ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC) VOL. 14, NO. 2 (2013)
PAGES 201-223

ENGINEERING DESIGN OPTIMIZATION USING A HYBRID PSO AND HS ALGORITHM

A. Kaveh[*, a] and A. Nasrollahi[b]
[a,b]Center of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran
[b]Building and Housing Research Center (BHRC), Tehran, Iran

Received: 28 February 2012; **Accepted:** 15 July 2012

ABSTRACT

In this paper, a new hybrid Particle Swarm Optimization (PSO) and Harmony Search (HS) algorithm, denoted by PSOHS is presented. This hybrid algorithm is designed to improve the efficiency of the PSO and remove some of the disadvantages which reduce the capability of the PSO. The main problem of the PSO is the lack of balance between exploration and exploitation of the algorithm. Another problem is how to handle the violating particles from feasible search space without reduction in the performance of the algorithm. The problem of unbalanced exploration and exploitation is solved using linear varying inertia weight. The second problem is solved in some other algorithms via reproduction of the violating particles using the HS algorithm. In this paper, these two approaches are combined to achieve a more efficient algorithm for engineering design problems. To show the higher capability of this approach compared to other works, several benchmark engineering examples, which have been considered previously and solved utilizing a variety of optimization algorithms, is solved by the present hybrid algorithm. Results illustrate a desirable performance of the *PSOHS* in both obtaining lower weight and having a higher convergence rate.

Keywords: Particle swarm optimization; harmony search algorithm; PSOHS; engineering optimization

1. INTRODUCTION

Nowadays, economy plays an important role in all aspects of human life. Since engineering projects are time, energy, and cost consuming. Economy has a great influence on them and optimization is an inevitable part of engineering design and practice. This is the why

[*]Email address of the corresponding author: alikaveh@iust.ac.ir (A. Kaveh)

optimization methods are growing with a high rate. An engineering project is considered valuable if it has two main features: to be "safe" and to be "optimal". The term of "safe" means that engineering projects must be consistent with their conditions, such as applied forces, temperature, chemical effects, quakes, and etc. with no problem for users. Also, the term "optimal" means that the project must be economically reasonable, and maximum efficiency with minimum possible costs should be achieved. Thus, the term optimization, engulf the whole aspects of the project and should be considered in three levels of the project: design, practice, and operation. The aim of this study is to present a new approach for optimization of engineering projects at the stage of designing. For the purpose of optimal design, there are two main methods: mathematical programming and meta-heuristic algorithms. Mathematical programming methods have many limitations. For example, initial values should be reasonable; search space should be continuous and etc. But meta-heuristics have not these limitations. Thus, meta-heuristic methods have become more popular and are used in optimization more than mathematical programming methods.

Particle Swarm Optimization *PSO* is a meta-heuristic optimization method which has been presented by Eberhart and Kennedy [1]. This algorithm is inspired from social interaction among animals and insects which are living in swarms and flocks having social behavior. In fact, in this algorithm each member (particle) of the society (swarm) tends to follow the member which has better position and this following causes search in space. PSO is hybridized with other meta-heuristic by Kaveh and Talatahari [2,3]. Some advantages of PSO consist of ease of implementation and directional search which results in popularity of the PSO. This algorithm has been extensively applied by researchers and has had different enhancements. But there is still an important problem which reduces the search capability and convergence rate of the PSO. The problem is how to handle the constraint violation from feasible search space.

Harmony Search (HS) algorithm is another meta-heuristic algorithm based on natural musical performance that occur when a musician searches for a better state of harmony, such as jazz improvisation. HS have been introduced by Geem, Kim and Loganathan [4] and works as: the engineers seek for a global optimum of an objective function, just like the musicians seek to find a musical pleasing harmony as determined by aesthetics [5]. This algorithm has been used by Kaveh and Talatahari [6] to handle the constraint violation in a hybrid algorithm from PSO, HS, and ant colony optimization ACO.

In this paper, a new hybrid algorithm consisting of an enhanced version of PSO and HS is presented. In fact, an enhanced PSO, which has a good performance, acts as the skeleton of the algorithm and HS is used to handle violating prticles from feasible search space to master the performance of the PSO. In order to prove the improvement of the PSO, some benchmark functions which have been considered with many different algorithms and are suitable for comparison, are selected and optimal design is performed utilizing the PSOHS. Results illustrate the success of this algorithm and better solutions are reached.

2. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a multi-agent meta-heuristic optimization algorithm which has been introduced by Eberhart and Kennedy [1]. This algorithm makes use of velocity vector to update the current position of each particle in the swarm. The velocity vector is updated utilizing a memory in which the best position of each particle and the best position among all particles are stored. This can be considered as an autobiographical memory. Therefore, the position of each particle in the swarm which adapts to its environment by flying in the direction of the best position of the entire particles and the best position of particle itself, provides the search of the PSO. The position of the *i*th particle at iteration *k+1* can be calculated using:

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i}.\Delta t \tag{1}$$

Where, x_{k+1}^{i} is the new position; x_{k}^{i} is the position at iteration *k*; v_{k+1}^{i} is the updated velocity vector of the *i*th particle; and Δt is the time step which is considered as unity. The velocity vector of each particle is determined using the following equation:

$$v_{k+1}^{i} = w.v_{k}^{i} + c_1.r_1.\frac{\left(p_{k}^{i} - x_{k}^{i}\right)}{\Delta t} + c_2.r_2\frac{\left(p_{k}^{g} - x_{k}^{i}\right)}{\Delta t} \tag{2}$$

Where, v_{k}^{i} is the velocity vector at iteration *k*; r_1 and r_2 are two random numbers between 0 and 1; p_{k}^{i} represents the best ever position of *i*th particle*, local best*; p_{g}^{k} is the *global best* position in the swarm up to iteration *k*; c_1 is the cognitive parameter; c_2 is the social parameter; and w is a constant named inertia weight.

With the above description, the PSO algorithm can be summarized as follow:

2.1 Initialization

Initial position, x_{0}^{i}, and velocities, v_{0}^{i}, of the particles are distributed randomly in the feasible search space.

$$x_{0}^{i} = x_{\min} + r.\left(x_{\max} - x_{\min}\right) \tag{3}$$

$$v_{0}^{i} = \frac{x_{\min} + r.\left(x_{\max} - x_{\min}\right)}{\Delta t} \tag{4}$$

Where, *r* is a random number uniformly distributed between 0 and 1; $x_{\min}$ and $x_{\max}$ are minimum and maximum possible variables for the *i*th particle, respectively.

2.2 Solution evaluation

Evaluate the objective function values for each particle, $f\left(x_{k}^{i}\right)$, using the design variables correspond to iteration k .

2.3 Updating memory
Update the local best of each particle, p_k^i, and the global best, p_g^k, at iteration k.

2.4 Updating positions
Update the position of each particle utilizing its previous position and the updated velocity vector as specified in Eqs. (1) and (2).

2.5 Stopping criteria
Repeat steps 2~4 until the stopping criteria is met.

3. HARMONY SEARCH ALGORITHM

Harmony Search (HS) algorithm is a meta-heuristic algorithm based on the natural musical performance that occurs when a musician searches for a better state harmony, such as jazz improvisation. This algorithm has been presented by Geem, Kim and Loganathan [4] and works as: the engineers seek for a global optimum of an objective function, just like the musicians seek to find a musical pleasing harmony as determined by aesthetics [5]. This seeking for a new improvised harmony is a search which if can be regulated in optimization it can find the global minimum of the objective function.

The HS algorithm includes a number of optimization operators, such as the harmony memory HM which is a memory that some best so far results are saved in it and if, in a stage, better solution is obtained, it is saved in the HM and the worst one is excluded from it; Harmony memory size HMS, which is the number of solution vectors saved in the HM; Harmony memory considering the rate HMCR 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 the pitch adjusting rate *PAR*. The pitch adjusting process is performed only after a value is chosen from HM and sets the rate of choosing a value from neighboring of the best vector. Steps of the HS are as follow:

A new harmony vector is improvised from the HM based on *HMCR* and *PAR*. With the probability of *HMCR,* the new vector is generated from HM and with the probability of *(1–HMCR)* the new vector is generated randomly from possible ranges of values. The pitch adjusting process is performed only after a value is selected from HM. The value *(1–PAR)* sets the rate of doing nothing. A *PAR* of *0.25* indicates that the algorithm will select a neighboring value with $0.25 \times HMCR$. It is recommended not to set *HMCR* as 1.0 because it is probable that the global minimum does not exist in the HM. With the aforementioned the search of the HM is summarized in Eq. (5), in which the term *"w.p."* represents *"with the probability"*.

If the generated harmony vector is better than a harmony vector in HM, judged in terms of the objective function value, the new harmony is included in the HM and the worst one is excluded from it.

$$x_{i,j} = \begin{cases} \textit{w.p. HMCR} & \\ & \Rightarrow \textit{ select a value for a variable from HM} \\ & \Rightarrow \textit{w.p. (1}-\textit{PAR) do nothing} \\ & \Rightarrow \textit{w.p. PAR choose a neighboring value} \\ \textit{w.p.} & \\ \textit{(1}-\textit{HMCR)} & \Rightarrow \textit{select a new variable randomly} \end{cases} \tag{5}$$

4. HYBRID PARTICLE SWARM OPTIMIZATION AND HARMONY SEARCH ALGORITHM

In this section, the hybrid PSO and HS is presented. For this purpose, it is necessary to explain why this modification is performed. There are two main problems in the PSO: first, the lack of balance between exploration and exploitation; second, there is no efficient approach to control the violating particles. For definition of the first problem it should be mentioned that in meta-heuristic optimization algorithms, there should be a balance between exploration and exploitation in a way that at initial iterations, the algorithm should perform a global search and this search should cover the whole search space in a suitable manner. In this stage, some points which are expected to be near the global minimum of the cost function are found. Then at the later iterations, the algorithm should perform a local search using the solution vectors found so far. Form Eq. (2) it can be seen that the velocity vector definition of the PSO which is the search engine of the algorithm has not this specification and at early iterations it is the same as in the subsequent iterations and this issue causes the lack of balance between exploration and exploitation of the PSO.

However, this problem has been solved using dynamic variation of inertia weight by linearly decreasing the value of w in each iteration of the algorithm presented by Shi and Eberhart [7] as

$$w_{k+1} = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k \tag{6}$$

Where, $w_{\max}$ is the maximum considered inertia weight, $w_{\min}$ is the minimum considered inertia weight, and $k_{\max}$ is the number of iterations.

In this paper, the linearly varying inertia weight is changed slightly and is defined as:

$$w_{k+1} = rand \times \left(w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k \right) \tag{7}$$

The multiplier *rand* results in a fast convergence because it prevents the particles to be dispersed when they are near the global minimum at initial iterations and the velocity of particles are high due to global search.

Utilizing Eq. (6) or Eq. (7), at initial iterations there will be a large value of inertia weight providing a global search and by progression of the algorithm, this value will reduce until at the latest iterations there will be only local search based on position of the best particle and the best ever position of particles as seen in Eq. (2).

The second problem with the PSO is that like many other optimization algorithms, the method for controlling the violation of constraints. One of the simplest approaches is utilizing the nearest limit values for the violating particle. Alternatively, one can force the violating particle to return to its previous position, or reduce the maximum value of the velocity to allow fewer particles to violate the variables in the boundaries. Although these approaches are simple, they are not efficient enough and may lead to the reduction of the exploration of the search space. This problem has previously been addressed and solved using the harmony search based handling approach [4]. According to this mechanism, any component of the solution vector violating the variable boundaries can be regenerated from the HM by using Eq. (5). This approach is an efficient one which improves the convergence rate of the algorithm because of simultaneous action of the two algorithms. If the particle is in the feasible search space, the PSO will work and if it violates the boundaries, HS will be activated. However, in the PSOHS it is necessary that the memory in which the global best is stored be extended and some of the best designed vectors stored. This memory can be utilized as the HM when a particle violates and the HS should become active.

With the above mentioned explanation, the steps of the PSOHS are shown in the flowchart of Figure 1.

In this paper, the linear varying inertia weight is changed slightly and is defined as follow:

$$w_{k+1} = rand \times \left(w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k \right) \tag{8}$$

The multiplier *rand* results in a fast convergence because it prevents the particles to be dispersed when they are near the global minimum at initial iterations and the velocity of particles are high due to global search.

5. NUMERICAL EXAMPLES

To show the higher performance of the PSOHS compared to the standard PSO and other algorithms, the present algorithm is applied to optimal design of five benchmark examples. These examples have been implemented using a variety of optimization algorithms and are suitable for comparison of new algorithms with the existing ones. For each example, 20 independent runs are performed and the number of particles is assumed to be 20. Then the best, the mean and the worst result along with the standard deviation of each example are obtained and compared to those of the other methods. Also, the number of iterations, as a criterion of the performance of algorithm, is compared to those of the other methods for some examples.

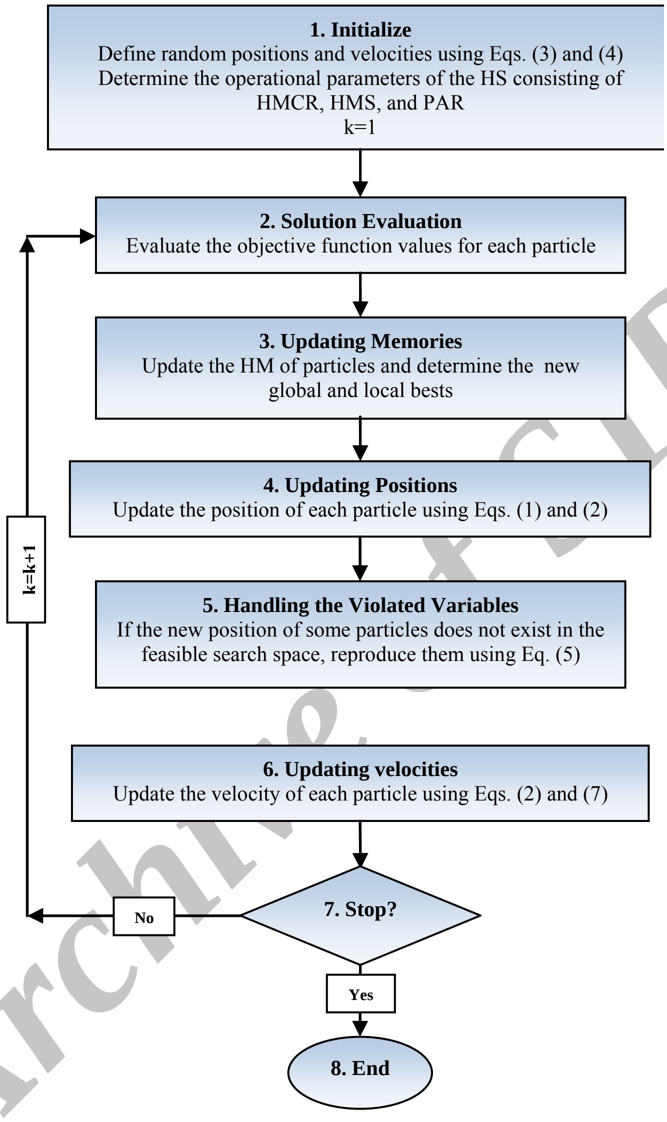

Figure 1. Flowchart of the PSOHS

5.1 A tension/compression spring design problem

This problem is optimized by Belegundu [7] and Arora [8]. It consists of minimizing the weight of a tension/compression spring subjected to constraints on shear stress, surge frequency, and minimum deflection, as shown in Figure 2.

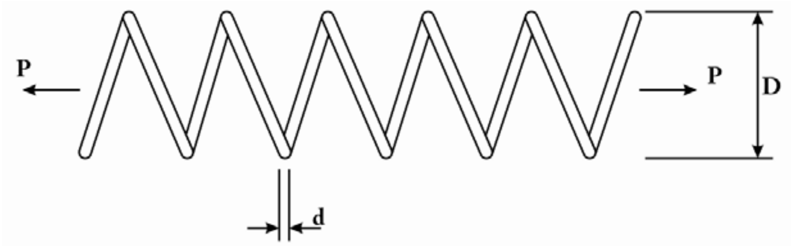

Figure 2. Geometry and the parameters of the tension/compression spring design

The design variables are the mean coil diameter $D(=x_1)$, the wire diameter $d(=x_2)$, and the number of active coils $N(=x_3)$. The problem can be stated with the cost function

$$f_{\cos t}(X)=(x_3+2)x_2x_1^2, \tag{9}$$

to be minimized at the presence of the following constraints:

$$\begin{aligned} g_1(X)&=1-\frac{x_2^3x_3}{71{,}785x_1^4}\le 0,\\ g_2(X)&=\frac{4x_2^2-x_1x_2}{12{,}566\left(x_2x_1^3-x_1^4\right)}+\frac{1}{5{,}108x_1^2}-1\le 0,\\ g_3(X)&=1-\frac{140.45x_1}{x_2^2x_3}\le 0,\\ g_4(X)&=\frac{x_1+x_2}{1.5}-1\le 0. \end{aligned} \tag{10}$$

The variables are selected from the following regions:

$$\begin{aligned} 0.1&\le x_1\le 2,\\ 0.25&\le x_2\le 1.3,\\ 2&\le x_3\le 15. \end{aligned} \tag{11}$$

This problem has been solved by Belegundu [7] using eight different mathematical optimization techniques (only the best results are shown). Arora [8] has also solved this problem using a numerical optimization technique called constraint correction at the constant cost. Coello [9] as well as Coello and Montes [10] solved this problem employing a GA-based method. Additionally, He and Wang [11] utilized a co-evolutionary particle swarm optimization (CPSO). Recently, Montes and Coello [12] and Coelho [13] used evolution strategies to solve this problem. Table 1 presents the best solution of this problem obtained using the PSOHS algorithm and compares the PSOHS results to the solutions reported by other researchers. From Table 1, it can be seen that the best feasible solution obtained by the PSOHS is better than those of the previously reported ones and it is equal to that of the Coelho [13].

Table 1: Optimal result of the tension/compression spring design

Author or method	x_1 (d)	x_2 (D)	x_3 (N)	Best result	Mean of results	Worst results	SD
Belegudu [7]	0.050000	0.315900	14.250000	0.012833	N/A	N/A	N/A
Arora [8]	0.053396	0.399180	9.185400	0.012730	N/A	N/A	N/A
Coello [9]	0.051480	0.351661	11.632201	0.012705	0.012769	0.012822	3.939E-05
Coello and Montes [10]	0.051989	0.363965	10.890522	0.012681	0.012742	0.012973	5.900E-05
He and Wang [11]	0.051728	0.357644	11.244543	0.012675	0.012730	0.012924	5.199E-05
Montes and Coello [12]	0.051643	0.355360	11.397926	0.012698	0.013461	0.164850	9.660E-04
Coelho [13]	0.051515	0.352529	11.538862	0.012665	0.013524	0.017759	1.268E-3
PSOHS (present work)	0.051625	0.355176	11.379955	0.012665	0.013550	0.017398	1.3864E-3

5.2. A pressure vessel design problem

A cylindrical vessel clapped at both ends by semispherical heads as shown in Figure 3 is considered as the second design example. The objective is to minimize the total cost, including the cost of material, forming and welding [14]:

$$f_{\cos t}(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \quad (11)$$

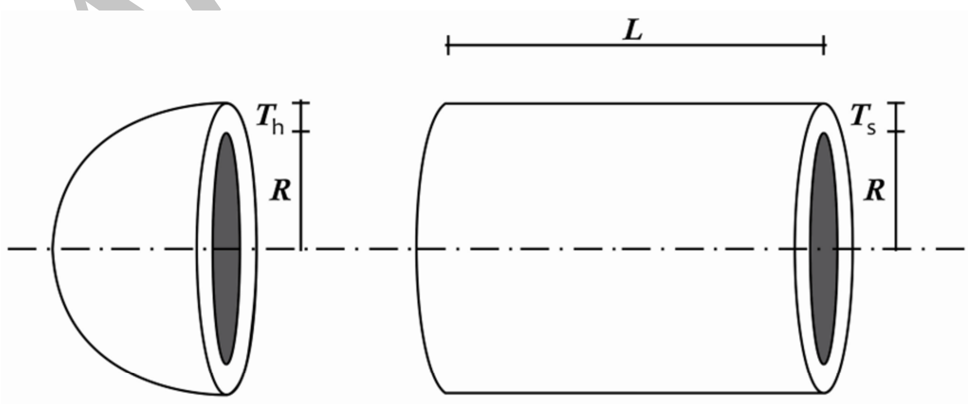

Figure 3. Geometry and parameters of the pressure vessel

Where x_1 is the thickness of the shell (T_s); x_2 is the thickness of the head (T_h), x_3 is the inner radius (R), and x_4 is the length of cylindrical section of the vessel (L), not including the head. T_s and T_h are integer multiples of 0.0625 inch, the available thickness of the rolled steel plates, R and L are continuous.

The constraints can be expressed as:

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(X) &= -x_2 + 0.00954x_3 \leq 0, \\ g_3(X) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \leq 0, \\ g_4(X) &= x_4 - 240 \leq 0. \end{aligned} \tag{12}$$

The design space is specified as:

$$\begin{aligned} 0 &\leq x_1 \leq 99, \\ 0 &\leq x_2 \leq 99, \\ 10 &\leq x_3 \leq 200, \\ 10 &\leq x_4 \leq 200. \end{aligned} \tag{13}$$

The approaches applied to this problem include a branch and bound technique [14], an augmented Lagrangian multiplier approach [15], genetic adaptive search [16], a GA-based co-evolution model [13], a feasibility-based tournament selection scheme [14], a co-evolutionary particle swarm optimization [15], an evolution strategy [16] and a Gaussian quantum-behaved PSO approach [17]. The best solutions obtained by the above mentioned approaches and their statistical simulation results are listed in Table 2. From Table 2, it can be seen that the best solution found by PSOHS is 2.50% less than the best solution among other methods. Also, from this table, it can be seen that the standard deviation quantity of the PSOHS is less than those of other methods.

5.3 A 25-bar element space truss

As the third example, a 25-bar space truss as transmission tower is considered as described by Schmit and Fleury [17] and shown in Figure 4. The design variables are the cross-sectional areas of the members, which are categorized into eight groups as shown in Table 3. Loading of the structure is shown in Table 4. Constraints are imposed to cross sectional areas of the members between 0.01 in^2 to 3.4 in^2, and to the allowable stresses which are included in Table 5. Another considered constraint is the allowable displacement which is taken as ± 0.35 in for every direction.

Table 2: Results of the optimal design of the pressure vessel

Author or Method	x_1 (T_s)	x_2 (T_h)	X_3 (R)	x_4 (L)	Best result	Mean of results	Worst result	SD
Sandgren [14]	1.125	0.625	47.7	117.701	8,129.10	N/A	N/A	N/A
Kannan and Kramer [15]	1.125	0.625	58.291	43.69	7,198.04	N/A	N/A	N/A
Deb and Gene [16]	0.9375	0.5	48.329	112.679	6,410.38	N/A	N/A	N/A
Coello [9]	0.8125	0.4375	40.3239	200	6,288.74	6,293.84	6,308.15	7.4133
Coello and Montes [10]	0.8125	0.4375	42.09739	176.654	6,059.95	6,177.25	6,469.32	130.9297
He and Wang [11]	0.8125	0.4375	42.09126	176.7465	6,061.08	6,147.13	6,363.80	86.4545
Montes and Coello [12]	0.8125	0.4375	42.09808	176.6405	6,059.75	6,850.00	7,332.88	426
Coelho [13]	0.8125	0.4375	42.0984	176.6372	6,059.72	6,440.38	7,544.49	448.4711
PSOHS (present work)	0.7943	0.3890	41.1578	188.65810	5,902.67	6,594.868	7264.889	424.4998

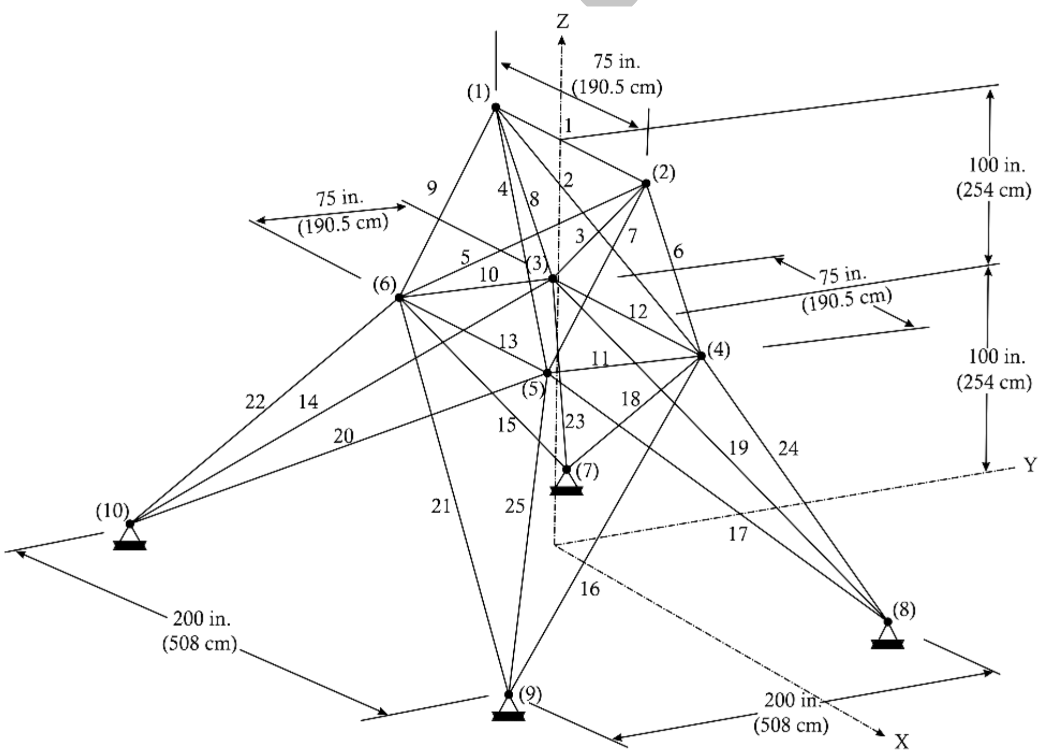

Figure 4. Geometry and element grouping of the 25-bar element space truss

Table 3: Truss member grouping of the 25-bar space truss members

Group	Truss members
1	1
2	2~5
3	6~9
4	10~11
5	12~13
6	14~17
7	18~21
8	22~25

Table 4: Nodal load of the 25-bar space truss

Node	F_x (lb)	F_y (lb)	F_z (lb)
1	10,000	−10,000	−10.000
2	0	−10,000	−10.000
3	500	0	0
6	600	0	0

Table 5: Allowable stresses for the 25-bar space truss members

Element group	Allowable compressive Stress ksi (MPa)	Allowable tensile stress ksi (MPa)
1	35.092 (241.96)	40.0 (275.80)
2	11.590 (79.913)	40.0 (275.80)
3	17.305 (119.31)	40.0 (275.80)
4	35.092 (241.96)	40.0 (275.80)
5	35.092 (241.96)	40.0 (275.80)
6	6.759 (46.603)	40.0 (275.80)
7	6.959 (47.982)	40.0 (275.80)
8	11.082 (76.410)	40.0 (275.80)

Table 6 shows the history of the solution of this example and also the result of the PSOHS is included. From this table it can be seen that PSOHS has also a better solution for this example. Thus, we can say PSOHS is more reliable and in a few runs we can be sure that a good optimum weight is obtained. The progression curves of PSOHS and some other works are shown in Figure 5 and Figure 6, respectively. In these figures it can be seen that the convergence rate of the PSOHS is significantly higher than that of PSO and PSOPC. As a comparison, in Figure 5 it is shown that the PSOHS is converged in 300 iterations where this number for the PSO and PSOPC is 3000. Also, a detailed result which is obtainable from Figures 5 and 6 is the search trend of the algorithms. In the curve resulted from PSOHS, a global search at initial iterations is obvious. This trend is performed by the algorithm when there is no information about the minimum cost function and in the latest iteration when the particles travel to near the global optimum, then the local search is performed. On the contrary, PSO and PSOPC algorithms have not efficient phase alternation between the exploration and exploitation. This is due to the following two reasons:

1. There is no difference in velocity definition of the PSO and PSOPC as the search engine of these algorithms in initial, median and latest iterations. Thus, in these algorithms only global search is performed and the local search is not done, and when the particles are near the minimum of cost function, due to the large value of the velocity, they get far away in the next iteration.
2. In most of the structural design optimization problems, the real minimum of the cost function is located out of the feasible space. For example, when the cost function is the sum of the element weights, when cross-sectional area of some elements become negative in value, penalty function, not only does not force the particles to return to the feasible space, but also motivates them to get far from the possible boundaries. In PSO and PSOPC, when the particles violate from feasible search space, they are returned to the previous location or reproduced randomly. This problem is more significant when global search is being performed because of searching many locations.

Both of these problems are solved in this work because of the following reasons:

1. The first problem is solved by the use of the linearly varying inertia weight. At initial iterations, a large value of velocity covers the search space. As iterations are continued, the particles get nearer to optimum location. Thus, proportional to the rate of algorithm, it is necessary to reduce the value of the velocity and perform the local search. Fortunately, in velocity definition of the PSO, the term $c_1.r_1.\frac{\left(p_k^i - x_k^i\right)}{\Delta t} + c_2.r_2\frac{\left(p_k^g - x_k^i\right)}{\Delta t}$ can do the local search and a separate velocity definition for local search is not requisite. Therefore, with linearly varying inertia weight, the PSO has a progressive phase transformation which alters global search to local search with progression of algorithm and provides a continuous stream for the optimization process.
2. The second problem is solved by the use of the HS. As mentioned previously, the probability of violation from feasible search space is significant at initial stages of the algorithm. When HS is used, two benefits are obtained. Firstly, the HS saves the progression of algorithm and the search is not returned or stopped using pre-mentioned methods of handling the violating particles. Secondly, the HS has a unique search in which the new vectors are generated from the best vectors and their neighbors from the HM. When HS is applied for handling the violating particles,

new particle is generated among best particles and their neighbors. Thus, it is more probable that the new variable be nearer to the optimum location.

Table 6: Results of the optimal design of the 25-bar space truss structure

Element Group	Zhou and Rozvani [18]	Haftka and Grdal [19]	Erbatur [20]	Zhu [21]	Wu and Chow [22]	Prez and Behdinan [23]	PSOHS (present work)
1	0.01	0.01	0.1	0.1	0.1	0.1	0.010
2	1.987	1.987	1.2	1.9	0.5	1.0227	0.393
3	2.994	2.991	3.2	2.6	3.4	3.4	3.389
4	0.01	0.01	0.1	0.1	0.1	0.1	0.01
5	0.01	0.012	1.1	0.1	1.5	0.1	1.992
6	0.684	0.683	0.9	0.8	0.9	0.6399	0.978
7	1.677	1.679	0.4	2.1	0.6	2.0424	0.479
8	2.662	2.664	3.4	2.6	3.4	3.4	3.399
Best Result (kips)	545.16	545.22	493.8	562.93	486.29	485.33	482.46
Mean of Results (kips)	N/A	N/A	N/A	N/A	N/A	N/A	487.7325
Worst Result (kips)	N/A	N/A	N/A	N/A	N/A	534.84	493.5095
SD	N/A	N/A	N/A	N/A	N/A	N/A	0.60718

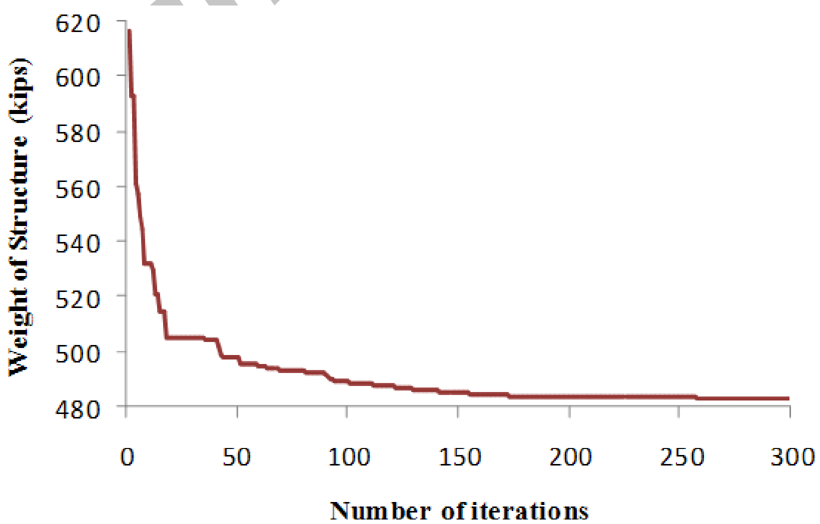

Figure 5. Convergence history of the PSOHS for the 25- bar space truss

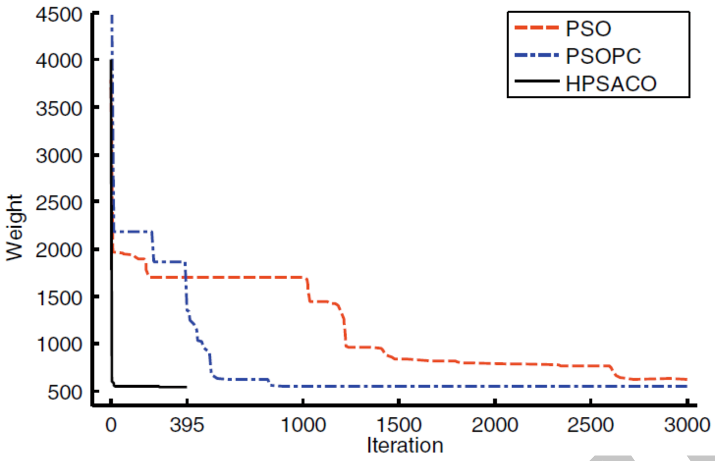

Figure 6. Convergence history of the 25-bar space truss [6]

5.4 120-bar dome space truss

Design of a 120-bar dome truss, shown in Figure 7, is considered as the next example in order to compare the practical capability of the PSOHS algorithm with those of the PSO and PSOPC. This dome is utilized in literature to find size optimum design. The modulus of elasticity is 210,000 MPa (30,450 ksi), and the material density is 7971.810 kg/m^3 (0.288 lb/in^3). The yield stress of steel is taken as 400 MPa (58.0 ksi). The dome is considered to be subjected to vertical loading at all the unsupported joints. These loads are taken as −60 kN (−13.49 kips) at node 1, −30 kN (−6.744 kips) at nodes 2 through 14, and −10 kN (−2.248 kips) at the rest of the nodes. The minimum cross sectional area of all members is 2 cm^2 (0.775 in^2) and the maximum cross-sectional area is taken as 129.03 cm^2 (20.0 in^2). The stress constraints of the structural members are calculated according to the AISC 1989 specifications as illustrated in Eq. (14). The 120 bar truss members are categorized into 7 groups as shown in Figure 7.

$$\begin{cases} \sigma_i^+ = 0.6F_y & for \quad \sigma_i \geq 0 \\ \sigma_i^- & for \quad \sigma_i < 0 \end{cases} \tag{14}$$

Where, σ_i^- is calculated according to the slenderness ratio using:

$$\sigma_i^- = \begin{cases} \left[\left(1 - \dfrac{\lambda_i^2}{2C_c^2}\right)F_y\right] \Big/ \left(\dfrac{5}{3} + \dfrac{3\lambda_i}{8C_c} - \dfrac{\lambda_i^3}{8C_c^3}\right) & for \quad \lambda_i < C_c \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & for \quad \lambda_i \geq C_c \end{cases} \tag{15}$$

Where, E is the modulus of elasticity; F_y is the yield strength of steel; C_c is the slenderness ratio which divides the elastic and inelastic buckling regions $\left(C_c = \sqrt{2\pi^2 E / F_y}\right)$; and λ_i is the slenderness ratio.

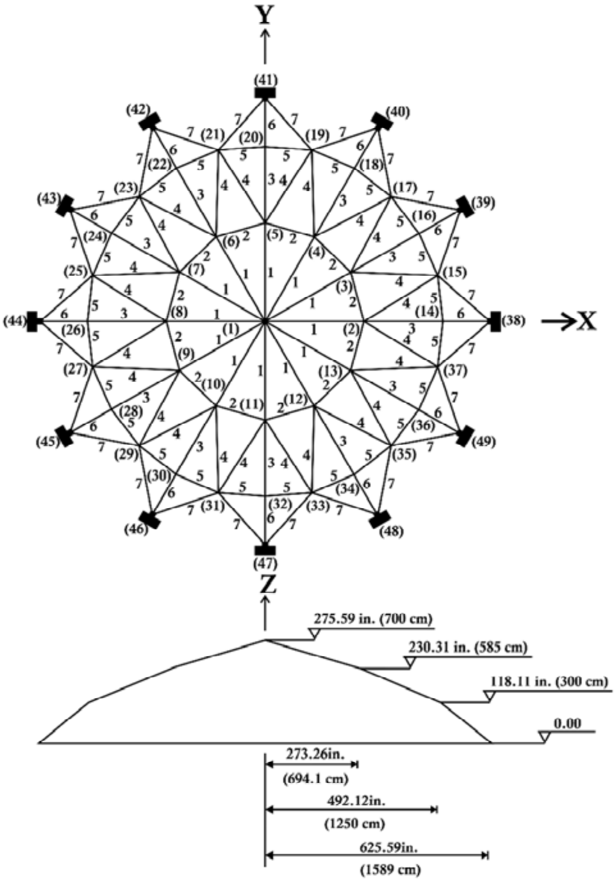

Figure 7. Geometry and element grouping of the 120-bar dome space truss

Figure 8 shows the convergence history of the PSOHS. In Figure 9, the PSO, PSOPC and HPSACO convergence curves are shown. In these figures, the convergence rate of the PSOHS is significantly higher than those of the PSO and PSOPC. In Figure 8 it can be shown that fewer than 300 iterations are sufficient for the PSOHS to converge, while nearly 3000 iterations are needed for the PSO and PSOPC for convergence. Results of different methods are included in Table 7 for comparison. From this table, similar to other examples, results obtained from PSOHS the are significantly better in weight. These results also indicate the improvement of the PSO when it is hybridized with HS. From Table 7, the HPSACO results are better than other methods for this dome design problem, and from convergence curve of this algorithm it can be concluded that fewer than 400 iterations are needed for convergence of the algorithm (see Figure 9). However, the weight obtained from PSOHS is less than that of the HPSACO. Thus, PSOHS performs well in both convergence and finding optimum design while HPSACO only has fast convergence.

Figure 10 shows the number of violated variables vs. number of iteration. In this figure, it can be seen that at initial iterations nearly half of the particles violate the feasibility boundaries and this amount decreases as the global search gets altered to the local search. Also, in this figure, it can be observed that in the 120-bar truss spatial dome problem, fewer particles tend to violate the boundaries compared to 25-bar truss structure. This reason becomes obvious from the obtained results. In the 25-bar truss structure, some members need the minimum amount of cross-sectional area and when the local search is done near the boundaries, some particles violate from search space, however, in the 120-bar truss dome problem, the variables are not on the boundaries. Thus fewer particles violate from the search space and the PSO have fewer halts.

Table 3: Results of the optimal design of the 120-bar truss dome

Element group	Lee and Geem [4]			Kaveh and Talatahari [6]	Present work
	HS	PSO	PSOPC	HPSACO	HCSSPSO
1	3.295	3.147	3.235	3.311	3.037
2	3.396	6.376	3.37	3.438	3.867
3	3.874	5.957	4.116	4.147	3.241
4	2.571	4.806	2.784	2.831	2.246
5	1.15	0.775	0.777	0.775	1.637
6	3.331	13.798	3.343	3.474	2.492
7	2.784	2.452	2.454	2.551	2.301
Best result (kips)	19707.77	32432.9	19618.7	19491.3	18292.8
Mean of results (kips)	N/A	N/A	N/A	N/A	18377.6
Worst result (kips)	N/A	N/A	N/A	N/A	18489.5
SD	N/A	N/A	N/A	N/A	176.525

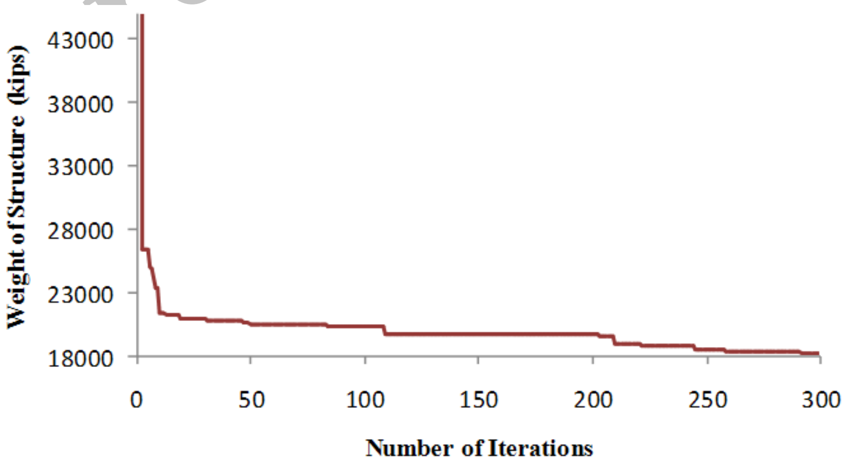

Figure 8. Convergence history of the PSOHS for the 120- bar dome space truss dome

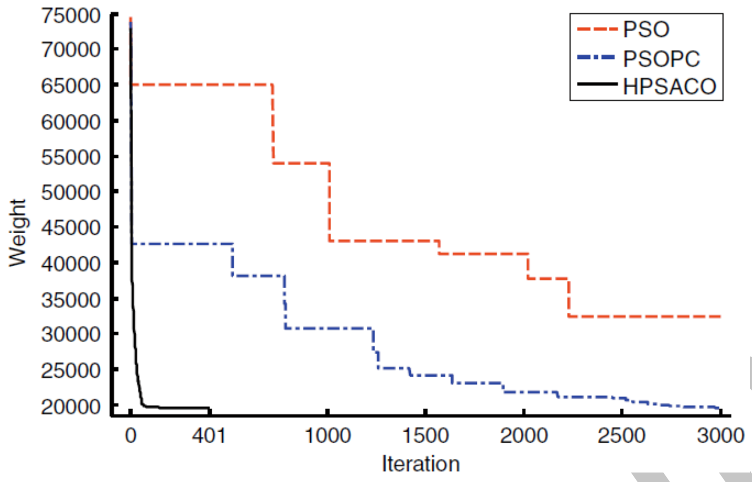

Figure 9. Convergence history of the 120-bar dome space truss [6]

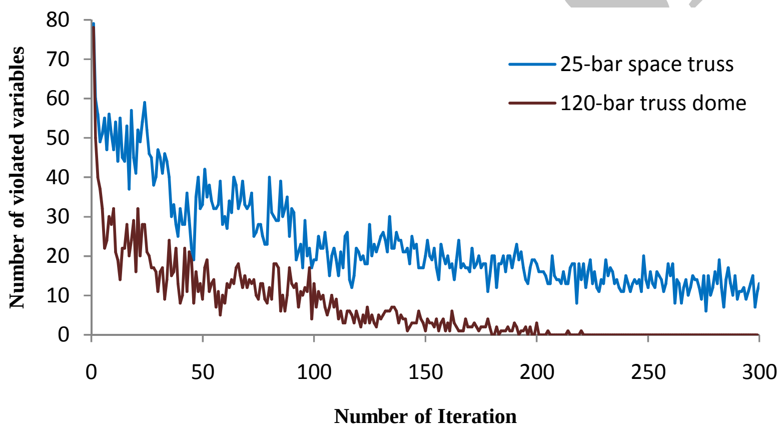

Figure 10. Number of violated variables vs. number of iteration for the 25-bar space truss problem and the 120-bar truss dome

5.5 A 10-story, 3-bay steel moment frame

The last example considered in this paper is a 10-story and 3-bay steel moment frame which is shown in Figure 11. The loading of the structure and element groups are depicted in this figure. For this example, the modulus of elasticity and yield stress of steel are assumed as 210 *GPa* and 230 *MPa*, respectively. The elements are selected from W-sections and moment of inertia of the elements and the elastic section modulus are considered based on the following relationships:

$$I = \alpha_1 A^{\beta_1} \tag{16}$$

$$S = \alpha_2 A^{\beta_2} \tag{17}$$

Where $\alpha_1 = 1.3162$, $\beta_1 = 2.165$, $\alpha_2 = 1.016$, and $\beta_2 = 1.571$.

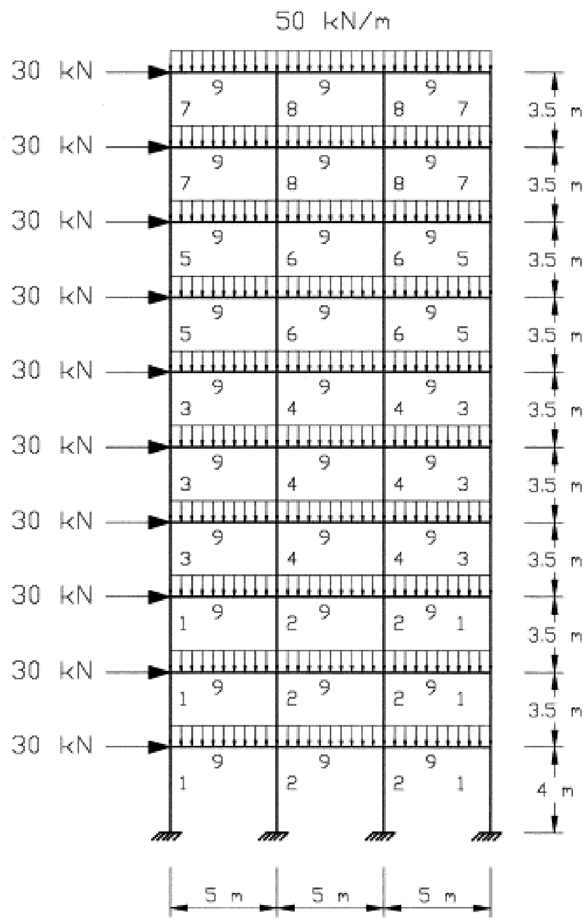

Figure 11. Geometry, element grouping, and loading of the 10-story and 3-bay steel moment frame problem

Table 8 shows the present and a previous solution of this example. From this table, it can be concluded that the PSOHS has the best solution compared to the other considered optimization methods in the obtained weight and standard deviation. The convergence history of the PSO, PSO with linearly varying inertia weight (LPSO) and PSOHS for the best solution and average of 20 independent runs are shown in Figure 12 and Figure 13, respectively. These figures show that the PSOHS has the fastest convergence rate because of an adjusted exploration and exploitation. From Figure 13, it can be noted that the average of 20 independents runs of the PSO is larger than other two methods, although it has a solution near to the LPSO and PSOHS. This figure and the standard deviations in the Table 8 show that the PSO has a disperse trend in different runs and is not a reliable algorithm. For further explanation, Figure 14 is presented which shows that the obtained weight vs. different runs. From this figure, it can be seen that the PSO provides a design result larger than LPSO and PSOHS and only in one run it has a design result near the others; but other methods have less fluctuation in different runs.

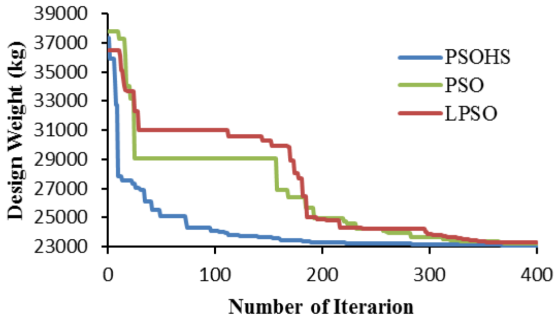

Figure 12. Convergence of algorithms in design of 10-story and 3-bay steel moment frame

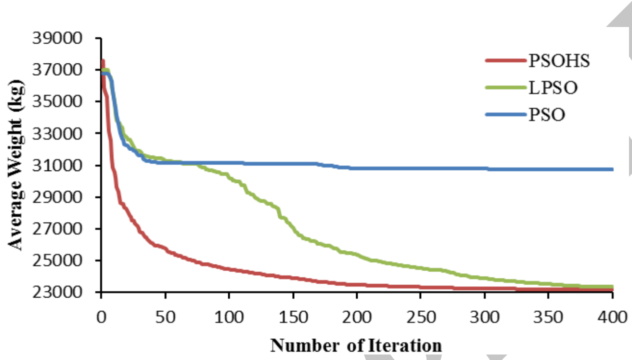

Figure 13. Convergence of algorithms in design of the 10-story and 3-bay steel moment frame in 20 independent runs

Table 4: Result of the design optimization of the 10-story, 3-bay steel moment frame

Design output	Saka & Kameshki [22]	PSO	LPSO	PSOHS
A1 (cm^2)	176.30	85.03	82.44	80.00
A2 (cm^2)	288.20	128.45	125.38	122.87
A3 (cm^2)	125.10	53.38	62.46	57.43
A4 (cm^2)	176.60	90.93	81.97	85.68
A5 (cm^2)	84.77	50.42	52.87	51.32
A6 (cm^2)	111.50	55.32	55.91	54.32
A7 (cm^2)	57.77	50.00	50.13	50.02
A8 (cm^2)	61.77	50.85	50.70	50.01
A9 (cm^2)	110.10	127.21	126.91	127.40
Best Result (kg)	29430.00	23272.58	23184.60	23095.20
Average of Results (kg)	N/A	30741.03	23338.60	23190.19
Worst Result (kg)	N/A	34284.83	23974.21	23326.23
Standard Deviation	N/A	2147.24	179.56	69.65

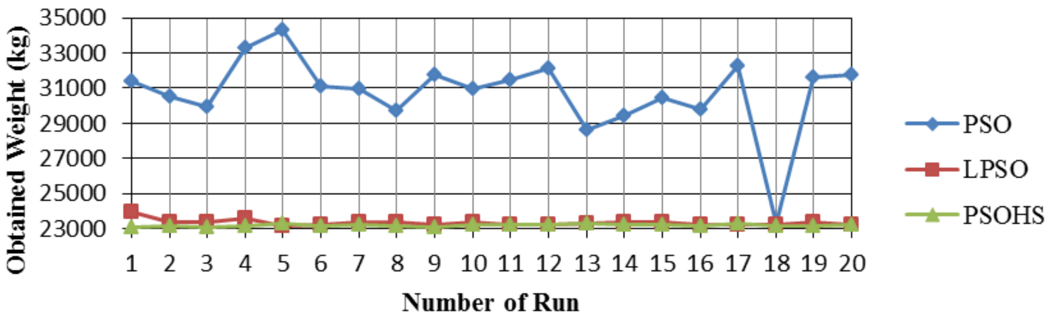

Figure 14. Design weight in different runs of the PSO, LPSO, and PSOHS

From Table 8, it is infermed that the differences of the ontained weight in different rus for LPSO are larger than those of the PSOHS. To portrait it more visible, Figure 15 is presented which shows that the differential of obtained weight from run of the LPSO and PSOHS. From this figure, it can be concluded that the PSOHS has less variation in different runs and thus it can provide a reliable designed frame in fewer runs.

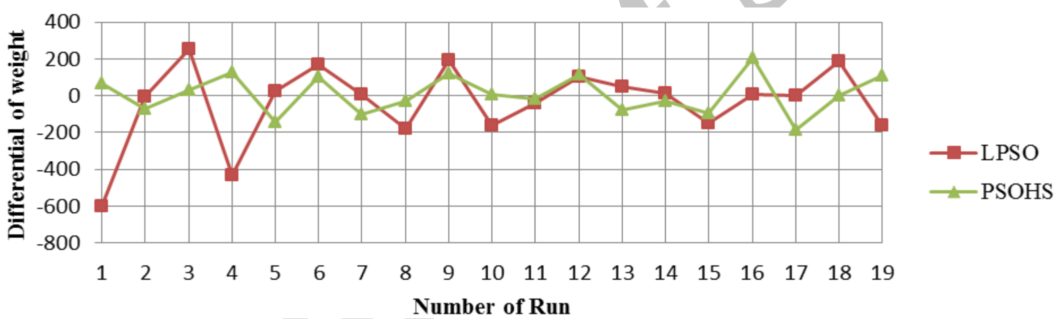

Figure 15. Differential of Weight vs. Number of runs for the LPSO and PSOHS in design optimization of the 10-story and 3-bay steel moment frame

6. CONCLUSIONS

In this paper, a new hybrid PSO and HS optimization algorithm is presented. PSO has some advantages consisting of having fewer parameters and its implementation is simple. Since velocity definition of the PSO is the search engine of this algorithm, constant parameters in velocity, provide a constant search phase. Therefore, PSO performs well only in global search. The most effective parameter in velocity of particles in the PSO is the inertia weight which controls the amount of global and local search in this algorithm. Thus, fixed inertia weight results in a fixed trend of search. The PSO with linear varying inertia weight can provide an efficient balance between global search and local search.

On the other hand, one of the most important problems in many meta-heuristic algorithms is the way of handling the violating particles. Since, in most of the engineering design optimization problems, the real minimum of objective function is located out of the feasible search space, thus it is probable that some particles defined in the PSO, violate from feasible boundaries. This problem has been solved previously by utilizing the HS. By this method of handling the violating particles, the algorithm has a continuous search.

In this paper, PSO with linear varying inertia weight has been hybridized with the HS. With this hybridized algorithm, all of the above mentioned problems are solved. The hybrid PSO and HS algorithm, *PSOHS*, have a good balance between exploration and exploitation, and violating particles are reproduced from a memory in which some of the so far best design variables are saved. To show the effectiveness of this way of hybridizing, several benchmark structures are considered and designed using this new approach. Results show a significant improvement in the designed weight of structure and the convergence rate.

REFERENCES

1. Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan 1995.
2. Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables, *Journal of Constructional Steel Research*, Nos (8-9), **65**(2009) 1558–68.
3. Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures *Asian Journal of Civil Engineering*, No. 4, **9**(2008) 325–44.
4. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search; *Simulation*, **76**(2001) 60–8.
5. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, **82**(2004) 781–98.
6. Kaveh A, Talatahari S. Particle swarm optimizer ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures*, **87**(2009) 267–83.
7. Belegundu AD. A study of mathematical programming methods for structural optimization. PhD thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa 1982.
8. Arora J S. *Introduction to Optimum Design*, McGraw-Hill, New York, NY 1989.
9. Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry,* **41**(2000) 113–27.
10. Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics* **16**(2002) 193–203.
11. He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence,* **20**(2007) 89–99.
12. Montes EM, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems,* No. 4, **37**(2008) 443–73.
13. Coelho LS. Gaussian quantum-behaved particle swarm optimization approaches for

constrained engineering design problems. *Expert Systems with Applications,* **37**(2010) 1676−83.

14. Sandgren E. Nonlinear integer and discrete programming in mechanical design. In *Proceedings of the ASME Design Technology Conference*, Kissimine, FL, 1988, pp. 95−105.
15. Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Transaction of the ASME, Journal of Machine Design*, **116**(1994) 318–20.
16. Deb K, Gene AS. A robust optimal design technique for mechanical component design. In Dasgupta D, Michalewicz Z. (eds.) Evolutionary Algorithms in Engineering Applications. Berlin: Springer 1997, pp. 497−514.
17. Schmit L, Fleury C. Discrete-continuous variable structural synthesis using dual methods. *AIAA Journal*, **18**(1980) 1515−24.
18. Zhou M, Rozvany G. An optimality criteria method for large systems. Part ii: Algorithm. *Structural and Multidisciplinary Optimization,* **6**(1993) 250−62.
19. Haftka R, Grdal Z. *Elements of Structural Optimization*. 3rd ed. Dordrecht: Kluwer Academic Publishers 1992.
20. Erbatur F, Hasantebi O, Tntnncn I, Kilt H. Optimal design of planar and space structures with genetic algorithms. *Computers and Structures*, **75**(2000) 209−24.
21. Zhu D. An improved templemans algorithm for optimum design of trusses with discrete member size. *Engineering Optimization*, **9**(1986) 303−12.
22. Wu S, Chow P. Steady-state genetic algorithm for discrete optimization of trusses. *Computers and Structures*, No. 6, 5**6**(1971) 979−91.
23. Perez RE, Behdinan K. Particle swarm approach for structural design optimization. *Computers and Structures*, Nos. (19-20), **85**(2007) 1579−88.
24. Saka MP, Kameshki ES. Optimum design of unbraced rigid frames. *Computers and Structures,* **69**(1998) 433–42.