Scientia Iranica A (2014) 21(2), 295–305

SCIENTIA
IRANICA

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
www.scientiairanica.com

Charged system search and particle swarm optimization hybridized for optimal design of engineering structures

A. Kaveh* **and A. Nasrollahi**

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Tehran, P.O. Box 16846-13114, Iran.

Received 25 February 2013; received in revised form 4 September 2013; accepted 24 September 2013

KEYWORDS
Hybrid metaheuristic algorithm;
Charged system search;
Particle swarm optimization;
Optimal design;
Engineering structures.

Abstract. In this paper, a new Hybrid Charged System Search and Particle Swarm Optimization, HCSSPSO, is presented. Although Particle Swarm Optimization (PSO) has many advantages, including directional search, it has also some disadvantages resulting in slow convergence rate and low performance. On the other hand, the Charged System Search (CSS) is a robust optimization algorithm which has been successfully utilized in many structural optimization problems. In this study, the goal is to incorporate the positive features of the PSO in CSS and make it more capable of solving optimization problems. The hybrid CSS and PSO is named HCSSPRO, and it uses the positive features of the PSO to further improve the CSS. In order to show the higher performance of the HCSSPSO, it is implemented and applied to some engineering problems. These structures are benchmark examples which are optimized by many other methods and are suitable for comparison. Results of the present algorithm show its better performance and higher convergence rate for the problem studied.

1. Introduction

In recent years, many meta-heuristic optimization algorithms have been developed and employed in the optimal design of engineering structures. Nowadays, there is considerable effort made towards designing engineering structures and machines using the least possible amount of materials and resources. For this purpose, many mathematical programming and meta-heuristic methods of optimization based on natural events, the social behavior of animals and insects, and physics laws, are applied to engineering design. Due to the simplicity and versatility of the meta-heuristic methods compared to mathematical programming approaches, these algorithms have become the focus of attention among many researchers.

In general, gradient-based methods converge faster and can obtain solutions with higher accuracy compared to stochastic approaches in fulfilling the local search task. However, for effective implementation of these methods, the variables and cost function of the generators should be continuous. Also, a good starting point is vital for these methods to be executed successfully. In many optimization problems, prohibited zones, side limits, and non-smooth or non-convex cost functions need to be considered. As a result, non-convex optimization problems cannot be solved by the traditional mathematical programming methods. Although dynamic programming or mixed integer non-linear programming and their modifications offer some facility in solving non-convex problems, these methods, in general, require considerable computational effort.

Particle Swarm Optimization is a multi-agent meta-heuristic optimization algorithm which was introduced by Eberhart and Kennedy [1]. This algorithm is inspired by social interaction among animals and

*. *Corresponding author. Tel.: +98 21 77240104;*
Fax: +98 21 77240398
E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

insects that are living in swarms and flocks, having social behavior. In this algorithm, each member (particle) of the society (swarm) tends to follow the member which has a better position and this directs the exploration of the search space. Some advantages of the PSO consist of the ease of its implementation and directional search, resulting in its popularity. However, a lack of balance between exploration and exploitation reduces its performance and convergence rate. Another problem with PSO is not having an efficient method for dealing with violated particles from the feasible search space. Kaveh and Nasrollahi [2] used a linear varying inertia weight for balancing exploration and exploitation, as well as a Harmony Search Based approach for reproducing violated particles.

On the other hand, the CSS is another multi-agent meta-heuristic optimization algorithm which was developed by Kaveh and Talatahari [3]. This algorithm is based on the laws of Coulomb and Gauss from electrostatics and the Newtonian laws of mechanics. The CSS was successfully applied to many structural optimization problems by Kaveh and Talatahari [4-8], Kaveh and Ahmadi [9], and Kaveh and Behnam [10]. Results from the above research show that CSS has a fast convergence rate and there is a suitable balance between exploration and exploitation. However, the main problem with CSS is the direction of movement of each Charged Particle (CP), since all CPs which can exert forces onto a CP, have the same influence, i.e. the best CPs have the same influence as other CPs. Previously, Kaveh and Laknejadi [11] introduced a hybrid CSS and PSO for multi-objective optimization. Kaveh et al. [12] also presented a new version of CSS by introducing magnetic, as well as electrostatic, forces.

In this paper, the directional search characteristic of the PSO is added to the CSS to master the influence of the best CP and the best position history of CP itself. This modification improves the capability of the CSS in finding the minimum of an objective function. The new algorithm is applied to four benchmark structural design problems and the higher performance of the HCSSPSO is shown compared to other methods. The obtained improvements correspond to both the design output and the convergence rate of the algorithm.

2. Particle swarm optimization

The PSO makes use of a velocity vector to update the current position of each particle in the swarm. The position of each particle in the swarm, which adapts to its environment by flying in the direction of the best position of all particles and the best position of the particle itself, provides the search process of the PSO. The position of the ith particle at iteration $k+1$ is calculated using the following relationship:

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

where, x_{k+1}^i is the new position; x_k^i is the position at the kth iteration; v_{k+1}^i is the updated velocity vector of the ith particle; and Δt is the time step which is considered as unity. The velocity vector of each particle is determined by:

$$v_{k+1}^i = w.v_k^i + c_1.r_1.\frac{(p_k^i - x_k^i)}{\Delta t} + c_2.r_2\frac{(p_k^g - x_k^i)}{\Delta t}, \tag{2}$$

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 particle i, 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 the inertia weight.

With the above description of the PSO, the algorithm can be summarized as follows.

The initial position, x_0^i, and the velocities, v_0^i, of the particles are distributed randomly in a feasible search space:

$$x_0^i = x_{\min} + r.(x_{\max} - x_{\min}), \tag{3}$$

$$v_0^i = \frac{x_{\min} + r.(x_{\max} - x_{\min})}{\Delta t}, \tag{4}$$

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

3. Charged system search

The Charged System Search (CSS) algorithm is an optimizer based on the Coulomb and Gauss electrostatics laws and the Newtonian Mechanics laws. A brief explanation of the CSS is as follows: This algorithm is a multi-agent approach, where each agent is a Charged Particle (CP). A CP is considered as a charged sphere with radius a, having a uniform volume charge density of magnitude:

$$q_i = \frac{\text{fit}(i) - \text{fitworst}}{\text{fitbest} - \text{fitworst}}, \quad i = 1, \ldots, N, \tag{5}$$

where fitbest and fitworst are the best fitness and the worst fitness of all the particles, fit(i) represents the fitness of the agent, i, and N is the total number of CPs.

CPs can impose electric forces on the others. The forces are considered as attractive and their magnitude for a CP located inside the sphere is proportional to the separation distance, r_{ij}, between the CPs. For a CP located outside the sphere, it is inversely proportional to the square of the separation distance between the particles. In continuous problems, it is sufficient to consider all the forces as attractive. Therefore, the forces can be obtained as follows:

$$F_j = q_j \sum_{i,i\neq j}^{N} \left(\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{r_{ij}^2}.i_2 \right) .p_{ij} \left(X_i - X_j \right), \quad (6)$$

where F_j is the resultant force acting on the jth CP, and r_{ij} is the separation distance between two charged particles, which is defined as:

$$r_{ij} = \frac{\|X_i - X_j\|}{\|(X_i + X_j)/2 - X_{\text{best}}\| + \varepsilon}, \quad (7)$$

where X_i and X_j are the positions of the ith CP and jth CP, respectively; X_{best} is the position of the best current CP, and ε is a small positive number. The initial positions of the CPs are determined randomly in the search space, and the initial velocities of charged particles are assumed to be zero.

p_{ij} determines the probability of moving each CP towards the others, as:

$$p_{ij} = \begin{cases} 1 & \frac{\text{fit}(i) - \text{fitbest}}{\text{fit}(j) - \text{fit}(i)} > \text{rand} \vee \text{fit}(j) < \text{fit}(i) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The resultant forces and motion laws determine the new location of the CPs. At this stage, each CP moves toward its new position under the action of the resultant forces and its previous velocity as:

$$X_{j,\text{new}} = \text{rand}_{j1}.k_a.\frac{F_j}{m_j}.\Delta t^2 + \text{rand}_{j2}.k_v.V_{j,\text{old}}.\Delta t + X_{j,\text{old}}, \quad (9)$$

where k_a is the acceleration coefficient; k_v is the velocity coefficient to control the influence of the previous velocity; and rand_{j1} and rand_{j2} are two random numbers uniformly distributed in the range of (0,1):

$$V_{j,\text{new}} = \frac{X_{j,\text{new}} - X_{j,\text{old}}}{\Delta t}. \quad (10)$$

To save the best design, a memory (Charged Memory or CM) is considered in the CSS. If a CP moves out of the allowable search space, its position is corrected using the Harmony Search handling method. According to this mechanism, any component of the solution vector violating boundaries can be regenerated from CM as:

$$x_{i,j} = \begin{cases} \text{w.p. CMCR} \\ \quad \Rightarrow \text{select a value for a variable from CM} \\ \quad \Rightarrow \text{w.p. (1-PAR) do nothing} \\ \quad \Rightarrow \text{w.p. PAR choose a neighboring value} \\ \text{w.p. (1-CMCR)} \\ \quad \Rightarrow \text{select a new variable randomly} \end{cases} \quad (11)$$

where $x_{i,j}$ is the ith component of the CP, j. The Charged Memory Considering Rate (CMCR), varying between 0 and 1, sets the rate of choosing a value in the new vector from the historic values stored in the CM, and (1-CMCR) sets the rate of choosing one random value from the possible range of values. The pitch adjusting process is performed only after a value is chosen from the CM. The value (1-PAR) sets the rate of doing nothing. Here, "w.p." stands for "with probability".

4. Hybrid charged system search and particle swarm optimization

In this section, a new hybrid Charged System Search and Particle Swarm Optimization is presented. For this purpose, some of the PSO parameters which provide good search for the PSO are added to the CSS. These modifications include:

1. Initial velocities of CPs are not considered as 0, but it is assumed to be defined by Eq. (4). This is necessary for the next stages.
2. For providing a balance between exploration and exploitation in the PSO, Fourie and Groenwold [13] have considered a dynamic varying of the inertia weight. In this approach, the inertia weight is not considered as constant. It varies by the progress of the algorithm and, in iteration $k+1$, the inertia weight is defined as:

$$w_{k+1} = k_w.w_k, \quad (12)$$

where w_k and w_{k+1} are the inertia weights in iteration k and $k+1$, respectively; and k_w is a constant multiplier.

However, in the CSS, the linear varying of force effect and velocity effect is considered as:

$$k_a = 0.5 \left(1 + \frac{\text{iter}}{\text{itermax}} \right), \quad (13)$$

$$k_v = 0.5 \left(1 - \frac{\text{iter}}{\text{itermax}} \right). \quad (14)$$

In this study, by inspiration of the dynamic varying of inertia weight, a new dynamically varying of the force effect and the velocity effect are introduced as follows:

$$k_a = (k_1)^{\frac{\text{iter}}{\text{itermax}}}, \quad (15)$$

$$k_v = (k_2)^{\frac{\text{iter}}{\text{itermax}}}, \quad (16)$$

where, k_1 and k_2 are two constants. Since, at the initial stages of the optimization there must be a focus on exploration, the velocity must have more significant effect and, in the latest iterations exploitation, must be performed by an algorithm, it is recommended to consider the values of k_1 and k_2 as 2.0 and 1.0E − 5, respectively. These values ensure a good balance between exploration and exploitation in the algorithm. By using a dynamically varying concept, the convergence rate of the CSS can be increased.

3. Since, in the CSS, the velocity vectors of particles depend on the amount of force, and, on the other hand, the forces exerted on a CP depend on the distances of other particles and the amount of the cost function corresponding to each CP, it is not ensured that the best particle has the most effect. This may cause reduction of the convergence rate and exploration. Thus, if we can master the effect of the best CP, the performance of the CSS will be improved. The key of this problem exists in the definition of the velocity in the PSO. If we consider the velocity of CPs by Eq. (17), the effect of the best CP will be raised, and the effect of the best position of CP up to the current iteration will be added to the CSS:

$$v^i_{k+1} = w.v^i_k + c_1.r_1.\left(p^i_k - x^i_k\right) + c_2.r_2.\left(p^g_k - x^i_k\right). \qquad (17)$$

However, in this approach, it is necessary to add a memory to the CSS for each particle in order to save the best position of the CP up to the current iteration. With these modifications, the performance of the CSS improves in such a way that by defining the dynamical variance, it is expected to raise the convergence rate, and by the use of the velocity definition of the PSO in the CSS, a better search will be performed. The stages of implementation of the HCSSPSO are similar to that of the CSS and only these modifications are applied. With the above definitions, the steps of optimization by the HCSSPSO are shown in the flowchart of Figure 1.

5. Numerical examples

To illustrate the effectiveness of the mentioned modifications, several benchmark examples that have previously been optimized by other researchers, are considered. These examples have been studied before using a variety of methods and are suitable for measuring the capability of an algorithm compared to existing ones. In these examples, the number of CPs is considered to be 20, and 20 independent runs are performed for each example. After some trial and error processes for finding suitable values for constants, w, c_1, and c_2, the best value for these parameters is found to be 1.0. In order to deal with the constraints, a penalty function approach is utilized.

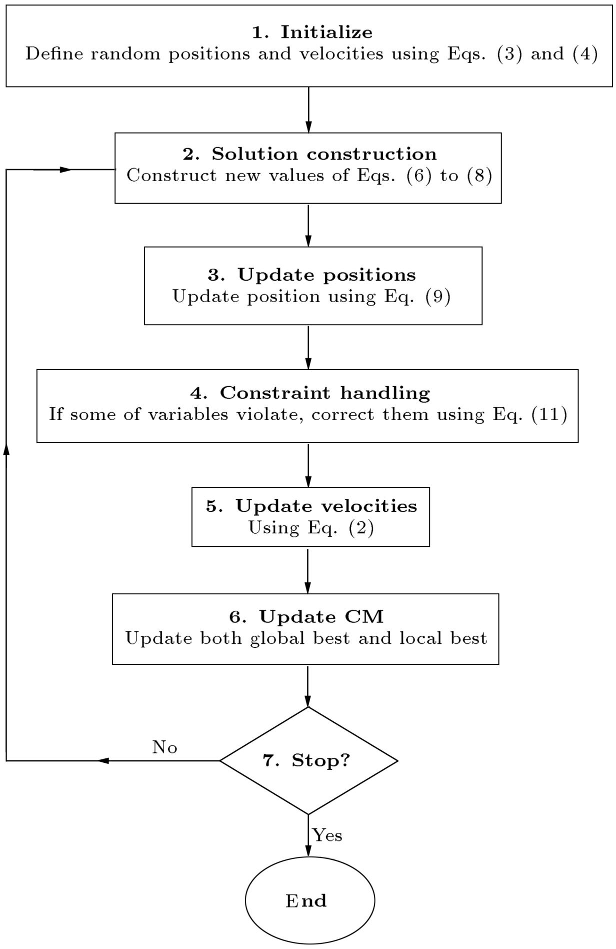
1. Initialize
Define random positions and velocities using Eqs. (3) and (4)
2. Solution construction
Construct new values of Eqs. (6) to (8)
3. Update positions
Update position using Eq. (9)
4. Constraint handling
If some of variables violate, correct them using Eq. (11)
5. Update velocities
Using Eq. (2)
6. Update CM
Update both global best and local best
No
7. Stop?
Yes
End

Figure 1. Flowchart of the HCSSPSO.

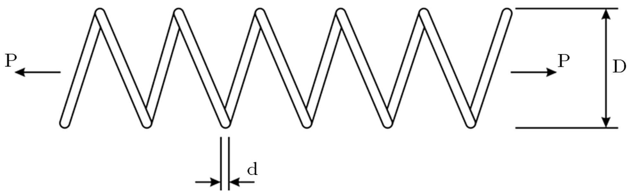

Figure 2. Geometry and the parameters of the tension/compression spring design.

5.1. *A tension/compression spring design problem*

This problem is described by Belegundu [14] and Arora [15]. 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.

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 expressed with the following cost function:

$$f_{\text{cost}}\,(X) = (x_3 + 2)\,x_2 x_1^2, \qquad (18)$$

to be minimized in the presence of the following constraints:

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71,785 x_1^4} \leq 0, \tag{19}$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12,566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5,108 x_1^2} - 1 \leq 0, \tag{20}$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0, \tag{21}$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0. \tag{22}$$

The variables are selected from the following regions:

$$0.1 \leq x_1 \leq 2, \tag{23}$$

$$0.25 \leq x_2 \leq 1.3, \tag{24}$$

$$2 \leq x_3 \leq 15. \tag{25}$$

This problem has been solved by Belegundu [14] using eight different mathematical optimization techniques (only the best results are shown). Arora [15] has also solved this problem using a numerical optimization technique, called constraint correction, at a constant cost. Coello [16], and Coello and Montes [17] solved this problem using a GA-based method. Additionally, He and Wang [18] utilized a Co-evolutionary Particle Swarm Optimization (CPSO). Recently, Montes and Coello [19], and Coelho [20] used evolution strategies to solve this problem. Table 1 presents the best solution of this problem obtained using the HCSSPSO algorithm and compares the HCSSPSO results to the solutions reported by other researchers. From Table 1, it can be seen that the best feasible solution obtained by the HCSSPSO is better than those previously reported. This issue is in both obtained cost function and standard deviation or the number of runs required for finding a reliable minimum by the algorithm. As can be seen, the results of the present algorithm compared to those of the CSS are better, and one can state that the advantages of the PSO are successfully added to the CSS.

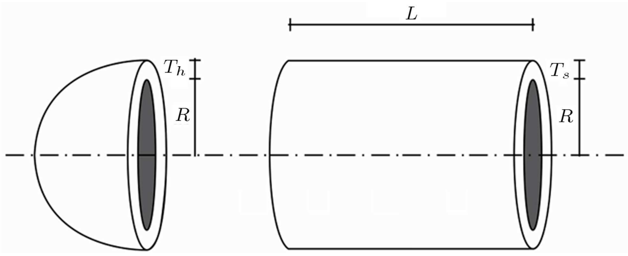

Figure 3. Geometry and parameters of the pressure vessel.

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 [21]:

$$f_{\text{cost}}(X) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3, \tag{26}$$

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 the cylindrical section of the vessel (L), not including the head. T_s and T_h are integer multiples of 0.0625 inches, the available thickness of the rolled steel plates, and R and L are continuous.

The constraint can be expressed as:

$$g_1(X) = -x_1 + 0.0193 x_3 \leq 0, \tag{27}$$

$$g_2(X) = -x_2 + 0.00954 x_3 \leq 0, \tag{28}$$

$$g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \tag{29}$$

$$g_4(X) = x_4 - 240 \leq 0. \tag{30}$$

Table 1. Optimal results of the tension/compression spring design.

Method	x_1(d)	x_2(D)	x_3(N)	Best result	Mean of the results	Worst results	SD
Belegudu [14]	0.050000	0.315900	14.250000	0.012833	N/A	N/A	N/A
Arora [15]	0.053396	0.399180	9.185400	0.012730	N/A	N/A	N/A
Coello [16]	0.051480	0.351661	11.632201	0.012705	0.012769	0.012822	3.939E-05
Coello and Montes [17]	0.051989	0.363965	10.890522	0.012681	0.012742	0.012973	5.900E-05
He and Wang [18]	0.051728	0.357644	11.244543	0.012675	0.012730	0.012924	5.199E-05
Montes and Coello [19]	0.051643	0.355360	11.397926	0.012698	0.013461	0.164850	9.660E-04
Coelho [20]	0.051515	0.352529	11.538862	0.012665	0.013524	0.017759	1.268E-3
CSS (present work)	0.052952	0.387880	9.671307	0.012694	0.013518	0.015435	1.088E-03
HCSSPSO (present work)	0.051799	0.359380	11.134561	0.012665	0.013142	0.015134	8.893E-04

Table 2. Result of the optimal design of pressure vessel.

Author or method	$x_1(T_s)$	$x_2(T_h)$	x_3 (R)	x_4 (L)	Best result	Mean of the results	Worst results	SD
Sandgren [21]	1.125	0.625	47.7	117.701	8,129.10	N/A	N/A	N/A
Kannan and Kramer [22]	1.125	0.625	58.291	43.69	7,198.04	N/A	N/A	N/A
Deb and Gene [23]	0.9375	0.5	48.329	112.679	6,410.38	N/A	N/A	N/A
Coello [16]	0.8125	0.4375	40.3239	200	6,288.74	6,293.84	6,308.15	7.4133
Coello and Montes [17]	0.8125	0.4375	42.09739	176.654	6,059.95	6,177.25	6,469.32	130.9297
He and Wang [18]	0.8125	0.4375	42.09126	176.7465	6,061.08	6,147.13	6,363.80	86.4545
Montes and Coello [19]	0.8125	0.4375	42.09808	176.6405	6,059.75	6,850.00	7,332.88	426
Coelho [20]	0.8125	0.4375	42.0984	176.6372	6,059.72	6,440.38	7,544.49	448.4711
CSS (present work)	0.8125	0.3871	40.9604	191.2691	6,052.89	6,539.16	7,174.35	408.312
HCSSPSO (present work)	0.8125	0.3916	41.4370	185.0857	6,003.56	6,172.06	6,807.91	271.8559

The design space is as follows:

$$0 \leq x_1 \leq 99, \tag{31}$$

$$0 \leq x_2 \leq 99, \tag{32}$$

$$10 \leq x_3 \leq 200, \tag{33}$$

$$10 \leq x_4 \leq 200. \tag{34}$$

The approaches applied to this problem include a branch and bound technique [21], an augmented Lagrangian multiplier approach [22], genetic adaptive search [23], a GA-based co-evolution model [16], a feasibility-based tournament selection scheme [17], a co-evolutionary particle swarm optimization [18], an evolution strategy [19], and a Gaussian quantum-behaved PSO approach [20]. 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 the HCSSPSO is better than the best solutions found by other techniques. Also, from this table, it can be seen that the average searching quality of the HCSSPSO is better than that of other methods.

5.3. A 25-bar element space truss

As the third example, a 25-bar space truss, as a transmission tower, is considered, as described by Schmit and Fleury [24], 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. The 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 3. Truss member grouping of the 25-bar 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 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 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)

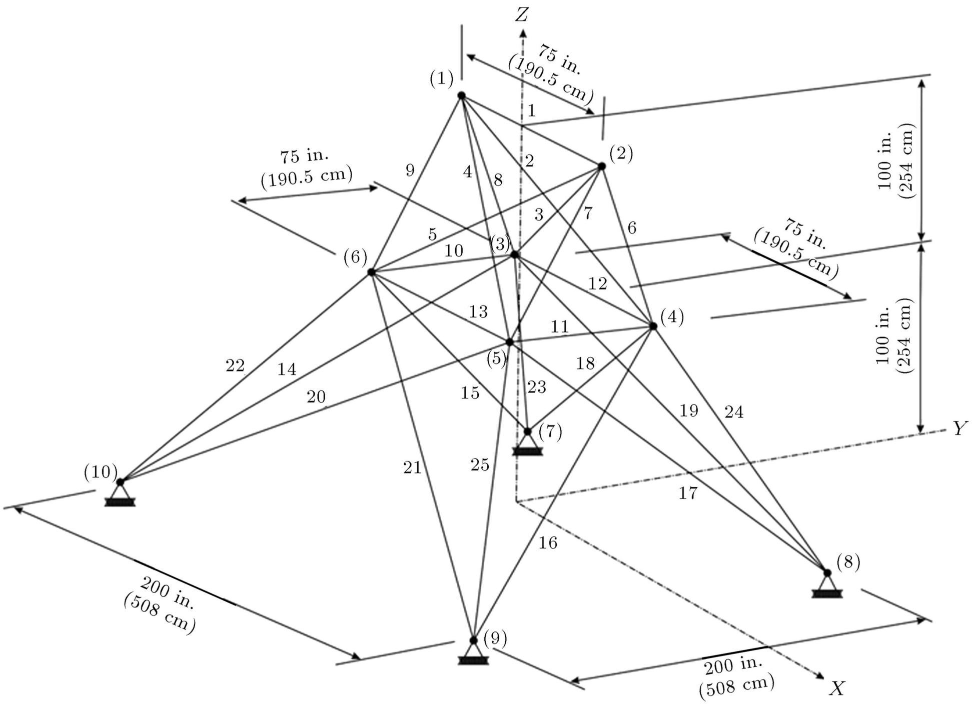

Figure 4. Geometry and element grouping of the 25-bar element space truss.

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

Element group	**Wu and Chow [25]**	**Zhu [26]**	**Haftka and Grudal [27]**	**Zhou and Rozvany [28]**	**Erbatur et al. [29]**	**Perez and Behdinan [30]**	**CSS (present work)**	**HCSSPSO (present work)**
1	0.1	0.1	0.01	0.01	0.1	0.1	0.104	0.1
2	0.5	1.9	1.987	1.987	1.2	1.0227	0.3475	0.4438
3	3.4	2.6	2.991	2.994	3.2	3.4	3.3667	3.3946
4	0.1	0.1	0.01	0.01	0.1	0.1	0.1	0.1
5	1.5	0.1	0.012	0.01	1.1	0.1	1.845	1.7925
6	0.9	0.8	0.683	0.684	0.9	0.6399	0.9538	0.9497
7	0.6	2.1	1.679	1.677	0.4	2.0424	0.5993	0.51
8	3.4	2.6	2.664	2.662	3.4	3.4	3.3921	3.3969
Best result (kips)	486.29	562.93	545.22	545.16	493.8	485.33	485.54	484.43
Mean of results (kips)	N/A	N/A	N/A	N/A	N/A	N/A	487. 90	485.04
Worst result (kips)	N/A	N/A	N/A	N/A	N/A	534.84	489.37	486.364
SD	N/A	N/A	N/A	N/A	N/A	N/A	1.732174	0.60718

Table 6 shows the chronic solution of this example and, also, results of the CSS and HCSSPSO are included. From this table, it can be seen that HCSSPSO has a better solution for this example. It is also obvious that the CSS has better results compared to other methods except HCSSPSO, and by the use of the HCSSPSO, a better solution than CSS is obtained. A smaller standard deviation indicates that the obtained weights from different runs are nearer than those of a higher standard deviation and this can be regarded as an additional capability of the algorithm. As shown in Table 6, the HCSSPSO has a smaller standard deviation compared to the CSS results. Thus, it can be stated that the HCSSPSO is more reliable and, in a few runs, we can be sure that a nearly optimum weight is obtained. The HCSSPSO and CSS progression curves

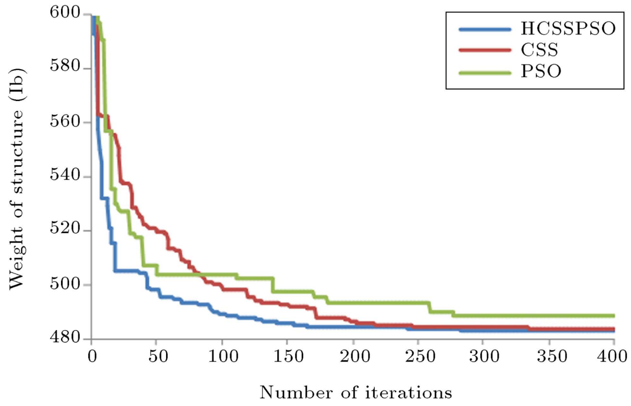

Figure 5. Convergence history of the 25-bar space truss.

have been shown in Figure 5. From this figure, it can be observed that the convergence rate of the HCSSPSO is higher than that of CSS and PSO.

5.4. 120-bar dome space truss

The design of a 120-bar dome truss, shown in Figure 5, is considered the last example to compare the practical capability of the HCSSPSO algorithm. This dome is utilized in literature to find a size design using metaheuristic algorithms. The modulus of elasticity is 30,450 ksi (210,000 MPa), and the material density is 0.288 lb/in^3 (7971.810 kg/m^3). The yield stress of steel is taken as 58.0 ksi (400 MPa). The dome is considered to be subjected to vertical loading at all the unsupported joints. These loads are taken as -13.49 kips (-60 kN) at node 1, -6.744 kips (-30 kN) at nodes 2 through 14, and -2.248 kips (-10 kN) at the remaining nodes. The minimum cross sectional area of all members is 0.775 in^2 (2 cm^2) and the maximum cross-sectional area is taken as 20.0 in^2 (129.03 cm^2). The stress constraints of the structural members are calculated as per AISC (1989) specifications, as illustrated in Eqs. (34) and (35). Also, displacement limitations of ± 0.1969 in (± 5 mm) are imposed on all nodes in x, y and z directions. The 120-bar truss members are categorized into 7 groups, as shown in Figure 6.

$$\begin{cases} \sigma_i^+ = 0.6F_y & \text{for } \sigma_i \geq 0 \\ \sigma_i^- & \text{for } \sigma_i < 0 \end{cases} \tag{35}$$

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

$$\sigma_i^- = \begin{cases} \left[\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y\right] / \left(\frac{5}{3}+\frac{3\lambda_i}{8C_c}-\frac{\lambda_i^3}{8C_c^3}\right) & \text{for } \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_c \end{cases} \tag{36}$$

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.

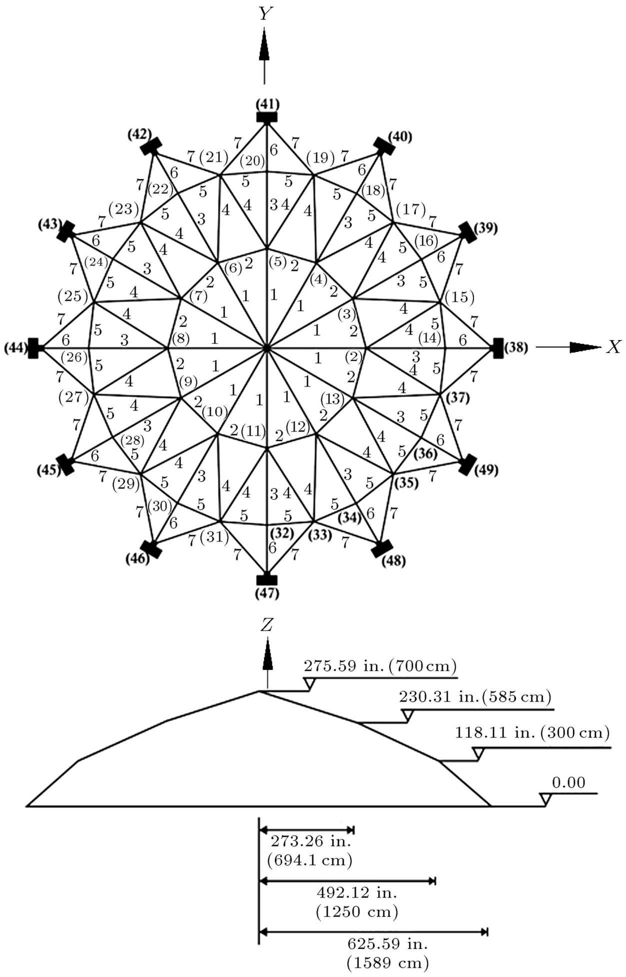

Figure 6. Geometry and element grouping of the 120-bar dome space truss.

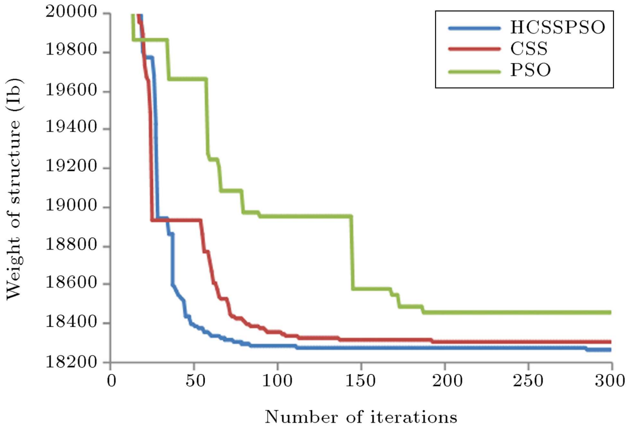

Figure 7. Convergence history of the 120-bar dome truss.

Figure 7 shows the convergence history of the CSS, HCSSPSO, and PSO. As can be seen, the convergence rate of the HCSSPSO is nearly the same as that of the CSS and higher than PSO. Also, results of the HCSSPSO are included in Table 7 for comparison. As shown, similar to other examples, HCSSPSO performs

Table 7. Results of the optimal design of the 120-bar truss dome.

Element group	Lee and Geem [31]			Kaveh and Talatahari [32]	Present work	
	HS	PSO	PSOPC	HPSACO	CSS	HCSSPSO
1	3.295	3.147	3.235	3.311	3.0189462	3.019056
2	3.396	6.376	3.37	3.438	4.1942215	3.737462
3	3.874	5.957	4.116	4.147	3.2376359	3.241314
4	2.571	4.806	2.784	2.831	2.2364713	2.236411
5	1.15	0.775	0.777	0.775	1.454316	1.728342
6	3.331	13.798	3.343	3.474	2.554482	2.483966
7	2.784	2.452	2.454	2.551	2.3006106	2.300618
Best result (lb)	19707.77	32432.9	19618.7	19491.3	18288.863	18268.6
Mean of results (lb)	N/A	N/A	N/A	N/A	18478.759	18377.6
Worst result (lb)	N/A	N/A	N/A	N/A	18773.834	18495.32
SD	N/A	N/A	N/A	N/A	192.67962	51.03623

much better, and the results obtained by the use of the present algorithm are better, in terms of both standard deviation and weight. These results also indicate the improvement of the CSS when the advantages of the PSO are added to it. However, as seen from Table 7, the CSS results are better than other methods for the design of this dome.

6. Mechanism of exploration and exploitation in the present algorithm

In order to show how the particles follow the global best and local best of the swarm, a 3-variable exponential function (two independent variables and one dependent variable) with a 2-dimensional search space is examined. This function is chosen because the search space can be visualized in a plane coordinate and thus, one can follow the search mechanism. The aforementioned function is in the following form:

$$f_{\text{cost}}(X) = -\exp\left(-0.5\left(x_1^2 + x_2^2\right)\right), X \in [-1,1]^2. \tag{37}$$

This function is visible in Figure 8 in the assumed search space and, from gradient based methods of finding the global minimum, it is evident that this function has a minimum of -1 at the point (0,0).

The search phase of the algorithms is shown in Figure 9. This figure shows two distinguished phases of search. The first phase takes place during the iterations 1 to 30 and is a global search or exploration. In this phase, the displacements of the particles are considerable and particles move from one side to

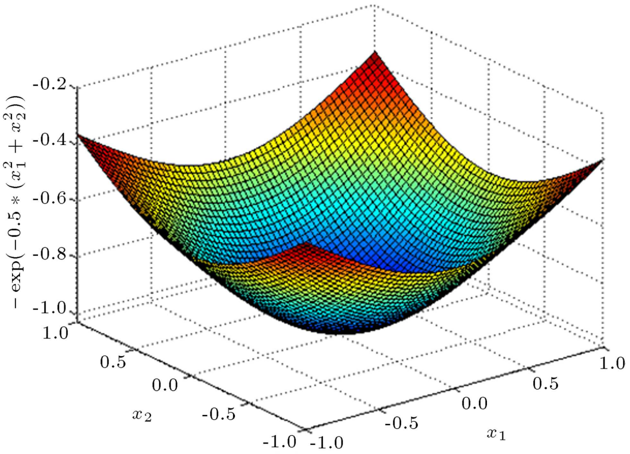

Figure 8. Graph of exponential function.

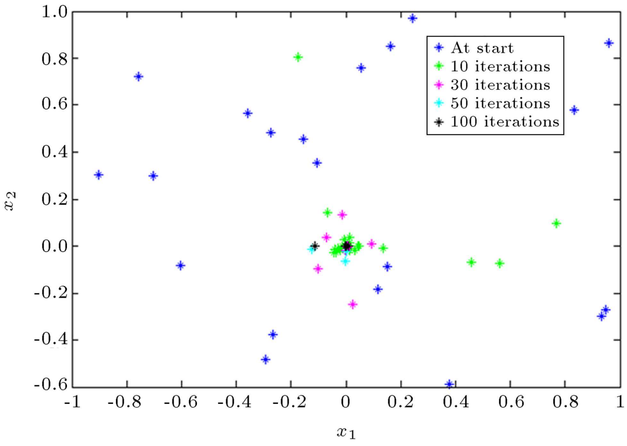

Figure 9. Exploration and exploitation mechanism of the HCSSPSO.

another with a direction towards the global minimum. The second phase starts from iteration 30, which is an exploitation involving the movement of the particles in a more contracted search space. This is a local search phase, which causes the correction of the particles positions and finally finds the correct point of the minimum value. From this figure, one can see that both exploration and exploitation of the HCSSPSO are performed accurately and properly and this mechanism makes the HCSSPSO a robust algorithm. This is obvious, since most of the particles are gathered at point (0,0).

7. Concluding remarks

In this paper, a new hybrid algorithm called HCSSPSO is presented. This algorithm consists of the hybridization of the CSS and PSO algorithms and can be considered an enhanced CSS with some advantages of the PSO. Since the definition of the velocity in CSS is affected by all the CPs exerting forces on each CP and is dependent, all the CPs have the same influence and the best CP does not, necessarily, have the most effect. This problem may cause a reduction in the search efficiency of the algorithm. On the other hand, the definition of the velocity in the PSO is dependent on the position of the best particle and the best position of the particle up to the current iteration. If the definition of velocity in the PSO is used in the CSS, it can work better. Another enhancement of the CSS, based on the PSO concept, is using a dynamic varying approach for the parameters, k_a and k_v, of the CSS. This modification has been previously implemented in inertia weight definition in the PSO, resulting in a higher convergence rate. These modifications are also added in the CSS. To prove positive modification of the HCSSPSO compared to CSS, four numerical examples extensively proposed by researchers are considered. Results illustrate the higher capability of the HCSSPSO compared to other methods of optimization and also CSS. Thus, it can be concluded that these improvements promote the performance of the CSS, and HCSSPSO can successfully be utilized in engineering design optimization problems.

Acknowledgement

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

References

1. Eberhart, R.C. and 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. and Nasrollahi, A. "Engineering design optimization using a hybrid PSO and HS algorithm", *Asian Journal of Civil Engineering (BHRC)*, **14**(2), pp. 201-223 (2013).
3. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: Charged system search", *Acta Mechanica*, **213**(3-4), pp. 267-289 (2009).
4. Kaveh, A. and Talatahari, S. "A charged system search with a fly to boundary method for discrete optimum design of truss structures", *Asian Journal of Civil Engineering*, **11**(3), pp. 277-293 (2010).
5. Kaveh, A. and Talatahari, S. "Charged system search for optimum grillage system design using the LRFD-AISC code", *Journal of Constructional Steel Research*, **66**(6), pp. 767-771 (2010).
6. Kaveh, A. and Talatahari, S. "Optimal design of skeletal structures via the charged system search algorithm", *Structural and Multidisciplinary Optimization*, **41**(6), pp. 893-911 (2010).
7. Kaveh, A. and Talatahari, S. "An enhanced charged system search for configuration optimization using the concept of fields of forces", *Structural and Multidisciplinary Optimization*, **43**(3), pp. 339-351 (2010).
8. Kaveh, A. and Talatahari, S. "Geometry and topology optimization of geodesic domes using charged system search", *Structural and Multidisciplinary Optimization*, **43**(2), pp. 215-229 (2011).
9. Kaveh, A. and Ahmadi, B. "Simultaneous analysis, design and optimization of structures using force method and supervised CSS algorithm", *Scientia Iranica*, **20**, pp. 65-76 (2013).
10. Kaveh, A. and Behnam, A.F. "Cost optimization of a composite floor system, one-way waffle slab, and concrete slab formwork using charged system search algorithm", *Scientia Iranica*, **19**, pp. 410-416 (2012).
11. Kaveh, A. and Laknejadi, K. "A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization", *Expert Systems with Applications*, **38**(12), pp. 15475-15488 (2011).
12. Kaveh, A., Motie Share, M.A. and Moslehi, M. "Magnetic charged system search: A new meta-heuristic algorithm for optimization", *Acta Mechanica*, **224**(1), pp. 85-107 (2013).
13. Fourie, P. and Geroenwold, A. "The particle swarm optimization algorithm in size and shape optimization", *Structural and Multidisciplinary Optimization*, **23**(4), pp. 259-267 (2002).
14. Belegundu, A.D. "A study of mathematical programming methods for structural optimization", PhD thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa (1982).
15. Arora, J.S., *Introduction to Optimum Design*, McGraw-Hill, New York (1989).
16. Coello, C.A.C. "Use of a self-adaptive penalty approach for engineering optimization problems", *Computers in Industry*, **41**, pp. 113-127 (2000).

17. Coello, C.A.C. and Montes, E.M. "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection", *Advanced Engineering Informatics*, **16**, pp. 193-203 (2002).

18. He, Q. and Wang, L. "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Engineering Applications of Artificial Intelligence*, **20**, pp. 89-99 (2007).

19. Montes, E.M. and Coello, C.A.C. "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems", *International Journal of General Systems*, **37**(4), pp. 443-473 (2008).

20. Coelho, L.S. "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems", *Expert Systems with Applications*, **37**, pp. 1676-1683 (2010).

21. Sandgren, E. "Nonlinear integer and discrete programming in mechanical design", In *Proceedings of the ASME Design Technology Conference*, Kissimine, FL, pp. 95-105 (1988).

22. Kannan, B.K. and Kramer, S.N. "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design", *Transactions of ASME Journal of Mechanical Design*, **116**, pp. 318-320 (1994).

23. Deb, K. and Gene, A.S. "A robust optimal design technique for mechanical component design", In Dasgupta, D., Michalewicz Z., Eds., *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, pp. 497-514 (1997).

24. Schmit, L. and Fleury, C. "Discrete-continuous variable structural synthesis using dual methods", *AIAA Journal*, **18**, pp. 1515-1524 (1980).

25. Wu, S. and Chow, P. "Steady-state genetic algorithm for discrete optimization of trusses", *Computers and Structures*, **56**(6), pp. 979-991 (1971).

26. Zhu, D. "An improved templemans algorithm for optimum design of trusses with discrete member size", *Engineering Optimization*, **9**, pp. 303-312 (1986).

27. Haftka, R. and Grudal, Z., *Elements of Structural Optimization*, 3rd Ed., Dordrecht: Kluwer Academic Publishers (1992).

28. Zhou, M. and Rozvany, G. "An optimality criteria method for large systems. Part II: Algorithm", *Structural and Multidisciplinary Optimization*, **6**, pp. 250-262 (1993).

29. Erbatur, F., Hasançebi, O., Tntnncn, I., and Kilt, H. "Optimal design of planar and space structures with genetic algorithms", *Computers and Structures*, **75**, pp. 209-224 (2000).

30. Perez, R.E. and Behdinan, K. "Particle swarm approach for structural design optimization", *Computers and Structures*, **85**(19-20), pp. 1579-1588 (2007).

31. Lee, K.S. and Geem, Z.W. "A new structural optimization method based on the harmony search algorithm", *Computers and Structures*, **82**(9-10), pp. 781-798 (2004).

32. Kaveh, A. and Talatahari, S. "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", *Computers and Structures*, **87**(5-6), pp. 267-283 (2009).

Biographies

Ali Kaveh was born in 1948 in Tabriz, Iran. He obtained a BS degree from the Department of Civil Engineering at the University of Tabriz in 1969, and MS, DIC and PhD degrees in 1970 and 1974, respectively, on Structures, from Imperial College of Science and Technology, London, UK. He is currently Professor of Structural Engineering at Iran University of Science and Technology, Tehran, Iran. Professor Kaveh is the author of 280 papers published in international journals and 135 papers presented at international conferences. He has authored 23 books in Farsi and 6 books in English published by Wiley, the American Mechanical Society, Research Studies Press, and Springer.

Amir Nasrollahi was born in 1985 in Borujerd, Iran. He obtained his BS degree in Civil Engineering from the Islamic Azad University, Kashan, Iran, in 2009, and his MS degree in Earthquake Engineering from the Building and Housing Research Center in 2011. At present, he is a PhD degree student of Earthquake Engineering at Iran University of Science and Technology, Tehran, Iran, and is working on development and application of meta-heuristic methods in structural analysis and design.