput Struct 2015;154:1–16.

[15] Sadollah A, Eskandar H, Bahreininejad A, Kim JH. Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures. Comput Struct 2015;149:1–16.

[16] Lieu QX, Do TTD, Lee J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 2018;195:99–112.

[17] Assimi H, Jamali A. A hybrid algorithm coupling genetic programming and Nelder-Mead for topology and size optimization of trusses with static and dynamic constraints. Expert Syst Appl 2018;95:127–41.

[18] Degertekin SO, Lamberti L, Ugur IB. Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. Appl Soft Comput 2018;70:903–28.

[19] Degertekin SO, Lamberti L, Ugur IB. Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm. Appl Soft Comput 2019;79:363–90.

[20] Kaveh A, Mahdavi VR. Multi-objective colliding bodies optimization algorithm for design of trusses. J Comput Des Eng 2019;6:49–59.

[21] Zhengtong H, Zhengqi G, Xiaokui M, Wanglin C. Multimaterial layout optimization of truss structures via an improved particle swarm optimization algorithm. Comput Struct 2019;222:10–24.

[22] Salcedo-Sanz S. Modern meta-heuristics based on nonlinear physics processes: a review of models and design procedures. Phys Rep 2016;655:1–70.

[23] Coelho LS, Mariani VC. Particle swarm optimization with quasi-Newton local search for solving economic dispatch problem. IEEE International Conference on Systems, Man and Cybernetics, 4. p. 3109–13.

[24] Coelho LS, Mariani VC, da Luz MVF, Leite JV. Novel gamma differential evolution approach for multiobjective transformer design optimization. IEEE Trans Magn 2013;49(5):2121–4.

[25] Ayala HVH, Keller P, Morais MF, Mariani VC, Rao RV. Design of heat exchangers using a novel multiobjective free search differential evolution paradigm. Appl Therm Eng 2016;94:170–7.

[26] Neto JXV, Guerra Junior EJ, Moreno SR, Ayala HVH, Mariani VC, Coelho LS. Wind turbine blade geometry design based on multi-objective optimization using metaheuristics. Energy 2018;162:645–58.

[27] Vasconcelos Segundo EH, Mariani VC, Coelho LS. Design of heat exchangers using Falcon Optimization Algorithm. Appl Therm Eng 2019;156:119–44.

[28] Vasconcelos Segundo EH, Mariani VC, Coelho LS. Metaheuristic inspired on owls behavior applied to heat exchangers design. Therm Sci Eng Prog 2019;14:100431.

[29] Pierezan J., Coelho LS. Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: IEEE World Conference on Computational Intelligence, Congress on Evolutionary Computation, Rio de Janeiro, Brazil, pp. 2633–2640; 2018.

[30] Pierezan J, Maidl G, Yamao EM, Coelho LS, Mariani VC. Cultural coyote optimization algorithm applied to a heavy duty gas turbine operation. Energy Convers Manage 2019;199:111932.

[31] Saxena A, Kumar R, Das S. β-chaotic map-enabled grey wolf optimizer. Appl Soft Comput 2019;75:84–105.

[32] Chen K, Zhou F, Liu A. Chaotic dynamic weight particle swarm optimization for numerical function optimization. Knowl-Based Syst 2018;139:23–40.

[33] Oliva D, El Aziz MA, Hassanien AE. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm. Appl Energy 2017;200:141–54.

[34] Ewees AA, Elaziz MA. Performance analysis of chaotic multi-verse harris hawks optimization: a case study on solving engineering problems. Eng Appl Artif Intell 2020;88:103370.

[35] Jiang J, Yang X, Meng X, Li K. Enhance chaotic gravitational search algorithm (CGSA) by balance adjustment mechanism and sine randomness function for continuous optimization problems. Physica A 2020;537:122621.

[36] Coelho LS, Mariani VC. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. Comput Math Appl 2012;64:2371–82.

[37] Mariani VC, Duck ARK, Guerra FA, Coelho LS, Rao RV. A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. Appl Therm Eng 2012;42:119–28.

[38] Yu H, Zhao N, Wang P, Chen H, Li C. Chaos-enhanced synchronized bat optimizer. Appl Math Model 2020;77:1201–15.

[39] Coelho LS, Ayala HVH, Mariani VC. A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization. Appl Math Comput 2014;234:452–9.

[40] Tian D, Zhao X, Shi Z. Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm Evol Comput 2019;51:100573.

[41] Davidchack RL, Lai YC, Klebanoff A, Bolt EM. Towards complete detection of unstable periodic orbits in chaotic systems. Phys Lett A 2001;287:99–104.

[42] Inoue K, Ohya M. Sato K Application of chaos degree to some dynamical systems. Chaos, Solitons Fractals 2000;11:1377–85.

[43] Nusse HE, Yorke JA. Dynamics: numerical explorations. Heidelberg, Germany: Springer; 1997.

[44] Chin VJ, Salam Z. Coyote optimization algorithm for the parameter extraction of photovoltaic cells. Sol Energy 2019;194:656–70.

[45] Wu S-J, Chow P-T. Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 1995;56:979–91.

[46] Zhang JQ, Sanderson AC. JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 2009;13:945–58.

[47] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 2020;191:1245–87.

[48] Lampinen J. A constraint handling approach for the differential evolution algorithm. In: Proceedings of the Congress on Evolutionary Computation, vol. 2, Washington, DC, USA; 2002.

[49] Lee KS, Geem ZW, Lee S, Bae K. The harmony search heuristic algorithm for discrete structural optimization. Eng Optim 2005;37:663–84.

[50] Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 2012;102–103:49–63.

[51] Kaveh A, Mahdavi VR. Colliding bodies optimization method for optimum design of truss structures with continuous variables. Adv Eng Softw 2014;70:1–12.

[52] Dede T. Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures. KSCE J Civ Eng 2014;18:1–9.

[53] Cheng MY, Prayogo D, Wu YW, Lukito MM. A hybrid harmony search algorithm for discrete sizing optimization of truss structure. Autom Constr 2016;69:21–33.

[54] Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. J Constr Steel Res 2009;65:1558–68.

[55] Kaveh A, Ghazaan MI. A comparative study of CBO and ECBO for optimal design of skeletal structures. Comput Struct 2015;153:137–47.

[56] Schmit LA, Farshi B. Some approximation concepts for structural synthesis. AIAA J 1974;12:692–9.

[57] Kaveh A, Mahdavi VR. Colliding-bodies optimization for truss optimization with multiple frequency constraints. J Comput Civil Eng 2015;29:1–10.

[58] Kaveh A, Zolghadr A. Democratic PSO for truss layout and size optimization with frequency constraints. Comput Struct 2014;130:10–21.

[59] Kaveh A, Zolghadr A. Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability. Comput Struct 2014;102–103:14–27.

[60] Tejani GG, Savsanin VJ, Patel VK. Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. J Comput Des Eng 2016;3:226–49.

[61] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;87:267–83.

[62] Lamberti L. An efficient simulated annealing algorithm for design optimization of truss structures. Comput Struct 2008;86:1936–53.

[63] Degertekin SO. Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 2012;92–93:229–41.

[64] Kim TH, Byun JI. Truss sizing optimization with a diversity-enhanced cyclic neighborhood network topology particle swarm optimizer. Mathematics 2020;8:1–21. Article 1087.