Computers and Structures 87 (2009) 267–283

Contents lists available at ScienceDirect

Computers and Structures

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

Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures

A. Kaveh [a,*], S. Talatahari [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 Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article history:
Received 18 August 2008
Accepted 6 January 2009
Available online 3 February 2009

Keywords:
Ant colony optimization
Harmony search
Particle swarm optimization
Passive congregation
Truss structures design
Size optimization

ABSTRACT

A heuristic particle swarm ant colony optimization (HPSACO) is presented for optimum design of trusses. The algorithm is based on the particle swarm optimizer with passive congregation (PSOPC), ant colony optimization and harmony search scheme. HPSACO applies PSOPC for global optimization and the ant colony approach is used to update positions of particles to attain the feasible solution space. HPSACO handles the problem-specific constraints using a fly-back mechanism, and harmony search scheme deals with variable constraints. Results demonstrate the efficiency and robustness of HPSACO, which performs better than the other PSO-based algorithms having higher converges rate than PSO and PSOPC.

1. Introduction

In the last decade, many new natural evolutionary algorithms have been developed for optimization of pin-connected structures, such as genetic algorithms (GAs) [1–5], particle swarm optimizer (PSO) [6,7], ant colony optimization (ACO) [8–10] and harmony search (HS) [11–13]. These methods have attracted a great deal of attention, because of their high potential for modeling engineering problems in environments which have been resistant to solution by classic techniques. They do not require gradient information and possess better global search abilities than the conventional optimization algorithms [14]. Having in common processes of natural evolution, these algorithms share many similarities: each maintains a population of solutions which are evolved through random alterations and selection. The differences between these procedures lie in the representation technique utilized to encode the candidates, the type of alterations used to create new solutions, and the mechanism employed for selecting new patterns.

Compared to other evolutionary algorithms based on heuristics including evolutionary algorithms (EAs), evolutionary programming (EP) and evolution strategies (ES) [15], the advantages of PSO consist of easy implementation and smaller number of parameters to be adjusted. However, it is known that the original PSO (or SPSO) had difficulties in controlling the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum) [16]. In order to improve this character of PSO, it is hybridized with other approaches such as ACO or HS. PSACO (a hybrid particle swarm optimizer and ant colony approach) which was initially introduced by Shelokar et al. [17] for the solution of the continuous unconstrained problems and recently utilized for truss structures [18], is applied to PSO as a global search technique and the idea of ant colony approach is incorporated as a local search for updating the positions of the particles by applied pheromone-guided mechanism. HPSO (a hybrid particle swarm optimizer and harmony search scheme) was proposed by Li et al. [7] for truss design employed the harmony memory (HM) operator for controlling the variable constraints.

The present paper hybridizes PSO, ACO and HS, and it is based on the principles of those two methods with some differences. We have applied PSOPC (a hybrid PSO with passive congregation [19]) instead of PSO to improve the performance of the new method. The relation of standard deviation in ACO stage is different with that of Ref. [17], and the inertia weight is changed in PSOPC stage. New terminating criterion is employed to increase the probability of obtaining an optimum solution in a smaller number of iterations. In the proposed method, similar to HPSO, HS is utilized for controlling the variable constraint. The resulted method has a good control on the exploration and exploitation compared to PSO and PSOPC. It increases the exploitation, and guides the exploration, and as a result, the convergence rate of the proposed algorithm is higher than other heuristic approaches.

* Corresponding author. Tel.: +98 21 44202710; fax: +98 21 77240398.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

0045-7949/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2009.01.003

There are some problem-specific constraints in truss optimization problems that must be handled. The penalty function method has been the most popular constraint-handling technique due to its simple principle and ease of implementation. The main difficulty of the penalty function method lies in that the appropriate values of penalty factors are problem-dependent and a large amount of effort is needed for fine-tuning of the penalty factors. Therefore, several techniques have been incorporated to handle the constraints. Compared to other constraint-handling techniques, fly-back mechanism is relatively simple and easy to implement into the PSO [7]. Therefore, this paper handles the problem-specific constraints by using this mechanism.

The present paper is organized as follows: In Section 2, we describe the PSO, ACO and HS. Statement of the optimization design problem is formulated in Section 3. In Section 4, the fly-back mechanism is described. In Section 5, the new method is presented. Various examples are studied in Section 6. The efficiency of HPSACO is investigated in Section 7. Conclusions are derived in Section 8.

2. Introduction to PSO, ACO and HS

In order to make the paper self-explanatory, before proposing HPSACO for truss design optimization, the characteristics of PSO, ACO and HS, are briefly explained in the following three sections:

2.1. Particle swarm optimization

Application of swarm intelligence in optimization was first suggested by Eberhart and Kennedy under the name of particle swarm optimization (PSO) [20]. The strength of PSO is underpinned by the fact that decentralized (without central supervision) biological creatures can often accomplish complex goals by cooperation. A

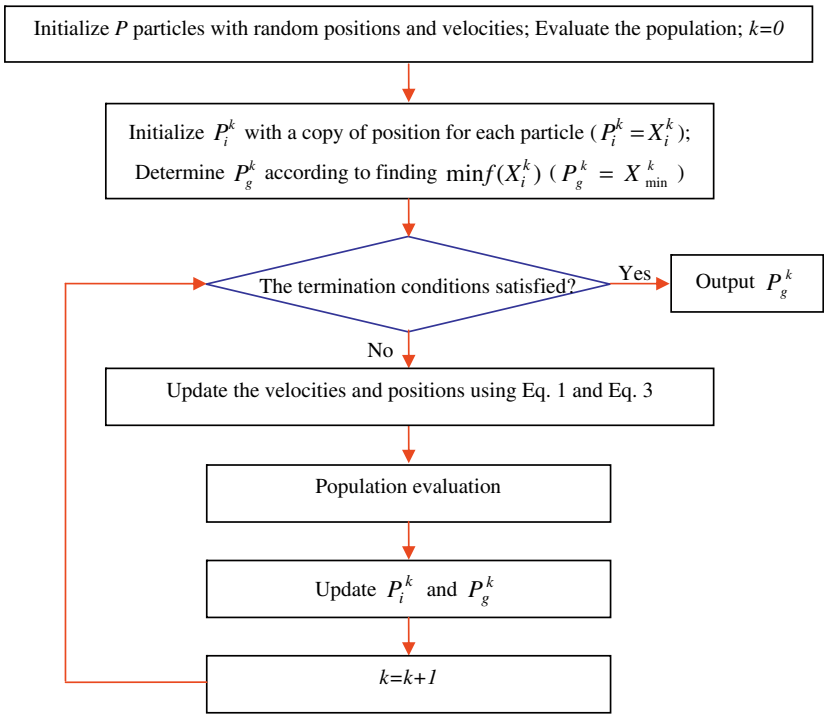

Fig. 1. The flow chart for PSOPC.

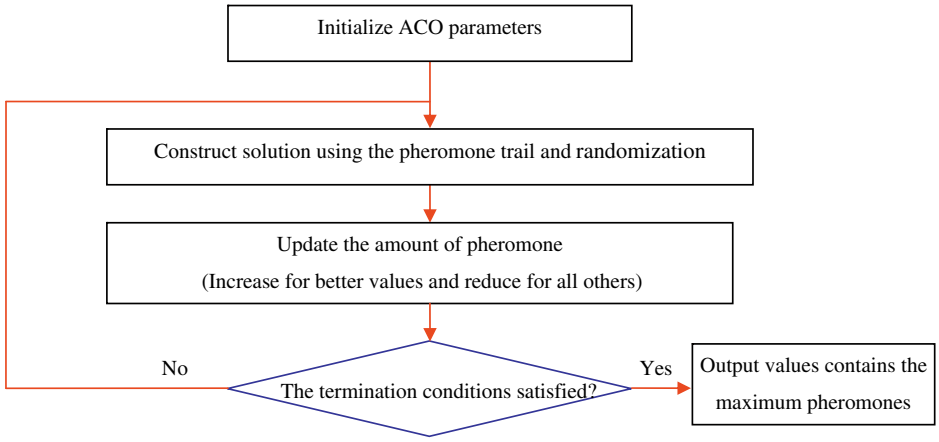

Fig. 2. The flow chart for ACO.

standard PSO algorithm is initialized with a population (swarm) of random potential solutions (particles). Each particle iteratively moves across the search space and is attracted to the position of the best fitness (evaluation of the objective function) historically achieved by the particle itself (local best) and by the best among the neighbors of the particle (global best). In essence, each particle continuously focuses and refocuses the effort of its search according to both local and global best. This behavior mimics the cultural adaptation of a biological agent in a swarm: it evaluates its own position based on certain fitness criteria, compares to others, and imitates the best in the entire swarm [21].

The update moves a particle by adding a change velocity V_i^{k+1} to the current position X_i^k as follows:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{1}$$

The velocity is a combination of three contributing factors: (1) previous velocity V_i^k, (2) movement in the direction of the local best P_i^k, and (3) movement in the direction of the global best P_g^k. The mathematical formulation can be expressed as

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \tag{2}$$

where ω is an inertia weight to control the influence of the previous velocity, r_1 and r_2 are two random numbers uniformly distributed in the range of $(0,1)$ and c_1 and c_2 are two acceleration constants [22].

Adding the passive congregation model to the PSO may increase its performance. He et al. [19] proposed a hybrid PSO with passive congregation (PSOPC). In this method, the velocity was defined as

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (R_i^k - X_i^k) \tag{3}$$

where R_i is a particle selected randomly from the swarm, c_3 is the passive congregation coefficient and r_3 is a uniform random sequence in the range $(0,1)$.

Several benchmark functions have been tested in Ref. [19]. The results show that the PSOPC has a better convergence rate and a higher accuracy than the PSO. Fig. 1 shows the flow chart for the PSOPC algorithm.

2.2. Ant colony optimization

Ant colony optimization (ACO) was first proposed by Dorigo [23] as a multi-agent approach to solve difficult combinatorial optimization problems. ACO was inspired by the observation of real ant colonies. Ants are social insects whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. An important behavior of ant colonies is their foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. Ants can smell pheromone and when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest-mates. When more paths are available from the nest to a food source, a colony of ants will be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back [24]. In fact, ACO simulates the optimization of ant foraging behavior. The ACO procedure is illustrated in Fig. 2.

2.3. Harmony search algorithm

Harmony search (HS) algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation [25]. The engineers seek for a global solution as determined by an objective function, just like the musicians seek to find musically pleasing harmony as determined by an aesthetic [11].

Fig. 3 shows the optimization procedure of the HS algorithm, which consists of the following steps [11]:

Step 1: Initialize the algorithm parameters and optimization operators. HS algorithm includes a number of optimization operators, such as the harmony memory (HM), the harmony memory size (HMS), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR). In the HS algorithm, the HM stores the feasible vectors, which are all in the feasible space. The harmony memory size determines the number of vectors to be stored.

Step 2: Improvise a new harmony from the HM. A new harmony vector is generated from the HM, based on memory considerations, pitch adjustments, and randomization. The HMCR

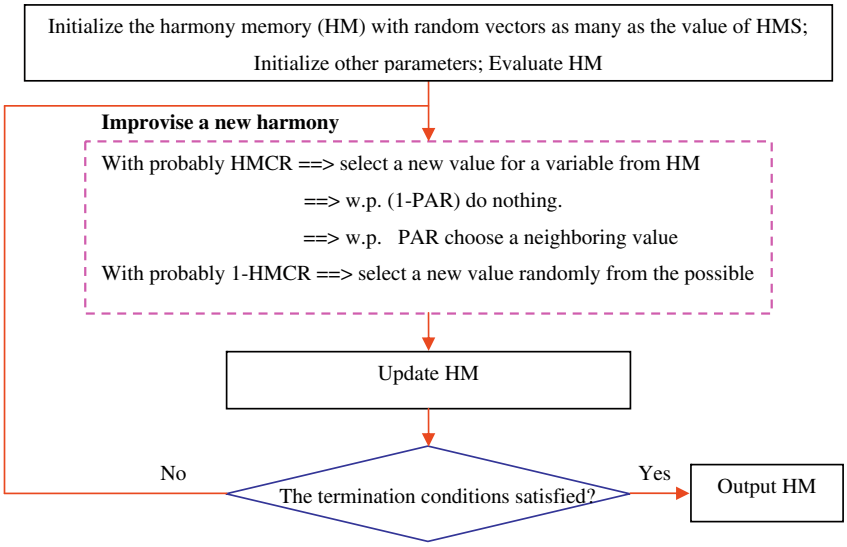

Fig. 3. The flow chart for HS.

varying between 0 and 1 sets the rate of choosing a value in the new vector from the historic values stored in the HM, and (1-HMCR) sets the rate of randomly choosing one value from the possible range of values. The pitch adjusting process is performed only after a value is chosen from the HM. The value (1-PAR) sets the rate of doing nothing. A PAR of 0.1 indicates that the algorithm will choose a neighboring value with 10% × HMCR probability.

Step 3: Update the HM. In Step 3, if a new harmony vector is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 4: Repeat Steps 2 and 3 until the terminating criterion is satisfied. The computations are terminated when the terminating criterion is satisfied. Otherwise, Steps 2 and 3 are repeated.

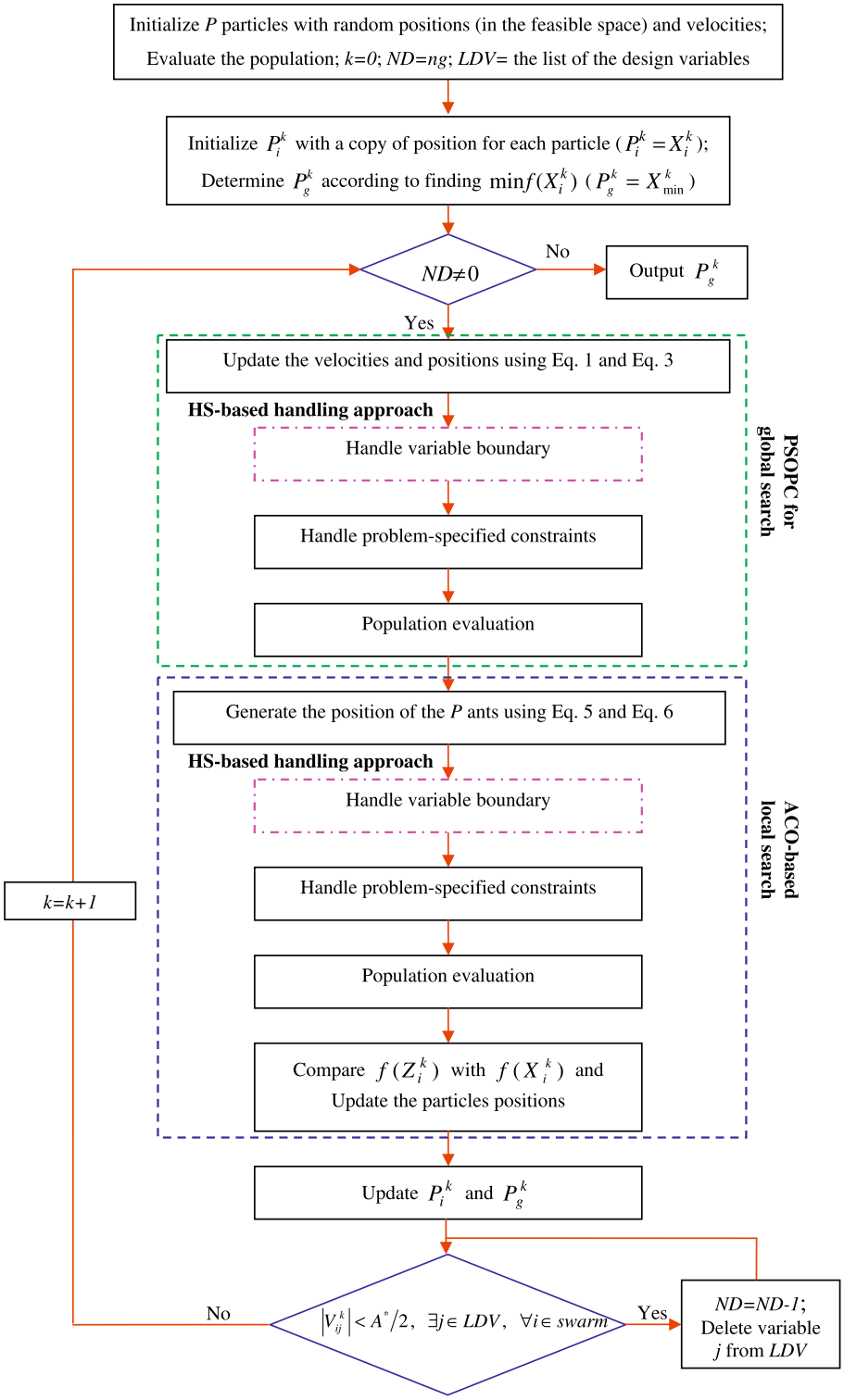

Fig. 4. The flow chart for HPSACO.

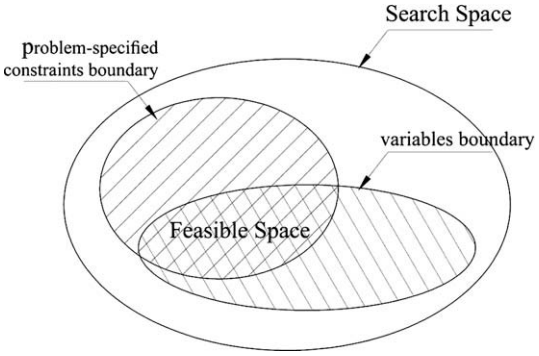

Fig. 5. Search space division.

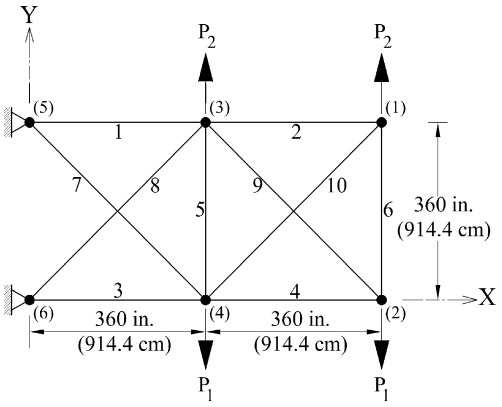

Fig. 6. A 10-bar planar truss.

3. Statement of the optimization design problem

Size optimization of truss structures involves arriving at optimum values for member cross-sectional areas A_i that minimize the structural weight W. This minimum design also has to satisfy inequality constraints that limit design variable sizes and structural responses [11]. Thus, the optimal design problem may be expressed as

$$\begin{aligned} \text{minimize} \quad & W(\{x\}) = \sum_{i=1}^{n} \gamma_i \cdot A_i \cdot L_i \\ \text{subject to} \quad & \delta_{\min} \le \delta_i \le \delta_{\max} \quad i = 1,2,\ldots,m \\ & \sigma_{\min} \le \sigma_i \le \sigma_{\max} \quad i = 1,2,\ldots,n \\ & \sigma_i^b \le \sigma_i \le 0 \quad i = 1,2,\ldots,nc \\ & A_{\min} \le A_i \le A_{\max} \quad i = 1,2,\ldots,ng \end{aligned} \tag{4}$$

Table 1
The pseudo-code for the HPSACO.

Set $k = 0$; ND = the number of design variables (ng);
Set LDV = the list of design variables;
Randomly initialize positions and velocities of all particles (from the range of $[A_{min}, A_{max}]$)
FOR(each particle i in the initial population)
  WHILE(the constraints are violated)
    Randomly re-generate the current particle X_i
  END WHILE
  Generate local best: Set $P_i^k = X_i^k$
  Generate global best: Find min$f(X_i^k)$, P_g^k is set to the position of $X_{\min}^k$
END FOR
WHILE($ND \neq 0$)
FOR(each particle (ant) i in the swarm(colony))
Generate the velocity and update the position of the current particle (vector) X_i^k
Variable boundary handling: Check whether each component of the current vector violates its
    corresponding boundary or not. If it does, select the corresponding component of the
    vector from P_i^k based on memory considerations, pitch adjustments, and randomization.
Problem-specified constraints handling: Check whether the current particle violates the problem
    constraints or not. If it does, reset it to the previous position X_i^{k-1}
Calculate the fitness value $f(X_i^k)$ of the current particle
Generate the position of the current ant $Z_i^k = N(P_g^k, \sigma)$
Variable boundary handling: Check whether each component of the current vector violates its
    corresponding boundary or not. If it does, select the corresponding component of the
    vector from P_i^k based on memory considerations, pitch adjustments, and randomization.
Problem-specified constraints handling: Check whether the current ant violates the problem
    constraints or not. If it does, reset it to the current particle X_i^k
Calculate the fitness value $f(Z_i^k)$ of the current ant
Update current particle position: Compare the fitness value of current ant with
    current particle. If the $f(Z_i^k)$ is better than the fitness value of $f(X_i^k)$,
    set $f(X_i^k) = f(Z_i^k)$ and $X_i^k = Z_i^k$
Update local best: Compare the fitness value of $f(P_i^k)$ with $f(X_i^k)$.
    If the $f(X_i^k)$ is better than the fitness value of $f(P_i^k)$,
    set P_i^k to the current position X_i^k
END FOR
Update global best: Find the global best position in the swarm. If the $f(X_i^k)$ is
    better than the fitness value of $f(P_g^k)$, P_g^k is set to the position
    of the current particle X_i^k
FOR(each $j \in LDV$)
IF($|V_{ij}^k| < A^*/2, \forall i \in swarm$)
    Set *ND=ND-1* and Delete variable j from LDV
END IF
END FOR
Set *k =k+1*
END WHILE

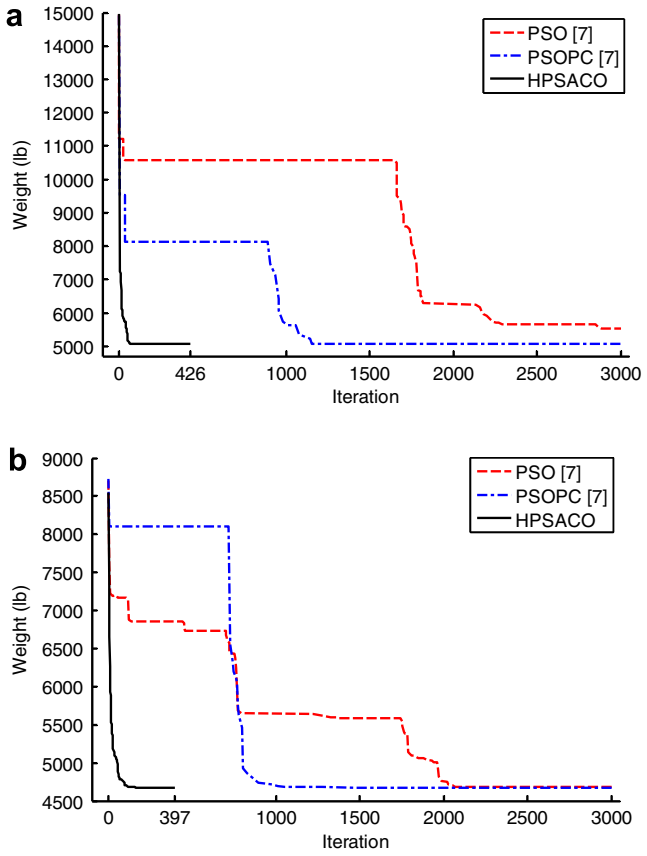

Fig. 7. Comparison of the convergence rates between the three algorithms for the 10-bar planar truss structure (a) Case 1 and (b) Case 2.

where $W(\{x\})$ is the weight of the structure, n is the number of members making up the structure, m is the number of nodes, nc is the number of compression elements, ng is the number of groups (number of design variables), γ_i is the material density of member i, L_i is the length of member i, A_i is the cross-sectional area of member i chosen between $A_{\min}$ and $A_{\max}$, min is the lower bound and max is the upper bound, σ_i and δ_i are the stress and nodal deflection, respectively and σ_i^b is the allowable buckling stress in member i when it is in compression.

4. Fly-back mechanism

Fly-back mechanism has been introduced by He et al. [26]. For most of the structural optimization problems, the global minimum locates on or close to the boundary of a feasible design space. The particles are initialized in the feasible region. When the particles fly in the feasible space to search the solution, if any one of them flies into the infeasible region, it will be forced to fly-back to the previous position to guarantee a feasible solution. The particle which flies back to the previous position may be closer to the boundary at the next iteration. This makes the particles to fly to the global minimum in a great probability. Although some experimental results have shown that it can find a better solution with fewer number of iterations than the other techniques [26], the fly-back mechanism has the difficulty of finding the first valid solutions for the swarm. However, if the first selections are limited to a neighborhood of the maximum value of permitted cross-sectional areas, one can expect, after a few iterations, a feasible swarm to be obtained. This neighborhood can be defined as:

$$\left[A_{\max} - \frac{A_{\max} - A_{\min}}{4}, A_{\max}\right]$$

Table 2
Optimal design comparison for the 10-bar planner truss (Case 1).

	Element group	Optimal cross-sectional areas (in.^{2})						
		Camp et al. [4]	Lee and Geem [11]	Li et al. [7]			HPSACO	
		GA	HS	PSO	PSOPC	HPSO	in.^{2}	cm^2
1	A_1	28.92	30.15	33.469	30.569	30.704	*30.307*	*(195.53)*
2	A_2	0.10	0.102	0.110	0.100	0.100	*0.100*	*(0.6452)*
3	A_3	24.07	22.71	23.177	22.974	23.167	*23.434*	*(151.19)*
4	A_4	13.96	15.27	15.475	15.148	15.183	*15.505*	*(100.03)*
5	A_5	0.10	0.102	3.649	0.100	0.100	*0.100*	*(0.6452)*
6	A_6	0.56	0.544	0.116	0.547	0.551	*0.5241*	*(3.3813)*
7	A_7	7.69	7.541	8.328	7.493	7.460	*7.4365*	*(47.977)*
8	A_8	21.95	21.56	23.340	21.159	20.978	*21.079*	*(135.99)*
9	A_9	22.09	21.45	23.014	21.556	21.508	*21.229*	*(136.96)*
10	A_{10}	0.10	0.100	0.190	0.100	0.100	*0.100*	*(0.6452)*
	Weight(lb)	5076.31	5057.88	5529.50	5061.00	5060.92	*5056.56*	*22502 N*

Table 3
Optimal design comparison for the 10-bar planner truss (Case 2).

	Element group	Optimal cross-sectional areas (in.^{2})					
		Lee and Geem [11]	Li et al. [7]			HPSACO	
		HS	PSO	PSOPC	HPSO	in.^{2}	cm^2
1	A_1	23.25	22.935	23.473	23.353	*23.194*	*(149.64)*
2	A_2	0.102	0.113	0.101	0.100	*0.100*	*(0.6452)*
3	A_3	25.73	25.355	25.287	25.502	*24.585*	*(158.61)*
4	A_4	14.51	14.373	14.413	14.250	*14.221*	*(91.748)*
5	A_5	0.100	0.100	0.100	0.100	*0.100*	*(0.6452)*
6	A_6	1.977	1.990	1.969	1.972	*1.969*	*(12.703)*
7	A_7	12.21	12.346	12.362	12.363	*12.489*	*(80.574)*
8	A_8	12.61	12.923	12.694	12.894	*12.925*	*(83.387)*
9	A_9	20.36	20.678	20.323	20.356	*20.952*	*(135.17)*
10	A_{10}	0.100	0.100	0.103	0.101	*0.101*	*(0.6516)*
	Weight(lb)	4668.81	4679.47	4677.70	4677.29	*4675.78*	*20807 N*

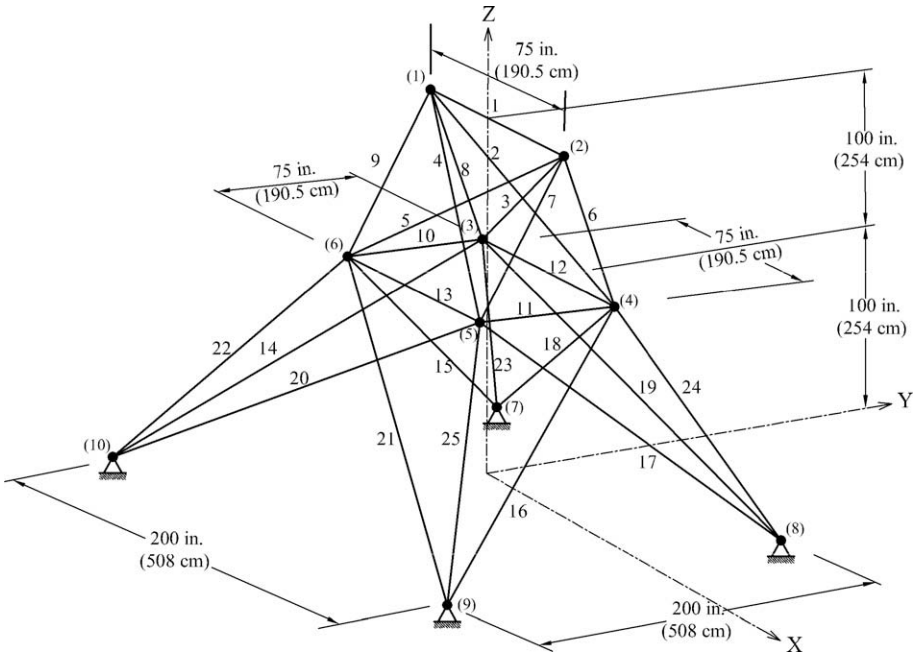

Fig. 8. A 25-bar spatial truss.

Table 4
Element information for the 25-bar spatial truss.

Element group number							
1	2	3	4	5	6	7	8
1:(1,2)	2:(1,4)	6:(2,4)	10:(6,3)	12:(3,4)	14:(3,10)	18:(4,7)	22:(10,6)
	3:(2,3)	7:(2,5)	11:(5,4)	13:(6,5)	15:(6,7)	19:(3,8)	23:(3,7)
	4:(1,5)	8:(1,3)			16:(4,9)	20:(5,10)	24:(4,8)
	5:(2,6)	9:(1,6)			17:(5,8)	21:(6,9)	25:(5,9)

Table 5
Loading conditions for the 25-bar spatial truss.

Node	Case 1			Case 2		
	P_X	P_Y kips (kN)	P_Z kips (kN)	P_X kips (kN)	P_Y kips (kN)	P_Z 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 6
Member stress limitation for the 25-bar spatial truss.

	Element group	Compressive stress limitations ksi (MPa)	Tensile stress limitations ksi (MPa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	$A_2 \sim A_5$	11.590 (79.913)	40.0 (275.80)
3	$A_6 \sim A_9$	17.305 (119.31)	40.0 (275.80)
4	$A_{10} \sim A_{11}$	35.092 (241.96)	40.0 (275.80)
5	$A_{12} \sim A_{13}$	35.092 (241.96)	40.0 (275.80)
6	$A_{14} \sim A_{17}$	6.759 (46.603)	40.0 (275.80)
7	$A_{18} \sim A_{21}$	6.959 (47.982)	40.0 (275.80)
8	$A_{22} \sim A_{25}$	11.082 (76.410)	40.0 (275.80)

5. A heuristic particle swarm ant colony optimization for truss design

The framework of heuristic particle swarm ant colony optimization (HPSACO) algorithm is illustrated in Fig. 4. HPSACO algorithm applies PSOPC for global optimization, while ACO works as a local search, wherein, ants apply pheromone-guided mechanism to refine the positions found by particles in the PSOPC stage. In HPSACO, a simple pheromone-guided mechanism of ACO is proposed and employed for the local search. The proposed ACO algorithm handles P ants equal to the number of particles in PSOPC [17].

In ACO stage, each ant generates a solution around P_g^k which can be expressed as

$$Z_i^k = N(P_g^k, \sigma) \tag{5}$$

In Eq. (5), $N(P_g^k, \sigma)$ denotes a random number normally distributed with mean P_g^k and variance σ, where

$$\sigma = (A_{\max} - A_{\min}) \times \eta \tag{6}$$

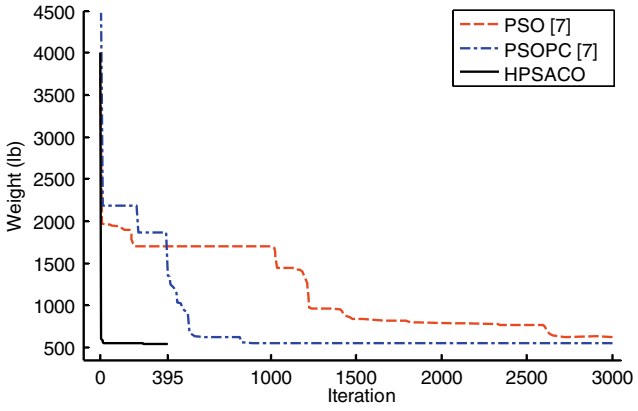

Fig. 9. Convergence rate comparison between the three algorithms for the 25-bar spatial truss structure.

Table 7
Optimal design comparison for the 25-bar spatial truss.

	Element group	Optimal cross-sectional areas (in.^{2})						
		Camp et al. [9]	Lee and Geem [11]	Li et al. [7]			HPSACO	
		ACO	HS	PSO	PSOPC	HPSO	in.^{2}	cm^2
1	A_1	0.010	0.047	9.863	0.010	0.010	*0.010*	*(0.0645)*
2	$A_2 \sim A_5$	2.000	2.022	1.798	1.979	1.970	*2.054*	*(13.252)*
3	$A_6 \sim A_9$	2.966	2.950	3.654	3.011	3.016	*3.008*	*(19.406)*
4	$A_{10} \sim A_{11}$	0.010	0.010	0.100	0.100	0.010	*0.010*	*(0.0645)*
5	$A_{12} \sim A_{13}$	0.012	0.014	0.100	0.100	0.010	*0.010*	*(0.0645)*
6	$A_{14} \sim A_{17}$	0.689	0.688	0.596	0.657	0.694	*0.679*	*(4.3806)*
7	$A_{18} \sim A_{21}$	1.679	1.657	1.659	1.678	1.681	*1.611*	*(10.394)*
8	$A_{22} \sim A_{25}$	2.668	2.663	2.612	2.693	2.643	*2.678*	*(17.277)*
	Weight(lb)	545.53	544.38	627.08	545.27	545.19	*544.99*	*2425.3 N*

η is used to control the step size. The normal distribution with mean P_g^k can be considered as a continuos pheromone which has the maximum in P_g^k and decreases with going away from it. In ACO algorithms, the probability of selecting a path with more pheromone is greater than other paths. Similarly, in the normal distribution, the probability of selecting a solution in the neighborhood of P_g^k is

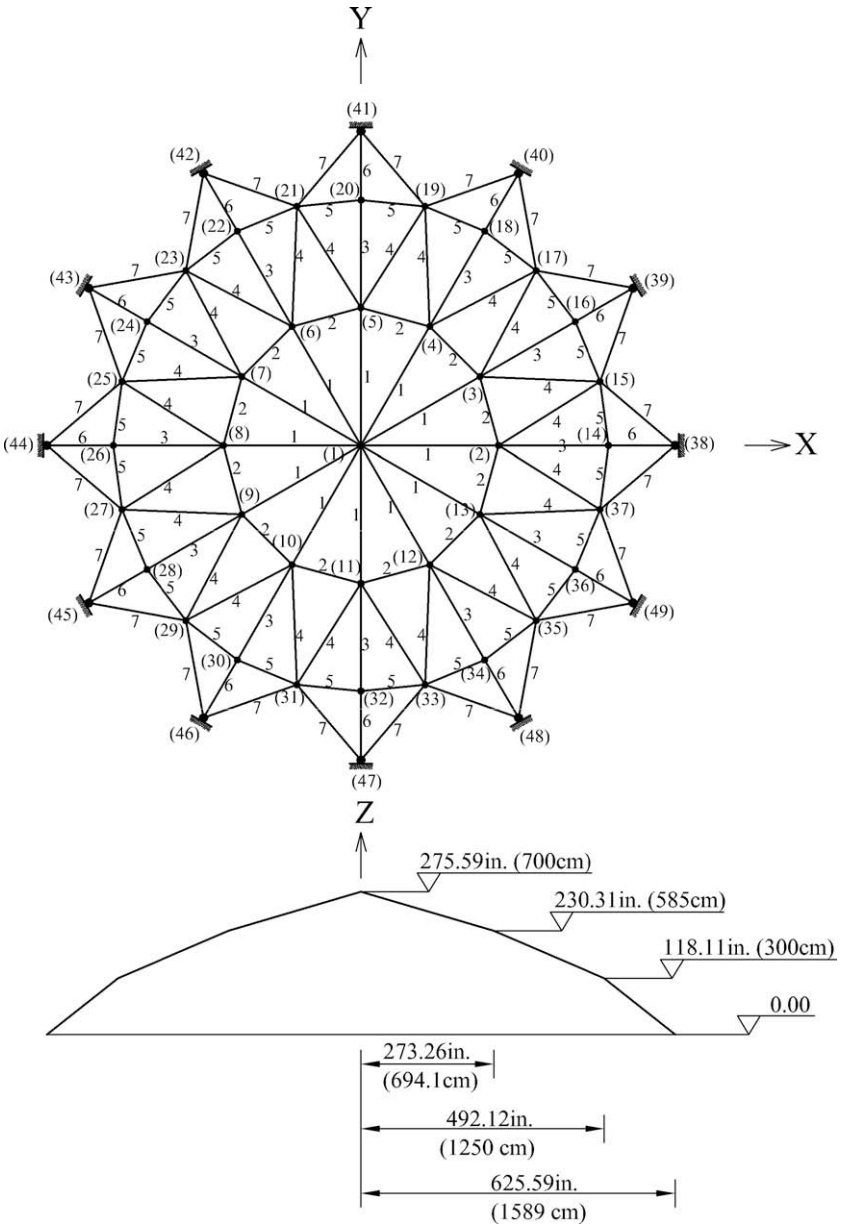

Fig. 10. A 120-bar dome truss.

Table 8
Optimal design comparison for the 120-bar dome truss (four Cases) Optimal cross-sectional areas (in^2).

Element group	Optimal cross-sectional areas ($in.^2$)									
	Case 1					Case 2				
	Lee and Geem [11]			HPSACO		Lee and Geem [11]			HPSACO	
	HS	PSO	PSOPC	$in.^2$	cm^2	HS	PSO	PSOPC	$in.^2$	cm^2
1	3.295	3.147	3.235	*3.311*	*(21.36)*	3.296	15.978	3.083	*3.779*	*(24.38)*
2	3.396	6.376	3.370	*3.438*	*(22.18)*	2.789	9.599	3.639	*3.377*	*(21.79)*
3	3.874	5.957	4.116	*4.147*	*(26.76)*	3.872	7.467	4.095	*4.125*	*(26.61)*
4	2.571	4.806	2.784	*2.831*	*(18.26)*	2.570	2.790	2.765	*2.734*	*(17.64)*
5	1.150	0.775	0.777	*0.775*	*(5.00)*	1.149	4.324	1.776	*1.609*	*(10.38)*
6	3.331	13.798	3.343	*3.474*	*(22.41)*	3.331	3.294	3.779	*3.533*	*(22.79)*
7	2.784	2.452	2.454	*2.551*	*(16.46)*	2.781	2.479	2.438	*2.539*	*(16.38)*
Weight (lb)	19707.77	32432.9	19618.7	*19491.3*	*86,736 N*	19893.34	41052.7	20681.7	*20078.0*	*89,347 N*
	Case 3					Case 4				
	Keleşoğlu and Ülker [27]	PSO	PSOPC	HPSACO		PSO	PSOPC	HPSACO		
				$in.^2$	cm^2			$in.^2$	cm^2	
1	5.606	1.773	2.098	*2.034*	*(13.12)*	12.802	3.040	*3.095*	*(19.96)*	
2	7.750	17.635	16.444	*15.151*	*(97.75)*	11.765	13.149	*14.405*	*(92.93)*	
3	4.311	7.406	5.613	*5.901*	*(38.07)*	5.654	5.646	*5.020*	*(32.39)*	
4	5.424	2.153	2.312	*2.254*	*(14.54)*	6.333	3.143	*3.352*	*(21.63)*	
5	4.402	15.232	8.793	*9.369*	*(60.44)*	6.963	8.759	*8.631*	*(55.68)*	
6	6.223	19.544	3.629	*3.744*	*(24.15)*	6.492	3.758	*3.432*	*(22.14)*	
7	5.405	0.800	1.954	*2.104*	*(13.57)*	4.988	2.502	*2.499*	*(16.12)*	
Weight (lb)	38237.83	46893.5	31776.2	*31670.0*	*140,931 N*	51986.2	33481.2	*33248.9*	*147,957 N*	

greater than the others. This principle is used in the HPSACO as a helping factor to guide the exploration and to increase the controlling in exploitation.

In the present method, objective function value $f(Z_i^k)$ is computed and the current position of ant i, Z_i^k, is replaced with the current position of particle i in the swarm, X_i^k, if $f(X_i^k) > f(Z_i^k)$ and the current ant is in the feasible space.

As illustrated in Fig. 5, a particle in the search space may violate either the problem-specific constraints or the limits of the variables. Since the fly-back mechanism is used to handle the problem-specific constraints, the particle will be forced to fly-back to its previous position no matter if it violates the problem-specific constraints. If it flies out of the variable boundaries, the solution cannot be used even if the problem-specific constraints are

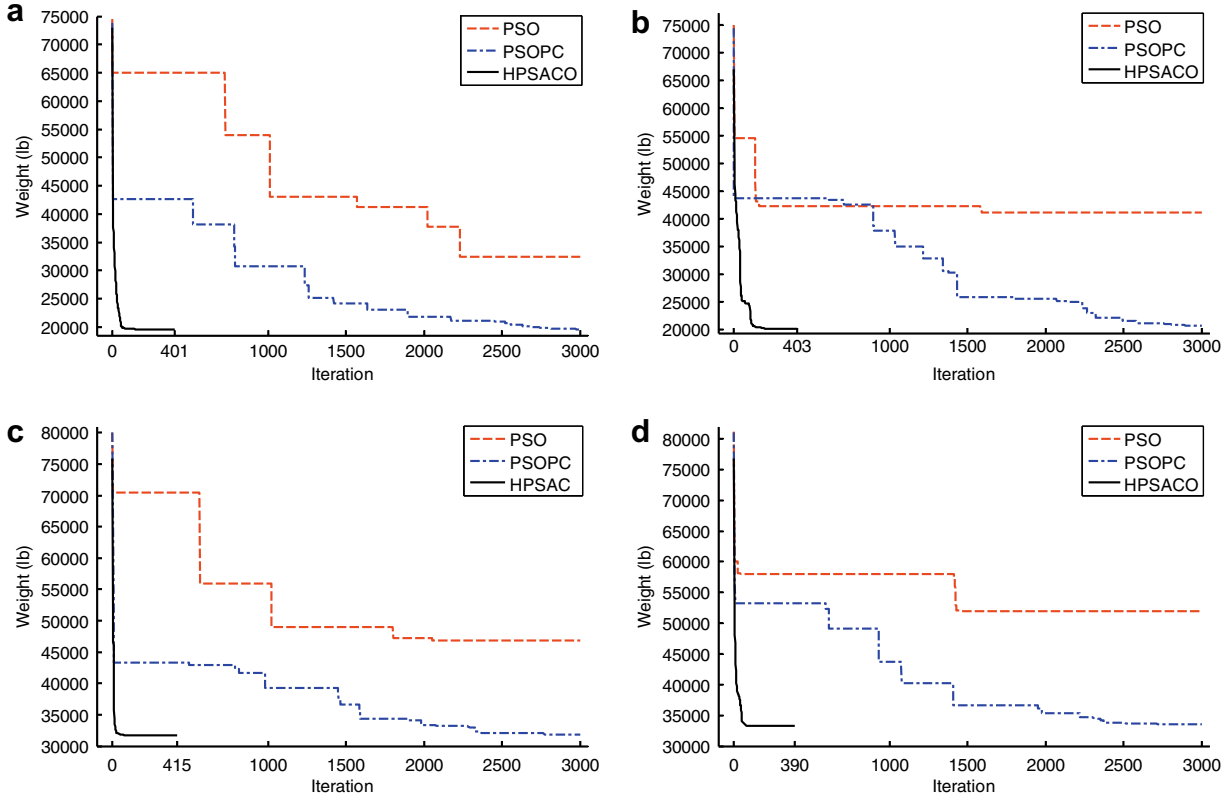

Fig. 11. Comparison of the convergence rates between the three algorithms for the 120-bar dome truss structure (a) Case 1, (b) Case 2, (c) Case 3 and (d) Case 4.

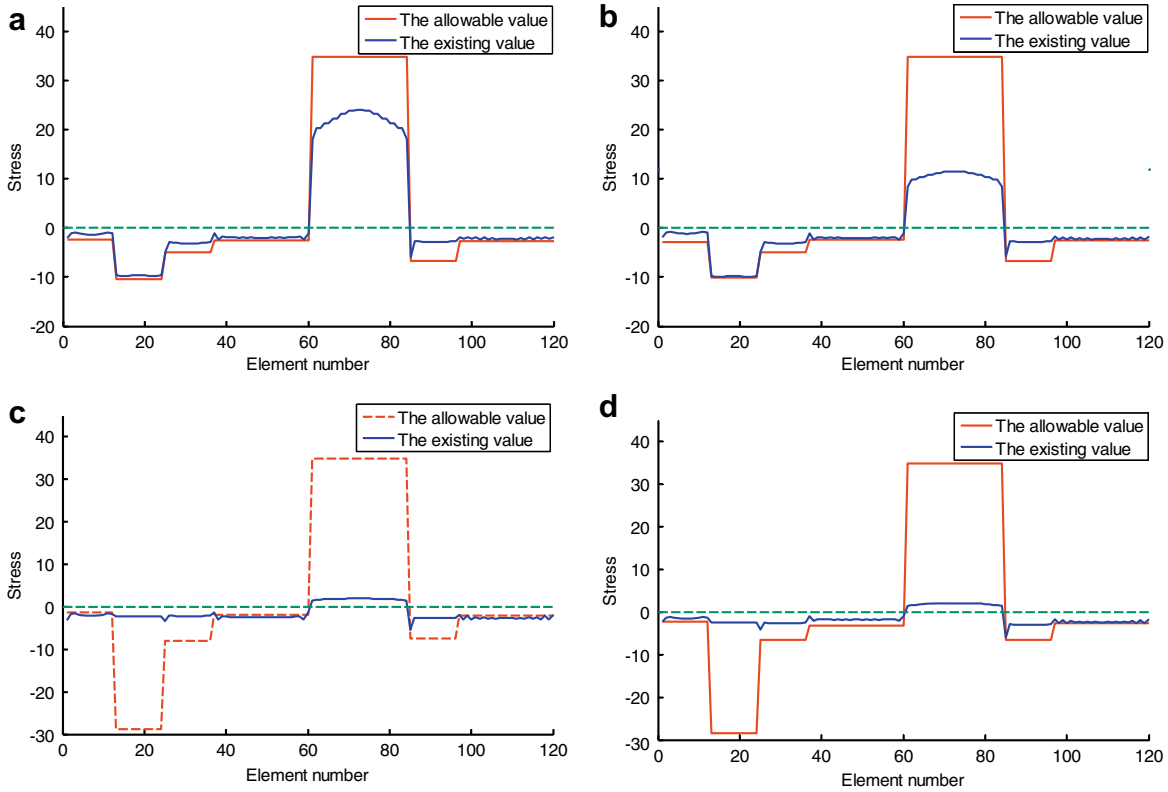

Fig. 12. Comparison of the allowable and existing stresses in the elements of the 120-bar dome truss structure using HPSACO.

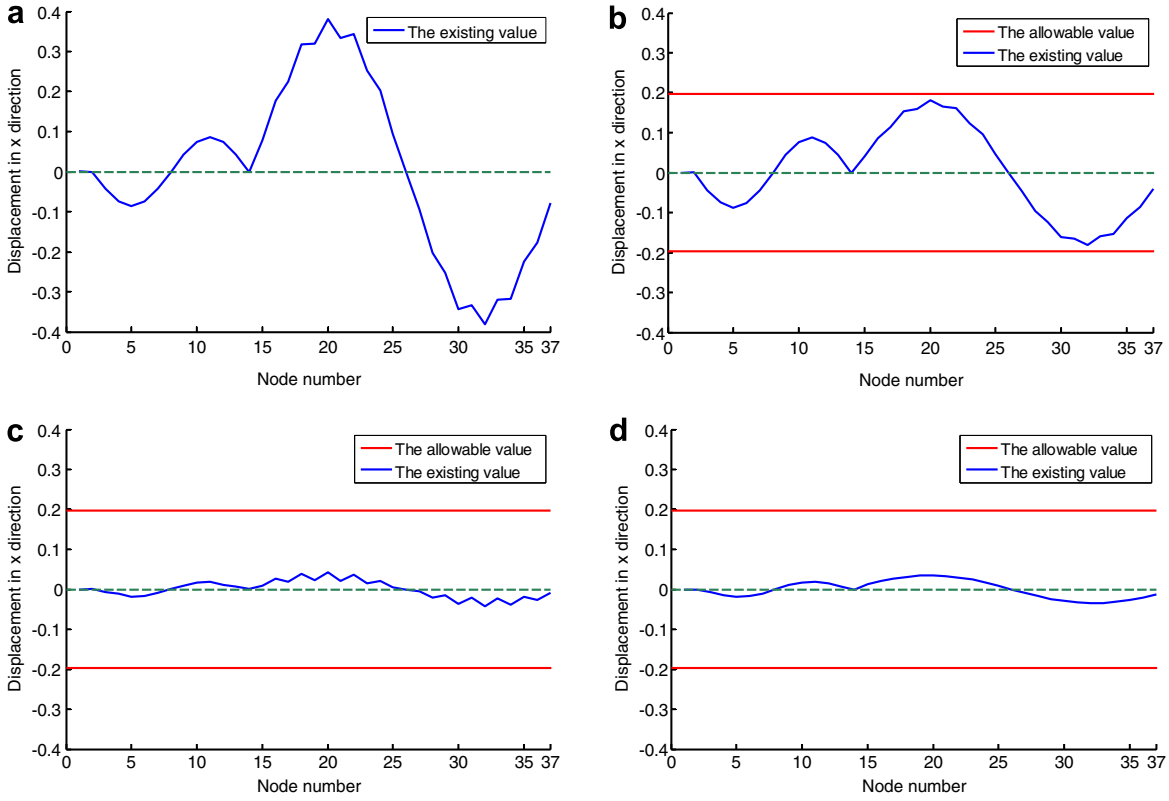

Fig. 13. Comparison of the allowable and existing displacements in the *x* direction for the nodes of the 120-bar dome truss structure using HPSACO.

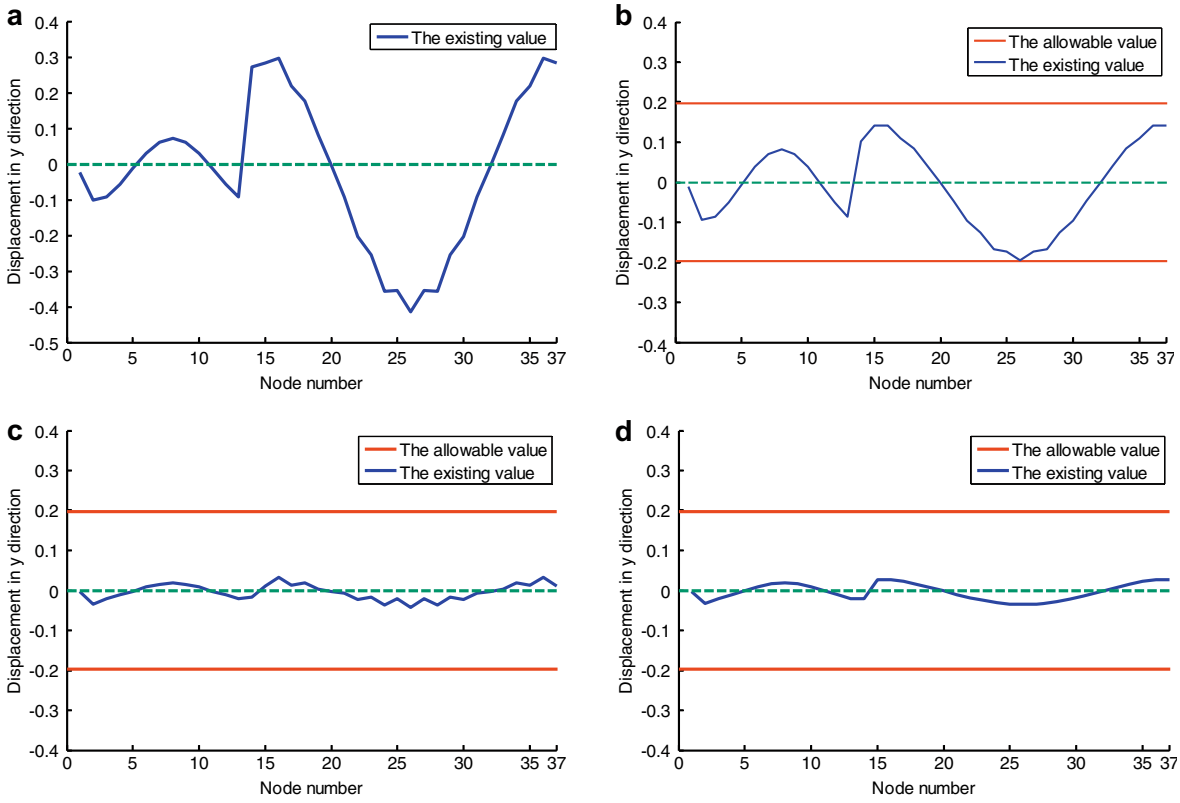

Fig. 14. Comparison of the allowable and existing displacements in the *y* direction for the nodes of the 120-bar dome truss structure using HPSACO.

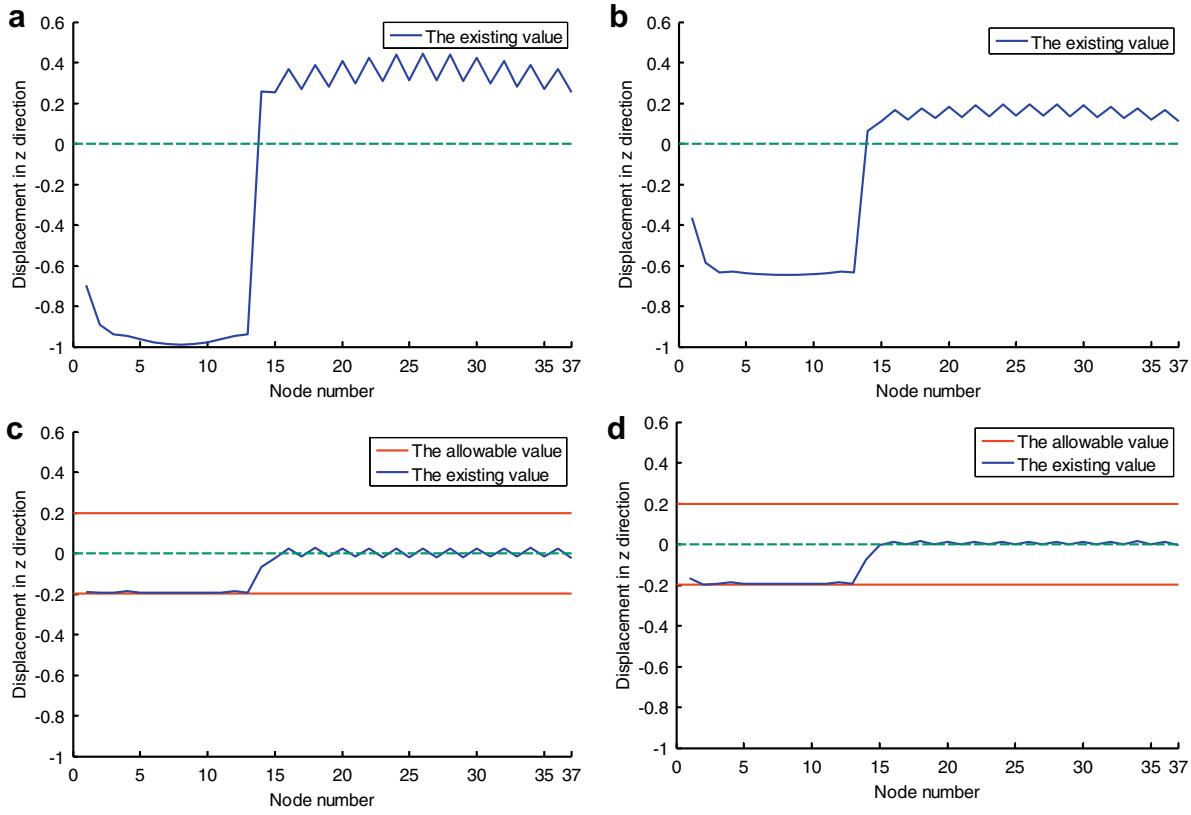

Fig. 15. Comparison of the allowable and existing displacements in the *z* direction for the nodes of the 120-bar dome truss structure using HPSACO.

satisfied. Therefore, the harmony search concept has been used to check whether they violate the variables' boundaries or not. Similarly, the PSO stores the feasible and good vectors (particles) in P_i^k, as does the harmony memory in the HS algorithm. Hence, any component of the vector (particle) violating the variables' boundaries can be generated randomly from P_i^k by such a technique, as explained in Step 2 for HS approach. In such a state, P_i^k works as the harmony memory.

In practical applications of truss optimization, a discrete solution is better than a continuous one. If $A_{\min}$, $A_{\max}$, A^* are the minimum cross-sectional area, the maximum cross-sectional area and the amount of increase in cross-sectional areas for a given truss, respectively, then discrete allowable series of allowable cross-sectional areas will be [10]

$$A_{\min}, A_{\min} + A^*, A_{\min} + 2A^*, \ldots, A_{\max} \tag{7}$$

A^* controls the exactitude of the solutions with a reverse relation; as A^* gets more, exactitude of the solutions decreases and the searching process must be stopped earlier and if the amount of A^* gets less, the searching process must be continued until an exact result is attained. Therefore, if in an iteration of search process, the absolute value of the component i in all of the particles' velocity vectors is less than $A^*/2$, continuing the search process can not change the amount of variable i; then the variable i reaches an optimum value and can be deleted from the virtual list of design variables. Therefore, the terminating criterion is defined as continuing the search process until all variables are deleted. The pseudocode for the HPSACO algorithm is listed in Table 1.

6. Numerical examples

In this section, common truss optimization examples as benchmark problems are optimized with the proposed method. The final results are compared to the solutions of other methods to demonstrate the efficiency of the present approach.

For the proposed algorithm, a population of 50 individuals is used for both particles and ants; the value of constants c_1 and c_2 are set 0.8 and the passive congregation coefficient c_3 is taken as 0.6. The value of inertia weight ($\omega(k)$) decreases linearly from 0.9 in first iteration to 0.4 in 3000th iteration for PSO and PSOPC [7]. Since the convergence rate of HPSACO is more than PSOPC, the solution is reached in less iterations. Thus, $\omega(k)$ for HPSACO is redefined as

$$\omega(k) = 0.9 - 0.0015 \times k \geq 0.4 \tag{8}$$

where k is the iteration number. In this way, the balance of $\omega(k)$ with fast rate of convergence in new method is saved and consequently, the performance improves. The amount of step size (η) in ACO stage is recommended 0.01 [18]. HMCR is set to 0.95 and PAR is taken as 0.10. A^* is considered as 0.001 for all the examples. The algorithms are coded in Matlab and structures are analyzed using the direct stiffness method.

6.1. Ten-bar planar truss

The 10-bar truss problem has become a common problem in the field of structural design to test and verify the efficiency of many different optimization methods. Fig. 6 shows the geometry and support conditions for this two dimensional, cantilevered truss with loading condition. The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3) and the modulus of elasticity is 10,000 ksi (68,950 MPa). The members are subjected to the stress limits of ±25 ksi (172.375 MPa) and all nodes in both vertical and horizontal directions are subjected to the displacement limits of ±2.0 in. (5.08 cm). There are 10 design variables in this example and a set of pseudo variables ranging from 0.1 to 35.0 in.^{2} (from 0.6452 cm^2 to 225.806 cm^2). Two cases are considered: Case 1, P_1 = 100 kips (444.8 kN) and P_2 = 0; and Case 2, P_1 = 150 kips (667.2 kN) and P_2 = 50 kips (222.4 kN).

For both load cases, the PSO and PSOPC algorithms achieve the best solutions after 3000 iterations (150,000 analyses) [7] and the HS algorithm reaches a solution after 20,000 analyses [11]. However, the HPSACO algorithm finds the best solution after about 426 and 397 iterations (10,650 and 9925 analyses), respectively for Case 1 and Case 2. Fig. 7 provides a comparison of the convergence rates of PSO, PSOPC and HPSACO. The best weights of PSACO are 5056.56 lb for Case 1 and 4675.78 lb for Case 2 while the best results of PSO and PSOPC are 5061.00 lb, 5529.50 lb for Case 1 and 4679.47 lb, 4677.70 lb for Case 2, respectively. Tables 2 and 3 compare the obtained results in this research with outcomes of other researches.

6.2. Twenty five-bar spatial truss

Fig. 8 shows the topology of a 25-bar spatial truss structure. In this example, designs are performed for a multiple loading case. The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3) and the modulus of elasticity is 10,000 ksi (68,950 MPa).

The structural members of this truss are arranged into eight groups, where all members in a group share the same material and cross-sectional properties. Table 4 defines each element group by member number (each member is defined by its start and end

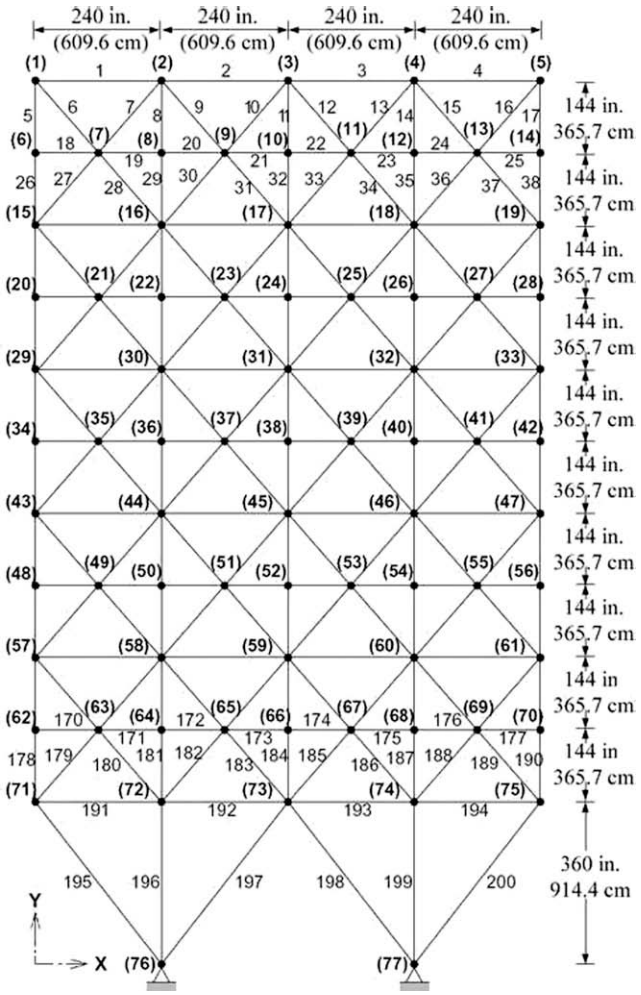

Fig. 16. A 200-bar planar truss.

node number). This spatial truss was subjected to the two loading conditions shown in Table 5. Maximum displacement limitations of ±0.35 in. (8.89 mm) were imposed on every node in every direction and the axial stress constraints vary for each group shown in Table 6. The range of cross-sectional areas varies from 0.01 to 3.4 in.^{2} (from 0.06452 cm^2 to 21.94 cm^2).

For this spatial truss structure, it takes about 1000 and 3000 iterations (50,000 and 150,000 analyses) for the PSOPC and the PSO algorithms to converge, respectively. The numbers of searches for the HS is 15,000 as provided in Ref. [11]. However the HPSACO algorithm takes 395 iterations (9875 analyses) to converge. Indeed, in this example, the PSO algorithm did not fully converge when the maximum number of iterations is reached [7]. In first iteration, the HPSACO algorithm achieves 633.87 lb while the PSO and PSOPC algorithms do not reach it until nearly 2,600 and 550 iterations, respectively. Fig. 9 compares the convergence rate of the three algorithms. Table 7 lists the optimal values of the eight size variables obtained by this research, and compares them with other results.

6.3. A 120-bar dome truss

A 120-bar dome truss, shown in Fig. 10, was first analyzed by Soh and Yang [5] to obtain the optimal sizing and configuration variables, i.e., the structural configuration optimization. In the example considered in this study similar to Lee and Geem [11] and Keleşoğlu and Ülker [27], only sizing variables to minimize the structural weight are considered. In addition, the allowable tensile and compressive stresses are used according to the AISC ASD (1989) [28] code, as follows:

$$\begin{cases} \sigma_i^+ = 0.6F_y & \text{for } \sigma_i \geq 0 \\ \sigma_i^- & \text{for } \sigma_i < 0 \end{cases} \tag{9}$$

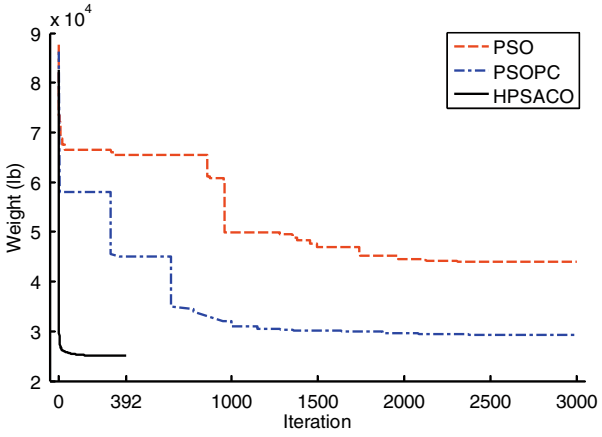

Fig. 17. Convergence rate comparison between the three algorithms for the 200-bar planar truss.

where σ_i^b is calculated according to the slenderness ratio

$$\sigma_i^- = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2C_c^2}\right)Fy\right] \Big/ \left(\frac{5}{3} + \frac{3\lambda_i}{8C_C} - \frac{\lambda_i^3}{8C_c^3}\right) & \text{for } \lambda_i < C_C \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geqslant C_C \end{cases} \tag{10}$$

where E is the modulus of elasticity, F_y is the yield stress of steel, C_c is the slenderness ratio (λ_i) dividing the elastic and inelastic buckling regions ($C_C = \sqrt{2\pi^2 E/F_y}$), λ_i is the slenderness ratio ($\lambda_i = kL_i/r_i$), k is the effective length factor, L_i is the member length and r_i is the radius of gyration.

The modulus of elasticity is 30,450 ksi (210,000 MPa) and the material density is 0.288 lb/in.^{3} (7971.810 kg/m^3). The yield stress

Table 9
Optimal design comparison for the 200-bar planar truss.

Group	Variables members (A_i, i = 1,...,200)	Optimal cross-sectional areas (in^2)				
		Lee and Geem [11]			HPSACO	
		HS	PSO	PSOPC	in.^{2}	cm^2
1	1,2,3,4	0.1253	0.8016	0.7590	*0.1033*	*(0.6666)*
2	5,8,11,14,17	1.0157	2.4028	0.9032	*0.9184*	*(5.9250)*
3	19,20,21,22,23,24	0.1069	4.3407	1.1000	*0.1202*	*(0.7756)*
4	18,25,56,63,94,101,132,139,170,177	0.1096	5.6972	0.9952	*0.1009*	*(0.6507)*
5	26,29,32,35,38	1.9369	3.9538	2.1350	*1.8664*	*(12.0415)*
6	6,7,9,10,12,13,15,16,27,28,30,31,33,34,36,37	0.2686	0.5950	0.4193	*0.2826*	*(1.8230)*
7	39,40,41,42	0.1042	5.6080	1.0041	*0.1000*	*(0.6452)*
8	43,46,49,52,55	2.9731	9.1953	2.8052	*2.9683*	*(19.1504)*
9	57,58,59,60,61,62	0.1309	4.5128	1.0344	*0.1000*	*(0.6452)*
10	64,67,70,73,76	4.1831	4.6012	3.7842	*3.9456*	*(25.4552)*
11	44,45,47,48,50,51,53,54,65,66,68,69,71,72,74,75	0.3967	0.5552	0.5269	*0.3742*	*(2.4141)*
12	77,78,79,80	0.4416	18.7510	0.4302	*0.4501*	*(2.90352)*
13	81,84,87,90,93	5.1873	5.9937	5.2683	*4.96029*	*(32.0018)*
14	95,96,97,98,99,100	0.1912	0.1000	0.9685	*1.0738*	*(6.9277)*
15	102,105,108,111,114	6.2410	8.1561	6.0473	*5.9785*	*(38.5708)*
16	82,83,85,86,88,89,91,92,103,104,106,107,109,110,112,113	0.6994	0.2712	0.7825	*0.78629*	*(5.0728)*
17	115,116,117,118	0.1158	11.1520	0.5920	*0.73743*	*(4.7576)*
18	119,122,125,128,131	7.7643	7.1263	8.1858	*7.3809*	*(47.6187)*
19	133,134,135,136,137,138	0.1000	4.4650	1.0362	*0.66740*	*(4.3058)*
20	140,143,146,149,152	8.8279	9.1643	9.2062	*8.3000*	*(53.5483)*
21	120,121,123,124,126,127,129,130,141,142,144,145,147,148,150,151	0.6986	2.7617	1.4774	*1.19672*	*(7.72074)*
22	153,154,155,156	1.5563	0.5541	1.8336	*1.0000*	*(6.4516)*
23	157,160,163,166,169	10.9806	16.1640	10.6110	*10.8262*	*(69.8463)*
24	171,172,173,174,175,176	0.1317	0.4974	0.9851	*0.1000*	*(0.6452)*
25	178,181,184,187,190	12.1492	16.2250	12.5090	*11.6976*	*(75.4682)*
26	158,159,161,162,164,165,167,168,179,180,182,183,185,186,188,189	1.6373	1.0042	1.9755	*1.3880*	*(8.9549)*
27	191,192,193,194	5.0032	3.6098	4.5149	*4.9523*	*(31.9506)*
28	195,197,198,200	9.3545	8.3684	9.8000	*8.8000*	*(56.7741)*
29	196,199	15.0919	15.5620	14.5310	*14.6645*	*(94.6099)*
	Weight (lb)	25447.1	44081.4	28537.8	*25156.5*	*111946.4N*

of steel is taken as 58.0 ksi (400 MPa). On the other hand, the radius of gyration (r_i) can be expressed in terms of cross-sectional areas, i.e., $r_i = aA_i^b$ [29]. Here, a and b are the constants depending on the types of sections adopted for the members such as pipes, angles, and tees. In this example, pipe sections (a = 0.4993 and b = 0.6777) were adopted for bars. All members of the dome are linked into seven groups, as shown in Fig. 10. The dome is considered to be subjected to vertical loading at all the unsupported joints. These were taken as −13.49 kips (60 kN) at node 1, −6.744 kips (30 kN) at nodes 2 through 14, and −2.248 kips (10 kN) at the rest of the nodes. The minimum cross-sectional area of all members is 0.775 in.^{2} (2 cm^2). In this example, four cases of constraints are considered: with stress constraints and no displacement constraints (Case 1), with stress constraints and displacement limitations of ±0.1969 in. (5 mm) imposed on all nodes in x- and y-directions (Case 2), no stress constraints but displacement limitations of ±0.1969 in. (5 mm) imposed on all nodes in z-directions (Case 3), and all constraints explained above (Case 4). For Case 1 and Case 2, the maximum cross-sectional area is 5.0 in.^{2} (32.26 cm^2) and for Case 3 and Case 4 is 20.0 in.^{2} (129.03 cm^2).

Table 8 gives the best solution vectors and the corresponding weights for all cases. Fig. 11 shows the convergence for all cases. In all cases, HPSACO needs nearly 10,000 analyses (400 iterations) to reach a solution which is less than 125,000 (2500 iterations) and 35,000 analyses for PSOPC and HS [11], respectively. Figs. 12–15 compare the allowable and existing stress and displacement constraint values of the HPSACO resulted for four cases. In Case 1, the stress constraints of some elements in the 4th and 7th groups are active. The maximum values of displacements in the x, y and z directions are 0.3817 in. (0.9695 cm), 0.4144 in. (1.0526 cm) and 0.988 in. (2.6855 cm), respectively. In Case 2, the stress constraints in the 2nd, 4th and 7th groups and the displacement of node 26 in y direction are active. The maximum value for displacement in the x direction is 0.1817 in. (0.4615 cm) which is near to the allowable value, and the maximum displacement in the z direction is 0.6448 in. (1.6378 cm). The displacement constraints in the x and y directions do not affect the results of Case 3 and Case 4. The active constraints for Case 3 are the displacements of the 1st to 13th nodes in z directions. If the stress constraints are considered for Case 3, the element stresses of 1st, 4th and 7th groups swerve from their limits. In Case 4, the stresses in the elements of the 7th group and the displacements of the 1st to 13th nodes in z directions affect the results.

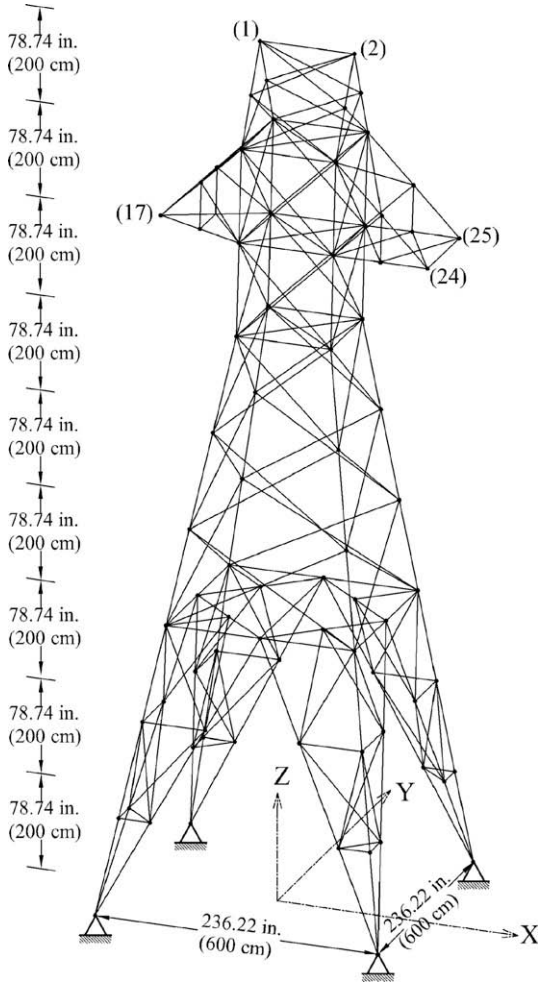

Fig. 18. A 244-bar transformation tower.

6.4. A 200-bar planar truss

The 200-bar plane truss, shown in Fig. 16, has been size optimized using HS by Lee and Geem [11]. 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 30,000 ksi (206,000 MPa), respectively. This truss is subjected to constraints only on stress limitations of ±10 ksi (68.95 MPa). There are three loading conditions: (1) 1.0 kip (4.45 kN) acting in the positive x-direction at nodes 1, 6, 15, 20, 29, 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 200 members of this truss are linked into twenty-nine groups, as shown in Table 9. The minimum cross-sectional area of all members is 0.1 in.^{2} (0.6452 cm^2) and the maximum cross-sectional area is 20 in.^{2} (129.03 cm^2).

The HPSACO algorithm founds the best solution vectors after 392 iterations (9800 analyses) while HS needs 48,000 analyses to reach the optimum solution. Table 9 compares the optimal results of HS [11] PSO, PSOPC and HPSACO and Fig. 17 compares the convergence rate between PSO, PSOPC and HPSO for this example.

6.5. A 244-bar transformation tower

Another space truss, a 244-bar transmission tower shown in Fig. 18, is examined as the final design problem to demonstrate the efficiency of the algorithm. Members of the transmission tower are initially collected into 26 groups as given by Saka [29] but in this study all members of the transmission tower are linked into 32 groups. The value of the modulus of elasticity is taken as 30,450 ksi (210,000 MPa) and the material density is 0.1 lb/in.^{3} (2767.990 kg/m^3). The allowable value of 20.30 ksi (140 MPa) is employed for tensile stresses and the formulation of buckling obeying AISC ASD (1989) [28] code is considered for compressive stresses (similar to the dome example). The displacement limitations of ±1.77 in. (4.5 cm) are imposed on nodes 1 and 2, and limitations of ±1.18 in. (3.0 cm) on nodes 17, 24 and 25 in x-direction.

Table 10
Loading conditions for the 244-bar transformation tower.

Node	Case 1			Case 2		
	P_X kips (kN)	P_Y kips (kN)	P_Z	P_X	P_Y kips (kN)	P_Z
1	−2.448 (10)	−6.744 (30)	0.0	0.0	−80.899 (360)	0.0
2	2.448 (10)	−6.744 (30)	0.0	0.0	−80.899 (360)	0.0
17	8.568 (35)	−20.224 (90)	0.0	0.0	−40.449 (180)	0.0
24	42.82 (175)	−10.112 (45)	0.0	0.0	−20.224 (90)	0.0
25	42.82 (175)	−10.112 (45)	0.0	0.0	−20.224 (90)	0.0

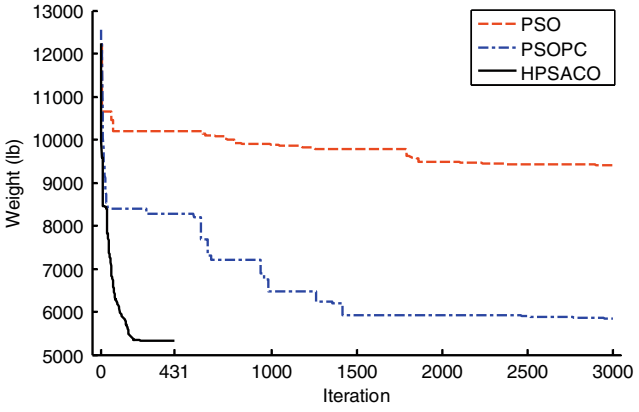

Fig. 19. Convergence rate comparison between the three algorithms for the 244-bar transformation tower.

These nodes are subjected to the displacement limits of ±0.59 in. (1.5 cm) in *y*-direction. The load cases considered is taken from the study by Saka [29] and are shown in Table 10.

The minimum cross-sectional area of all members is 0.775 in.^{2} (5.0 cm^2) and the maximum cross-sectional area is 20.0 in.^{2} (129.03 cm^2).

The HPSACO achieves a good solution after 431 iterations (10,775 analyses) and founds an optimum weight of 5324.2 lb while the PSOPC algorithm achieves 5847.9 lb after 3000 (150,000 analyses) iterations and the optimal weight reported by Saka [29] is 5614.4 lb. In this case, the PSO algorithm is not fully converged when the maximum number of iterations is reached. In 50th iteration, the HPSACO algorithm achieves 7361.6 lb while the PSOPC algorithm does not reach it until 627 iterations. Fig. 19 shows the convergence history for the minimum weight of 244-bar transformation tower.

7. Discussion on the efficiency of the HPSACO

Solution of a number of design examples shows that the performance of HPSACO is significantly better than other PSO-based algorithms. The major reasons for the improvements obtained by the present method can be summarized as follows:

(a) *Increasing the exploitation*: In truss optimization, usually there are some local optimums in the neighborhood of a desirable solution. Thus, the probability of finding a desirable optimum increases with additional searches around the local optimums. HPSACO does extra search (exploitation) around the local optimums, P_g^k, (by using Eq. 5) and therefore obtains the desirable solution with higher probability in a smaller number of iterations.

The difference between the best and the worst results of the 10-bar truss (Case 1) for PSOPC in 50 tests is 365.2 lb (7.21%), the average weight is 5173.45 lb, and the standard deviation is 81.17 lb (see Table 11). With adding the ACO principles to the PSOPC (PSACO [18]), these values are reduced to 3.2 lb (0.06%), 5058.23 lb, and 1.46 lb, respectively. In addition, although PSO is a week approach, applying ACO principles in PSO results in improvement of its performance. The average weight of PSO + ACO in 50 runs is 5079.19 lb, and the standard deviation is 4.76 lb, which are better than PSOPC'. Therefore, increasing the exploitation by applied pheromone-guided mechanism for updating the positions of the particles, not only can improve the results, but also reduces the standard deviation drastically.

(b) *Guiding the exploration*: To our knowledge, heuristic methods utilize two factors: the random search factor and the information collected from the search space during optimization process. In early iterations, the random search factor has more power than the collective information factor, but the increase in the number of iterations gradually abates the power of the random search factor and increases the power of the collective information factor. In HPSACO, ACO stage plays an auxiliary role in rapidly increasing the collective information factor; consequently, the convergence rate increases faster.

PSOPC requires 3000 iterations to reach a solution for 10-bar truss (Case 1). However, the number of required iterations to reach a solution for PSOPC + ACO (PSACO [18]) in 50 runs in average is 635.2 iterations. Also, PSO + ACO in average needs 567 iterations to reach the optimum solution, while PSO can not reach to an appropriate solution until the maximum number of iterations is achieved (3000 iterations).

Table 11
Investigation on the performance of various PSO-based algorithms for the 10-bar truss (Case 1) in 50 runs.

Algorithm	Minimum iterations	Maximum iterations	Average iterations	Best weight (lb)	Worst weight (lb)	Average weight (lb)	Standard deviation (lb)
PSOPC	3000	3000	3000	5061.00	5406.26	5173.45	81.17
PSOPC + HS (HPSO)	3000	3000	3000	5060.92	5103.63	5078.69	13.05
PSO + ACO	373	567	439.6	5065.23	5092.71	5079.19	4.76
PSO + ACO + HS	226	414	296.3	5065.61	5078.26	5070.86	2.87
PSOPC + ACO (PSACO) [18]	619	655	635.2	5057.36	5060.61	5058.23	1.46
PSOPC + ACO + HS (HPSACO)	405	436	420.3	5056.56	5061.12	5057.66	1.42

Table 12
Investigation on the performance of various PSO-based algorithms for the 200-bar truss in 50 runs.

Algorithm	Minimum iterations	Maximum iterations	Average iterations	Best weight (lb)	Worst weight (lb)	Average weight (lb)	Standard deviation (lb)
PSO	3000	3000	3000	44081.4	48941.8	54128.5	4920.8
PSOPC	3000	3000	3000	28537.8	30063.8	33746.8	3256.5
HPSACO	388	440	424.8	25156.5	25786.2	26421.6	830.5

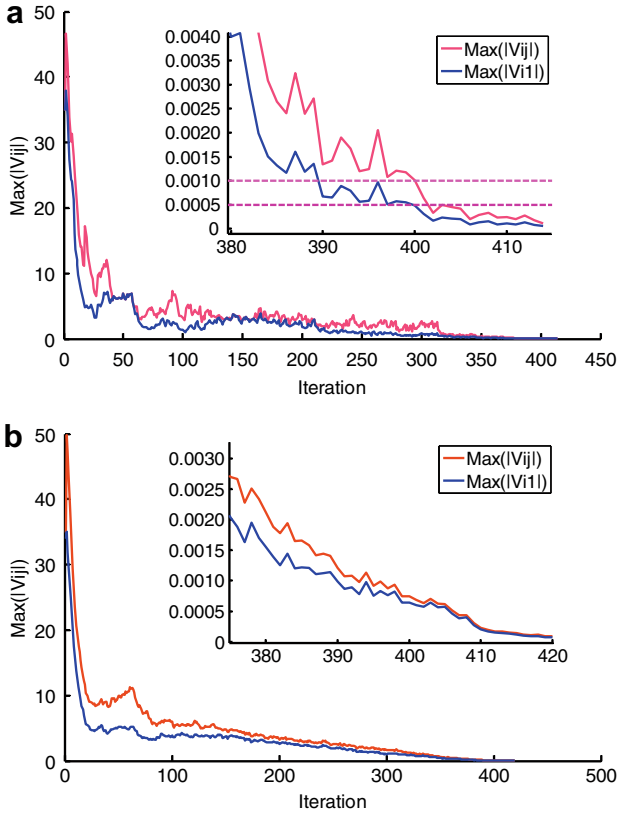

Fig. 20. The history of $\max(|V_{ij}^k|)$ and $\max(|V_{i1}^k|)$ in 50 tests for the 10-bar truss (Case 1).

Although minimizing the maximum value of the velocity can make fewer particles to violate the variable boundaries, it may also prevent the particles to cross the problem-specific constraints and can cause the reduction in exploration. Using the harmony search-based handling approach, this problem is dealt with.

In order to investigate the advantages of HS-based handling approach, the comparison of the performance of PSOPC with HPSO (PSOPC + HS), or PSO + ACO with PSO + ACO + HS, or PSOPC + ACO (PSACO) with PSOPC + ACO + HS (HPSACO) can be helpful. Table 11 summarizes the performances of all the above mentioned PSO-based approaches for the 10-bar truss (Case 1) in 50 runs for each algorithm. Although the results and standard deviations of HPSACO and PSACO do not differ much, however, the convergence rate of HPSACO is higher than PSACO. In average, HPSACO needs 420.3 iterations to reach a solution, while for PSACO [18] the average number of iterations to reach a solution is 635.2. Similarly, Table 12 compares the performance of PSO, PSOPC and HPSACO for the 200-bar truss.

(c) *Efficient terminating criterion*: In optimization problems, particularly in truss optimization, the number of iterations is highly important. The terminating criterion is a part of the search process which can be used to eliminate additional unnecessary iterations. This paper proposes a new terminating criterion to fulfill this goal.

Authors in Ref. [18] have shown that if $\max(|V_{ij}^k|)$ is less than A^*, the search process can be stopped. However, this terminating criterion causes some unnecessary searches in comparison to the present algorithm. Fig. 20 shows the average and a typical maximum absolute value of velocity for all design variables (i.e. $\max(|V_{ij}^k|)$) and for the first design variable (i.e. $\max(|V_{i1}^k|)$) in 50 tests for the 10-bar truss (Case 1) without considering the proposed terminating criterion. As shown in Fig. 20a, generally $\max(|V_{i1}^k|)$ is a decreasing function with a slight disorder. When $\max(|V_{i1}^k|)$ reaches to less than A^*, there is a probability (even slight) that $\max(|V_{i1}^k|)$ in the next iterations is more than A^*. Instead, if the upper bound of $\max(|V_{i1}^k|)$ is selected as $A^*/2$, it is very slight probability that $\max(|V_{i1}^k|)$ in the next iterations becomes more than A^* and it has a neglectable value and as a result, continuing the search process can not help to improve the results (without considering that it is an appropriate solution or not).

8. Concluding remarks

In this paper HPSACO is developed for optimal design of trusses. HPSACO is based on PSOPC, ACO and HS. In this method, ACO helps PSO process not only to efficiently perform the global exploration for rapidly attaining the feasible solution space but also effectively helps to reach optimal or near optimal solution. In order to make the particles remain in the feasible space, fly-back mechanism and HS are used. Fly-back mechanism handles the problem-specific constraints, and the HS deals with the variable constraint.

Since the rate of convergence for the proposed method is higher than the PSO and PSOPC, a new relation is defined for the inertia weight. A new terminating criterion is defined in the way that after decreasing the movements of particles, the search process stops. When the variation of a variable is less than a determined value, this criterion deletes it from the virtual list of variables. When this list is emptied, the search process stops. With these alterations, the number of iterations decreases. The comparisons based on several well-studied benchmark trusses demonstrate the effectiveness, efficiency and robustness of the proposed method.

Acknowledgement

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

References

[1] Kaveh A, Kalatjari V. Genetic algorithm for discrete sizing optimal design of trusses using the force method. Int J Numer Methods Eng 2002;55:55–72.
[2] Kaveh A, Kalatjari V. Size/geometry optimization of trusses by the force method and genetic algorithm. Z Angew Math Mech 2004;84(5):347–57.
[3] Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm, force method, and graph theory. Int J Numer Methods Eng 2003;58(5):771–91.
[4] Camp C, Pezeshk S, Cao G. Optimized design of two dimensional structures using a genetic algorithm. J Struct Eng ASCE 1998;124(5):551–9.
[5] Soh CK, Yang J. Fuzzy controlled genetic algorithm search for shape optimization. J Comput Civil Eng ASCE 1996;10(2):143–50.
[6] Li LJ, Ren FM, Liu F, Wu QH. An improved particle swarm optimization method and its application in civil engineering. In: Topping BHV, Montero G, Montenegro R, editors. Proceedings of the fifth international conference on engineering computational technology. Stirlingshire, United Kingdom: Civil-Comp Press; 2006 [Paper 42].
[7] Li LJ, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 2007;85:340–9.
[8] Kaveh A, Shojaee S. Optimal design of skeletal structures using ant colony optimization. Int J Numer Meth Eng 2007;5(70):563–81.
[9] Camp C, Bichon J. Design of space trusses using ant colony optimization. J Struct Eng ASCE 2004;130(5):741–51.
[10] Kaveh A, Farahmand Azar B, Talatahari S. Ant colony optimization for design of space trusses. Int J Space Struct 2008;23(3):167–81.
[11] Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. Comput Struct 2004;82:781–98.
[12] Saka MP. Optimum geometry design of geodesic domes using harmony search algorithm. Adv Struct Eng 2007;10(6):595–606.
[13] Saka MP. Optimum design of steel sway frames to BS5950 using harmony search algorithm. J Construct Steel Res, in press. Available online 3 April 2008.
[14] 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 2002;191(11–12). 1CA.245–87.

[15] Rechenberg I. Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart-Bad Cannstatt; 1973.
[16] Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. In: Proceeding of the evolutionary programming conference, San Diego, USA; 1998.
[17] Shelokar PS, Siarry P, Jayaraman VK, Kulkarni BD. Particle swarm and ant colony algorithms hybridized for improved continuous optimization. Appl Math Comput 2007;188:129–42.
[18] Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures. Asian J Civil Eng 2008;9(4):329–48.
[19] He S, Wu QH, Wen JY, Saunders JR, Paton RC. A particle swarm optimizer with passive congregation. Biosystem 2004;78:135–47.
[20] 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.
[21] Kennedy J, Eberhart RC, Shi Y. Swarm intelligence. San Francisco (CA): Morgan Kaufman Publishers; 2001.
[22] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proceedings of IEEE international conference on evolutionary computation; 1997.
[23] Dorigo M. Optimization, learning and natural algorithms (in Italian), PhD Thesis. Dipartimento di Elettronica, Politecnico di Milano, IT; 1992.
[24] Dorigo M, Caro DG, Gambardella LM. Ant algorithms for discrete optimization. Artif Life 1999;5(3):137–72.
[25] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76:60–8.
[26] He S, Prempain E, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems. Eng Optimiz 2004;36(5):585–605.
[27] Keleşoğlu O, Ülker M. Fuzzy optimization geometrical nonlinear space truss design. Turkish J Eng Environ Sci 2005;29:321–9.
[28] American Institute of Steel Construction (AISC). Manual of steel construction-allowable stress design. 9th ed. Chicago, IL; 1989.
[29] Saka MP. Optimum design of pin-jointed steel structures with practical applications. J Struct Eng ASCE 1990;116(10):2599–620.