INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2011; 3:377-395

AN EFFICIENT HYBRID ALGORITHM BASED ON PARTICLE SWARM AND SIMULATED ANNEALING FOR OPTIMAL DESIGN OF SPACE TRUSSES

A. Hadidi[1], A. Kaveh[2,*,†], B. Farahmand Azar[1], S. Talatahari[3] and C. Farahmandpour[1]
[1]*Department of Civil Engineering, University of Tabriz, Tabriz, Iran*
[2]*Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran*
[3]*Marand Faculty of Engineering, University of Tabriz, Tabriz, Iran*

ABSTRACT

In this paper, an efficient optimization algorithm is proposed based on Particle Swarm Optimization (PSO) and Simulated Annealing (SA) to optimize truss structures. The proposed algorithm utilizes the PSO for finding high fitness regions in the search space and the SA is used to perform further investigation in these regions. This strategy helps to use of information obtained by swarm in an optimal manner and to direct the agents toward the best regions, resulting in possible reduction of the number of particles. To show the computational advantages of the new PSO-SA method, some benchmark numerical examples are studied. The PSO-SA algorithm converges to better or at least the same solutions, while the number of structural analyses is significantly reduced compared to the standard PSO and some other existing algorithms in the literature.

Received: 5 February 2011; Accepted: 20 August 2011

KEY WORDS: Optimum design; particle swarm optimization; simulated annealing; space trusses

1. INTRODUCTION

Demand for lightweight, efficient and low cost structures seems mandatory because of growing realization of the rarity of raw materials and rapid depletion of convention energy

[*]Corresponding author: A. Kaveh, Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
[†]E-mail address: alikaveh@iust.ac.ir

sources. This requires engineers to be aware of techniques for weight and cost optimization of structures [1]. Recently new design techniques for structural optimization have emerged [2-4]. These stochastic search techniques make use of the ideas inspired from nature and do not suffer the discrepancies of mathematical programming based optimum design methods [5] such as continuous objective function, requirement for the calculation of the gradients of the objective function and constraints, thus global optimum design (or near it) can be archived only with assessment of the objective function.

Particle Swarm Optimization (PSO), as one of the successful algorithms, is based on simulation of the social behavior of bird flocking and fish schooling. The PSO is a population based technique which involves a population of particles that are randomly initialized in the search space. Each particle flies through search space by means of its velocity vector. The velocities determine particles flying direction and they are updated based on the best positions of individual particles and the best position across all others in each iteration. The best position of the population (swarm) is found by evaluating objective function of all the particles. This strategy increases the probability of migration to regions of high fitness. The PSO has some advantages such as few parameters implementation and simple programming in computers. Meanwhile, it does not require specific domain knowledge information, internal transformation of variables or other manipulations to handle the constraints [6].

On the other hand, simulated annealing is another optimization technique that simulates the physical annealing process in the field of combinatorial optimization. The idea is based on a simulated annealing process that starts with heating up the metal very rapidly and then cooling it slowly in order to obtain a very pure crystal structures with a minimum energy. In the primary steps, the movement of particles is accelerated by high temperature and during the cooling period an optimal place within the crystal structure can be found [7]. The similarities between simulated annealing and optimization were first recognized by Kirkpatrick et al. [8] and Cerny [9] in which solutions in an optimization problem are equivalent to configurations of a physical system and the fitness of a solution is equivalent to the energy of a configuration. The main characteristic of the simulated annealing (SA) is to avoid the algorithm to be trapped in local optimum by uphill move.

Recently, a few studies have been devoted to hybrid the PSO with SA in order to improve the features of the PSO in optimization of variety of problems. Idoumghar et al. [10] presented a hybrid algorithm that combines the exploration ability of the PSO with exploitation feature of the SA to avoide the premature convergence in the PSO algorithm. This hybrid algorithm was applied to well-known benchmark mathematical functions as well as a problem of reducing energy consumption in embedded systems memories. The results show that the hybrid algorithm works well in terms of accuracy, convergence rate, stability and robustness. Also, Zhao et al. [11] hybridized the PSO and SA to solve multi-objective problems of partner selection in virtual enterprise. The present hybrid algorithm combines the high speed of the PSO with the powerful ability of the SA to avoid being trapped in local minimum. The results show that this hybrid algorithm works quite efficiently in this field. In addition, Chaojun and Zulian [12] introduced Metropolis acceptance probability criterion to the particle swarm to update best position of each particle and the best position of the entire swarm. Simulation results on function benchmarks show the capability of the hybrid algorithm in overcoming the problem of getting trapped in the local best point.

The main drawback of the PSO and other non-gradient, probabilistic search algorithms is that they typically require numerous function evaluations in comparison to the gradient-based algorithms [13]. For structural optimization, this means a lot of finite elements analyses which is time consuming and the efficiency of aforementioned methods in optimal structural design is questionable [14]. This paper presents a new hybrid algorithm based on the PSO and SA to reduce the number of structural analysis while saving the reliability of achieving the optimum design.

2. STATEMENT OF THE STRUCTURAL OPTIMIZATION PROBLEM

Size optimization of truss structures includes finding optimum values for member cross-sectional areas x_i that minimize the structural weight W. This minimum design also has to satisfy inequality constraints that limit design variable sizes and structural responses. Thus, the optimal design of a truss is formulated as [15]:

$$\begin{aligned} &\text{Minimize } W(\{x\}) = \sum_{i=1}^{n} g_i . A_i . l_i \\ &\text{Subject to:} \quad g_{\min} \le g_i(\{x\}) \le g_{\max} \qquad i = 1,2,3,...,m \end{aligned} \tag{1}$$

where $W(\{x\})$ is the weight of the structure; n is the number of members making up the structure; $g(\{x\})$ denotes the constraints considered for the structure containing stress of elements and nodal deflection; *max* and *min* denote upper and lower bounds, respectively.

For the proposed method, it is essential to transform the constrained optimization problem to an unconstraint one. A detailed review of some constraint-handling approaches is available in [16,17]. In this study, a penalty function method is utilized for handling the design constraints which is calculated using the following formulas [2]:

$$\begin{cases} g_{\min} < g_i < g_{\max} & \Rightarrow \Phi_g^{(i)} = 0 \\ Otherwise & \Rightarrow \Phi_g^{(i)} = \dfrac{g_i - g_{\min/\max}}{g_{\min/\max}} \end{cases} \tag{2}$$

The objective function that determines the fitness of each particle is defined as:

$$Mer^{\,k} = e_1 \cdot W^{\,k} + e_2 \cdot (\sum \Phi_g^{(i)})^{e_3} \tag{3}$$

where *Mer* is the objective function (merit function); e_1, e_2 and e_3 are the coefficients of merit function; $\Phi_g^{(i)}$ denotes the summation of penalties. In this study e_1 and e_2 are set to 1 and W (weight of structure), respectively, while the value of e_3 is considered as 0.85 in order

to achieve a feasible solution.

3. REVIEW OF THE UTILIZED ALGORITHMS

3.1. Particle Swarm Optimization

The PSO algorithm, inspired by social behavior simulation, was initially proposed by Kennedy and Eberhart in 1995 [18,19]. The PSO is one of the evolutionary computation techniques such as genetic algorithms, evolutionary programming, evolution strategies, and genetic programming [20] that the main idea behind it is that the social sharing of information among members offers an evolutionary advantage. In fact, in the PSO, instead of using more traditional genetic operators, each particle adjusts its flying according to its own flying experience and its companions' flying experience [20].

The original PSO that was designed and developed by Kennedy and Eberhart based on the following two simple equations:

$$v_i^d(k+1) = v_i^d(k) + c_1 \times rand_{1i}^d \times (pbest_i^d(k) - x_i^d) + c_2 \times rand_{2i}^d \times (gbest^d(k) - x_i^d) \quad (4)$$

$$x_i^d(k+1) = x_i^d(k) + v_i^d(k+1) \quad (5)$$

Eq. (4) calculates the velocity of each particle according to three factors:

(1) previous velocity ($v_i^d(k)$);

(2) direction of best position ($pbest_i^d(k)$) visited by each particle itself;

(3) direction of the best position of swarm ($gbest^d(k)$) up to iteration k.

Eq. (5) updates each particle's position in the search space.

In these equations x_i, v_i represent the current position and the velocity of the ith particle, respectively; $rand_{1i}^d$ and $rand_{2i}^d$ represent random numbers between 0 and 1; $gbest^k(k)$ corresponds to the global best position in the swarm up to iteration k ; $c1$, $c2$ represent cognitive and social parameters, respectively.

After many numerical simulations, Eberhart and Shi [21] added a weighting factor to Eq. (4) to control the trade-off between the global exploration and the local exploitation abilities of the flying particles as:

$$v_i^d(k+1) = w \times v_i^d(k) + c_1 \times rand_{1i}^d \times (pbest_i^d(k) - x_i^d) + c_2 \times rand_{2i}^d \times (gbest^d(k) - x_i^d) \quad (6)$$

A larger inertia weight makes the global exploration easier while a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area [20]. By using the linearly decreasing inertia weight, the PSO lacks global search ability at the end of run even when the global search ability is required to jump out of the local minimum in some cases.

Nevertheless the results shown in literature illustrate that by using a linearly decreasing inertia weight the performance of the PSO can be improved greatly and have better results than that of both standard PSO and evolutionary programming as reported in [22,23].

3.2. Simulated Annealing

Simulated Annealing is a probabilistic heuristic based on thermodynamics considerations. The technique is motivated by an analogy to the statistical mechanics of annealing in solids. A low energy state usually means a highly ordered state, such as crystal lattice. To accomplish this, the material is first melted, i.e. heated to a temperature that permits many random, disordered and high-energy atomic rearrangements. Then it is cooled slowly and as the temperature is reduced the atoms migrate to a more ordered state with lower energy. The final degree of order depends on the temperature cooling rate [24]. If the annealing process be fast and is not given enough time to complete the cooling process, the melting metal still solidifies but with higher disordered atomic arrangement. For slowly cooled process, on the other hand, system is able to reach lower energy configurations.

A computational algorithm that simulates the annealing process was proposed by Metropolis et al. [25], and is referred to as the Metropolis algorithm. Then, the analogy between the simulated annealing and the optimization of functions with many variables was established by Kirkpatrick et al. [8] and Cerny [9]. By replacing the energy state with an objective function, and using variables for the configurations of the particles, the Metropolis algorithm can be applied to optimization problems [1]. Further details of the algorithm can be found in [26].

4. NEW HYBRID ALGORITHM

In structural optimization with high complexity, the number of particles should be selected high in order to further investigation in the search space and accomplish better solutions. From structural optimization point of view, this means numerous finite element analyses should be performed which is both costly and time-consuming. On the other hand, if the number of particles is reduced, the probability of finding desired solutions will decrease the intensely and the algorithm may become trapped in local minimums. Therefore, in this study a hybrid algorithm is utilized to overcome this difficulty. In the present algorithm, in order to reduce the number of structural analyses, the number of required particles is decreased. However the quality of the solutions obtained by the new algorithm, called the PSO-SA, is saved compared to the PSO algorithm. Figure 1 shows the optimization procedure of the present algorithm, which consist of the following steps:

Step 1: The PSO-SA has some parameters that must be adjusted in order to obtain optimal solution. These parameters are introduced and set in the following steps. In this step the position and velocity of the particles in the initial swarm are randomly generated and the best particle is determined.

Step 2: The new positions and velocities of the particles are determined according to Eq. (5) and Eq. (6), respectively. Then comparing the merit value of each particle with the

previous best one, *pbest* and *gbest* are updated.

Step 3: Due to the high convergence rate of the PSO in early iterations [22], the proposed algorithm utilizes only the PSO as the optimizer in early iterations and only if the difference between the best solutions obtained by the PSO in two successive iterations becomes smaller than a given value (shown by e), the SA is utilized along with the PSO. To fulfill this aim, as a criterion to use the SA in cooperation with the PSO, the best obtained solutions is checked after each 10 iteration; here, the value of (e) is set to 0.01.

Step 4: When the criterion in the step 3 is satisfied, the SA algorithm is used to further search around *gbest* point that obtained by the PSO. In order to generate candidate designs, in the neighborhood of *gbest*, a normal distribution is used as:

$$gbest^{a} = gbest^{c} + N(0, s) \tag{7}$$

where $gbest^{a}$ and $gbest^{c}$ represent candidate design and current design, respectively. $N(0, s)$ is a random number corresponding to normal distribution with mean value 0 and standard deviation s . The value of s is obtained by:

$$s = h \cdot (x_{\max} - x_{\min}) \tag{8}$$

In which h is set to 0.02.

Each time a candidate design is generated, its merit function value (Mer) is computed and compared to the current design. If the new candidate provides a better solution, it is accepted automatically and it replaces the current design. Otherwise, the so-called Metropolis test is employed to determine the winner, in which case the probability of accepting a poor candidate (*P*) is assigned as follows:

$$P(\Delta Mer) = e^{(-\frac{\Delta Mer}{K_B T})} \text{ , where } \Delta Mer = Mer_a - Mer_c \tag{9}$$

Mer_c and Mer_a are the merit function values of the current and candidate designs, respectively; T is the current temperature of the process; and K is the Boltzman parameter which is manipulated as the working average of positive ΔMer values.

In this procedure, each design variable is selected only once in a random order, until all design variables are selected. This is referred to as a single iteration of the cooling cycle. Generally, a cooling cycle is iterated a certain number of times in the same manner to ensure that the objectives function is reduced to a reasonably low value associated with the temperature of the cooling cycle.

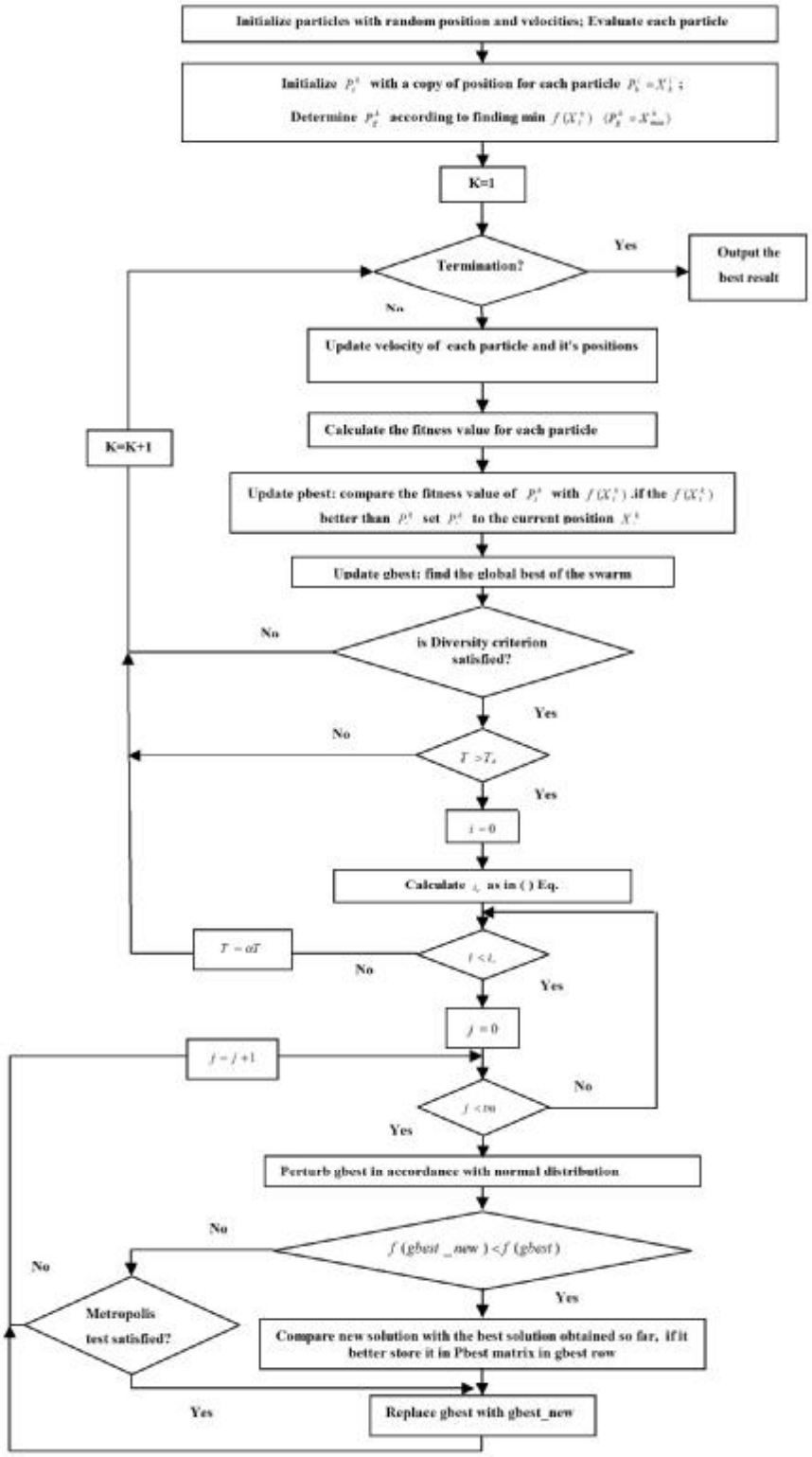


Figure 1. The flowchart for the PSO-SA.

Having selected the iterations of the starting and final cooling cycles (i_s and i_f), the iteration of a cooling cycle (i_c) at a given temperature T is determined by a linear interpolation between i_s and i_f as follows:

$$i_c = i_f + (i_f - i_s)\left(\frac{T - T_f}{T_f - T_s}\right) \tag{10}$$

where T is the current temperature of the process; T_s, T_f referred to as starting temperature and final temperature, respectively. In order to set the cooling schedule for the SA algorithm, the approach suggested by Balling [27] is used as:

$$T_s = -\frac{1}{\ln(P^s)}, T_f = -\frac{1}{\ln(P^f)} \tag{11}$$

In Eq. (11), P^s, P^f are starting acceptance probability and final acceptance probability, respectively. According to the recommendations given in [26], P^s and P^f are set to 0.5 and 1E-7, respectively. In this paper, the i_s and i_f are taken as 1 and 3, respectively [26]. This step has two unique features that are explained bellow.

It should be noted that when the number of particles in the PSO algorithm is reduced, after some iteration the population diversity reduces remarkably and this leads to a premature convergence. In order to surmount this problem, after completing the application of the SA, regardless of havng better or worse than initial *gbest*, the solution obtained by the SA is applied to calculate the subsequent velocities of the particles as *gbest*. This strategy results in wandering the particles and can restore the diversity loss of the particles when the SA works in cooperation with the PSO and improves the global search capacity of the algorithm.

Another feature of the algorithm is that the best obtained solution in the SA procedure is recorded in the Pbest matrix in the row corresponding to the *gbest*. Since the best solution obtained by the SA is stored in the *pbest*, in subsequent iteration the particles move and if a better result is not obtained, the best previous solution obtained by the SA is taken as the *gbest* and this helps the algorithm not to loose the good search direction.

Step 5: If Step 4 is utilzed, the temperature should be then reduced by the ratio of the cooling factor a as follows:

$$T^{r+1} = a \times T^{r} \tag{12}$$

where T^r and T^{r+1} denote the temperature at the rth and (r+1)-th cooling cycles, respectively. The cooling factor a is obtained by the following formula as in [27]:

$$a = \left[\frac{\ln(P^s)}{\ln(P^f)}\right]^{\frac{1}{(N_c - 1)}} \tag{13}$$

In Eq. (13), P^S and P^f are starting and final acceptance probability respectively, and N_c is the number of cooling cycle. In this paper, the number of cooling cycle is set to $N_c=150$.

Step 6: Depending to the condition satisfaction of each step, Step 2 to Step 5 are repeated.

In the present algorithm, by means of cooperation and share of information among the particles, the PSO finds rich sectors of search space, and the SA searches the best region found by the PSO. Therefore, the main characteristic of the algorithm is that further analyses are taken place in the high fitness region of search space. This feature of the algorithm results in using few particles. Therefore, instead of using many particles to seek the search space, a few particles are used and the regions with high fitness are investigated with higher accuracy.

5. NUMERICAL EXAMPLES

In this section, truss optimization problems have been studied to demonstrate the efficiency of proposed algorithm. The algorithms are coded in Matlab and structures are analyzed using the direct stiffness method. For both algorithms, the inertia weight w decreases linearly from 0.9 to 0.4; the value of acceleration constants $c1$ and $c2$ are both set to 2; the maximum velocity is set to the difference between the upper bound and lower bound of variables. The maximum number of iterations is limited to 500. For the standard PSO algorithm, the population of 50 particles is used and this amount is reduced to 10 particles for the PSO-SA algorithm. It should be noted that since w is a function of iteration, therefore the total number of finite element analyses in 500 iterations is reported to demonstrate the number of analyses that are performed to achieve the best solution.

5.1. A twenty-two-bar space truss

The topology and nodal numbering of a 22-bar spatial truss structure are shown in Figure 2. The structure was previously studied by Khan and Willmert [28], Sheu and Schmit [29], Lee and Geem [30]. The detailed information of the examples is available in [30].

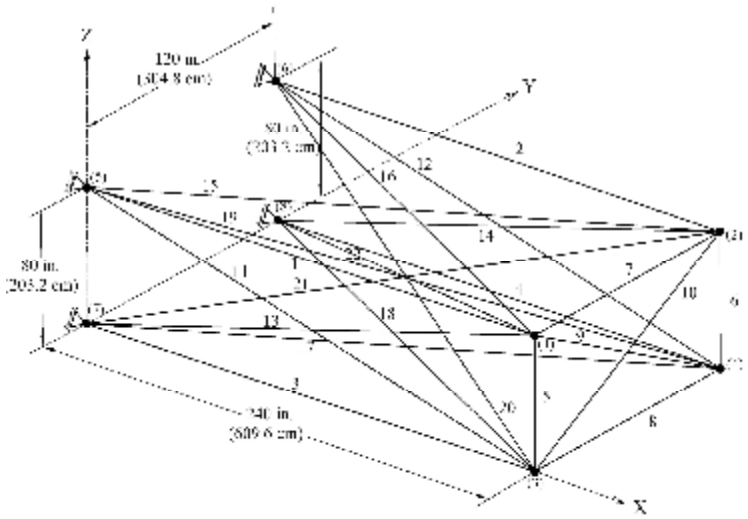


Figure 2. Twenty-two bar space truss

Table 1 lists the optimal values of the seven size variables obtained by the standard PSO and PSO-SA algorithms, and compares them with earlier results. For this spatial truss structure, the total number of finite element analyses to achieve the best solution is 7,618. This amount is significantly smaller than that for the PSO. Also, the HS algorithm in this example requires 10,000 analyses to obtain the best result. Table 2 shows the results of 50 runs for both of algorithms in which the standard deviation of the PSO-SA is smaller than that of the PSO. Figure 3 provides a comparison of the convergence rates for the two algorithms. It can be seen that the convergence rate of the PSO-SA algorithm is obviously higher than PSO.

Table 1. Comparison of the optimal designs for the 22-bar spatial truss structure

Element group		Optimal cross-sectional areas (in^2)						
		Sheu & Schmit [29]	Khan & Willert [28]	Lee & Geem [30]	PSO		PSO-SA	
					in^2	cm^2	in^2	cm^2
1	A_1~A_4	2.629	2.563	2.588	2.5799	16.64	2.5958	16.75
2	A_5~A_6	1.162	1.553	1.083	1.1312	7.30	1.2290	7.93
3	A_7~A_8	0.343	0.281	0.363	0.3472	2.24	0.3452	2.23
4	A_9~A_{10}	0.423	0.512	0.422	0.4212	2.72	0.4187	2.7
5	A_{11}~A_{14}	2.782	2.626	2.827	2.8330	18.28	2.8119	18.14
6	A_{15}~A_{18}	2.173	2.131	2.055	2.0946	13.51	2.1591	13.93
7	A_{19}~A_{22}	1.952	2.213	2.044	2.0205	13.03	1.9517	12.59
Weight (lb)		1024.80	1034.74	1022.23	1024	4554.75	1024.1	4555.20

Table 2. Performance of the PSO and PSO-SA algorithms for 22-bar truss in 50 run

Algorithm	Best	Average	Worst	Best-average difference	Best-worst difference	Standard deviation
PSO	1024	1033.79	1093.12	0.95 %	6.75 %	17.29
PSO-SA	1024.1	1033.38	1056.74	0.91 %	3.19 %	10.30

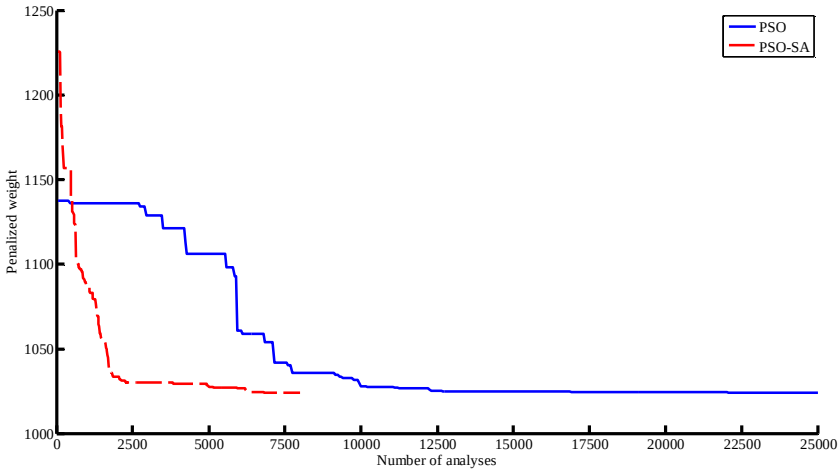


Figure 3. Comparing the convergence rates between the two algorithms for 22-bar spatial truss

5.2. Criterion of Success for the examples

Figure 4 shows the topology and nodal numbering of a 25-bar spatial truss structure. More details can be found in [30].

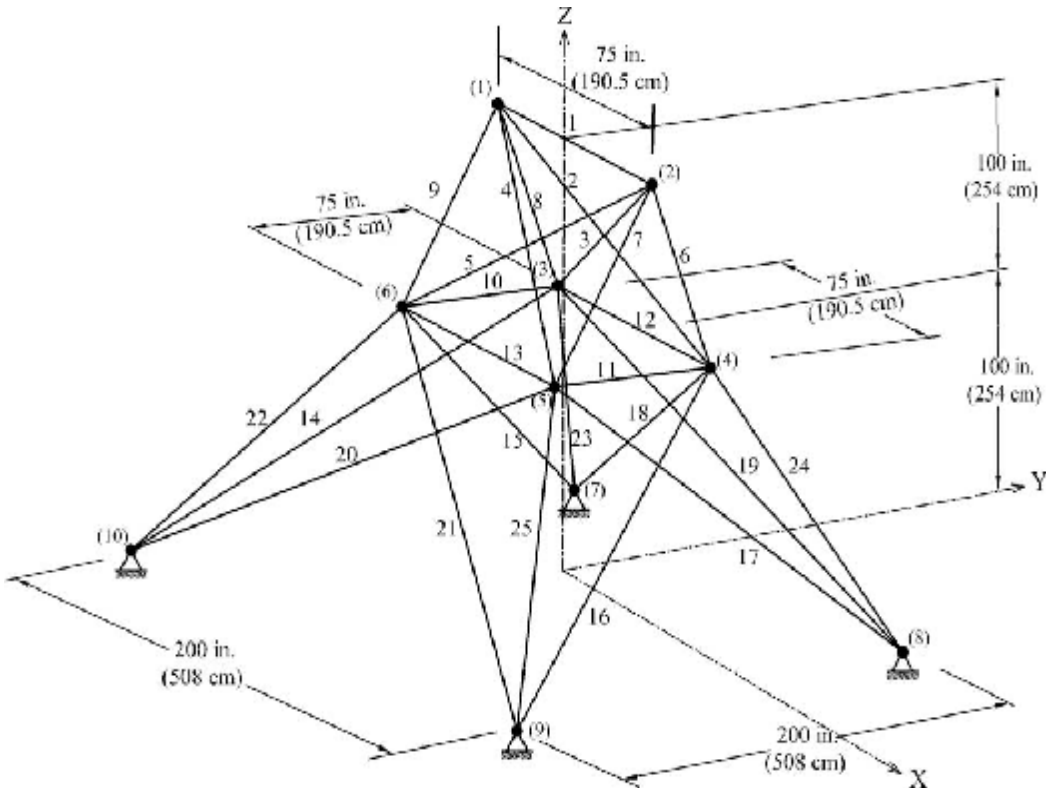


Figure 4. 25-bar spatial truss

Table 3 provides a comparison between the optimal design results reported in the literature and the present works. Also, Table 4 presents the statistical results of 50 independent runs of the standard PSO and PSO-SA. Obviously, the PSO-SA leads to better solutions for both best and worst results. The average and the standard deviation of the solutions obtained by the PSO-SA are also remarkably less than those of the PSO. Comparison of the convergence rates between the PSO and PSO-SA algorithms for this truss is presented in Figure 5. The number of searches for the HS is 15,000 as provided in [30]. Also the ACO algorithm require 16,500 analyses to obtain a solution [31]. The HBB-BC [32] algorithm achieves best solution after 12,500 searches. However, the number of required structural analyses during the optimization process by the PSO-SA is 7,992. Hence, it can be seen that using the PSO-SA results in a high reduction of computational burden of optimization process compared to other algorithms in the literature.

Table 3. Comparison of optimal designs for the 25-bar spatial truss structure

Element group		Optimal cross-sectional areas (in^2)						
		Kaveh & Talatahari [32]	Lee & Geem [30]	Camp et al. [31]	PSO		PSO-SA	
					in^2	cm^2	in^2	cm^2
1	A_1	0.010	0.047	0.010	0.0100	0.06	0.0100	0.06
2	A_2~A_5	1.993	2.022	2.000	1.9503	12.58	1.9935	12.86
3	A_6~A_9	3.056	2.950	2.966	3.0408	19.62	2.9819	19.24
4	A_{10}~A_{11}	0.010	0.010	0.010	0.0100	0.06	0.0100	0.06
5	A_{12}~A_{13}	0.010	0.014	0.012	0.0100	0.06	0.0100	0.06
6	A_{14}~A_{17}	0.665	0.688	0.689	0.6929	4.47	0.6802	4.39
7	A_{18}~A_{21}	1.642	1.657	1.679	1.6866	10.88	1.6772	10.82
8	A_{22}~A_{25}	2.679	2.663	2.668	2.6362	17.01	2.6700	17.22
Weight (lb)		545.16	544.38	545.53	545.22	2425.14 (N)	545.17	2424.92 (N)

Table 4. Performance of PSO and PSO-SA algorithms for 25-bar truss in 50 run

Algorithm	Best	Average	Worst	Best-average difference	Best-worst difference	Standard deviation
PSO	545.22	549.96	594.53	0.87%	9.04%	9.91
PSO-SA	545.17	546.3	550.44	0.21 %	0.97 %	1.04

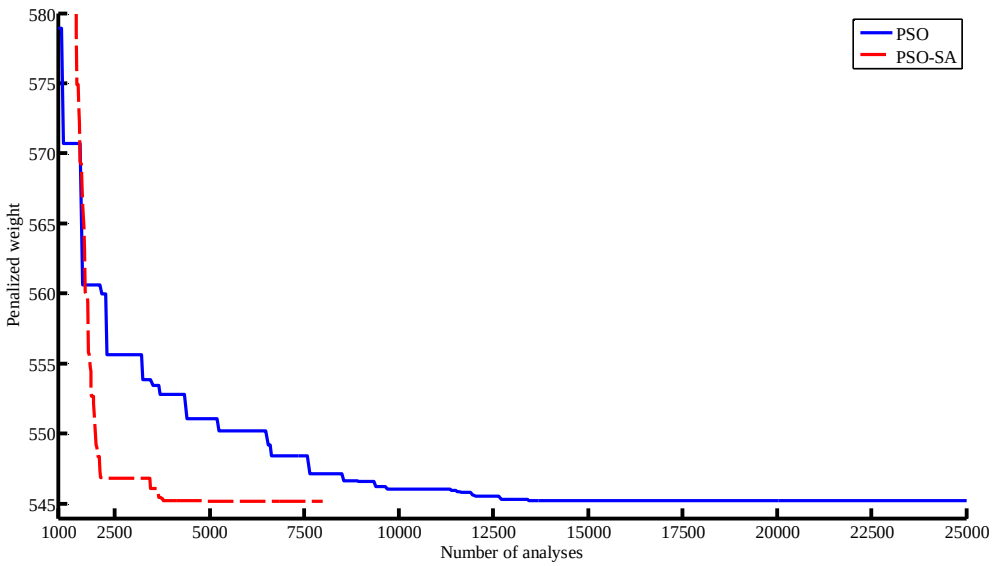


Figure 5. Comparing the convergence rates between the two algorithms for 25-bar truss.

5.3. A seventy two-bar spatial truss

The 72-bar spatial truss structure is shown in Figure 6. The detailed information of this example can be found in [30]. The minimum cross-sectional area of 0.01 in^2 is considered.

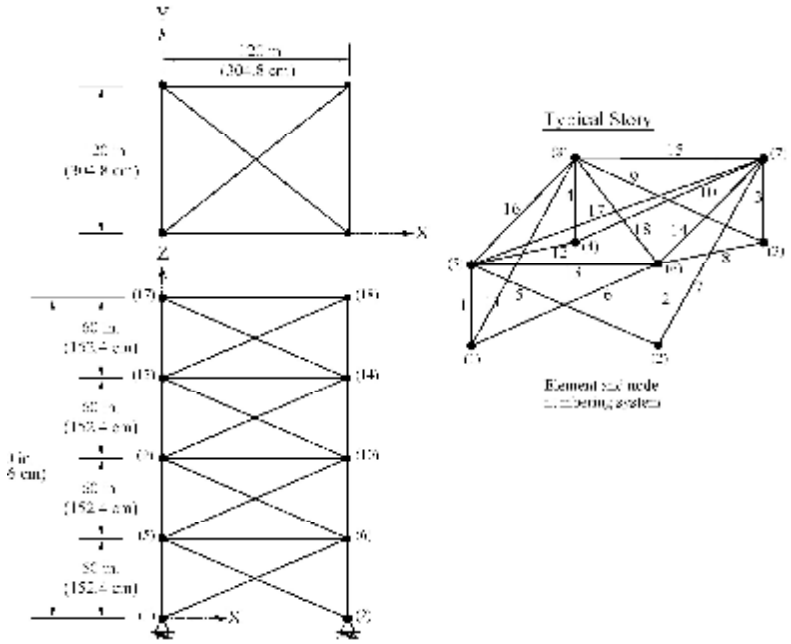


Figure 6. Seventy-two bar space truss.

The computational performance of the PSO-SA and PSO in this example is investigated through 50 independent runs and the results are given in Table 5. Also, Table 6 compares the obtained results of this article with those of Ref. [30] for this example. It can be observed from this table that the average of results and worst weight obtained by the PSO is considerably reduced through using the PSO-SA algorithm. The percent difference between the best and average solution is decreased from 13.35% to 0.21%. Also the difference between the best and the worst solution is decreased from a high value of 81.96% to 0.46%. Moreover, the standard deviation of solutions has decreased remarkably. It can be seen from Table 5 that the optimal designs obtained by the PSO and PSO-SA are slightly better than the previous design results. For this case the number of structural analyses reported for the HS is 20,000 while the number of finite element analyses for the PSO-SA in optimization process is 10,984. Figure 7 compares the convergence rate for the PSO and PSO-SA methods.

Table 5. Performance of PSO and PSO-SA algorithms for 72-bar truss in 50 run

Algorithm	Best	Average	Worst	Best-average difference	Best-worst difference	Standard deviation
PSO	364	412.61	662.34	13.35 %	81.96 %	64.9
PSO-SA	364	364.77	365.7	0.21 %	0.46 %	0.43

Table 6. Comparison of optimal designs for the 72-bar spatial truss structure

Element group		Optimal cross-sectional areas (in^2)				
		Lee & Geem [30]	PSO		PSO-SA	
			in^2	cm^2	in^2	cm^2
1	A_1~A_4	1.963	1.9067	12.3	1.8946	12.22
2	A_5~A_{12}	0.481	0.5311	3.43	0.5156	3.33
3	A_{13}~A_{16}	0.010	0.0100	0.064	0.0100	0.064
4	A_{17}~A_{18}	0.011	0.0100	0.064	0.0100	0.064
5	A_{19}~A_{22}	1.233	1.3154	8.49	1.2784	8.25
6	A_{23}~A_{30}	0.506	0.5168	3.33	0.5173	3.34
7	A_{31}~A_{34}	0.011	0.0100	0.64	0.0100	0.064
8	A_{35}~A_{36}	0.012	0.0100	0.64	0.0100	0.064
9	A_{37}~A_{40}	0.538	0.5345	3.45	0.5733	3.70
10	A_{41}~A_{48}	0.533	0.5055	3.26	0.5245	3.38
11	A_{49}~A_{52}	0.010	0.0114	0.07	0.0100	0.064
12	A_{53}~A_{54}	0.167	0.1055	0.68	0.1095	7.06
13	A_{55}~A_{58}	0.161	0.1672	1.08	0.1668	1.08
14	A_{59}~A_{66}	0.542	0.5322	3.43	0.5234	3.38
15	A_{67}~A_{70}	0.478	0.4422	2.85	0.4546	2.93
16	A_{71}~A_{72}	0.551	0.5591	3.61	0.5631	3.63
Weight (lb)		364.33	364	1619.07 (N)	364	1619.07 (N)

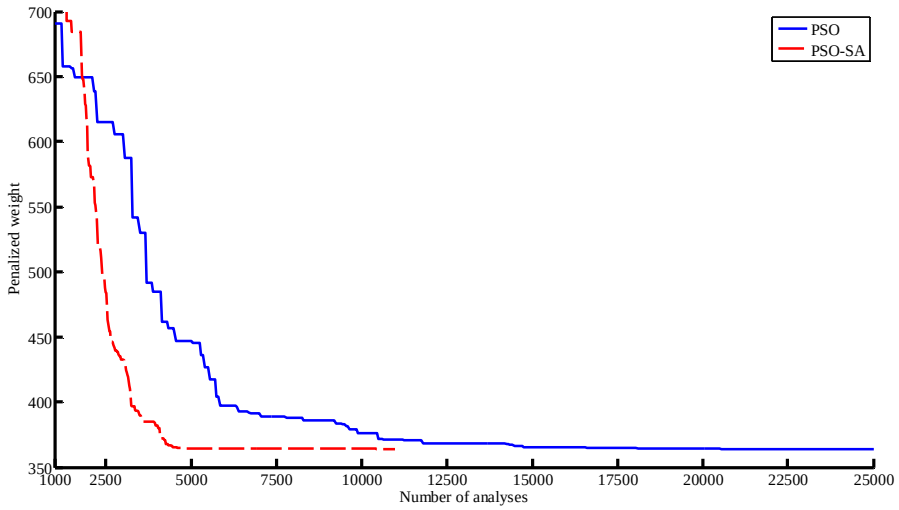


Figure 7. Comparing the convergence rates between the two algorithms for 72-bar truss

5.4. One hundred twenty-bar dome truss

Figure 8 shows the topology and group numbers of a 120-bar dome truss. In this example, two cases of constraints are considered: with stress constraints and displacement constraints of 0.1969 in both x and y directions (Case 1), with stress constraints and displacement limitations of 0.1969 in. (5 mm) imposed on all nodes in x-, y- and z directions (Case 2). For Case 1, the maximum cross-sectional area is 5.0 in (32.26 cm^2) and for Case 2, it is taken as 20.0 in (129.03 cm^2).

Table 7 provides the best solution vectors and the corresponding weights for the cases and Figure 9 shows the convergence histories. In both cases, the PSO-SA needs 7618 analyses to obtain a result which is less than 125,000, 10,000, 10,000 and 35,000 for the PSOPC, HPSACO [33], HBB-BC [32], and HS [30], respectively. Table 8 shows the computational performance of the PSO-based and other heuristic algorithms for this example. It can be seen that the PSO-SA algorithm achieves clearly better results than the PSO.

Table 7. Comparison of optimal designs for the 120-bar truss (Cases 1 and 2)

Optimal cross-sectional areas (in^2)								
Case1								
Element group	**Lee & Geem [30]**	**Kaveh & Talatahari [33]**			**PSO**		**PSO-SA**	
		PSO	**PSOPC**	**HPSACO**	**in^2**	**cm^2**	**in^2**	**cm^2**
1	3.296	3.147	3.235	3.799	3.3183	21.41	3.3183	21.41
2	2.789	6.376	3.370	3.377	2.4749	15.97	2.4749	15.97
3	3.872	5.957	4.116	4.125	4.2899	27.67	4.2899	27.67
4	2.570	4.806	2.784	2.734	2.8110	18.13	2.8110	18.13
5	1.149	0.775	0.777	1.609	0.7750	5	0.7750	5
6	3.331	13.798	3.343	3.533	3.5241	22.73	3.5241	22.73
7	2.781	2.452	2.454	2.539	2.3833	15.38	2.3833	15.38
Weight (lb)	19893.34	32432.9	19618.7	20078.0	19802.77	88082.7	19802.77	88082.7
Case 2								
	Kaveh & Talatahari [33]			**PSO**		**PSO-SA**		
	HBB-BC	PSOPC	HPSACO	in^2	cm^2	in^2	cm^2	
1	3.037	3.040	3.095	3.0252	19.52	3.0249	19.51	
2	14.431	13.149	14.405	14.8108	95.55	14.7900	95.42	
3	5.130	5.646	5.020	5.1531	33.24	5.1240	33.06	
4	3.134	3.143	3.352	3.1345	20.22	3.1342	20.22	
5	8.591	8.759	8.631	8.3977	54.18	8.4236	54.34	
6	3.377	3.758	3.432	3.2933	21.25	3.3095	21.35	
7	2.500	2.502	2.499	2.4955	16.10	2.4957	16.10	
Weight (lb)	33287.9	33481.2	33248.9	33251.95	147904.7(N)	33250.88	147899.9(N)	

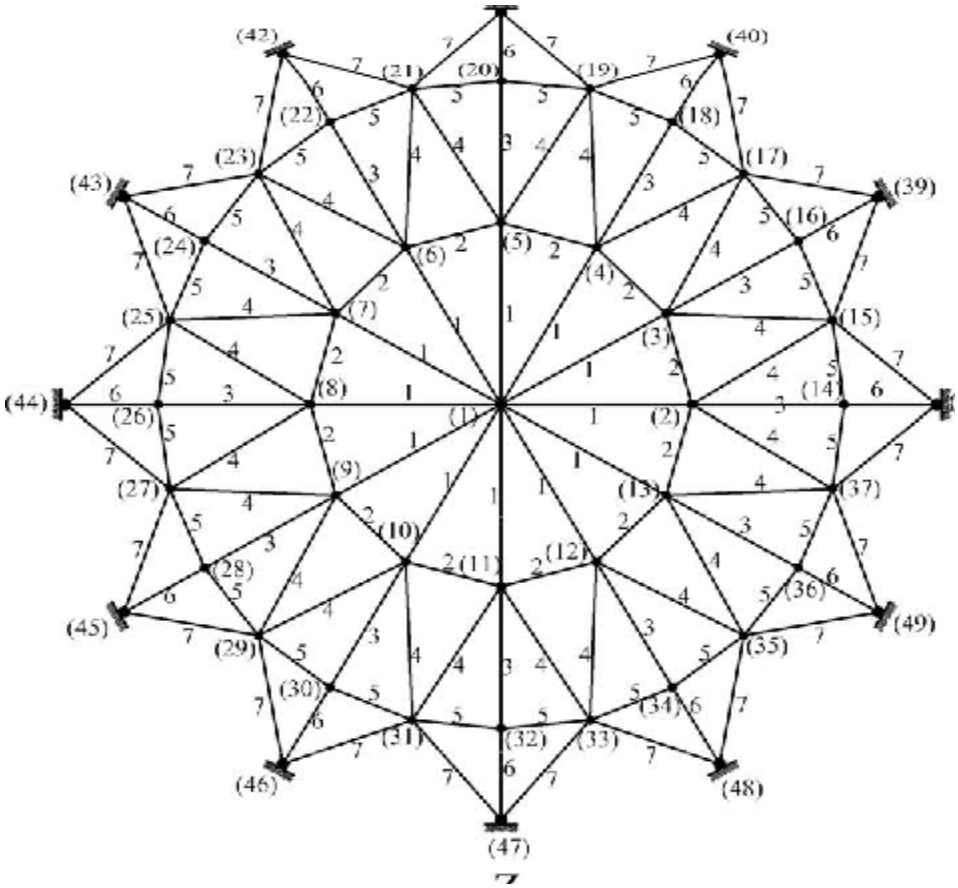
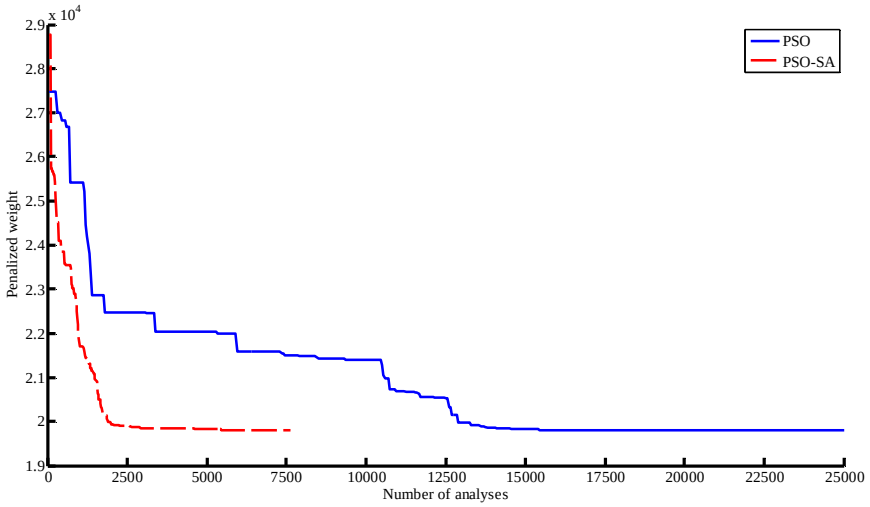


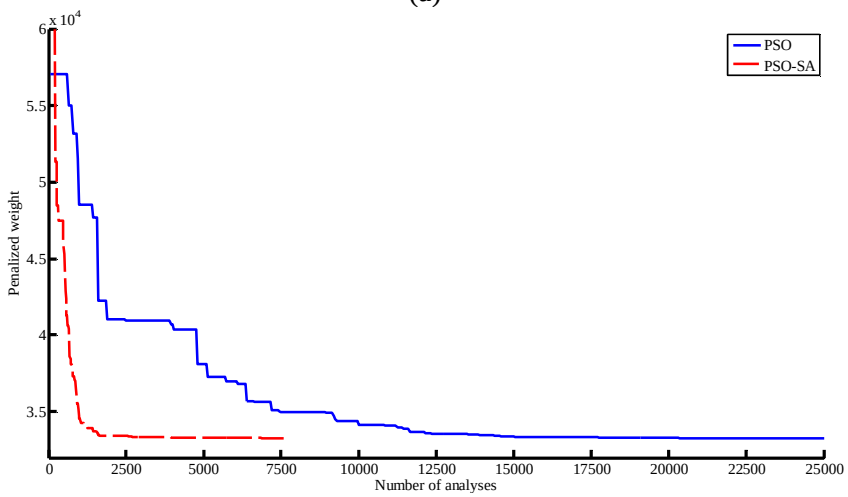
Figure 8. One-hundred-twenty-bar dome truss

Table 8. Performance of PSO and PSO-SA algorithms for 120-bar truss in 50 run

Algorithm	Best	Average	Worst	Best-average difference	Best-worst difference	Standard deviation
			Case 1			
PSO	20422	22423.18	25513.43	9.80 %	24.93 %	1521.73
PSO-SA	19802.77	19817.86	20013.57	0.07 %	1.06 %	46.68
			Case 2			
PSO	33251.96	33666.04	40231.33	1.24 %	21 %	1031.26
PSO-SA	33250.88	33303.47	33469.8	0.16 %	0.66 %	36.54



(a)



(b)

Figure 9. Comparing the convergaence rates between the two algorithms for 120-bar truss: a) Case 1; b) Case 2

6. CONCLUSION

In this paper an efficient optimization algorithm based on the PSO and SA is proposed for optimal design of structures. The new developed optimization algorithm, called the PSO-SA, has capability of finding global optima using few structural analyses. The PSO has high convergence rate, in early iteration. The proposed algorithm makes use of this feature of the PSO algorithm in such a way that in early iterations only the PSO works as a single optimizer and with this strategy the number of structural analyses is reduced further. After that when the optimization process proceeds, the PSO and SA work together. In this stage the PSO finds

optimum region of search space by using the information sharing among its particles and the SA seeks a better solution in this high fitness region. Therefore, instead of using numerous particles to seek the search space, a few numbers of particles is used and the regions with high fitness are identified for more searches. This results in reducing the required number of finite element analyses considerably due to reduction in the number of agents. In order to assess the robustness and effectiveness of the PSO-SA algorithm, some structural optimization problems are presented. The results show that the PSO-SA has efficiency and computational advantageous in comparison with the standard PSO and some other heuristic algorithms. In addition, it should be noted that the reliability of the algorithm to achieve the best solution is increased remarkably in comparison to the PSO algorithm.

REFERENCES

1. Haftka RT, Gurdal Z. *Elements of Structural Optimization*, Kluwer Academic Publishers, 1992.
2. Kaveh A, Talatahari S. An improved ant colony optimization for constrained engineering design problems, *Eng Comput* 2010; **27**(1): 155-82.
3. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**(3-4):267-289.
4. Kaveh A, Talatahari S. Optimal design of Schwedler and ribbed domes via hybrid Big Bang–Big Crunch algorithm, *J Constr Steel Res* 2010; **66**(3):412-9.
5. Saka MP. Optimum design of steel sway frames to BS5950 using harmony search algorithm, *J Constr Steel Res* 2008; **65**(1): 36-43
6. Perez RE, Behdinan K. Particle swarm approach for structural design optimization, *Comput Struct* 2007; **85**(19-20): 1579-88.
7. Tamilarasi A, Kumar TA. An enhanced genetic algorithm with simulated annealing for job-shop scheduling, *Int J Eng Sci Tech*, 2010; **2**(1): 144-51.
8. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing, *Science*, 1983; **220**(4598): 671-80.
9. Cerny V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, *J Optim Theory Appl* 1985; **45**(1) 41-51.
10. Idoumghar L, Melkemi M, Schott R, Aouad MI. Hybrid PSO-SA type algorithms for multimodal function optimization and reducing energy consumption in embedded systems, *Appl Comput Intelligence Soft Comput*, 2011; doi:10.1155/2011/138078
11. Zhao F, Zhang Q, Yu D, Chen X, Yang Y. A hybrid algorithm based on PSO and simulated annealing and its Applications for Partner Selection in Virtual Enterprise, *Proceedings of International Conference on Intelligent Computing* 2005: 380-389.
12. Chaojun D, Zulian Q. Particle swarm optimization algorithm based on the idea of simulated annealing, *Int J Comput Science Network Security* 2006; **6**(10): 152-7.
13. Venter G, Sobieszczanski-Sobieski J. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization, *Struct Multidiscip Optim* 2004; **26**(1-2): 121-31.
14. Chen TY, Su JJ. Efficiency improvement of simulated annealing in optimal structural

designs, *Adv Eng Softw*2002; **33**(7-10): 675-80.
15. Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures, *Asian J Civil Eng* 2008; **9**(4): 329-48.
16. Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput Method Appl Mech Eng* 2002; **191**(11-12): 1245-87.
17. Michalewicz Z. A survey of constraint handling techniques in evolutionary computation methods, *Proceedings of 4th Annual Conference on Evolutionary Programming*, MIT Press, Cambridge , MA,1995, pp. 135-55.
18. Kennedy J, Eberhart RC. Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Network*, 1995, Vol. **4**, pp. 1942-48.
19. 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, pp. 39-43.
20. Eberhart RC, Shi Y. Comparison between genetic algorithms and particle swarm optimization, *Proceedings of the 7th International Conference on Evolutionary Programming* VII, 1998, pp. 611–6.
21. Shi Y, Eberhart RC. A modified particle swarm optimizer, *Proceedings of IEEE International Conference on Evolutionary Computation*, Alaska, 1998, 69-73.
22. Angeline PJ. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference, *Proceedings of Annuale Conference on Evolutionary programming*, San Diego, 1998, pp. 601-10.
23. Shi Y, Eberhart RC. Empirical study of particle swarm optimization, *Proceedings of the 1999 IEEE Congress on Evolutionary Computation* 1999, Vol. **3**, pp. 1945-50.
24. Błachut J, Optimal barreling of steel shells via simulated annealing algorithm, *Comput Struct* 2003; **81**(18-19): 1941-1953.
25. [25] N. Metropolis, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines, *J Chem Phys* 1953; **21**(6): 1087-92.
26. Hasancebi O, Erbatur F. On efficient use of simulated annealing in complex structural optimization problems, *Acta Mech* 2002; **157**(1-4): 27-50.
27. Balling RJ. Optimal steel frame design by simulated annealing, *J Struct Eng ASCE* 1991; **117**(6):1780-95.
28. Khan MR, Wilmert KD, Thronton WA. An optimality criterion method for large-scale structures, *AIAA J 1979*; **17**(7):753-61.
29. Sheu CY, Schmit LA Jr. Minimum weight design of elastic redundant trusses under multiple static load conditions, *Comput Method Appl Mech Eng* 1973; **2**(3): 255-64.
30. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm, *Comput Struct* 2004; **82**(9-10): 781-98.
31. Camp C, Bichon BJ. Design of space trusses using ant colony optimization, *J Struct Eng ASCE* 2003; **130**(5): 741-51.
32. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang–Big Crunch algorithm, *Comput Struct* 2009; **87**(17-18): 1129-40.
33. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Comput Struct* 2009; **87**(5-6):

267-83.