Structural Engineering and Mechanics, *Vol. 54, No. 3 (2015) 453-474*
DOI: http://dx.doi.org/10.12989/sem.2015.54.3.453

Hybrid PSO and SSO algorithm for truss layout and size optimization considering dynamic constraints

A. Kaveh*, T. Bakhshpoori and E. Afshari

Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

(Received May 3, 2014, Revised November 5, 2014, Accepted November 11, 2014)

Abstract. A hybrid approach of Particle Swarm Optimization (PSO) and Swallow Swarm Optimization algorithm (SSO) namely Hybrid Particle Swallow Swarm Optimization algorithm (HPSSO), is presented as a new variant of PSO algorithm for the highly nonlinear dynamic truss shape and size optimization with multiple natural frequency constraints. Experimentally validation of HPSSO on four benchmark trusses results in high performance in comparison to PSO variants and to those of different optimization techniques. The simulation results clearly show a good balance between global and local exploration abilities and consequently results in good optimum solution.

Keywords: hybrid meta-heuristics; Hybrid Particle Swallow Swarm Optimization algorithm; truss optimization; frequency constraints

1. Introduction

Natural frequencies of a structure provide useful information about the dynamic behavior of the system. In fact, in most of the low frequency vibration problems, the response of the structure is primarily a function of its fundamental frequencies and mode shapes (Grandhi 1993). In particular, it is sometimes desirable to control the natural frequencies of a structure in order to keep out the unwelcome resonance phenomenon. Frequency constraints are highly non-linear, non-convex and implicit with respect to the design variables. The highly nonlinear dynamic truss shape and size optimization with multiple natural frequency constraints has been studied since the 1980s with the paper of Bellagamba and Yang (Grandhi 1993). In spite of difficulties in addressing this type of problem, considerable progress has been achieved in solution methods, where the geometry of the structure is prescribed and cross-sectional areas have to be optimized, i.e., size optimization of trusses (Grandhi and Venkayya 1998, Khot 1985, Tong and Liu 2001, Sedaghati *et al*, 2002). However, until nowadays, relatively little technical papers (Wang *et al.* 2004, Lingyun *et al.* 2005, Lingyun *et al.* 2011, Gomes 2011, Zuo *et al.* 2011, Miguel and Miguel 2012, Kaveh and Zolghadr 2013, Gholizadeh, and Barzegar 2013, Kaveh and Zolghadr 2014), among others, are available on structural shape and sizing optimization with multiple natural frequency constraints, in spite of the natural frequencies of a structure to be much more sensitive to the shape changes and combining

*Corresponding author, Professor, E-mail: alikaveh@iust.ac.ir

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

shape and sizing variables with different orders of magnitude (Miguel and Miguel 2012).

In this paper the Particle Swarm Optimization algorithm (PSO) is hybridized with the Swallow Swarm Optimization algorithm (SSO), and named as Hybrid Particle Swallow Swarm Optimization (HPSSO). This algorithm is the applied to shape and size optimization of trusses with multiple natural frequency constraints. PSO incorporates swarming behaviors, from which the idea is emerged initially by Kennedy and Eberhart (2001). The system is initialized first as a set of randomly generated potential solutions and then guided random search is performed for the optimum one iteratively such as other population based meta-heuristic algorithms. The search is based on an idea that particles move through the search space from their current positions with velocities dynamically adjusted according to their current velocity, best self-experienced position and best global-experienced position with some impression of randomness. Neshat *et al*. (2013) have recently presented a new swarm intelligence-based technique, Swallow Swarm Optimization (SSO), which was conceptualized based on modeling swallow swarm movement and their other behaviors. The algorithm shares some common features with PSO, but with several significant differences. In this algorithm the colony is divided into internal subcolonies with a commonly best experienced particle as local leader. The movement velocity of individuals is altered similar to the PSO, with an additional component taken from the best experienced inter subcolony position associated with each individual. In addition SSO benefits utilizing other type of particles (aimless particles) as an operation for further exploration task. In the real colony these swallows increase the chance of finding food outside the internal areas.

Speed of convergence and global search ability are the two important criteria for evaluating the performance of stochastic search techniques (Kaveh 2014). In the standard PSO, all particles learn from the best global-experienced particle in updating velocities and positions. Hence the algorithm exhibits a fast-converging behavior in the truss shape and size optimization with multiple natural frequency constraints due to the presence of many design variables with different orders of magnitude, many constraints, and large size and highly nonlinearity of the search space considering the frequency constraints within the shape optimization. To overcome this problem, the HPSSO algorithm is presented in this paper that tries to add new features of the SSO algorithm to the PSO. HPSSO very recently has been applied successfully by the authors in the optimizing of benchmark mathematical functions and sizing optimization of trusses (Kaveh *et al*. 2014). HPSSO provides a mechanism for particles to learn not only from the best global-experienced particle but also from other promising particles. Getting to work some particles with predetermined tasks is another utilized feature. For performance evaluation of the HPSSO, we do experiments on 4 benchmark trusses taken from literature. Numerical results reveal that the HPSSO performs much better compared to the standard PSO, a very recently developed effective variant of PSO, and other search techniques available for the literature.

The remaining sections of this paper are organized as follows. In Section 2 we will present HPSSO algorithm together with outlining the PSO and SSO algorithms. In subsequent section, the new method is implemented and applied to four benchmark truss examples. Finally the paper is concluded in Section 4.

2. A hybrid of PSO with SSO (HPSSO)

2.1 Particle swarm optimization (PSO)

PSO is a population based meta-heuristic algorithm developed by Kennedy and Eberhart (2001) that simulates social behaviors of animals. Similar to other meta-heuristic methods, PSO is initialized with a population of random designs, named particles, that is updated in each generation to search the optimum. Each particle is associated with a velocity vector adaptively changed in the optimization process. Particles move through the search space from their current positions with velocity vectors that are dynamically adjusted according to their current velocity, best self-experienced position and the best global-experienced position. PSO algorithm constitutes the simple conduct rules for search ability of each particle as follows

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

The new position of particles X_i^{k+1} is obtained by adding the new velocity V_i^{k+1} to the current position X_i^k. V_i^k , P_i^k and P_g^k are previous velocity, the best position visited by each particle itself and the best solution the swarm has found so far, respectively. 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 learning factors which control the influence of the cognitive and social components.

2.2 Swallow swarm optimization (SSO)

SSO has been developed recently by Neshat *et al.* (2013) as a new swarm intelligence based algorithm reproducing the behavior of swallow swarms. Studies conducted on various species of swallows revealed peculiar features that have been taken as the basis of the SSO algorithm: the very social life and migration of large groups; high-speed flying which can affect convergence speed; there are few floating swallows that fly out of the colony or between subcolonies to search and inform the rest of the swarm on food sources or on the attack of hunters; organization of the swarm in several subcolonies each of which has an experienced leader.

SSO has common features with PSO but also several significant differences (Neshat *et al.* 2013). An initial population of particles is randomly generated and progressively updated in the process of optimization. Three types of particles are considered: leader, explorer and aimless particles. Leader particles are categorized into two types: Local Leaders (*LL*) that conduct the related internal subcolonies and show a local optimum point, and Head Leader (*HL*) that is responsible for the leadership of the entire colony and indicates the global optimum point. Explorer particles, that represent the largest part of the population, take care of the exploration of design space. In each optimization iteration (k), particles play different roles according to their type.

Each swallow arriving at an extreme point emits a special sound to guide the group toward there. If that place is the best in the design space, that particle becomes the Head Leader, $HL^{(k)}$. If the particle reaches a good position (yet not the best) compared with its neighboring particles, it is chosen as a local leader, $LL^{(k)}$. Otherwise, the particle is an explorer one and has to change its position in the search space. The new position of explorer particles, $X_i^{(k+1)}$, is obtained by adding a change velocity $V_i^{(k+1)}$ to the current position $X_i^{(k)}$ considering $VHL_i^{(K+1)}$ (change velocity vector of particle toward Head Leader), and $VLL_i^{(K+1)}$ (change velocity vector of particle toward Local Leader). The change velocity of explorer particles toward head leader and corresponding local leader are dynamically adjusted according to the current velocity vector of the particle toward leaders ($VHL_i^{(k)}$ and $VLL_i^{(k)}$, respectively), best self-experienced position ($X_{best}^{(k)}$), and leader's

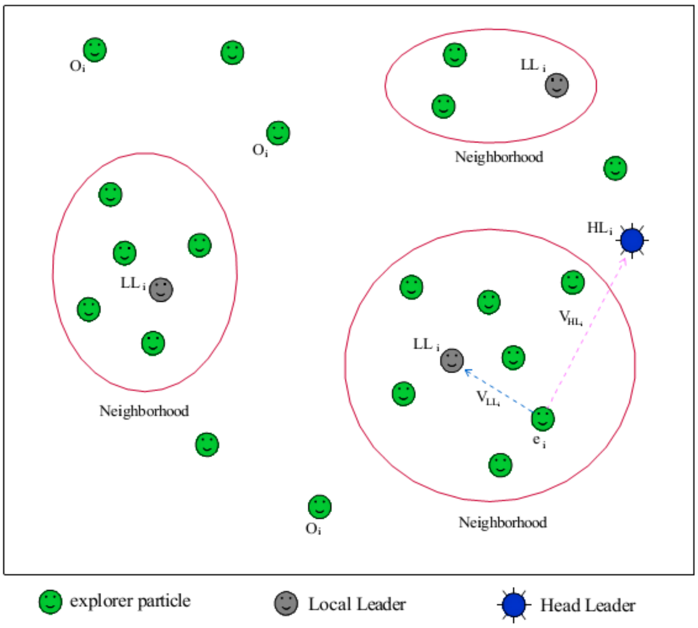

Fig. 1 Types of particles and movements of the explorer particles

position ($HL^{(k)}$ and $LL^{(k)}$, respectively). This is shown schematically in Fig. 1 and modeled mathematically as follows

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

$$V_i^{k+1} = VHL_i^{k+1} + VLL_i^{k+1}$$

$$VHL_i^{k+1} = VHL_i^k + \alpha_{HL}\, rand()\,(Xbest_i^k - X_i^k) + \beta_{HL}\, rand()\,(HL^k - X_i^k) \tag{2}$$

$$VLL_i^{k+1} = VLL_i^k + \alpha_{LL}\, rand()\,(Xbest_i^k - X_i^k) + \beta_{LL}\, rand()\,(LL_i^k - X_i^k)$$

where α_{HL}, β_{HL}, α_{LL} and β_{LL} are acceleration control coefficients adaptively defined (Neshat *et al*. 2013), while *rand*() is a random number uniformly distributed in (0,1).

Aimless particles $o_{(i)}$ also carry out exploration but have nothing to do with head leader and local leaders. They simply move back and forth with respect to their previous positions by displacing by a random fraction of the allowable step defined by the upper and lower bound of design variables. That is

$$o_i^{k+1} = o_i^k + \left[rand(\{-1,1\}) \times \frac{rand(\min_s, \max_s)}{1 + rand()} \right] \tag{3}$$

2.3 Hybrid particle swallow swarm optimization

Hybrid particle swallow swarm optimization (HPSSO) includes two important features of the SSO being added to the basic PSO formulation, considering a specific number of subcolonies and a certain number of particles for specific task. Similar to SSO, there are leaders (global leader and local leaders), explorers, and aimless particles. The size of population N is specified along with the number of subcolonies $N_{subcolony}$ and aimless particles $N_{aimless}$; the number of explorer particles ($N_{e.p}$) is determined as a consequence. HPSSO starts with a set of particles randomly positioned in the design space and with random velocities. The position and velocity of each particle are progressively updated to search the optimum. In each iteration, particles are sorted based on the value of the cost function (usually, pseudo-cost function including penalty terms or fitness). The best particle is set as the head leader and $N_{subcolony}$ subsequent particles are set as local leaders going from top to bottom. $N_{aimless}$ particles are then selected from the worst ones going from bottom to top. The remaining particles are set as explorers. The search of each explorer particle is performed by adding the updated velocity vector to the current position of that particle. Compared with PSO, the velocity vector includes an additional term to account for the contribution of local leaders

$$
\begin{aligned}
&X_i^{k+1} = X_i^k + V_i^{k+1} \\
&V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (P_{l(i)}^k - X_i^k)
\end{aligned}
\tag{4}
$$

where $P_{l(i)}{}^k$ is the local leader of the subcolony including the ith particle, r_3 is a uniform random number in the (0,1) interval and c_3 is the learning factor controlling the influence of the proximity cognition. In each iteration, the position of a particle included in a subcolony can be changed so as to move away from the current local leader and join the leader of another group. The distance between each explorer particle and local leaders is used to determine the related subcolony so that each explorer particle can be placed near the closest local leader. That is:

$$
\begin{aligned}
&dist_{i,j} = \left| X_i - P_{l,j} \right|, \quad i = 1, 2, \ldots, N_{e.p} \ , \quad j = 1, 2, \ldots, N_{sub-colony} \\
&dist_{i,j} = \sqrt{(X_i^1 - P_{l,j}^1)^2 + (X_i^2 - P_{l,j}^2)^2 + \cdots + (X_i^{ng} - P_{l,j}^{ng})^2}
\end{aligned}
\tag{5}
$$

where ng is the number of design variables, $dist_{i,j}$ is the distance between the ith explorer particle, and jth local leader.

Three possible options can be considered for aimless particles: (i) they perform just a random search in the same way as it is done in SSO; (ii) they perform a local search in the neighborhood of local leaders; (iii) they perform a dynamic search in the neighborhood of the global leader. If option (ii) is chosen, the number of aimless particles coincides with the number of subcolonies and hence an aimless particle should be defined for each subcolony. In this case, the distance between the worst particles and local leaders is the criterion to assign each aimless particle to its corresponding subcolony. This strategy is most effective in truss optimization problems (Kaveh *et al*. 2014).

Aimless particles perform their search in the neighborhood of the local leader of their subcolony according to the following rule

$$o_i^{k+1} = P_{l(i)}^k + rand(-1,1) \times \left[\lambda^k \times (\max_s - \min_s) \right], \; i = 1,2,3,...,N_{aimless} \tag{6}$$

where: $rand(-1,1)$ is a uniform random number between −1 and 1; $\min_s$ and $\max_s$, respectively, are the lower and upper bound of design variables; λ^k is a parameter defined to generate the effective search range about local leaders. That is

$$\lambda^k = \lambda_{\max} - (\lambda_{\max} - \lambda_{\min}) \times iter / iter_{\max} \tag{7}$$

where $\lambda_{\max}$ and $\lambda_{\min}$, respectively, are the values of λ in the first and last iterations of the algorithm, set in the present study as 0.01 and 0.001; *iter* is the number of the current iteration; $iter_{\max}$ is the total number of optimization iterations.

Fig. 2 illustrates the transition between two consecutive generations. The population is updated by: (i) copying first the head leader and local leaders from one generation to the subsequent (in some way, this can be interpreted as an elitist strategy); (ii) performing search with explorer particles to move population toward the best regions of design space (exploration phase or global search); (iii) performing a dynamic local search with aimless particles in the neighborhood of the head leader or local leaders. The flowchart of the HPSSO algorithm is presented in Fig. 3.

3. Numerical examples

The HPSSO algorithm presented in this research was tested in four truss shape and size classical optimization problems with multiple natural frequency constraints of a planar 10-bar

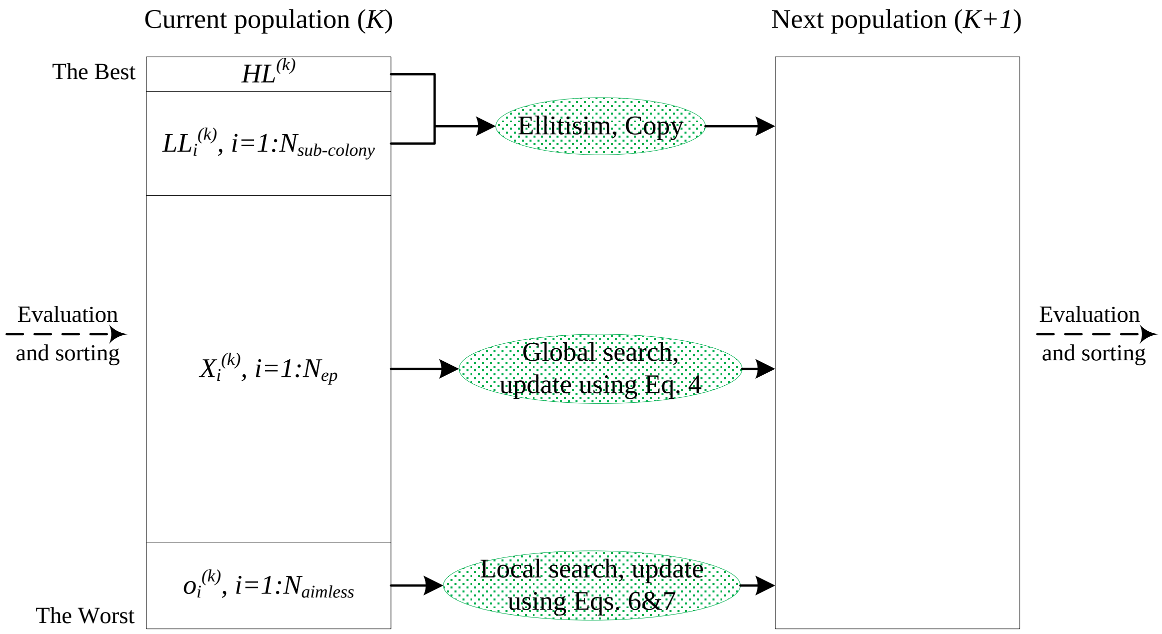

Fig. 2 Schematic representation of the transition from the current iteration to the subsequent iteration

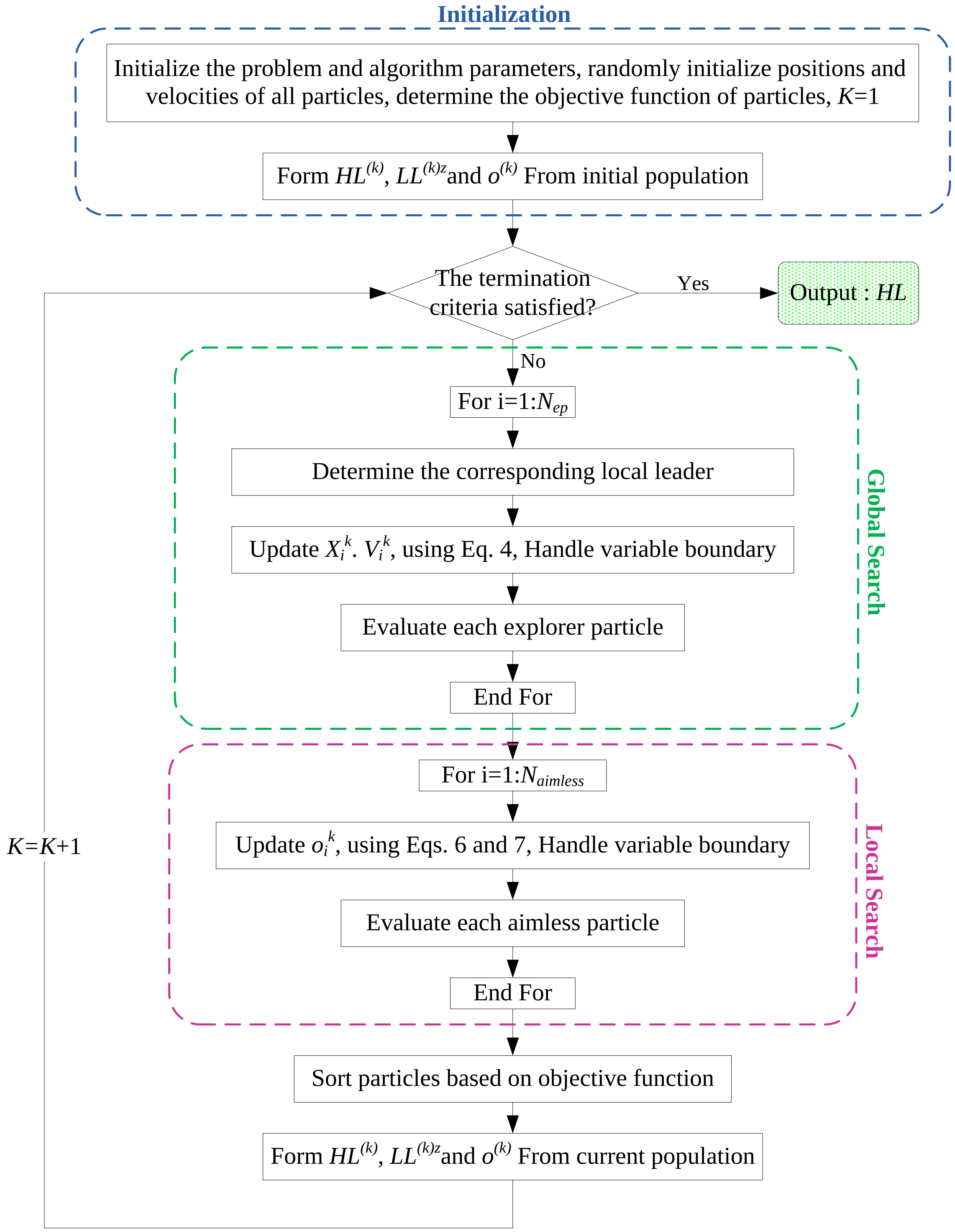

Fig. 3 Flowchart of the HPSSO algorithm

Table 1 Properties of problems 1 and 4 considered from the literature

	planar 10-bar truss			72-bar space truss		
	E (GPa)	ρ (kg/m^3)	additional Mass (kg)	*E* (GPa)	ρ (kg/m^3)	Additional Mass (kg)
PSO	69.8	2770.0	454.0	69.8	2770.0	2270
CSS	69.8	2770.0	454.0	69.8	2770.0	2270
FA	68.95	2767.99	453.6	68.95	2767.99	2268
HS	68.95	2767.99	453.6	68.95	2767.99	2268
DPSO	68.9	2770.0	454.0	-	-	-

truss, a simply supported 37-bar plane truss, a 52-bar space dome shaped truss, and a spatial 72-bar space truss. These test cases were frequently used in structural design optimization to test optimization algorithms as follows among many others: by, Grandhi (1993) using evolutionary node shift methods, Grandhi and Venkaya (1998) using an optimality algorithm, Sedaghati *et al*. (2002) utilizing a sequential quadratic programming and finite element force method, Wang *et al*. (2004) employing a niche hybrid genetic algorithm, Lingyun *et al*. (2005) using Niche Hybrid Genetic Algorithm (NHGA), Lingyun *et al*. (2011) based on parallel genetic algorithm, Gomes (2011) utilizing standard PSO, Zuo *et al*. (2011) using adaptive eigenvalue reanalysis methods, Miguel and Miguel (2012) employing both harmony search (HS) and firefly algorithm (FA) methods, Kaveh and Zolghadr (2011, 2013, 2014) utilizing Democratic PSO algorithm (DPSO) as an advanced variant of PSO and standard and enhanced charged system search algorithm (CSS). Optimization results were compared with recently presented studies using new effective optimization methods which: FA, HS, CSS, and in particular with standard PSO and with an advanced PSO variant. Kaveh and Zolghadr (2013) have recently developed a Democratic Particle Swarm Optimization (DPSO) which enables all eligible particles have the right to be involved in decision making to enhance exploration capabilities with respect to standard PSO. It should be noted the first and last problems have been considered slightly different in these studies. Table 1 represents the details (elastic modulus (E); specific mass (ρ); and additional mass) considered for these problems. Considering lower specific and additional mass and bigger modulus of elasticity will generally result in relatively lighter structures. For comparing the proposed method with all of these algorithms (PSO (Gomes 2011), CSS (Kaveh and Zolghadr 2011), FA and HS (Miguel and Miguel 2012), and DPSO (Kaveh and Zolghadr 2014)) we will consider the properties according to each study listed in Table 1.

The best combination of internal parameters was determined by carrying out a sensitivity analysis on the 37-bar plane truss: population included 30 particles and 5 subcolonies, 5 aimless particles; c_1 and c_3 were set equal to 0.8, c_2 was set equal to 2, the inertia weight decreased from 0.9 to 0.7. Table 2 presents the results of sensitivity analysis of 30 independent runs, carried out for adjusting the population parameters. As it is clear, considering the number of subcolonies at least as one tenth of the population size, leads to lighter designs. The lightest design is obtained for N=30 and $N_{subcolony}$=5. Learning parameters are adjusted considering various amounts of each one: (0.3, 0.6, 0.8, 1.0, 1.3, 1.5, 1.7, 2.0, 2.5). Table 3 tabulates the best optimum weight achieved from 30 independent runs for each combination case of learning parameters and two adjacent combinations. The best optimum design is obtained for c_1=c_3= 0.8, and c_2= 2. By proper selection of ω, it is possible to achieve a good balance between global and local exploration abilities, thus finding better designs. The inertia weight ω can range between 0 and 1 and strongly affects the

Table 2 Results of sensitivity analysis carried out to find the best combination of internal population parameters of HPSSO for the 37-bar plane truss

	Nll=2	*Nll*=3	*Nll*=5	*Nll*=7	*Nll*=10	*Nll*=15	*Nll*=20
N=20	360.3409	360.1410	360.2016	360.5238	–	–	–
N=30	360.9523	360.5205	360.0889	360.5217	361.0257	–	–
N=50	364.5873	362.5915	361.3069	360.3522	360.3533	360.6950	361.4856
N=100	382.1692	372.0556	367.4492	361.6171	361.0441	361.0816	361.5903
N=150	387.4601	379.7737	379.1532	372.6632	366.3754	363.2179	367.3366

Table 3 Results of sensitivity analysis carried out to find the best combination of learning parameters of HPSSO for the 37-bar plane truss

C_1=0.6	C_3=1.3	C_3=1.5	C_3=1.7
C_2=1.5	360.9940	360.3556	360.4090
C_2=1.7	360.2080	360.1403	360.3280
C_2=2.0	360.2069	360.7642	360.4093
C_1=0.8	C_3=0.6	C_3=0.8	C_3=1
C_2=1.7	360.4796	360.3226	360.6734
C_2=2.0	360.4262	360.0985	360.507
C_2=2.5	360.5259	360.7922	360.6627
C_1=1.0	C_3=0.3	C_3=0.6	C_3=0.8
C_2=1.7	360.4494	360.6422	360.3555
C_2=2.0	360.3747	360.1526	360.3601
C_2=2.5	360.5217	360.3193	360.3951

convergence behavior. As shown by Eberhat and Shee (2000), considering a reducing ω linearly from 0.9 to 0.7 is an efficient approach. In this study, reducing ω linearly from 0.9 to 0.7 during the optimization process leads to better results.

The stopping criterion was based on the maximum number of optimization iterations ($iter_{max}$) set equal to 200, 300 and 400 depending to the test cases size. In order to investigate the effect of the initial population on the optimization process, each test problem was solved independently 30 times starting from a different population randomly generated. This allows to account for the random nature of the HPSSO algorithm. The optimization and finite element structural analyses code were coded in the MATLAB software environment.

3.1 Problem statement

In a frequency constraint truss layout and size optimization problem, the aim is to minimize the weight of the structure while satisfying some constraints on natural frequencies. The design variables are considered to be the cross-sectional areas of the members and/or the coordinates of some nodes. Prescribing the truss topology and assuming to be unchanged, the optimization problem can be stated as follows

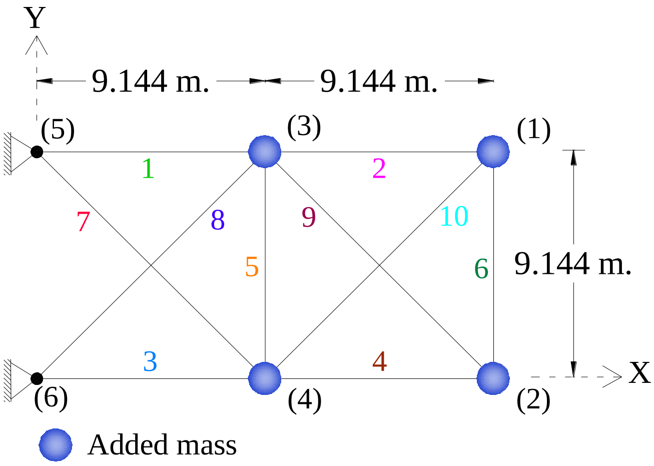

Fig. 4 Schematic of the planar 10-bar truss structure

$$\text{Find}\ \{V\} = [v_i] \quad i = 1, 2, \ldots, k,\ \ v_{\min} \le v_k \le v_{\max},$$
$$\text{To minimize:}\ \ W\{V\} = \sum_{e=1}^{n} L_e \rho_e A_e,\ \ \text{Subject to:} \tag{8}$$

$$g_1\{V\}: \omega_j^* - \omega_j \le 0 \quad \text{for some natural frequencies j} \tag{9}$$

$$g_2\{V\}: \omega_{jj} - \omega_{jj}^* \le 0 \quad \text{for some natural frequencies jj} \tag{10}$$

where $\{V\}$ is the set of design variables; k is the number of independent design variables, v_i, including either a shape or sizing variable must take a value between its lower bound v_{min} and upper bound v_{max}, respectively. W is the total weight of truss, and total number of elements is denoted by n. L_e, ρ_e and A_e are respectively length, material density, and cross sectional area of the eth element. First frequency constraint (g_1) represents that some natural frequencies ω_j, should exceed the prescribed lower limits. Second frequency constraint (g_2) represents that other natural frequencies, should be less than the prescribed upper limits. In order to handle optimization constraints, a penalty approach is utilized in this study by introducing the following pseudo-cost function (Kaveh and Bakhshpoori 2013)

$$f_{\text{cost}}(\{V\}) = (1 + \varepsilon_1 . \upsilon)^{\varepsilon_2} \times W(\{V\}), \quad \upsilon = \sum_{j=1}^{2} \max[0,\ g_j(\{V\})] \tag{11}$$

where υ is the total constraint violation. Constants ε_1 and ε_2 must be selected considering the exploration and the exploitation rate of the search space. In this study, ε_1 was set equal to one while ε_2 was selected so as to decrease the total penalty. Thus, ε_2 increased from the value of 1.5 set in the first steps of the search process to the value of 3 set toward the end of the optimization process.

3.2 Planar 10-bar truss

Truss geometry including node and element numbering, non-structural mass of 453.6 kg (1000

Table 4 Optimization results (cm^2) obtained by other meta-heuristic methods in the 10-bar truss problem

Member	FA	HS	CSS	PSO	DPSO
1	36.198	34.282	38.811	37.712	35.944
2	14.030	15.653	9.0307	9.959	15.530
3	34.754	37.641	37.099	40.265	35.285
4	14.900	16.058	18.479	16.788	15.385
5	0.654	1.069	4.479	11.576	0.648
6	4.672	4.740	4.205	3.955	4.583
7	23.467	22.505	20.842	25.308	23.610
8	25.508	24.603	23.023	21.613	23.599
9	12.707	12.867	13.763	11.576	13.135
10	12.351	12.099	11.414	11.186	12.357
Mass (kg)	531.28	534.99	531.95	537.98	532.39

Table 5 Optimization results (cm^2) obtained by HPSSO in the 10-bar truss problem

	HPSSO					
According to	FA and HS		CSS and PSO		DPSO	
Member	$iter_{max}$=200	$iter_{max}$=300	$iter_{max}$=200	$iter_{max}$=300	$iter_{max}$=200	$iter_{max}$=300
1	35.91463	35.44069	35.10490	35.14366	35.95107	35.81050
2	14.74970	14.80652	14.51721	14.78617	14.61329	14.97440
3	35.13693	35.71416	35.06363	35.44331	35.51132	35.37693
4	15.00478	14.97475	14.74008	14.67259	15.03865	14.76242
5	0.64503	0.64500	0.64586	0.64526	0.64500	0.64502
6	4.62533	4.62048	4.57284	4.55385	4.63732	4.63220
7	23.78835	23.81563	23.51944	23.83466	23.71204	24.33330
8	24.31428	24.25257	24.03368	23.58726	24.62326	23.88956
9	12.60807	12.59062	12.50740	12.20288	12.57024	12.62065
10	12.58117	12.52555	12.36574	12.37694	12.40856	12.60052
Mass (kg)	530.8289	530.7610	524.4870	524.4868	532.1052	532.0762
Number of structural analyses	4,806	6,894	4,806	7,206	4,806	7,206

lb) is attached to all free nodes (1-4), and kinematic constraints is shown in Fig. 4. The material is aluminum, with elastic modulus equal to 68.95 GPa and specific mass of 2767.99 kg/m^3. These properties are according to study by Miguel and Miguel (2012) using FA and HS. The natural frequency constraints are $\omega_1 \geq 7$ Hz, $\omega_2 \geq 15$ Hz, and $\omega_3 \geq 20$ Hz. The allowable lower and upper bound of the cross sectional area (m^2) is 0.645×10^{-4} and 50×10^{-4}.

Table 4 presents the best optimized designs and the corresponding masses found by different methods (FA, HS, CSS, standard PSO and DPSO), and Table 5 lists the optimized designs, the required number of structural analyses and the corresponding masses obtained by HPSSO according to the details reported using other methods. Table 6 represents the corresponding natural frequencies. Table 7 presents the optimization results based on the HPSSO obtained for 30 independent runs carried out from different initial populations randomly generated and other methods. The number of independent runs considered as 5 for FA and HS, 10 for CSS and 30 for PSO and DPSO.

Table 6 Optimum design of natural frequencies (HZ) for the 10-bar truss from various methods

Frequency number	FA	HS	CSS	PSO	DPSO	HPSSO*	
						$iter_{max}$=200	$iter_{max}$=300
1	7.0002	7.0028	7.000	7.000	7.000	7.000	7.000
2	16.1640	16.7429	17.442	17.786	16.187	16.175	16.180
3	20.0029	20.0548	20.031	20.000	20.000	20.004	20.001
4	20.0221	20.3351	20.208	20.063	20.021	20.010	20.008
5	28.5428	28.5232	28.261	27.776	28.470	28.568	28.545
6	28.9220	29.2911	31.139	30.939	29.243	28.977	28.957
7	48.3538	49.0342	47.704	47.297	48.769	48.535	48.556
8	50.8004	51.7451	52.420	52.286	51.389	51.045	51.057

*According to the details used by FA and HS (Miguel and Miguel 2012)

Table 7 Comparison (kg) of robustness and reliability of HPSSO and other meta-heuristic methods in the 10-bar truss problem

Algorithm	According to	Best	Average	Worst	SD
FA		-	535.07	-	3.64
HS		-	537.68	-	2.49
CSS		-	536.39	-	3.32
PSO		-	540.89	-	6.84
DPSO		-	537.80	-	4.02
HPSSO ($iter_{max}$=200)	FA and HS	530.8289	535.97	539.14	2.72
HPSSO ($iter_{max}$=200)	CSS and PSO	524.4870	528.72	532.44	2.98
HPSSO ($iter_{max}$=200)	DPSO	532.1052	536.25	539.75	3.13
HPSSO ($iter_{max}$=300)	FA and HS	530.7610	534.16	537.78	3.07
HPSSO ($iter_{max}$=300)	CSS and PSO	524.4868	527.91	531.71	2.95
HPSSO ($iter_{max}$=300)	DPSO	532.0762	535.54	539.99	3.15

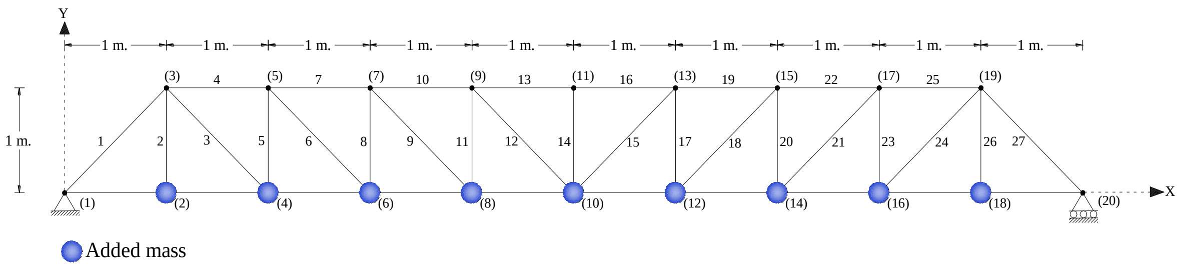

Fig. 5 Schematic of the 37-bar truss structure

3.3 Planar 37-bar truss

The second optimization problem solved in this study regards the simply supported planar 37-bar truss shown in Fig. 5. A non-structural mass of m=10 kg is attached at each of the free nodes on the lower chord. The truss is made of steel with modulus of elasticity of 210 Gpa and material density of 37800 kg/m^3. The truss is optimized on shape and size for its mass minimization with multiple frequency constraints. Nodal coordinates in the upper chord and cross-sectional areas of

Table 8 Comparison of optimization results (Y coordinates: m; and areas: cm^2) obtained by HPSSO and other meta-heuristic methods in the 37-bar truss problem

Member	FA	HS	CSS	PSO	DPSO	HPSSO	
						$iter_{max}$=300	$iter_{max}$=400
Y3, Y19	0.9392	0.8415	0.8726	0.9637	0.9482	1.00000	1.00000
Y5, Y17	1.3270	1.2409	1.2129	1.3978	1.3439	1.347531	1.357692
Y7, Y15	1.5063	1.4464	1.3826	1.5929	1.5043	1.50689	1.531195
Y9, Y13	1.6086	1.5334	1.4706	1.8812	1.6350	1.63223	1.666696
Y11	1.6679	1.5971	1.5683	2.0856	1.7182	1.705748	1.734591
A1, A27	2.9838	3.2031	2.9082	2.6797	2.6208	2.821098	2.911875
A2, A26	1.1098	1.1107	1.0212	1.1568	1.0397	1.001201	1.00000
A3, A24	1.0091	1.1871	1.0363	2.3476	1.0464	1.00000	1.00000
A4, A25	2.5955	3.3281	3.9147	1.7182	2.7163	2.624512	2.539312
A5, A23	1.2610	1.4057	1.0025	1.2751	1.0252	1.248281	1.268065
A6, A21	1.1975	1.0883	1.2167	1.4819	1.5081	1.277638	1.135538
A7, A22	2.4264	2.1881	2.7146	4.6850	2.3750	2.750126	2.546305
A8, A20	1.3588	1.2223	1.2663	1.1246	1.4498	1.251069	1.392601
A9, A18	1.4771	1.7033	1.8006	2.1214	1.4499	1.3896	1.432117
A10, A19	2.5648	3.1885	4.0274	3.8600	2.5327	2.763805	2.492398
A11, A17	1.1295	1.0100	1.3364	2.9817	1.2358	1.174415	1.174892
A12, A15	1.3199	1.4074	1.0548	1.2021	1.3528	1.333822	1.352078
A13, A16	2.9217	2.8499	2.8116	1.2563	2.9144	2.509497	2.57735
A14	1.0004	1.0269	1.1702	3.3276	1.0085	1.00000	1.00000
Mass (kg)	360.05	361.50	362.84	377.20	360.40	360.0752	359.975
Number of structural analyses	-	-	-	-	-	7,038	9,342

Table 9 Optimum design of natural frequencies (HZ) for the 37-bar truss from various methods

Frequency number	FA	HS	CSS	PSO	DPSO	HPSSO	
						$iter_{max}$=300	$iter_{max}$=400
1	20.0024	20.0037	20.0000	20.0001	20.0194	20.0065	20.0092
2	40.0019	40.0050	40.0693	40.0003	40.0113	40.0194	40.0222
3	60.0043	60.0082	60.6982	60.0001	60.0082	60.0054	60.0186
4	77.2153	77.9753	75.7339	73.0440	76.9896	77.1575	76.2377
5	96.9900	99.2564	97.6137	89.8240	97.2222	95.3689	95.5098

members are considered as design variables. All members on the lower chord have fixed cross sectional areas of 4×10^{-3} m^2 and the others have initial cross sectional areas of 1×10^{-4} m^2 (also as the lower bound). In the optimization process, nodes on the upper chord can be shifted vertically. In addition, nodal coordinates and member areas are linked to maintain the structural symmetry. Therefore, only five shape variables and fourteen sizing variables will be redesigned for optimization. The natural frequency constraints are $\omega_1\geq20$ Hz, $\omega_2\geq40$ Hz, and $\omega_3\geq60$ Hz.

Table 8 presents the best optimized designs and the corresponding masses found by HPSSO and different methods (FA, HS, CSS, standard PSO and DPSO), and Table 9 represents the

Table 10 Comparison (kg) of the robustness and reliability of the HPSSO and other meta-heuristic methods for the 37-bar truss problem

Algorithm	Best	Average	Worst	SD
FA	-	360.37	-	0.26
HS	-	362.04	-	0.52
CSS	-	366.77	-	3.742
PSO	-	381.2	-	4.26
DPSO	-	362.21	-	1.68
HPSSO ($iter_{max}$=300)	360.0752	367.212	400.7903	8.907
HPSSO ($iter_{max}$=400)	359.975	364.1593	398.4291	7.864

Table 11 Comparison of the optimization results (*Y* coordinates: m; and areas: cm^2) obtained by HPSSO and other meta-heuristic methods in the 52-bar truss problem

Member	FA	HS	CSS	PSO	DPSO	HPSSO	
						$iter_{max}$=200	$iter_{max}$=300
Z1	6.4332	4.7374	5.2716	5.5344	6.1123	5.787564	5.908632
X2	2.2208	1.5643	1.5909	2.0885	2.2343	2.181542	2.210574
Z2	3.9202	3.7413	3.7093	3.9283	3.8321	3.714817	3.774155
X6	4.0296	3.4882	3.5595	4.0255	4.0316	3.916504	3.985947
Z6	2.5200	2.6274	2.5757	2.4575	2.5036	2.502532	2.50074
A1	1.0050	1.0085	1.0464	0.3696	1.0001	1.000000	1.000000
A2	1.3823	1.4999	1.7295	4.1912	1.1397	1.189154	1.17997
A3	1.2295	1.3948	1.6507	1.5123	1.2263	1.268856	1.26862
A4	1.2662	1.3462	1.5059	1.5620	1.3335	1.502314	1.426836
A5	1.4478	1.6776	1.7210	1.9154	1.4161	1.447167	1.438013
A6	1.0000	1.3704	1.0020	1.1315	1.0001	1.000000	1.000000
A7	1.5728	1.4137	1.7415	1.8233	1.5750	1.702234	1.555282
A8	1.4153	1.9378	1.2555	1.0904	1.4357	1.282983	1.408335
Mass (kg)	197.53	214.94	205.237	228.38	195.351	195.4308	195.1085

corresponding natural frequencies. Table 10 presents the statistical results based on the HPSSO obtained for 30 independent runs carried out from different initial populations randomly generated and other methods. The number of independent runs considered as 5 for FA and HS, 10 for CSS and 30 for both standard and democratic PSO.

3.4 The 52-bar dome shaped truss

Simultaneous layout and size optimization of a 52-bar domelike truss is considered as the third example. Initial layout of the structure is depicted in Fig. 6. This test case, is described in detail in Miguel and Miguel (2012). The optimized designs found by the different algorithms are compared in Table 11 that shows also the corresponding structural weights, and Table 12 represents the corresponding natural frequencies. Statistical results of independent optimization runs are presented in Table 13.

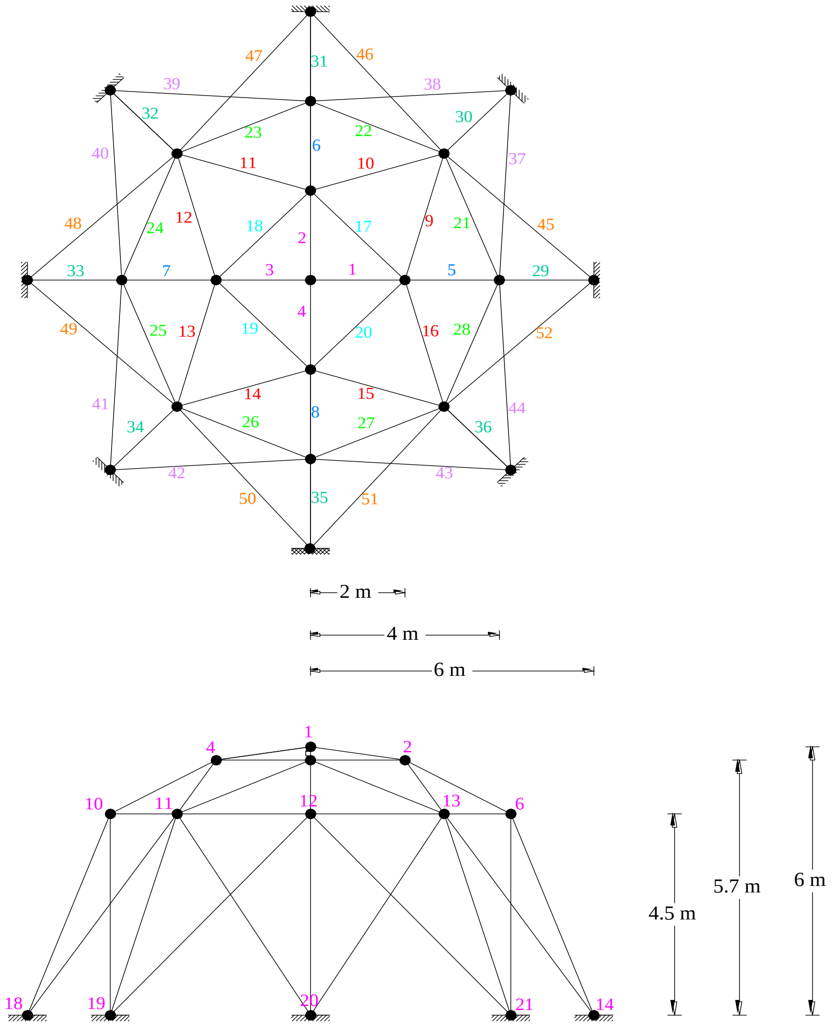

Fig. 6 Schematic of the dome shaped 52-bar truss structure

Table 12 Optimum design of natural frequencies (HZ) for the 52-bar truss from various methods

Frequency number	FA	HS	CSS	PSO	DPSO	HPSSO	
						$iter_{max}$=200	$iter_{max}$=300
1	11.3119	12.2222	9.246	12.751	11.315	11.21629	11.4099
2	28.6529	28.6577	28.648	28.649	28.648	28.6521	28.6483
3	28.6529	28.6577	28.699	28.649	28.648	28.67663	28.6490
4	28.8030	28.6618	28.735	28.803	28.650	28.71777	28.7166
5	28.8030	30.0997	29.223	29.230	28.688	29.13155	29.1050

3.5 The 72-bar space truss

The spatial 72-bar truss optimized in the last test problem is schematized in Fig. 7. In the four nodes on the top of the structure (nodes 1-4) a non-structural mass of 2270 kg is attached. The design variables are the member cross sectional areas, treated as continuous design variables, which are linked into 16 groups in order to maintain the structural symmetry. Member linking detail is available in Table 14 from Miguel and Miguel (2012).The material is aluminum, with elastic modulus being equal to 68.95 GPa and specific mass of 2770 kg/m^3. These properties are according to the study made by Miguel and Miguel (2012) using FA and HS. The natural frequency constraints are ω_1=4 Hz and $\omega_3 \geq 6$ Hz. The allowable minimum area of the cross sectional is 0.645×10^{-4} m^2.

Table 13 Comparison (kg) of the robustness and reliability of HPSSO and other meta-heuristic methods in the 52-bar truss problem

Algorithm	Best	Average	Worst	SD
FA	-	212.80	-	17.98
HS	-	229.88	-	12.44
CSS	-	213.101	-	7.391
PSO	-	234.3	-	5.22
DPSO	-	198.71	-	13.85 kg
HPSSO ($iter_{max}$=200)	195.4308	241.8956	389.2015	53.2531
HPSSO ($iter_{max}$=300)	195.1085	214.0870	270.0908	19.8910

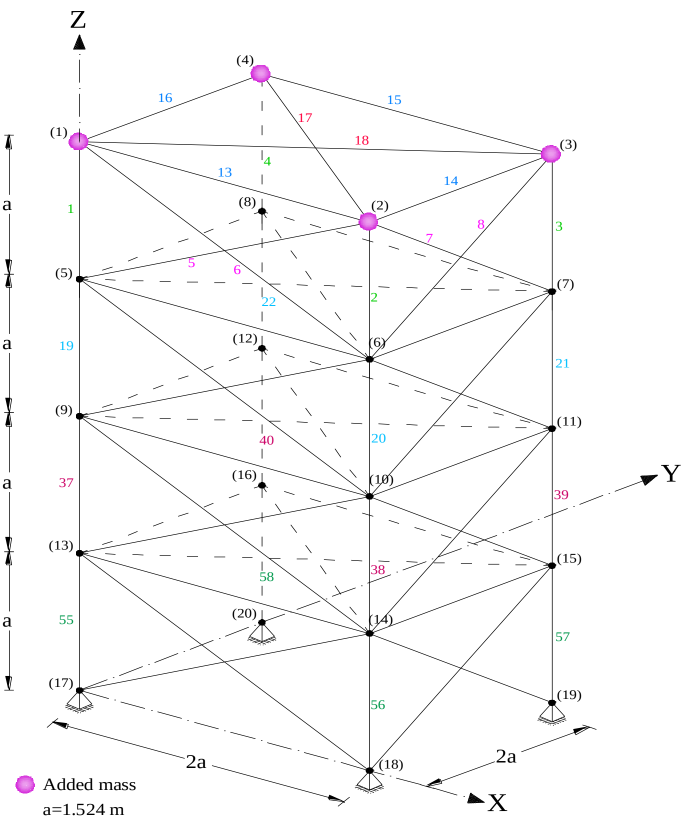

Fig. 7 Schematic of the spatial 72-bar truss structure

Table 14 Optimization results (cm^2) obtained by different meta-heuristic methods for the 72-bar truss problem

Member	FA	HS	CSS	PSO
1	3.3411	3.6803	2.528	2.987
2	7.7587	7.6808	8.704	7.849
3	0.6450	0.6450	0.645	0.645
4	0.6450	0.6450	0.645	0.645
5	9.0202	9.4955	8.283	8.765
6	8.2567	8.2870	7.888	8.153
7	0.6450	0.6450	0.645	0.645
8	0.6450	0.6461	0.645	0.645
9	12.0450	11.4510	14.666	13.45
10	8.0401	7.8990	6.793	8.073
11	0.6450	0.6473	0.645	0.645
12	0.6450	0.6450	0.645	0.645
13	17.3800	17.4060	16.464	16.684
14	8.0561	8.2736	8.809	8.159
15	0.6450	0.6450	0.645	0.645
16	0.6450	0.6450	0.645	0.645
Mass (kg)	327.691	328.334	328.814	328.823

Table 15 Optimization results (cm^2) obtained by other meta-heuristic methods for the 72-bar truss problem

	HPSSO			
According to	FA, HS		CSS and PSO	
Member	$iter_{max}$=200	$iter_{max}$=300	$iter_{max}$=200	$iter_{max}$=300
1	3.5330	3.5329	4.0983	3.4041
2	7.8391	8.0157	7.9589	7.5881
3	0.6450	0.6450	0.6450	0.6451
4	0.6450	0.6450	0.6501	0.6451
5	8.0181	8.0510	8.5625	8.2960
6	7.9707	7.9363	8.2050	7.7144
7	0.6450	0.6450	0.6450	0.6450
8	0.6450	0.6450	0.6450	0.6450
9	12.4962	12.6954	12.8024	12.4260
10	8.0792	8.0952	7.5955	8.2415
11	0.6450	0.6450	0.6450	0.6450
12	0.6450	0.6470	0.6450	0.6455
13	17.6322	17.3953	15.9458	17.0557
14	8.2570	8.0887	8.0497	8.2833
15	0.6450	0.6455	0.6450	0.6450
16	0.6452	0.6450	0.6569	0.6450
Mass (kg)	327.7704	327.6923	325.0476	324.7630
Number of structural analayses	4,806	7,206	4,806	7,206

Table 16 Optimum design of natural frequencies (HZ) for the 72-bar truss from various methods

Frequency number	FA	HS	CSS	PSO	TLBO	HPSSO	
						$iter_{\max}$=200	$iter_{\max}$=300
1	4.0000	4.0000	4.000	4.000	4.0000	4.0000	4.0000
2	4.0000	4.0000	4.000	4.000	4.0000	4.0002	4.0001
3	6.0000	6.0000	6.006	6.000	6.0000	6.0010	6.0004
4	6.2468	6.2723	6.21	6.219	6.2567	6.2407	6.2459
5	9.0380	9.0749	8.684	8.976	9.0984	9.0639	9.0801

*According to the details used by FA and HS (Miguel and Miguel 2012)

Table 17 Comparison (kg) of robustness and reliability of HPSSO and other meta-heuristic methods for the 72-bar truss problem

Algorithm	According to	Best	Average	Worst	SD
FA		-	329.89	-	2.59
HS		-	332.64	-	2.39
CSS		-	337.70	-	5.42
PSO		-		-	
HPSSO ($iter_{max}$=200)	FA and HS	327.7704	329.89	378.68	9.22
HPSSO ($iter_{max}$=200)	CSS and PSO	325.0476	348.33	454.36	36.77
HPSSO ($iter_{max}$=300)	FA and H	327.6923	331.81	390.61	14.41
HPSSO ($iter_{max}$=300)	CSS and PSO	324.7630	330.98	401.68	15.57

Table 18 Comparison (kg) of the best achieved optimum design by HPSSO and other meta-heuristic methods

Algorithm	Problem 1	Problem 2	Problem 3	Problem 4
FA	531.28 (+0.1 %)	360.050 (+0.02 %)	197.530 (+1.2 %)	327.691 (-0.0 %)
HS	534.99 (+0.8 %)	361.500 (+0.4 %)	214.940 (+10.2 %)	328.334 (+0.2 %)
CSS	531.95 (+1.4 %)	362.840 (+0.8 %)	205.237 (+5.2 %)	328.814 (+1.2 %)
PSO	537.98 (+2.6 %)	377.200 (+4.8 %)	228.380 (+17.1 %)	328.823 (+1.3 %)
DPSO	532.39 (+0.06 %)	360.400 (+0.1 %)	195.351 (+0.1 %)	-

Table 14 presents the best optimized designs and the corresponding masses found by different methods (FA, HS, CSS, standard PSO and DPSO), and Table 15 lists the optimized designs and the corresponding masses obtained by HPSSO according to the details reported using other methods. Table 16 represents the corresponding natural frequencies. Table 17 provides the optimization results based on the HPSSO obtained for 30 independent runs carried out from different initial populations randomly generated and other methods. The number of independent runs considered as 5 for FA, HS and TLBO, 10 for CSS and 30 for PSO and DPSO.

3.6 Evaluation of HPSSO and comparison with other optimization methods

Table 18 compares weight (kg) of the best reported optimum design by other methods (Fire Fly algorithm (FA) and Harmony Search method (HS) (Miguel and Miguel 2012), Charged System Search (CSS) (Kaveh and Zolghadr 2011), standard Particle Swarm Optimization (PSO) (Gomes

2011), and Democratic Particle Swarm Optimization (DPSO) (Kaveh and Zolghadr 2014)) against the HPSSO. It is clear that the HPSSO is in general the most efficient algorithm except for the 72-bar spatial truss in which HPSSO yields the optimum design 327.6923 kg which is practically same as the one (327.691 kg) obtained by the FA.

In order to analyze convergence behavior, optimization histories of the best and worst particles, and the average optimization history of all particles corresponding to the best run of the HPSSO are presented in Fig. 8. Upper bound of the Y axis is limited for the sake of clarity. It appears that HPSSO reaches an effective balance between global search and local search. It is always seen that in the early optimization cycles where randomness plays the main rule, the distance between the average and worst particle diagrams is small. HPSSO tries to increase this distance in the next iterations: this demonstrates the diversification capability of the optimization search. The distance then decreases again as the search process progresses and finally becomes negligible upon reaching the optimum design: this demonstrates the intensification ability of HPSSO. The big distance between the best particle and both average and worst particle diagrams can be due to the constrained nature of the problem. Such a behavior is observed in all test problems. The previous discussion demonstrates that HPSSO achieves an effective balance between diversification and intensification.

In order to further evaluate the performance of the algorithm, Fig. 9 shows the optimization histories of the first and third test problems for the best run seen for the HPSSO, standard PSO (Gomes 2011) and Democratic PSO (Kaveh and Zolghadr 2014). HPSSO performs better from

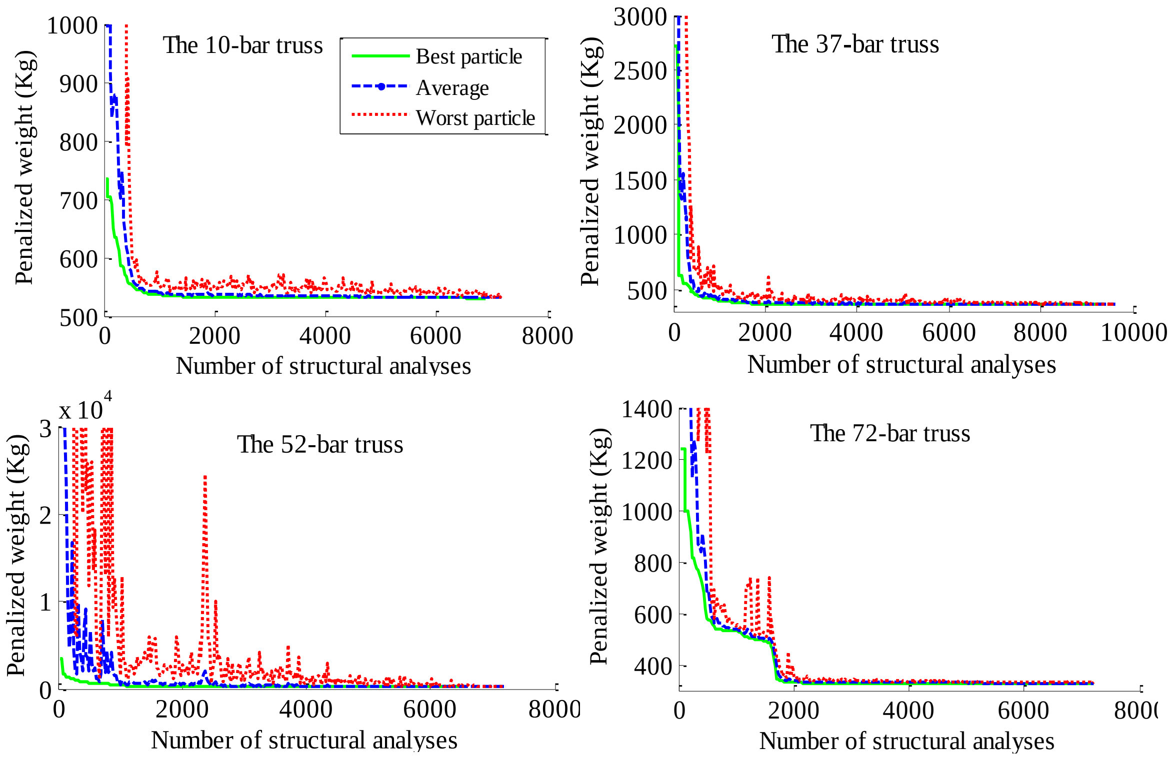

Fig. 8 Convergence curves of the best run recorded for test problems: Comparison of the best and worst particles and average of all particles

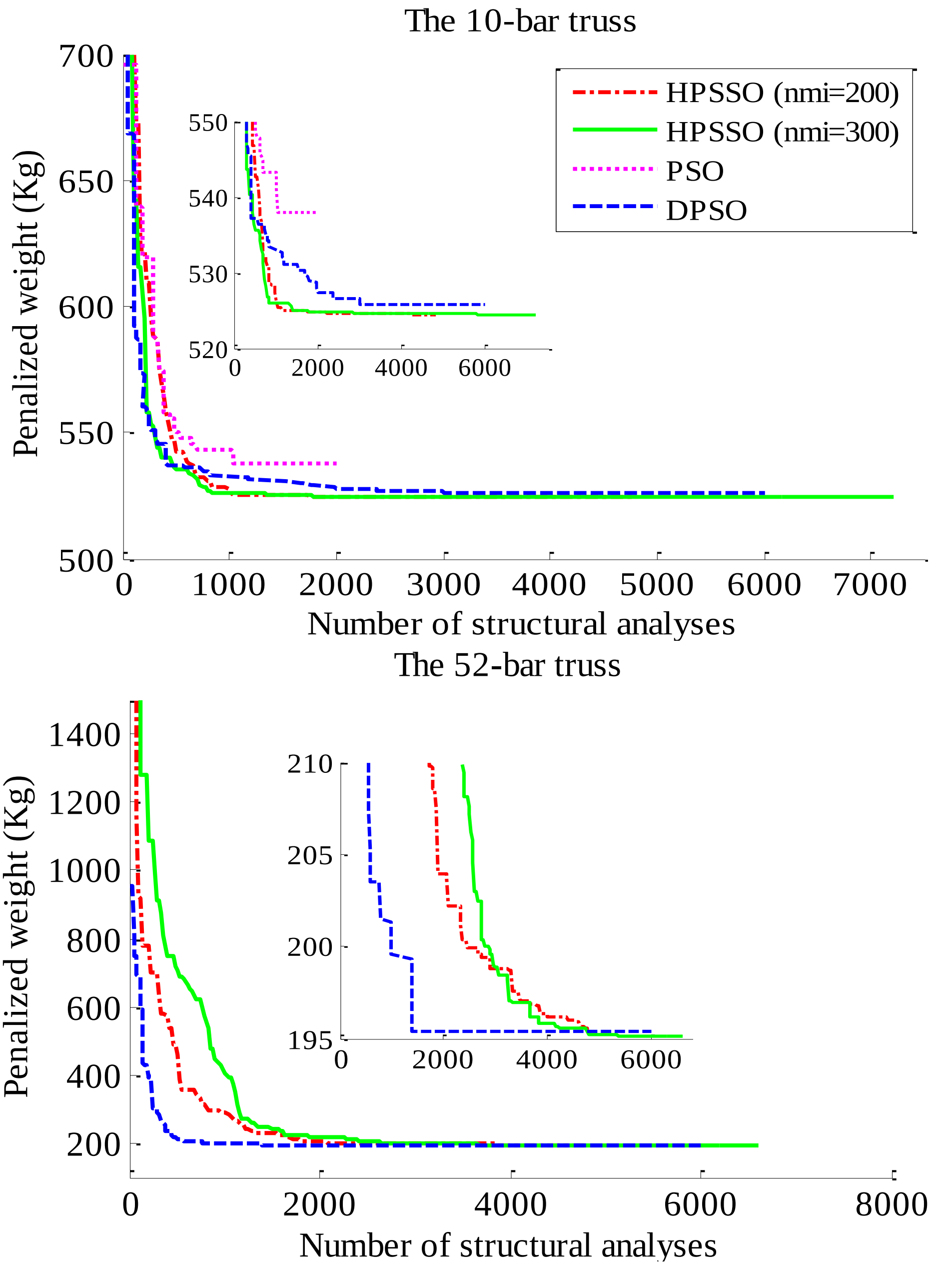

Fig. 9 Convergence curves recorded for the first and third trusses, comparing the convergence rates of the algorithms

accuracy point of than both standard and democratic PSO. The HPSSO diagrams show how this algorithm benefits all of its defined number of iterations as the stopping criteria for deeply search the design space in all test problems. Continuous step like movements of the HPSSO until the next iteration of algorithm shows how it can find all of the local optimums. This is another proof to show how HPSSO can balance its local and global search ability.

4. Conclusions

A new efficient hybrid swarm intelligence based algorithm combining Swallow Swarm Optimization and Particle Swarm Optimization is presented as a new variant of the PSO algorithm for the highly nonlinear dynamic truss shape and size optimization with multiple natural frequency constraints. Since population is divided into subcolonies, particles can learn not only from the best globally-experienced particle, but also from the best particle of each subcolony. The new HPSSO algorithm included elitism as it selects and preserves best particles (i.e., global and local leaders) in the process of updating population; makes a good balance between global and local search as explorer particles have the ability of learning from self, social and proximity cognition; utilizes aimless particles to further adjust local search ability.

HPSSO is tested on four truss design optimization problems. Numerical results demonstrate the efficiency of the proposed optimization algorithm that outperforms the standard PSO and some other state-of-the-art meta-heuristic algorithms.

References

Eberhart, R.C. and Shi, Y. (2000), “Comparing inertia weights and constriction factors in particle swarm optimization”, *Proceedings of the 2000 Congress on Evolutionary Computation*, 84-88.

Gomes, H.M. (2011), “Truss optimization with dynamic constraints using a particle swarm algorithm”, *Expert Syst. Appl.*, **38**(1), 957-968.

Grandhi, R.V. (1993), “Structural optimization with frequency constraints-a review”, *AIAA J.*, **31**(12), 2296-2303.

Grandhi, R.V. and Venkayya, V.B. (1988), “Structural optimization with frequency constraints”, *AIAA J.*, **26**, 858-866.

Gholizadeh, S. and Barzegar, A., (2013), “Shape optimization of structures for frequency constraints by sequential harmony search algorithm”, *Eng. Optim.*, **45**(6), 627-646.

Kaveh, A. (2014), *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland.

Kaveh, A. and Bakhshpoori, T. (2013), “Optimum design of space trusses using cuckoo search algorithm with Lévy flights”, *IJST, Tran. Civil. Eng.*, **37**(C1), 1-15.

Kaveh, A., Bakhshpoori, T. and Afshary, E. (2014), “An efficient hybrid particle swarm and swallow swarm optimization algorithm”, *Comput. Struct.*, **143**, 40-59.

Kaveh, A. and Zolghadr, A. (2011), “Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm”, *Asian J. Civil Eng.*, **12**(4), 487-509.

Kaveh, A. and Zolghadr, A. (2013), “Topology optimization of trusses considering static and dynamic constraints using the CSS”, *Appl. Soft Comput.*, **13**(5), 2727-2734.

Kaveh, A. and Zolghadr, A. (2014), “Democratic PSO for truss layout and size optimization with frequency constraints”, *Comput. Struct.*, **130**, 10-21.

Kennedy, J. and Eberhart, R. (2011), *Swarm Intelligence*, Morgan Kaufmann, San Francisco.

Khot, N.S. (1985), “Optimization of structures with multiple frequency constraints”, *Comput. Struct.*, **20**, 869-876.

Lingyun, W., Mei, Z., Guangming, W. and Guang, M. (2005), “Truss optimization on shape and sizing with frequency constraints based on genetic algorithm”, *Comput. Mech.*, **25**, 361-368.

Lingyun, W., Tianbing, T., Xianghong, X. and Wenjie, S. (2011), “Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm”, *Struct. Multidiscip. Optim.*, **43**, 665-682.

Miguel, L.F.F. and Miguel, L.F.F. (2012), “Shape and size optimization of truss structures considering dynamic constraints through modern meta-heuristic algorithms”, *Expert Syst. Appl.*, **39**(10), 9458-9467.

Neshat M., Sepidnam, G. and Sargolzaei, M. (2013), "Swallow swarm optimization algorithm: a new method to optimization", *Neural. Comput. Applic.*, **23**, 429-454.

Sedaghati, R., Suleman, A. and Tabarrok, B. (2002), "Structural optimization with frequency constraints using finite element force method", *AIAA J.*, **40**, 382-388.

Tong, W.H. and Liu, G.R. (2001), "An optimization procedure for truss structure with discrete design variables and dynamic constraints", *Comput. Struct.*, **79**, 155-162.

Wang, D., Zhang, W.H. and Jiang, J.S. (2004), "Truss optimization on shape and sizing with frequency constraints", *AIAA J.*, **42**, 622-630.

Zuo, W., Xu, T., Zhang, H. and Xu, T. (2011), "Fast structural optimization with frequency constraints by genetic algorithm using adaptive eigenvalue reanalysis methods", *Struct. Multidiscip. Optim.*, **43**, 799-810.

CC