Advances in Engineering Software 67 (2014) 136–147

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Chaotic swarming of particles: A new method for size optimization of truss structures

A. Kaveh [a,*], R. Sheikholeslami [b], S. Talatahari [c], M. Keshvari-Ilkhichi [b]

[a] Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran
[b] Department of Civil and Environmental Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
[c] Department of Civil Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 29 August 2012
Received in revised form 13 September 2013
Accepted 22 September 2013
Available online 13 October 2013

Keywords:
Meta-heuristic algorithm
Swarm intelligence
Particle swarm optimizer
Chaos theory
Truss structures
Size optimization

ABSTRACT

A new combination of swarm intelligence and chaos theory is presented for optimal design of truss structures. Here the tendency to form swarms appearing in many different organisms and chaos theory has been the source of inspiration, and the algorithm is called chaotic swarming of particles (CSP). This method is a kind of multi-phase optimization technique which employs chaos theory in two phases, in the first phase it controls the parameter values of the particle swarm optimization (CPVPSO) and the second phase is utilized for local search (CLSPSO). Some truss structures are optimized using the CSP algorithm, and the results are compared to those of the other meta-heuristic algorithms showing the effectiveness of the new method.

1. Introduction

Truss structures form a broad category of man-made structures, including bridges, towers, cranes, roof support trusses, building exoskeletons, and temporary construction frameworks. Trusses derive their utility and distinctive look from their simple construction: rod elements (bars) which exert only axial forces, connected concentrically with welded, pinned or bolted joints. These utilitarian structures are ubiquitous in the industrialized world and can be extremely complex and thus difficult to model. For example, the Eiffel Tower, perhaps the most famous truss structure in the world, contains over 15,000 girders connected at over 30,000 points [1] and even simpler structures, such as railroad bridges, routinely contain hundreds of members of varying lengths [2]. Therefore due to the increasing tendency on the prices of materials, the utilization of the modern design optimization tools becomes a necessity. In engineering the main objective of optimization is to comply with basic standards but also to achieve good economic results. Design variables involved in optimum design of truss structures can be considered as sizing (finding the optimal sections for elements), lay out (the optimum location of the joints in the structure), and topology (finding the number of members of the structure and the way in which these members are connected to each other) variables.

On the other hand, optimal design of (minimizing the weight) a structure while at the same time satisfying various requirements on structural response, cost, aesthetics, and manufacturing is a complicated task. Experienced engineers may be able to come up with solutions that fulfill some of the requirements, but they will seldom be able to come up with the optimal structure [3]. In order to both optimize the structure and meet the given requirements, researchers use various techniques. One of them is design optimization with meta-heuristic algorithms. Meta-heuristics are developed to tackle complex optimization problems where other optimization methods have failed to be either effective or efficient. Meta-heuristic algorithms which are created by the simulation of the natural processes try to find the optimal solution in a stochastic manner and avoid local optimum solutions. These algorithms impose fewer mathematical requirements and they do not require very well defined mathematical models.

Among these phenomenon-mimicking methods, algorithms inspired from the collective behavior of species such as ants, bees, wasps, termite, fishes, and birds are referred as swarm intelligence algorithms [4]. Here, we introduce a new combination of chaos theory and swarm intelligence which is called the chaotic swarming of particles (CSP) for optimum design of truss structures which can be considered as a suitable field to investigate the efficiency of this algorithm. One of these swarm intelligence algorithms is particle swarm optimization (PSO) which is a population-based meta-heuristic discovered through simulation of social models of

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

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.09.006

bird flocking, fish schooling, and swarming able to find optimal solution(s) to the non-linear numeric problems. PSO was first introduced in 1995 by Eberhart and Kennedy [5], and has attracted much attention in various fields. However, PSO can easy be trapped in local optimal point when dealing with some complex and multimodal functions.

Recently, chaos and PSO have been combined in different studies for different purposes. Some of the works have intended to show the chaotic behaviors in the PSO process. In some of the works, chaos has been used to overcome the limitations of PSO [6]. Hence previous research can be categorized into two types. In the first type, chaos is embedded into the PSO velocity updating equation, *i.e.*, a chaotic map is used to control the value of parameters in the velocity updating equation [7–12]. In the second type, chaotic search is inserted in the PSO formulation [13–16]. In this article, we combined the two types of improvements to control the value of the parameters and to increase the local PSO search capability. This enhances search behavior and allows to avoid local optima.

The present article consists of five sections. After the introduction in Section 1, the proposed method is described in Section 2. The formulation of truss sizing problems is presented in Section 3. Benchmark examples are studied in Section 4 and the results of the proposed method are compared with literature. The paper is concluded in Section 5.

2. Chaotic swarming of particles

2.1. Standard particle swarm optimizer

PSO involves a number of particles, which are initialized randomly in the space of the design variables. These particles fly through the search space and their positions are updated based on the best positions of individual particles and the best position among all particles in the search space which in truss sizing problems corresponds to a particle with the smallest weight [5]. In PSO, a swarm consists of N particles moving around in a D-dimensional search space. The position of the jth particle at the kth iteration is used to evaluate the quality of the particle and represents candidate solution(s) for the search or optimization problems. The update moves a particle by adding a change velocity V_j^{k+1} to the current position X_j^k as follows:

$$\begin{aligned} V_j^{k+1} &= wV_j^k + c_1 \times r_{1j}^k \otimes (P_j^k - X_j^k) + c_2 \times r_{2j}^k \otimes (P_g^k - X_j^k) \\ X_j^{k+1} &= X_j^k + V_j^k \end{aligned} \tag{1}$$

where w is an inertia weight to control the influence of the previous velocity; r_{1j}^k and r_{2j}^k are random numbers uniformly distributed in the range of (0,1); c_1 and c_2 are two acceleration constants namely called cognitive and social parameter, respectively; P_j^k is the best position of the jth particle up to iteration k; P_g^k is the best position among all particles in the swarm up to iteration k. In order to increase PSO's exploration ability, the inertia weight is now modified during the optimization process with the following equation:

$$w^{k+1} = w^k \times D_r \times rand \tag{2}$$

where D_r is the damping ratio which is a constant number in the interval (0,1); and *rand* is a uniformly distributed random number in the range of (0,1).

In standard PSO algorithm, the information of local best and global best were shared by next generation particles. In this paper we use an improved PSO, which uses the dynamic inertia weight that decreases according to iterative generation increasing (Eq. (2)). A larger inertia weight facilitates global exploration and a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. The inertia weight, w, controls the momentum of the particle by weighing the contribution of the previous velocity: basically it controls how much memory of the previous flight direction should influence the new velocity [17]. Fig. 1 shows the flowchart of standard PSO.

2.2. Chaotic update of PSO internal parameters (CPVPSO)

Chaos and random signals share the property of long term unpredictable irregular behavior. Many random generators in programming softwares as well as the chaotic maps are deterministic. However, chaos can help order to arise from disorder. Similarly, many optimization algorithms are inspired from biological systems where order arises from disorder. In these cases disorder often indicates both non-organized patterns and irregular behavior, whereas order is the result of self-organization and evolution and often arises from a disorder condition or from the presence of dissymmetry [18]. On the other hand self-organization and evolution are two key factors of many stochastic optimization techniques such as PSO. Due to these common properties between chaos and optimization algorithms, simultaneous use of these concepts seems to improve the performance of the optimizer. Utilizing chaotic sequences for particle swarm optimization [7], harmony search algorithm [19], artificial bee colony [20], Big Bang–Big Crunch [21], imperialist competitive algorithm [18,22], and charged system search [23] are some familiar examples of this combination. Seemingly the benefit of such combination is a generic for other stochastic optimization and experimental studies have confirmed this; however, this has not yet mathematically been proven.

In this phase, when a random number is needed by PSO algorithm, it can be generated by iterating one step of the chosen chaotic map (cm) being started from a random initial condition of the first iteration of PSO. One-dimensional non-invertible maps are the simplest systems with capability of generating chaotic motion [24]. One of the well-known chaotic maps is the Logistic map. Logistic map is a polynomial map [25]. It is often cited as an example of how complex behavior can arise from a very simple non-linear dynamics equation. This map is defined by,

$$cm^{k+1} = \alpha \times cm^k(1 - cm^k) \tag{3}$$

where cm^k is the kth chaotic number, with k denoting the iteration number. It is trivial to show that if $0 < \alpha \leqslant 4$ then the interval (0,1) is mapped into itself, *i.e.* if $cm^0 \in (0,1)$ then $cm^k \in (0,1)$. It is proven that when $\alpha = 4$, Eq. (3) is totally in chaos state. The mathematical explanation is that all values between 0 and 1 except the fixed points (0.25, 0.5, 0.75) are produced randomly by iteration. Utilizing the chaos characteristic, which is sensitive to the initial value and setting n different initial values between 0 and 1 (except the fixed point) to cm^k in Eq. (3), one can get n chaos variables of different orbits.

In order to control values of PSO parameters by using chaotic maps, the approach described in [7] is followed in this research; r_{1j}^k, r_{2j}^k, and *rand* are generated from the iterations of chaotic map instead of using classical random number generator.

$$\begin{aligned} V_j^{k+1} &= w^k \times V_j^k + c_1 \times cm^k \otimes (P_j^k - X_j^k) + c_2 \times cm^k \otimes (P_g^k - X_j^k) \\ w^{k+1} &= w^k \times D_r \times cm^k \end{aligned} \tag{4}$$

2.3. Chaotic local search algorithm (CLSPSO)

In this phase, chaotic search is introduced in the PSO formulation. This is a kind of multi-phase optimization technique [14] because chaotic optimization and PSO coexist and are switched to each other according to certain conditions. Here, chaotic local

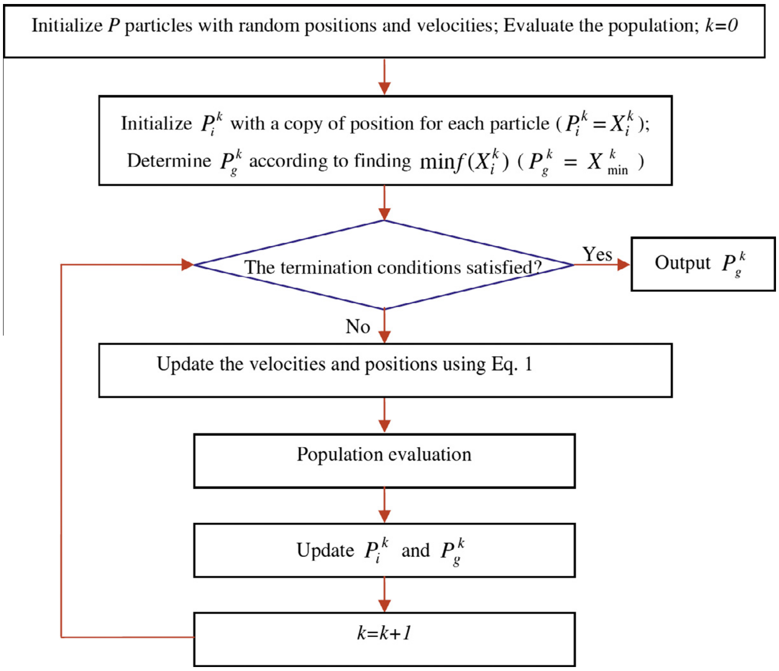

Initialize P particles with random positions and velocities; Evaluate the population; $k=0$
Initialize P_i^k with a copy of position for each particle ($P_i^k = X_i^k$);
Determine P_g^k according to finding $\min f(X_i^k)$ ($P_g^k = X_{\min}^k$)
The termination conditions satisfied?
Yes
Output P_g^k
No
Update the velocities and positions using Eq. 1
Population evaluation
Update P_i^k and P_g^k
$k=k+1$

Fig. 1. Flowchart of the standard PSO algorithm.

search (CLS) that uses Logistic map for the particle is incorporated to enhance search behavior and skip local optima.

Fig. 2 shows the ergodic property and the probability distribution of the Logistic map when $cm^0 = 0.3$ (iteration number is 2000). The figure indicates that the chaotic iteration is more random and chaotic search can more easily escape from local minima than other stochastic optimization algorithms. In other word, due to the non-repetition of chaos, it can carry out global searches at higher speeds [26]. Statistical properties of the logistic map are provided in Appendix A. The main drawback of this approach is that chaotic search works well in a small search space but generates unacceptable optimization time in a large search space [27]. The CLSPSO process is now described.

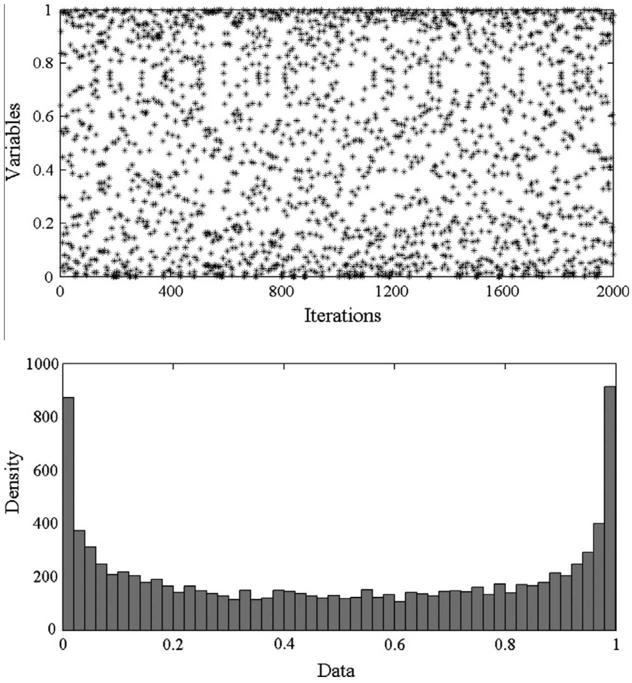


Fig. 2. Property of the logistic map.

- *First chaotic search process (CLSPSO1)*:

Step 1: Set $t = 0$. Initialize the number of the first chaotic search N_1, initial value of chaos variables (cm^0), the lower and upper bound of the decision variables ($X_{\min}$ and $X_{\max}$), and the number of particles. Determine the initial design variables for the jth particle as

$$X_j^0 = X_{\min} + cm_j^0 (X_{\max} - X_{\min}) \tag{5}$$

Step 2: Evaluate the objective function and determine X_g^0 by finding $f^* = \min f(X_j^0)$.
Step 3: Map chaotic variables cm^t into the variance range of the optimization variables by the following equation:

$$X_j^{t+1} = X_g^t + (2cm_j^t - 1)(X_g^t - X_j^t) \tag{6}$$

Step 4: Evaluate the new position (X_j^{t+1}).
Step 5: If the new solution is better than the initial solution $f(X_j^{t+1}) \leqslant f^*$, then $f^* = f(X_j^{t+1})$.
Step 6: Generate the next values of the chaotic variables by a chaotic map and set $t = t + 1$.
Step 7: If $t < N_1$ go to step 3, else stop the first chaotic search process and obtain the output X_g and f^* as the result of the CLSPSO1.
Step 8: Set X_g as the global best (P_g).

- *Second chaotic search process (CLSPSO2)*:

Step 1: Initialize the number of the second chaotic search N_2 and set $i = 1$.
Step 2: Using the PSO algorithm, generate the global best P_g^k.
Step 3: Set $X_g^i = P_g^k$.
Step 4: Update the global best position of the particles using the chaotic map by the following equation:

$$X_g^{i+1} = X_g^i + (2cm^i - 1)\frac{X_{\max} - X_{\min}}{k} \tag{7}$$

Step 5: If the new solution is better than the initial solution $f(X_g^{i+1}) \leqslant f(X_g^i)$, then $f^* = f(X_g^{i+1})$ and $P_g^k = X_g^{i+1}$.

Step 6: Generate the subsequent values of the chaotic variables by a chaotic map and set $i = i + 1$.

Step 7: If $i < N_2$ go to step 4, else stop the second chaotic search process and obtain the output P_g^k and f^* as the result of the CLSPSO2.

2.4. Combination of CPVPSO and CLSPSO

The framework of the CSP is illustrated in Fig. 3. In the CLSPSO1 phase, the initial positions of the particles are determined chaotically in the search space. Also the values of the fitness function for the particles are calculated. The best particle among the entire set of particles is treated as a global best (X_g). The first step is actually a special form of the "Scatter Search" method proposed already in 1977. There are a lot of algorithms using the same idea, such as "Self-Organizing Migration Algorithm".

After reaching a pre-defined number of iterations (N_1), the CLSPSO1 is stopped and switched to PSO while CPVPSO applies for controlling the value of parameters in the velocity updating equation. In the second phase, the CLSPSO2 (updating process) is activated if PSO stops moving. CLSPSO2 causes the particles to escape from local minima using the logistic map. After a better solution is found by the CLSPSO2 or after a fixed number of iterations (N_2), the PSO will continue. The algorithm is terminated when the termination criterion has been met: that is, if there is no significant improvement in the solution.

For many heuristic algorithms it is a common feature that if all the agents get gathered in a small space, *i.e.*, if the agents are trapped in part of the search space, escaping from this may be very difficult. In order to further understand how CSP can solve this problem, let us consider the following mathematical minimization problem:

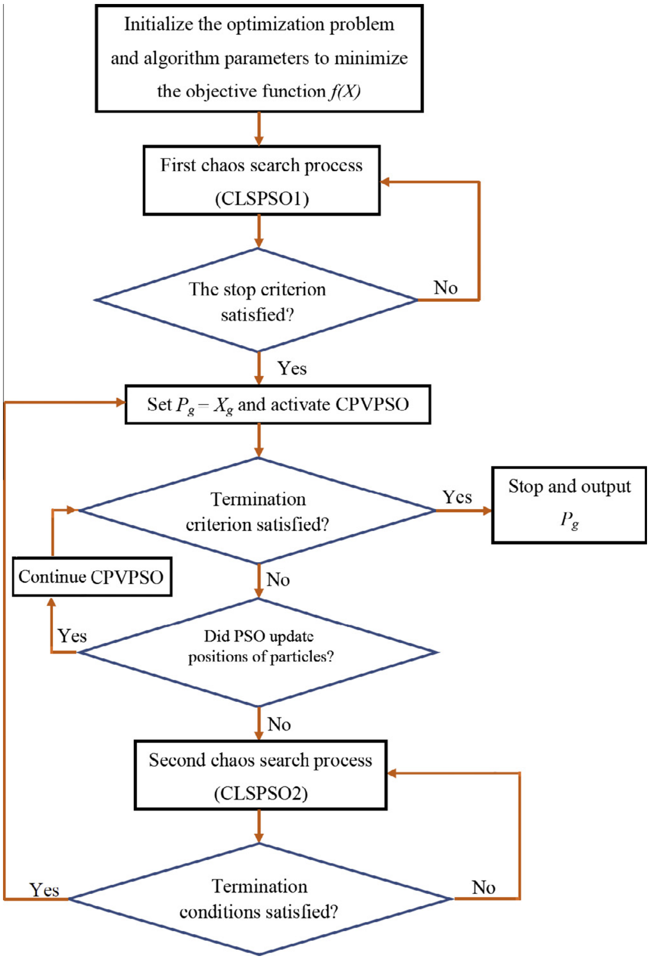

Initialize the optimization problem and algorithm parameters to minimize the objective function f(X)
First chaos search process (CLSPSO1)
The stop criterion satisfied?
No
Yes
Set $P_g = X_g$ and activate CPVPSO
Termination criterion satisfied?
Yes
Stop and output P_g
Continue CPVPSO
No
Yes
Did PSO update positions of particles?
No
Second chaos search process (CLSPSO2)
Yes
Termination conditions satisfied?
No

Fig. 3. Flowchart of the CSP algorithm developed in this research.

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{8}$$

The objective function is the Six-Hump Camelback function which is one of the standard test functions in optimization. This function has six local optima, two of which are global and are located at either $X_g = (-0.08984, 0.71266)$ or $X_g = (0.08984, -0.71266)$, each with a corresponding function value equal to $f(X_g) = -1.0316285$. Possible value bounds between -10.0 and 10.0 are used for the two design variables, x_1 and x_2, shown in Eq. (8). The total number of particles is 5, and the w^0 (inertia weight) and the D_r (damping ratio) are 0.9 and 0.99, respectively. Also $c_1 = 1.31$ and $c_2 = 4 - c_1$. After 50 searches, the CLSPSO1 was stopped with a function value of -1.0315957, and switched to the PSO. After 200 searches when $f = -1.0316279$, CLSPSO2 is activated. CSP finally found a near-optimal vector, $X = (-0.08984, 0.71265)$, that has a function value of -1.0316284. In Fig. 4 a plot of trajectory is presented that is obtained using the CSP for this example. This figure shows the chaotic behavior of the particle. Notice how the oscillations increase in amplitude, but then rapidly decrease. Near the end of the sample ($k = 300$) the amplitude increases again, but it eventually decreases. The observed increase in amplitude is caused by the chaotic nature of the parameters. In short, this implies that the trajectory of the particle will converge most of the time, occasionally taking divergent steps. The plot shows that this system "recovers" after taking large divergent steps. The recovery is caused by the fact that approximately 50% of the time the particle takes a step along a convergent trajectory. Therefore, chaotically divergent behavior can have a positive influence on the diversity of the solutions. This property is especially valuable when optimizing functions that contain many local minima. In designing the CSP algorithm, two opposing criteria are taken into account: exploration of the search space and exploitation of the best solutions found. In exploration, new regions must be searched to make sure that all regions of the search space are fairly explored and that the search is not limited to only a reduced number of regions. In exploitation, the smaller regions are explored more thoroughly to find better solutions. Thus in order to have a reasonable balance between the exploration and exploitation, we use the CPVPSO and CLSPSO mechanisms.

3. Statement of truss sizing optimization problems

Size optimization of truss structures involves determining optimum values for member cross-sectional areas, A_i, that minimize the structural weight W. This minimum weight design should also satisfy the inequality constraints that limit design variable sizes

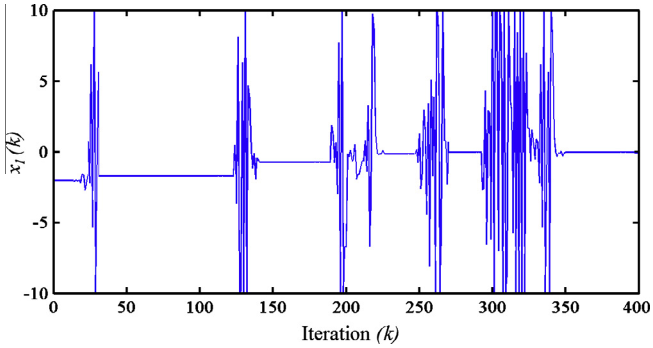


Fig. 4. Chaotic trajectory of a particle, obtained using the CSP method.

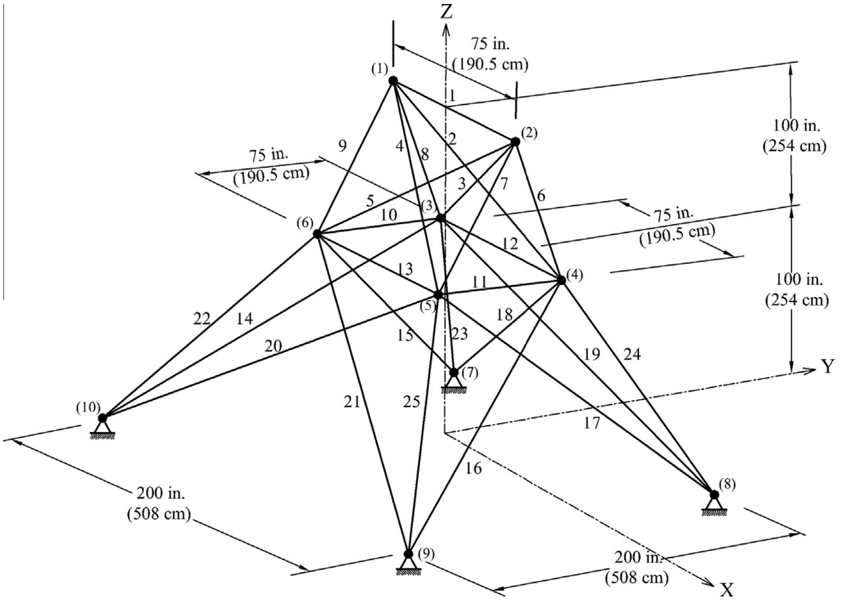


Fig. 5. Schematic of the spatial 25-bar truss structure.

Table 1
Loading conditions for the spatial 25-bar truss structure.

Node	Case 1			Case 2		
	P_X	P_Y	P_Z	P_X	P_Y	P_Z
	kips (kN)	kips (kN)	kips (kN)	kips (kN)	kips (kN)	kips (kN)
1	0.0	20.0 (89)	−5.0 (22.25)	1.0 (4.45)	10.0 (44.5)	−5.0 (22.25)
2	0.0	−20.0 (89)	−5.0 (22.25)	0.0	10.0 (44.5)	−5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 2
Stress limits for the spatial 25-bar truss structure.

Element group		Compressive stress limits ksi (MPa)	Tensile stress limits ksi (MPa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	A_{2-5}	11.590 (79.913)	40.0 (275.80)
3	A_{6-9}	17.305 (119.31)	40.0 (275.80)
4	A_{10-11}	35.092 (241.96)	40.0 (275.80)
5	A_{12-13}	35.092 (241.96)	40.0 (275.80)
6	A_{14-17}	6.759 (46.603)	40.0 (275.80)
7	A_{18-21}	6.959 (47.982)	40.0 (275.80)
8	A_{22-25}	11.082 (76.410)	40.0 (275.80)

Table 3
Comparison of optimized designs for the 25-bar truss problem.

Element group		Optimal cross-sectional areas (in.^{2})					
		ACO [28]	BB-BC [29]	PSO [30]	HPSACO [31]	CSS [32]	CSP
1	A_1	0.010	0.010	0.010	0.010	0.010	0.010
2	A_2–A_5	2.042	1.993	2.052	2.054	2.003	1.910
3	A_6–A_9	3.001	3.056	3.001	3.008	3.007	2.798
4	A_{10}–A_{11}	0.010	0.010	0.010	0.010	0.010	0.010
5	A_{12}–A_{13}	0.010	0.010	0.010	0.010	0.010	0.010
6	A_{14}–A_{17}	0.684	0.665	0.684	0.679	0.687	0.708
7	A_{18}–A_{21}	1.625	1.642	1.616	1.611	1.655	1.836
8	A_{22}–A_{25}	2.672	2.679	2.673	2.678	2.660	2.645
Best weight (lb)		545.03	545.16	545.21	544.99	545.10	545.09
Average weight (lb)		545.74	545.66	546.84	545.52	545.58	545.20
Std. dev. (lb)		0.940	0.491	1.478	0.315	0.412	0.487
No. of analyses		3520	12,500	9596	9875	7000	17,500

Table 4
Results of sensitivity analysis carried out to find the best combination of the parameters of CSP for the 25-bar truss problem.

Cases	N	D_r	C_1	C_2	N_1	N_2	Weight (lb)
1	10	0.80	2.00	2.00	20	40	584.31
2	10	0.99	1.70	2.30	10	50	630.72
3	20	0.80	2.00	2.00	20	40	553.87
4	20	0.99	1.60	2.40	10	50	584.60
5	30	0.80	2.00	2.00	20	40	561.37
6	30	0.99	1.50	2.50	10	50	549.17
7	40	0.80	1.40	2.60	20	40	550.71
8	40	0.99	1.41	2.59	10	50	546.79
9	50	0.80	1.30	2.70	20	40	546.95
10	50	0.99	1.31	2.69	10	50	545.09

and structural responses. The optimal design of a truss can be formulated as:

$$\text{minimize} \quad W(\{x\}) = \sum_{i=1}^{n} \gamma_i \cdot A_i \cdot L_i \tag{9}$$

$$\begin{aligned} \text{subject to} \quad & \delta_{\min} \leqslant \delta_i \leqslant \delta_{\max} \quad i = 1,2,\ldots,m \\ & \sigma_{\min} \leqslant \sigma_i \leqslant \sigma_{\max} \quad i = 1,2,\ldots,n \\ & A_{\min} \leqslant A_i \leqslant A_{\max} \quad i = 1,2,\ldots,ng \end{aligned} \tag{10}$$

where $W(\{x\})$ = weight of the structure; n = number of members making up the structure; m = number of nodes; ng = number of groups (number of design variables); γ_i = material density of member i; L_i = length of member i; A_i = cross-sectional area of member i chosen between $A_{\min}$ and $A_{\max}$; min = lower bound and max = upper bound; σ_i and δ_i are the element stress and nodal displacement, respectively.

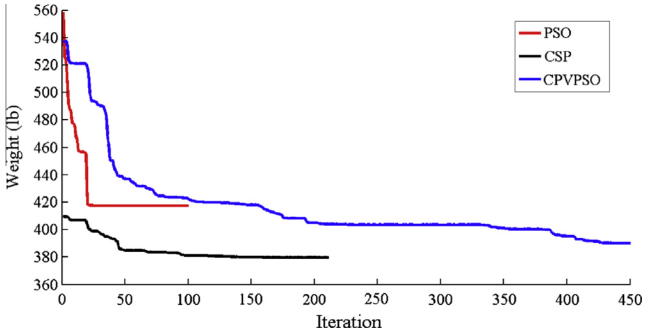


Fig. 7. Comparison of convergence curves of algorithm variants for the 72-bar truss structure.

To handle the side constraints if the solutions are not in the allowable limits, the particles positions are determined as

$$\begin{cases} A_i > A_{\max} \Rightarrow A_i^{new} = A_{\max} \\ A_i < A_{\min} \Rightarrow A_i^{new} = A_{\min} \end{cases} \tag{11}$$

An appropriate penalty function was utilized to handle non-linear constraints: if the design is feasible, the penalty is zero. Conversely, the amount of penalty is computed by normalizing constraint violation with respect to the corresponding allowable

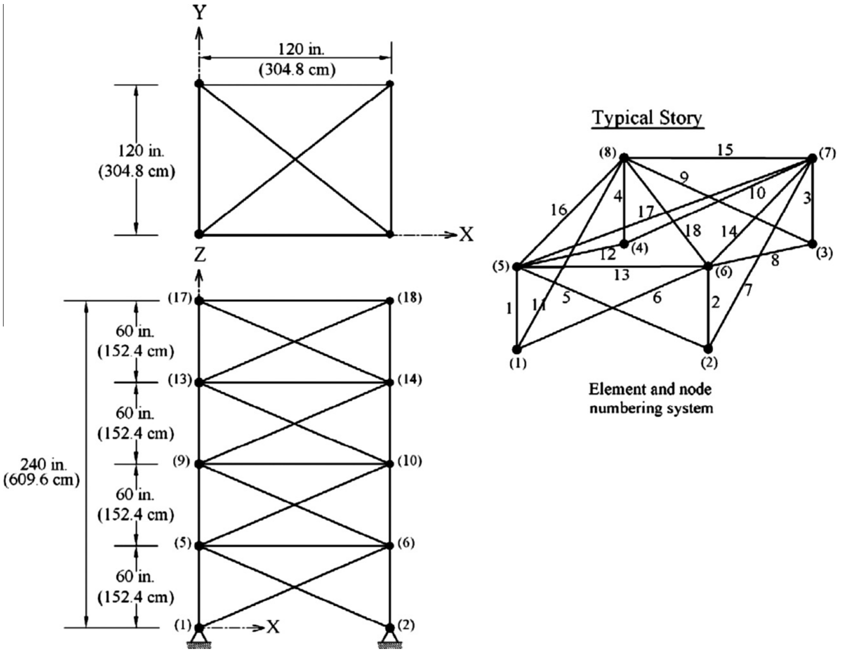


Fig. 6. Schematic of the spatial 72-bar truss structure.

Table 5
Loading conditions for the spatial 72-bar truss structure.

Node	Case 1			Case 2		
	P_X kips (kN)	P_Y kips (kN)	P_Z kips (kN)	P_X kips (kN)	P_Y kips (kN)	P_Z kips (kN)
17	5.0 (22.25)	5.0 (22.25)	−5.0 (22.25)	0.0	0.0	−5.0 (22.25)
18	0.0	0	0	0.0	0.0	−5.0 (22.25)
19	0.0	0.0	0.0	0.0	0.0	−5.0 (22.25)
20	0.0	0.0	0.0	0.0	0.0	−5.0 (22.25)

Table 6
Comparison of optimized designs for the 72-bar truss problem.

Element group		Optimal cross-sectional areas (in.^{2})				
		GA [33]	PSO [34]	BB-BC [29]	SAHS [37]	CSP
1	A_1–A_4	1.755	1.743	1.9042	1.860	1.94459
2	A_5–A_{12}	0.505	0.518	0.5162	0.521	0.50260
3	A_{13}–A_{16}	0.105	0.100	0.1000	0.100	0.10000
4	A_{17}–A_{18}	0.155	0.100	0.1000	0.100	0.10000
5	A_{19}–A_{22}	1.155	1.308	1.2582	1.271	1.26757
6	A_{23}–A_{30}	0.585	0.519	0.5035	0.509	0.50990
7	A_{31}–A_{34}	0.100	0.100	0.1000	0.100	0.10000
8	A_{35}–A_{36}	0.100	0.100	0.1000	0.100	0.10000
9	A_{37}–A_{40}	0.460	0.514	0.5178	0.485	0.50674
10	A_{41}–A_{48}	0.530	0.546	0.5214	0.501	0.51651
11	A_{49}–A_{52}	0.120	0.100	0.1000	0.100	0.10752
12	A_{53}–A_{54}	0.165	0.109	0.1007	0.100	0.10000
13	A_{55}–A_{58}	0.155	0.161	0.1566	0.168	0.15618
14	A_{59}–A_{66}	0.535	0.509	0.5421	0.584	0.54022
15	A_{67}–A_{70}	0.480	0.497	0.4132	0.433	0.42229
16	A_{71}–A_{72}	0.520	0.562	0.5756	0.520	0.57941
Best weight (lb)		383.12	381.91	379.66	380.62	379.97
Average weight (lb)		N/A	N/A	381.85	382.42	381.56
Std. dev. (lb)		N/A	N/A	1.912	1.380	1.803
No. of analyses		N/A	N/A	13,200	13,742	10,500

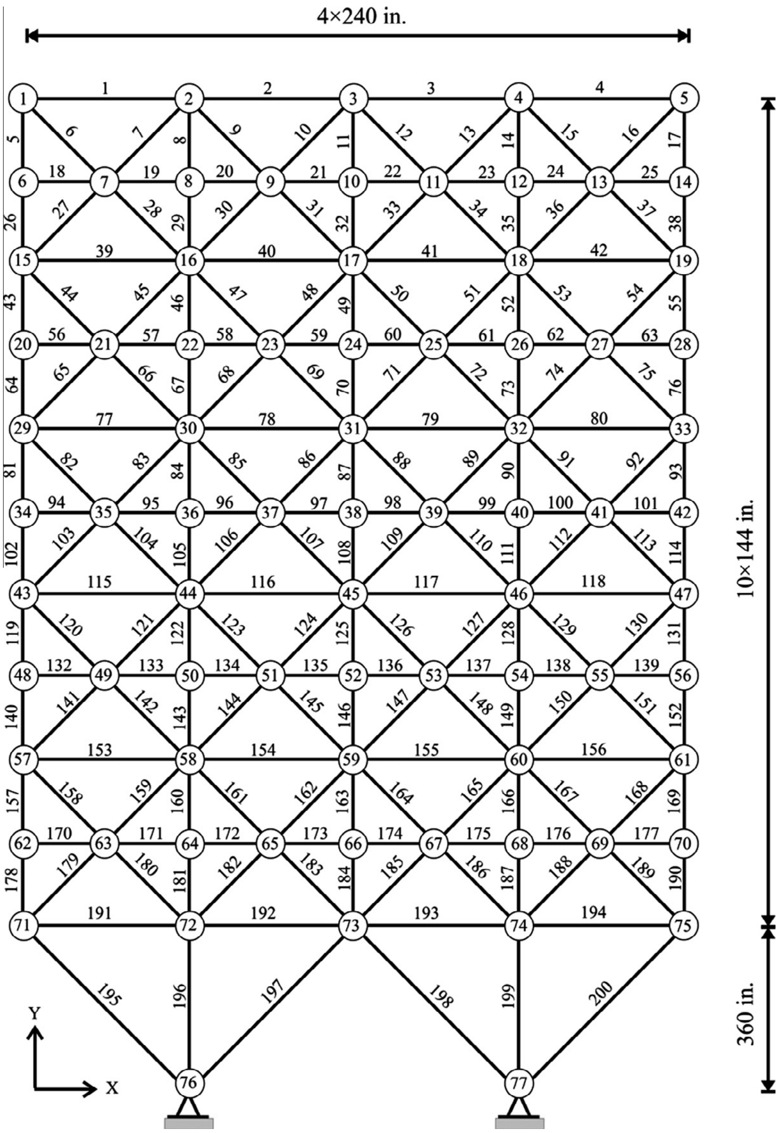


Fig. 8. Schematic of the planar 200-bar truss structure.

Table 7
Comparison of optimized designs for the 200-bar truss problem.

Element group	CMLPSA [35]	GA [36]	SAHS [37]	CSP
A_{1-4}	0.1468	0.3469	0.154	0.1480
$A_{5,\ 8,\ 11,\ 14,\ 17}$	0.9400	1.0810	0.941	0.9460
A_{19-24}	0.1000	0.1000	0.100	0.1010
$A_{18,\ 25,\ 56,\ 63,\ 94,\ 101,\ 132,\ 139,\ 170,\ 177}$	0.1000	0.1000	0.100	0.1010
$A_{26,\ 29,\ 32,\ 35,\ 38}$	1.9400	2.1421	1.942	1.9461
$A_{6,\ 7,\ 9,\ 10,\ 12,\ 13,\ 15,\ 16,\ 27,\ 28,\ 30,\ 31,\ 33,\ 34,\ 36,\ 37}$	0.2962	0.3470	0.301	0.2979
A_{39-42}	0.1000	0.1000	0.100	0.1010
$A_{43,\ 46,\ 49,\ 52,\ 55}$	3.1042	3.5650	3.108	3.1072
A_{57-62}	0.1000	0.3470	0.100	0.1010
$A_{64,\ 67,\ 70,\ 73,\ 76}$	4.1042	4.8050	4.106	4.1062
$A_{44,\ 45,\ 47,\ 48,\ 50,\ 51,\ 53,\ 54,\ 65,\ 66,\ 68,\ 69,\ 71,\ 72,\ 74,\ 75}$	0.4034	0.4400	0.409	0.4049
A_{77-80}	0.1912	0.4400	0.191	0.1944
$A_{81,\ 84,\ 87\ 90,\ 93}$	5.4284	5.9520	5.428	5.4299
A_{95-100}	0.1000	0.3470	0.100	0.1010
$A_{102,\ 105,\ 108,\ 111,\ 114}$	6.4284	6.5720	6.427	6.4299
$A_{82,\ 83,\ 85,\ 86,\ 88,\ 89,\ 91,\ 92,\ 103,\ 104,\ 106,\ 107,\ 109,\ 110,\ 112,\ 113}$	0.5734	0.9540	0.581	0.5755
$A_{115-118}$	0.1327	0.3470	0.151	0.1349
$A_{119,\ 122,\ 125,\ 128,\ 131}$	7.9717	8.5250	7.973	7.9747
$A_{133-138}$	0.1000	0.1000	0.100	0.1010
$A_{140,\ 143,\ 146,\ 149,\ 152}$	8.9717	9.3000	8.974	8.9747
$A_{120,\ 121,\ 123,\ 124,\ 126,\ 127,\ 129,\ 130,\ 141,\ 142,\ 144,\ 145,\ 147,\ 148,\ 150,\ 151}$	0.7049	0.9540	0.719	0.70648
$A_{135-156}$	0.4196	1.7639	0.422	0.4225
$A_{157,\ 160,\ 163,\ 166,\ 169}$	10.8636	13.3006	10.892	10.8685
$A_{171-176}$	0.1000	0.3470	0.100	0.1010
$A_{178,\ 181,\ 184,\ 187,\ 190}$	11.8606	13.3006	11.887	11.8684
$A_{158,\ 159,\ 161,\ 162,\ 164,\ 165,\ 167,\ 168,\ 179,\ 180,\ 182,\ 183,\ 185,\ 186,\ 188,\ 189}$	1.0339	2.1421	1.040	1.035999
$A_{191-194}$	6.6818	4.8050	6.646	6.6859
$A_{195,\ 197,\ 198,\ 200}$	10.8113	9.3000	10.804	10.8111
$A_{196,\ 199}$	13.8404	17.1740	13.870	13.84649
Best weight (lb)	25445.6	28533.1	25491.9	25467.9
Average weight (lb)	N/A	N/A	25610.2	25547.6
Std. dev. (lb)	N/A	N/A	141.85	135.09
No. of analyses	9650	51,360	19,670	31,700

Table 8
Results of sensitivity analysis carried out to find the best combination of the parameters of CSP for the 200-bar truss problem.

Cases	N	D_r	C_1	C_2	N_1	N_2	Weight (lb)
1	50	0.80	2.00	2.00	20	40	28065.39
2	50	0.99	1.70	2.30	10	50	28427.47
3	60	0.80	2.00	2.00	20	40	26321.65
4	60	0.99	1.60	2.40	10	50	26142.10
5	70	0.80	2.00	2.00	20	40	25893.64
6	70	0.99	1.50	2.50	10	50	25607.09
7	80	0.80	1.45	2.65	20	40	25791.78
8	80	0.99	1.40	2.60	10	50	25569.11
9	100	0.99	1.30	2.70	10	50	25467.95
10	120	0.99	1.25	2.85	10	50	25502.45

limit. Nodal displacements and element stresses can be obtained from structural analysis. These values are compared to the allowable limits to calculate the penalty functions as

$$\begin{cases} \delta_i^{\min} < \delta_i < \delta_i^{\max} \Rightarrow \Phi_\delta^{(i)} = 0 \\ \delta_i^{\min} > \delta_i \text{ or } \delta_i^{\max} < \delta_i \Rightarrow \Phi_\delta^{(i)} = \left| \frac{\delta_i - \delta_i^{\min/\max}}{\delta_i^{\min/\max}} \right| \end{cases} \quad i = 1, 2, \ldots, m \tag{12}$$

$$\begin{cases} \sigma_i^{\min} < \sigma_i < \sigma_i^{\max} \Rightarrow \Phi_\sigma^{(i)} = 0 \\ \sigma_i^{\min} > \sigma_i \text{ or } \sigma_i^{\max} < \sigma_i \Rightarrow \Phi_\sigma^{(i)} = \left| \frac{\sigma_i - \sigma_i^{\min/\max}}{\sigma_i^{\min/\max}} \right| \end{cases} \quad i = 1, 2, \ldots, n \tag{13}$$

where Φ_σ and Φ_δ are the values of stress penalty and nodal deflection penalty, respectively.

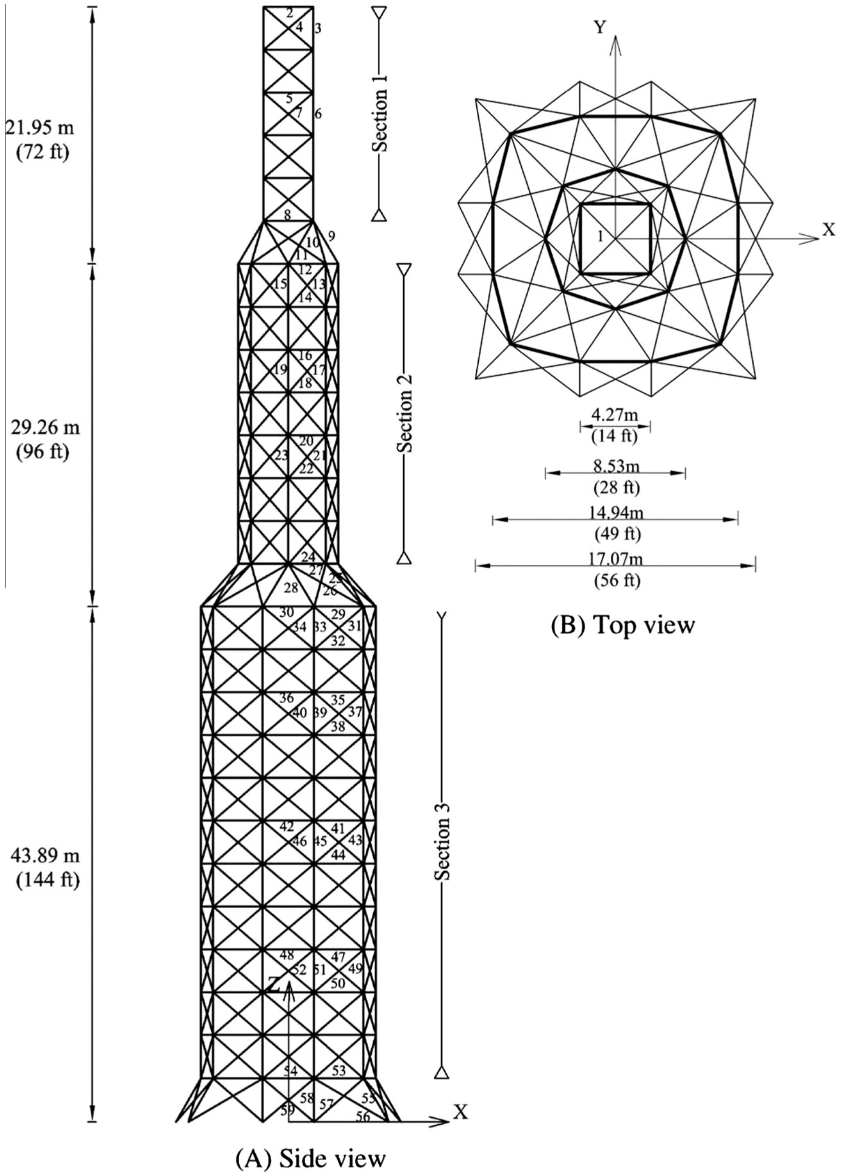
The objective function and optimization constraints are merged into the following pseudo-cost function that must be minimized:

$$f_{cost}(\{x\}) = (1 + \varepsilon_1 \cdot \sum \Phi)^{\varepsilon_2} \times W(\{x\}) \tag{14}$$

The penalty function method has some limitations. For example, penalty parameters are problem dependent and proper tuning of parameters is needed. If penalty parameters are large, penalty functions tend to become ill-conditioned near the boundary of the feasible domain and this may result in a local optimal solution or an infeasible solution. In this case, repeated runs are suggested by varying the penalty parameter until satisfactory results are obtained. Here, constants ε_1 and ε_2 are selected considering the exploration and the exploitation rate of the search space. Thus ε_1 is set to unity, and ε_2 is selected in a way that it decreases the penalties and reduces the cross-sectional areas. In the first steps of the search process ε_2 is set to 1.5, and ultimately increased to 3 [18]. Since CSP is independent of the type of penalty function, one can easily utilize another approach in the application of the CSP.

4. Test problems and results

In order to test the efficiency of the proposed method, some test problems previously treated by other investigators are studied: the weight minimization of 25-bar truss, 72-bar truss, 200-bar, and 942-bar truss. For all test cases, after a sensitivity analysis, the CSP internal parameters are set to: $w^0 = 0.9$, damping ratio $(D_r) = 0.99$, number of the first chaotic search $(N_1) = 50$ and number of the second chaotic search $(N_2) = 10$. Also for the first two examples, the maximum number of iteration is set to 1200, number of particles $(N) = 50$, and $c_1 = 1.31$, $c_2 = 2.69$. For the last two examples the maximum number of iteration is set to 2500, number of particles $(N) = 100$, and $c_1 = 1$, $c_2 = 3$. Therefore the maximum number of structural analyses is 63,000 for the first two examples, and 256,000 for the last two examples if all chaotic search iterations



(A) Side view

(B) Top view

Fig. 9. Schematic of the 26-story space tower: (a) side view; (b) top view.

were completed in each optimization cycle. The algorithms were implemented in Matlab and optimization runs were carried out on a computer with an Intel Core i5 CPU, 2.53 GHz processor, and 3.00 GB RAM. Also a direct stiffness method is utilized to analyze the structures.

Sensitivity analysis was performed for internal parameters of the CSP algorithm including D_r, c_1, c_2, N_1, N_2 and N (number of particles). In order to avoid the possible randomness of the search process due to the use of different initial solutions (*i.e.* initial populations and initial values of cm^0), each problem was solved 10 times. Because of the large computational time, only sensitivity analysis of the 25-bar truss and 200-bar truss problems were carried out and best combination of parameters obtained for these problems were used for the 72-bar truss and 942-bar truss problems, respectively.

4.1. Spatial 25-bar truss problem

The spatial 25-bar truss structure is schematized in Fig. 5. The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3) and the modulus of elasticity is 10 Msi (68.950 GPa). Elements can be divided into eight groups, as: (1) A_1, (2) A_{2-5}, (3) A_{6-9}, (4) A_{10-11}, (5) A_{12-13}, (6) A_{14-17}, (7) A_{18-21}, and (8) A_{22-25}. This spatial truss is subjected to two loading conditions as shown in Table 1. Maximum displacement limitations of ±0.35 in. (±8.89 mm) are imposed on all nodes in all coordinate directions. Stress constraints set for each element group are presented in Table 2. The range of cross sectional areas varies from 0.01 in.^{2} (0.0645 cm^2) to 3.4 in.^{2} (21.94 cm^2).

In Table 3, the optimum cross sectional areas and statistical information of the solution obtained by the CSP and the different methods containing ACO [28], BB-BC [29], PSO [30], HPSACO [31], and CSS [32] are presented. The best weight obtained by the CSP is 545.0977 lb after 350 iterations and 17,500 analyses which is about 28% of the maximum number of analyses. CSP outperforms standard PSO and BB-BC. The best weight optimized by CSP is consistent with literature but the average weight is slightly lower. It should be added that the CSP violates displacement constraints by a small amount of 0.657%. The results of the sensitivity analysis carried out to find the best combination of the parameters for CSP are presented in Table 4.

4.2. Spatial 72-bar truss problem

The spatial 72-bar truss structure shown in Fig. 6 has material density of 0.1 lb/in.^{3} (2767.990 kg/m^3) and modulus of elasticity of 10 Msi (68.950 GPa). The members are subjected to the stress limits of ±25 ksi (±172.375 MPa). The uppermost nodes are subjected to the displacement limits of ±0.25 in. (±0.635 cm) in both the x and y directions. Elements are divided into 16 groups using symmetry: (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}, (16) A_{71}–A_{72}. The minimum permitted cross-sectional area for each member is 0.10 in.^{2} (0.6452 cm^2), and the maximum cross-sectional area of each member is 4 in.^{2} (25.81 cm^2). Table 5 lists the different load cases applied to the structure.

Fig. 7 compares the convergence characteristic curve of the CSP, PSO with that of the CPVPSO, and standard PSO. For this spatial truss structure, it takes about 210, 450, and 100 iterations for the CSP, CPVPSO, and PSO algorithms to converge, respectively. Therefore, after 17%, 36%, and 8% of maximum number of analyses, the CSP, CPVPSO, and PSO are found as the best solutions, respectively. The best weight obtained by the PSO and CPVPSO is 416.21 lb and 390.14 lb while the best result of the CSP is 379.97 lb. These results demonstrate the performance of the CLSPSO mechanism in CSP. The best weight of GA [33], PSO [34], and SAHS [37] is 383.12, 381.91, and 380.62 lb, respectively. In addition, the average weight of CSP is 381.56 lb, while it is 381.85 and 382.42 lb for BB-BC and SAHS [29]. Table 6 compares CPS with other meta-heuristic algorithms presented in literature: the new algorithm is absolutely competitive with the other optimization methods.

4.3. Planar 200-bar truss problem

The planar 200-bar truss structure shown in Fig. 8 is designed for minimum weight. Truss elements are divided into 29 groups (design variables) All members are made of steel: the material density and modulus of elasticity are 0.283 lb/in.^{3} (7933.410 kg/m^3) and 30 Msi (206 GPa), respectively. Element stresses must not exceed ±10 ksi (68.95 MPa). There are three independent loading conditions: (1) 1.0 kip (4.45 kN) acting in the positive x-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62, and 71; (2) 10 kips (44.5 kN) 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 (3) conditions 1 and 2 acting together.

The minimum weight and the values of the cross sectional area obtained by CSP and some other previous studies reported in the literature such as a modified simulated annealing algorithm (CMLPSA) [35], an improved genetic algorithm (GA) [36], and self adaptive harmony search algorithm (SAHS) [37] are presented in Table 7. It can be seen that the CSP algorithm found an optimum weight of 25467.95 lb after approximately 317 iterations and 31,700 analyses which is about 12% of the maximum number of analyses. The optimal design obtained using the CSP algorithm showed an excellent agreement with the previous designs reported in the literature. The results of the sensitivity analysis carried out to find the best combination of the parameters for CSP are presented in Table 8.

4.4. Spatial 942-bar tower truss problem

The 26-story-tower space truss shown in Fig. 9 has 942 elements and 244 nodes. Fifty-nine design variables are used to represent the cross-sectional areas of 59 element groups in this structure, employing the symmetry of the structure. Fig. 9 shows the geometry and the 59 element groups. The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3) and the modulus of elasticity is 10 Gsi (68.950 GPa). The members are subjected to the stress limits of ±25 ksi (172.375 MPa) and the four nodes of the top level in the x, y and z directions are subjected to the displacement limits of ±15.0 in. (38.10 cm) (about 1/250 of the total height of the tower). The allowable cross-sectional areas in this example are selected from 0.1 to 200.0 in.^{2} (from 0.6452 to 1290.32 cm^2). Loading conditions are presented in Table 9.

Table 9
Loading conditions for the spatial 26-story tower.

Case number	Direction	Load
1	Vertical	3 kips (13.344 kN)
2	Vertical	6 kips (26.688 kN)
3	Vertical	9 kips (40.032 kN)
4	Horizontal (*X* direction)	1 kips (4.448 kN)
5	Horizontal (*X* direction)	1.5 kips (6.672 kN)
6	Horizontal (*Y* direction)	1 kips (4.448 kN)
7	Horizontal (*Y* direction)	1 kips (4.448 kN)

Table 10
Optimized designs obtained for the 26-story tower problem.

Members	Area	Members	Area	Members	Area
A_1	14.0925	A_{21}	4.3475	A_{41}	0.6235
A_2	8.6965	A_{22}	1.1995	A_{42}	2.9045
A_3	6.1505	A_{23}	6.2555	A_{43}	12.3365
A_4	0.9095	A_{24}	9.2665	A_{44}	1.2195
A_5	0.6245	A_{25}	8.9865	A_{45}	4.9785
A_6	4.6535	A_{26}	4.4975	A_{46}	1.0685
A_7	1.0435	A_{27}	2.9485	A_{47}	0.7465
A_8	13.0025	A_{28}	4.2215	A_{48}	1.4485
A_9	9.4465	A_{29}	5.9315	A_{49}	16.4445
A_{10}	6.7035	A_{30}	9.8325	A_{50}	1.8985
A_{11}	0.6035	A_{31}	13.8705	A_{51}	5.0325
A_{12}	1.2095	A_{32}	1.5125	A_{52}	1.0255
A_{13}	3.0725	A_{33}	3.0985	A_{53}	11.6285
A_{14}	1.0005	A_{34}	1.1185	A_{54}	15.4075
A_{15}	8.2485	A_{35}	0.5965	A_{55}	16.6135
A_{17}	0.7215	A_{37}	1.6875	A_{57}	3.1965
A_{18}	8.2665	A_{38}	8.0155	A_{58}	2.6845
A_{19}	1.0625	A_{39}	1.1215	A_{59}	4.3205
A_{20}	6.5035	A_{40}	4.7895		

Table 11
Comparison of optimization results for the 26-story tower problem.

	GA [29]	PSO [29]	BB-BC [29]	HBB-BC [29]	CSP
Best weight (lb)	56,343	60,385	53,201	52,401	52,200
Average weight (lb)	63,223	75,242	55,206	53,532	53,147
Std. dev. (lb)	6640.6	9906.6	2621.3	1420.5	1256.2
No. of analyses	50,000	50,000	50,000	30,000	48,500

After 485 iterations and 48,500 analyses (about 19% of the maximum number of analyses) CSP found an optimum weight corresponding to the design listed in Table 10. The best weights are 56,343 lb, 60,385 lb, 53,201 lb, and 52,401 lb for the GA, PSO, BB-BC and HBB-BC [29], respectively. In addition, CSP is more efficient in terms of average weight and standard deviation of optimized weight. The average weight obtained by CSP is 53,147 lb which is 15.94%, 29.36%, 3.73% and 0.72% lighter than GA, PSO, BB-BC, and HBB-BC, respectively. Table 11 provides the statistic information for this example.

5. Conclusions

Many meta-heuristic approaches are inspired by natural phenomena. This paper presented a new nature based algorithm. The method is called chaotic swarming of particles, and it is inspired from the chaotic and collective behavior of species such as bees, fishes, and birds. Here chaos theory is used to control the value of the parameters of PSO and to increase the local search capability of the PSO in order to enhance search behavior and skip local optima. The proposed approach not only performs exploration by using the population-based evolutionary searching ability of PSO, but also performs exploitation by using the chaotic local searching behavior.

Acknowledgement

The first author is grateful to the Iran National Science Foundation for the support.

Appendix A. Statistical properties of the logistic map

(The content of this section is based on Refs. [38,39]).

The logistic map is a transformation g: [0,1] → [0,1] defined by $g(y, a) = ay(1 - y)$, where a is a control parameter taking on values in the interval [0;4]. For the purpose of this article, the logistic map with $a = 4$ is used. In this case, the map has good qualities as a pseudo-random number generator.

For $a = 4$, trajectories determined by $g(y, 4)$ behave chaotically for almost all y_0 in [0,1]. Thus, in spite of the sensitivity of trajectories to initial states, this is not usually reflected in the distribution of states within long-term trajectories. In general, it is well-known that for maps exhibiting chaotic dynamics, the precise long-term predictions of individual trajectories are impossible due to sensitive dependence of trajectories on initial conditions. Therefore, it is natural to investigate the statistical properties of deterministic maps with chaotic dynamical behavior. The statistical properties of the logistic map with $a = 4$ are now described.

Let ([0,1], $\boldsymbol{B}$, μ) be a measure space where $\boldsymbol{B}$ is the Borel σ-algebra on [0,1] and μ is the Lebesgue measure on ([0,1], $\boldsymbol{B}$). Then the measure

$$v(A) = \int_A \frac{1}{\pi\sqrt{y(1-y)}}\mu(dy) \tag{A1}$$

is invariant under the transformation $g(y, 4)$, that is, $v(A) = v(g^{-1}(A))$ for all A in $\boldsymbol{B}$. The density of v, given by

$$\rho^*(y) = \frac{1}{\pi\sqrt{y(1-y)}}, \quad y \in [0,1] \tag{A2}$$

is the stationary density of the Frobenius–Perron operator P corresponding to the transformation $g(y, 4)$. This operator is defined by

$$\begin{aligned} P\rho(x) &= \frac{d}{dy}\int_{f^{-1}([0,1])} \rho(u)du \\ &= \frac{1}{4\sqrt{1-y}}\left[\rho\left(\frac{1}{2} - \frac{1}{2}\sqrt{1-y}\right) + \rho\left(\frac{1}{2} + \frac{1}{2}\sqrt{1-y}\right)\right] \end{aligned} \tag{A3}$$

A few elementary calculations confirm that:

- $\mu = \langle y \rangle = \int_0^1 y\rho^*(y)dy = 1/2$
- $\langle y^2 \rangle = \int_0^1 y^2\rho^*(y)dy = 3/8$
- $\langle y^2 \rangle - \langle y \rangle^2 = 1/8$
- $\sigma = \left[\langle y^2 \rangle - \langle y \rangle^2\right]^{1/2} = \frac{1}{2\sqrt{2}} = 0.35$
- $\|P^n\rho - \rho^*\| \to 0 \quad \text{as} \quad n \to \infty$

The logistic map $g(y, 4)$ is an ergodic transformation. As a consequence

$$\overline{Q} = \lim_{n\to\infty}\frac{1}{n}\sum_{k=0}^{n-1} Q(g^k(x)) = \int Q(y)\gamma(dy) = \langle y \rangle \tag{A4}$$

for any integrable function Q, where $g^k(y) = g(g^{k-1}(y))$. This result is the famous Birkhoff's ergodic theorem.

References

[1] Harriss J. The tallest tower – Eiffel and the Belle Epoque. Houghton Mifflin; 1975.
[2] Smith J, Hodgins J, Oppenheim I, Witkin A. Creating models of truss structures with optimization. ACM Trans Graph 2002;21(3):295–301.
[3] Sigmund O. Topology optimization: a tool for the tailoring of structures and materials. Philos Trans R Soc A 2000;358:211–27.
[4] Talbi EG. Metaheuristics: from design to implementation. New Jersey: John Wiley & Sons; 2009.
[5] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. Nagoya, Japan; 1995. p. 1942–8.
[6] Alatas B, Akin E. Chaotically encoded particle swarm optimization algorithm and its applications. Chaos Solit Fract 2009;41:939–50.
[7] Alatas B, Akin E, Ozer AB. Chaos embedded particle swarm optimization algorithms. Chaos Solit Fract 2009;40:1715–34.
[8] Park JB, Jeong YW, Kim HH, Shin JR. An improved particle swarm optimization for economic dispatch with valve-point effect. Int J Innov Energy Syst Power 2006;1(1):1–7.
[9] Jiang C, Etorre B. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization. Math Comput Simul 2005;68:57–65.
[10] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proceedings of the IEEE international conference on computational intelligence; 1998. p. 69–73.
[11] Jiang C, Bompard E. A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment. J Energy Convers Manage 2005;46:2689–96.
[12] Coelho LS, Mariani VC. A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch. Chaos Solit Fract 2009;39:510–8.
[13] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. Chaos Solit Fract 2005;25(5):1261–71.
[14] Xiang T, Liao X, Wong K. An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Appl Math Comput 2007;190:1637–45.
[15] Coelho LDS. A quantum particle swarm optimizer with chaotic mutation operator. Chaos Solit Fract 2008;37(5):1409–18.
[16] Meng H, Zheng P, Wu R, Hao X, Xie Z. A hybrid particle swarm algorithm with embedded chaotic search. In: Proceedings of the IEEE conference on cybernetics and intelligent systems. Singapore; 2004. p. 367–71.
[17] van den Bergh F, Engelbrecht AP. A study of particle swarm optimization particle trajectories. Inform Sci 2006;175:937–71.
[18] Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi AH. Imperialist competitive algorithm combined with chaos for global optimization. Commun Nonlinear Sci Numer Simul 2012;17:1312–9.
[19] Alatas B. Chaotic harmony search algorithm. Appl Math Comput 2010;29(4):2687–99.
[20] Alatas B. Chaotic bee colony algorithms for global numerical optimization. Expert Syst Appl 2010;37:5682–7.
[21] Alatas B. Uniform big bang-chaotic big crunch optimization. Commun Nonlinear Sci Numer Simul 2011;16(9):3696–703.
[22] Talatahari S, Kaveh A, Sheikholeslami R. Chaotic imperialist competitive algorithm for optimum design of truss structures. Struct Multidiscip Optim 2012;46:355–67.
[23] Talatahari S, Kaveh A, Sheikholeslami R. Engineering design optimization using chaotic enhanced charged system search algorithms. Acta Mech 2012;223:2269–85.
[24] Ott E. Chaos in dynamical systems. Cambridge, UK: Cambridge University Press; 2002.
[25] May RM. Simple mathematical models with very complicated dynamics. Nature 1976;261:459–74.
[26] Coelho L, Mariani V. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Syst Appl 2008;34:1905–13.
[27] Jia D, Zheng G, Qu B, Khurram Khan M. A hybrid particle swarm optimization algorithm for high-dimensional problems. Comput Indust Eng 2011;61:1117–22.
[28] Camp C, Bichon J. Design of space trusses using ant colony optimization. J Struct Eng ASCE 2004;130(5):741–51.
[29] Kaveh A, Talatahari S. Size optimization of space trusses using big bang-big crunch algorithm. Comput Struct 2009;87:1129–40.

[30] Schutte JF, Groenwold AA. Sizing design of truss structures using particle swarms. Struct Multidiscip Optim 2003;25:261–9.

[31] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;87(5–6):267–83.

[32] Kaveh A, Talatahari S. Optimal design of skeletal structures via the charged system search algorithm. Struct Multidiscip Optim 2010;41(6):893–911.

[33] Erbatur F, Hasançebi O, Tütüncü I, Kiliç H. Optimal design of planar and space structures with genetic algorithms. Comput Struct 2000;75:209–24.

[34] Perez RE, Behdinan K. Particle swarm approach for structural design optimization. Comput Struct 2007;85:1579–88.

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

[36] Togan V, Daloglu AT. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. Comput Struct 2008;86:1204–18.

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

[38] Krohling RA, Mendel E, Campos A. Swarm algorithms with chaotic jumps for optimization of multimodal functions. Eng Optim 2011;43(11):1243–61.

[39] Hilborn RC. Chaos and nonlinear dynamics. USA: Oxford University Press; 2000.