ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 10, NO. 3 (2009)
PAGES 321-334

A REFINEMENT OF DISCRETE PARTICLE SWARM OPTIMIZATION FOR LARGE-SCALE TRUSS STRUCTURES

E. Doğan[a], O. Hasançebi[b] and M. Polat Saka[*c]
[a]Middle East Technical University, Department of Engineering Sciences, 06531 Ankara, Turkey
[b]Middle East Technical University, Department of Civil Engineering, 06531 Ankara, Turkey

ABSTRACT

This paper proposes a refined version of particle swarm optimization technique for the optimum design of steel structures. Swarm is composed of a number of particles and each particle in the swarm represents a candidate solution of the optimum design problem. Design constraints in accordance with ASD-AISC (Allowable Stress Design Code of American Institute of Steel Institution) are imposed by the particle swarm optimization based optimum design algorithm developed. A constraint handling method called the 'penalty function method' is introduced to maintain acceptable solutions. The refined version of the particle swarm optimization algorithm proposed in this paper is easy to implement and the results and convergence performance are better than the simple particle swarm optimization algorithm and some other meta-heuristic optimization techniques. The effect of different inertia weight parameters in finding the optimum design is also tested in two numerical examples.

Keyword: Meta-heuristic optimization; discrete particle swarm; trusses; discrete particle swarm; steel structures

1. INTRODUCTION

Metaheuristic optimization techniques transform ideas taken from nature such as survival of the fittest, immune system or cooling of molten metals through annealing into a numerical optimization algorithms [1-3]. These techniques are shown to be quite effective in finding the optimum solution of optimization problems where the design variables are discrete [4]. Saka [5] carries out a detailed review of these techniques and their applications. Particle swarm optimization technique is one of the recent additions to the meta-heuristic optimization techniques. It is based on the swarm intelligence [6]. In nature fish school, birds flock and bugs swarm not only for reproduction but for other reasons such as finding food and escaping predators. There are implicit rules that each member of bird flock and fish school has to abide by so that they can move in a synchronized way without colliding. Each

[*]Email address of corresponding author: mpsaka@metu.edu.tr (M.P. Saka)

individual in a flock maintains optimum distance from the neighboring individuals so that the flock can move smoothly from one place to another. Particle swarm optimizer is a simulator of social behavior that is used to realize the movement of a birds' flock. It is population based optimization algorithm. Its population is called a swarm and each individual in the swarm is called a particle. Each particle flies through the problem space to search for optimum. In this study optimum design algorithm based on the particle swarm optimizer is presented that determines the optimum W- sections for the load carrying members of truss from the steel section list of ASD-AISC (Allowable Stress Design Code of American Institute of Steel Institution) [7].

Particle swarm optimization technique [8-15] is originally formulated as a continuous optimization technique. According to this algorithm, which is originally developed by Kennedy and Eberhart [8], real numbers are assumed as design variables. This assumption is utilized in most of the applications of the particle swarm optimization algorithm to the structural optimization problems in the literature. Continuous applications of particle swarm optimization have been reported in Tasgetiren et al. [9], He et al. [10] and Arumugam et al [11]. Very few studies considered the discrete valued variables in PSO algorithm. However many optimization problems such as steel design problems using ready steel sections do need discrete set. There are two approaches that can deal with discrete variables. These are binary numbers technique and rounding off method that can be used to obtain integer numbers from continuous ones. Kennedy and Eberhart [12] have been the first researchers using binary numbers in particle swarm optimization to achieve discrete set. Liu et al.[13] used rounding off method in their research. In the present study, the rounding off method is implemented due to its simplicity. However, it is noticed that this technique raise problems in the optimum design of large-scale steel frames. To overcome this problem, a refined version of particle swarm optimization method is developed and presented. This refined PSO algorithm is compared with simple PSO, simple genetic algorithm and ant colony optimization techniques in two real size truss problems.

2. FORMULATION OF DISCRETE OPTIMUM DESIGN FOR TRUSS STRUCTURES

Formulation of the optimum design problem according to Allowable Stress Design Code (ASD-AISC) [7] for a pin jointed steel structure consisting of N_m members that are grouped into N_d design variables can be expressed as the following discrete programming problem.

$$\text{Minimize } W = \sum_{m=1}^{N_m} \rho_m L_m \tag{1}$$

In Eqn (1) W is the weight of the structure ρ_m *and* L_m are the unit weight and the length of member *m*, respectively. The objective is to find standard steel sections for the members of a steel frame such that the overall weight of the frame becomes the minimum. For the determination of such a section belonging to each member of structure, algorithm necessitates the selection of ready made steel sections which are sequenced with the corresponding integer values in a given profile list. Hence the sequence number of steel sections in the profile list is adopted as design variable. At the optimum, the algorithm developed comes up with the sequence numbers of steel sections that are to be adopted for the frame members so that the frame has the minimum weight.

It is the merit of the study that while minimizing Eqn (1), it is also necessary to satisfy the following restrictions:

$$\sigma_m \le (\sigma_m)_{all}, \quad m = 1,.., N_m \tag{2}$$

$$\lambda_m \le (\lambda_m)_{all}, \quad m = 1,.., N_m \tag{3}$$

$$d_{j,k} \le (d_{j,k})_{all}, \quad j = 1,.., N_j \tag{4}$$

Eqn (2) represents stress check for the members subjected to compression or tension. σ_m and $(\sigma_m)_{all}$ are the computed and allowable axial stresses for the *m*-th member, respectively.

Eqn (3) and Eqn (4) are restrictions being bounds on slenderness ratios and displacements, respectively; λ_m and $(\lambda_m)_{all}$ are the slenderness ratio and its upper limit for *m*-th member, respectively; $d_{j,k}$ and $(d_{j,k})_{all}$ are the displacements computed in the *k*-th direction of joint *j* and its permissible value, respectively, finally N_j is the total number of joints. The maximum slenderness ratio is limited to 300 and 200 for tension members and compression members respectively due to ASD-AISC [7] design code.

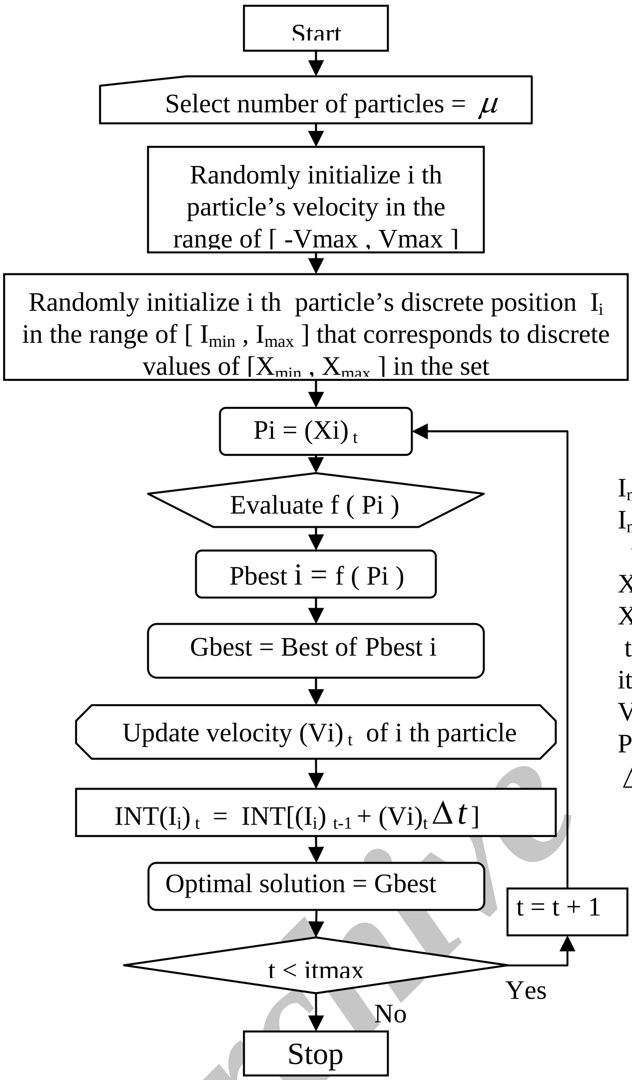
A detailed review of all constraint-handling approaches is presented in [16, 17]. In this study penalty function method is used for handling the design constraints mentioned above. If constraint violations are encountered frequently in an optimization problem, it is very challenging for the process to continue. In such a case, it may be more beneficial to keep a slightly infeasible particle in the solution. These particles having one or more constraints slightly infeasible are utilized in the design process that might provide a new particle that may be feasible. Compatible with this idea, penalty function method is a powerful handling method having been a part of the literature on constrained optimization for decades. It is easy to implement and efficient with a proper parameterization. For the present study a penalty integrated objective function is defined to evaluate infeasible design(s) in proportion to the sum of the constraint violation, as formulated in Eqn. (5).

$$f = W\left[1 + \alpha\left(\sum_{m=1}^{N_m} g_m + \sum_{m=1}^{N_m} \lambda_m + \sum_{j=1}^{N_j}\sum_{k=1}^{3} d_{j,k}\right)\right] \tag{5}$$

where f represents the constrained objective function, and α is referred to as the penalty coefficient, used to tune the intensity of penalization as a whole.

3. DISCRETE PARTICLE SWARM OPTIMIZER

Particle swarm optimization is a population based stochastic optimization technique which is inspired by social behavior of bird flocking or fish schooling. This behavior is concerned with grouping by social forces that depend on both the memory of each individual as well as the knowledge gained by the swarm. The procedure involves a number of particles which represents the swarm which are initialized randomly in the search space of an objective function. Each particle in the swarm represents a candidate solution to the optimum design problem. Originally particle swarm optimizer is developed for continuous design variables. To be able to use the method for discrete design variables some adjustments are required to be carried out. Firstly the discrete values among which the values of design variables are to be selected in set are arranged in ascending sequence. The sequence number of these values is then treated as design variables. At any stage of design cycle, once a sequence number is generated by the algorithm, then the real value of the design variable is easily taken from the discrete set. The steps of the algorithm are outlined in the following and the flowchart is given in Figure 1.



I_{min} =1
I_{max} : Last sequence number in the discrete set
X_{min} : lower bound of variables
X_{max} : upper bound of variables
t : cycle number
itmax : total cycle number
V_{max} : maximum velocity
Pi :Particles' discrete position
Δt : time step

Figure 1. Simple particle swarm optimizer for discrete variable

Step 1. Initializing Particles: A swarm is composed of a pre-selected number of particles called as swarm size (μ). A position (design) vector **I** and a velocity vector **v** (Eqn. 6) are two set of components that each particle should have. The positions of design variables are retained by the position vector **I**, while the velocity vector **v** is used to change these positions during the search. Random initialization is used to set up each particle in the swarm such that all initial positions $I_i^{(0)}$ and velocities $v_i^{(0)}$ are assigned from Eqns. (7-8):

$$\mathbf{P} = (\mathbf{I}, \mathbf{v}), \quad \mathbf{I} = [I_1, I_2, ..., I_{N_d}] \quad , \quad \mathbf{v} = [v_1, v_2, ..., v_{N_d}] \tag{6}$$

$$I_i^{(0)} = I_{\min} + r(I_{\max} - I_{\min}), \quad i = 1,.., N_d \tag{7}$$

$$v_i^{(0)} = \frac{I_{\min} + r(I_{\max} - I_{\min})}{\Delta t}, \quad i = 1,.., N_d \tag{8}$$

In Eqns. (7-8), $I_{\min}$ and $I_{\max}$ are the sequence numbers of the first and last standard steel sections in the profile list, respectively, r represents a random number between 0 and 1; Δt is referred to as the time step increment.

Step 2. Evaluating Particles: All the particles are analyzed, and using (Eqn 5) their objective function values are calculated.

Step 3. Updating the Particles' Best and the Global Best: Particle's best (Pbest) refers to the particle's best position which is the best design having minimum objective function during iteration so far. Each particle has a vector $\mathbf{B}$ containing the particle's best. Another vector $\mathbf{G}$ stores the best feasible design obtained by any particle since the beginning of the process, which is the global best position (gbest). Both the particles' bests and the global best are updated at the current iteration k.

$$\mathbf{B}^{(k)} = \left[B_1^{(k)},...B_i^{(k)}..,B_{N_d}^{(k)}\right] \ \mathbf{G}^{(k)} = \left[G_1^{(k)},...G_i^{(k)}..,G_{N_d}^{(k)}\right] \tag{9}$$

Step 4. Updating a Particle's Velocity Vector: (Eqn. 10) updates the velocity vector of each particle considering the particle's current position, the particle's best position and global best position.

$$v_i^{(k+1)} = wv_i^{(k)} + c_1 r_1 \left(\frac{G_i^{(k)} - I_i^{(k)}}{\Delta t}\right) + c_2 r_2 \left(\frac{B_i^{(k)} - I_i^{(k)}}{\Delta t}\right) \tag{10}$$

where, w is the inertia of the particle which controls the exploration properties of the algorithm; r_1 and r_2 are random numbers between 0 and 1; and c_1 and c_2 are the trust parameters, indicating how much confidence the particle has in itself and in the swarm, respectively.

Step 5. Updating a Particle's Position Vector: Using the updated velocity vector, the position vector of each particle is updated (Eqn. 11), which is rounded to nearest integer value for discrete variables.

$$I_i^{(k+1)} = I_i^{(k)} + v_i^{(k+1)} \Delta t \tag{11}$$

Step 6. Termination: The steps 2 through 5 are repeated in the same way for N_{ite} iterations.

4. REFINED PARTICLE SWARM OPTIMIZATION (RPSO)

Particle swarm optimization technique based optimum design algorithm necessitates to update the positions of all the particles using Eqns.(8-9). During the procedure particles' velocities and positions change and these changes lead to revisions of particle and global bests. Numerical applications show that, in large-scale structural optimization problems all the particles in the swarm are eventually dragged to the position identified by the global best. Therefore, the current and best positions of all the particles become identical to the global best, resulting in almost zero velocity vectors. In such a case, particles cannot fly anymore and the search gets stuck in a very poor design point. The present study considers the reformulated version of Eqn. (8) as in Eqns. (10-11) where an additional velocity term is defined and added to let the particle move randomly in certain directions in the close neighborhood of its current position.

$$v_i^{(k+1)} = wv_i^{(k)} + c_1 r_1 \left(\frac{G_i^{(k)} - I_i^{(k)}}{\Delta t} \right) + c_2 r_2 \left(\frac{B_i^{(k)} - I_i^{(k)}}{\Delta t} \right) + \hbar_i r_3 \frac{\sqrt{N_s}}{\Delta t} \tag{12}$$

$$\hbar_i = \begin{cases} 1 & if \;\; r \le \dfrac{1}{2N_d} \\ 0 & if \;\; r > \dfrac{1}{2N_d} \end{cases} \tag{13}$$

where, r_3 is a random number between 0 and 1; N_s is referred to as the number of steel sections in the profile list; and $\hbar_i$ is 0-1 heaviside function implemented by sampling a random number r between 0 and 1. Equation (12) implies that by means of a random velocity term added, in every two iterations only one design variable represented by a particle is allowed to change its position to new one. This makes it possible for particles to continue the search for optimum. The reformulated equation has been observed to eliminate the aforementioned drawback and greatly improve the efficiency of the technique. The improvements in the technique are demonstrated by numerical examples solved.

5. NUMERICAL EXAMPLES

The design algorithms are computerized in four design softwares that are all compiled in Borland Delphi source code. These design algorithms are based on ant colony optimization (ACO), simple genetic algorithm (SGA), standard particle swarm optimizer (PSO) and refined particle swarm optimizer RPSO). These softwares are automated to interact with SAP2000 v7.4 structural analysis program for generating and screening the structural models of the problems under consideration as well as carrying out a displacement based finite element analysis for each solution sampled during optimization process.

Two design examples namely 230-member plane truss bridge and 564-member space truss tower are solved to test the performance of proposed algorithm for the particle swarm optimizer and compare the results with the ones obtained from ant colony optimization and simple genetic algorithm techniques [18-21]. The parameters used are listed in Table 1. Each example is independently designed three times with each technique, resulting in a different final design in each run due to stochastic natures of the techniques. The results obtained for both examples are carried out and the design history graphs demonstrating the improvement of the feasible best design in the search process with all the techniques are plotted. The number of structural analysis performed (the number of design points sampled) is shown in the horizontal axis of these graphs, whereas the vertical axis represents the variation of the best feasible design weight obtained thus far during the search. The number of structural analysis is taken as 50,000 to make sure that all the meta-heuristic techniques are given the equal opportunity to grasp the global optimum, and that it is not a restraint for not being able to reach the global optimum. In all the examples, the following material properties of the steel are used: modulus of elasticity (*E*) = 29000ksi (203893.6MPa) and yield stress (F_y) = 36ksi (253.1MPa).

Table 1. The parameter data set.

Values of parameter set for particle swarm optimizer		
$\mu = 50$	$\Delta t = 1.0$	$c_1 = 1.5$
$w = 0.25$	$N_{ite} = 1000$	$c_2 = 1.5$

5.1 230-member plane truss bridge

The geometry of a 230-member, two-span bridge with a total length of 380ft (115.824m) is given in Figure 2. The 230 truss members are grouped into 48 independent design variables considering the symmetry about centerline. The grouping of members is also shown in Figure 2. A single design loading is considered such that traffic loads combined with dead loads of the floor system have resulted in an equivalent point load of 80kips (355.86kN) at each panel point on the lower chord. A discrete set of 137 economical standard steel sections selected from W-shape profile list based on area and radii of gyration properties is used to size the variables. The lower and upper bounds on size variables are taken as 6.16in^2 (39.74cm^2) and 215.0in^2 (1387.09cm^2), respectively. The stress and stability limitations of the members are calculated according to the provisions of ASD-AISC [7]. In addition, the displacements of all nodes in any direction are restricted to a maximum value of 1.44in (3.658cm) which is equal to 1/400 of maximum height of the bridge.

The 230-member plane truss bridge is separately designed three times by ant colony optimization (ACO), simple genetic algorithm, standard particle swarm optimizer and refined particle swarm optimizer. The design history of these runs is shown in Fig. 3. The best runs obtained from for these four softwares are given in Table 2. It is apparent from this table that the improved particle swarm optimizer produced the lightest truss that has the minimum weight of 309791.83lb. This result is 3%, 5.6% and 43% lighter than the one obtained with ant colony, genetic algorithm and simple particle swarm optimizer respectively. The best design obtained by the refined particle swarm optimizer is tabulated in Table 3 with section designations attained for each member group, and is considered to be the optimum solution of the problem reached in the present study.

Table 2. The best runs obtained with for algorithms for 230-member plane truss bridge and 564-member space truss tower.

Designs of three different runs	**RPSO**	**ACO**	**SGA**	**PSO**
Best runs for 230 member truss bridge	309791.83	318042.73	328209.92	545257.22
Best runs for 564 member truss tower	910354.08	924059.50	1015819.65	1715803.46

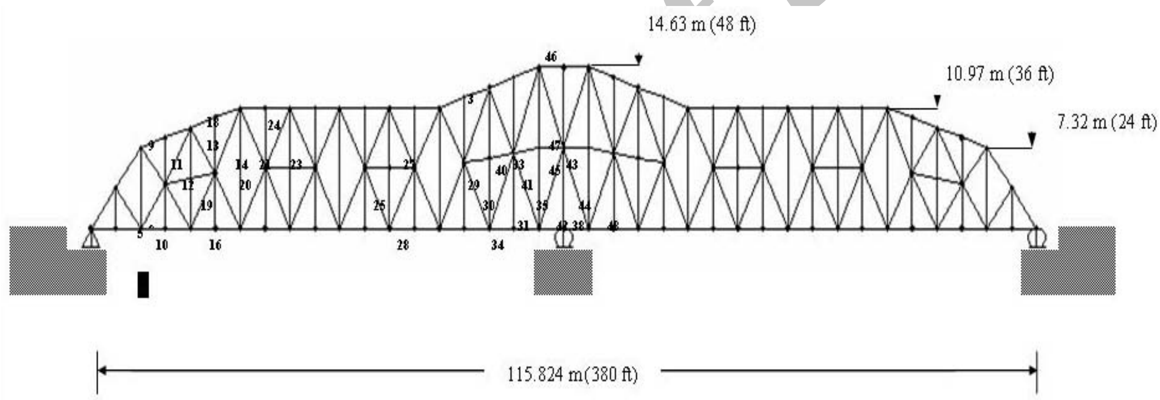


Figure 2. 230-member plane truss bridge

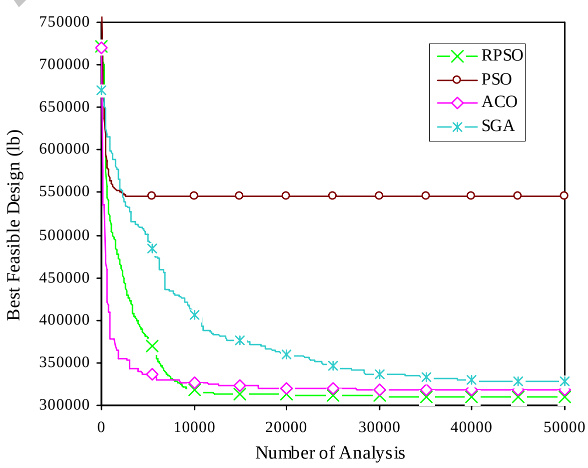


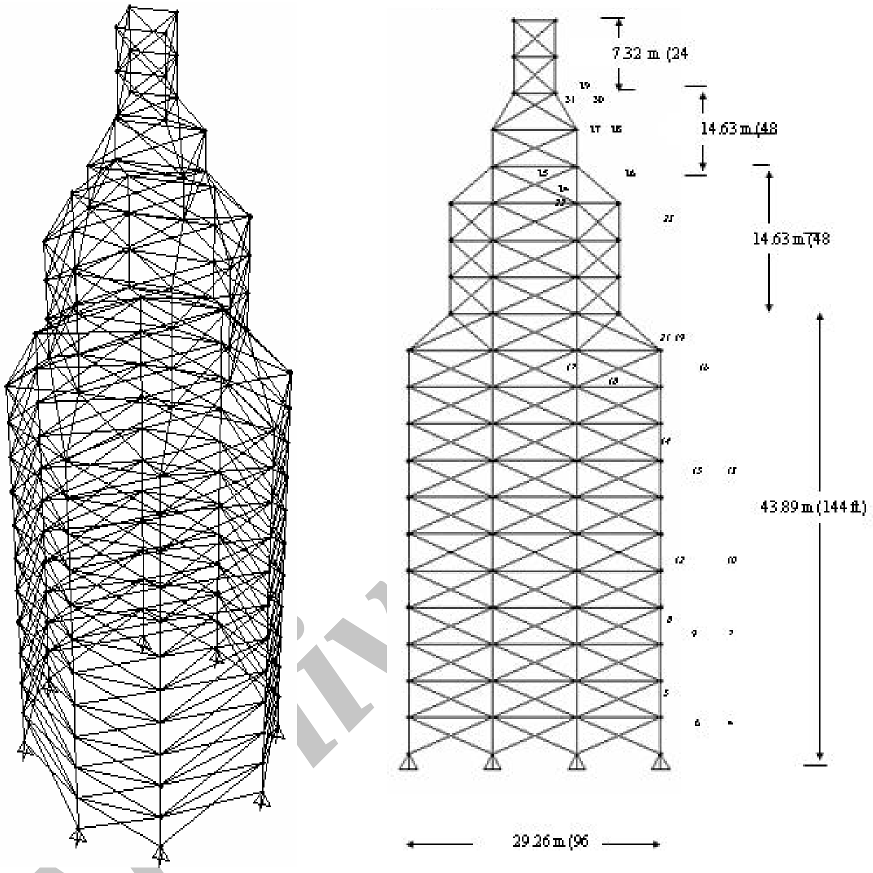
Figure 3. The design history graph obtained with meta-heuristic techniques for the 230-member plane truss bridge.

Table 3. The optimum design obtained with RPSO for the 230-member plane truss bridge.

Size Variable	Ready Section	Area, cm^2 (in^2)	Size Variable	Ready Section	Area, cm^2 (in^2)
1	W30X211	401 (62.16)	25	W24X104	197 (30.54)
2	W27X178	337 (52.24)	26	W12X79	150 (23.2)
3	W8X21	40 (6.2)	27	W8X21	40 (6.2)
4	W8X24	45.8 (7.01)	28	W8X21	40 (6.2)
5	W21X132	250 (38.8)	29	W8X24	45.8 (7.01)
6	W12X53	101 (15.7)	30	W21X122	231 (35.81)
7	W30X191	362 (56.11)	31	W14X132	250 (38.75)
8	W10X54	102 (15.81)	32	W10X112	213 (33.02)
9	W12X190	360 (55.8)	33	W8X21	40 (6.2)
10	W8X21	40 (6.2)	34	W8X21	40 (6.2)
11	W12X40	75.9 (11.77)	35	W21X111	211 (32.71)
12	W8X21	40 (6.2)	36	W14X233	442 (68.51)
13	W8X21	40 (6.2)	37	W24X117	222 (34.41)
14	W8X21	40 (6.2)	38	W24X117	222 (34.41)
15	W10X112	213 (33.02)	39	W40X221	400 (62)
16	W8X21	40 (6.2)	40	W8X21	40 (6.2)
17	W8X21	40 (6.2)	41	W8X21	40 (6.2)
18	W12X190	360 (55.8)	42	W8X21	40 (6.16)
19	W8X21	40 (6.2)	43	W8X21	40 (6.2)
20	W8X24	45.8 (7.01)	44	W8X24	45.8 (7.01)
21	W10X45	85.5 (13.25)	45	W24X162	308 (47.78)
22	W30X191	362 (56.11)	46	W40X324	614.8 (95.3)
23	W8X21	40 (6.16)	47	W8X21	40 (6.2)
24	W12X45	85.1 (13.19)	48	W33X424	800 (124)
		Weight **140500 kg** (309791,83lb)			

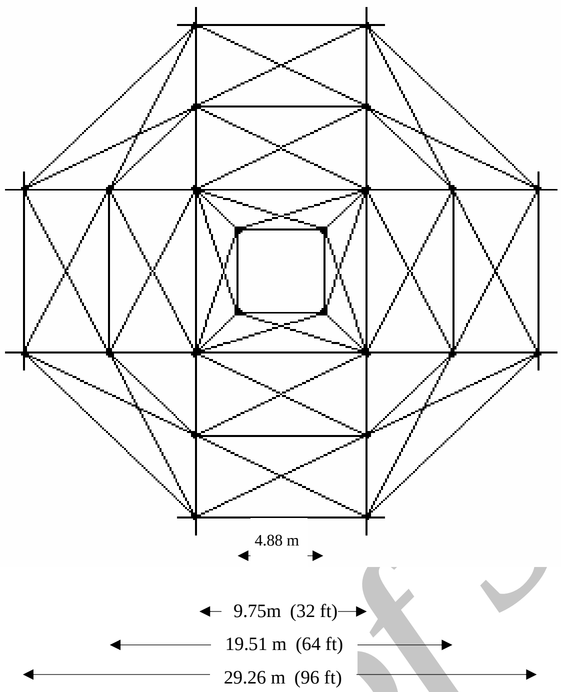
5.2 564-member space truss tower

564-member space truss tower is considered as the second numerical example. Figure 4 shows the geometry of the structure.. The symmetry of the tower around *x* and *y*-axes is considered to group the 564 members into 31 independent size variables. A single load case is considered such that it consists of lateral loads of 4.45kN (1 kips) applied in both x and y directions and a vertical load of -13.35kN (-3 kips) applied in the *z* direction at all nodes of the tower. A discrete set of 137 economical standard steel sections selected from W-shape profile list based on area and radii of gyration properties is used to size the variables. The lower and upper bounds on size variables are taken as 6.16in^2 (39.74cm^2) and 215.0in^2 (1387.09cm^2), respectively. The stress and stability limitations of the members are imposed according to the provisions of ASD-AISC [7]. In addition, the displacements of all nodes are limited to 7.315 cm (2.88 in) in any direction. The bests of three designs obtained by each meta-heuristic technique are given in Table 2. Results show that the same as the previous example minimum weight for truss tower is obtained with refined PSO algorithm, which is 910354.08lb. This is 1.5%, 10.38% and 46.9% lighter than the design produced by ant colony, genetic algorithm and simple particle swarm optimizer respectively. The design history graph is shown in Figure 5 and the sections which belong to this design are designated in Table 4.



a) 3D view

b) side view



c) side view

Figure 4. 564-member space truss tower a) 3D view b) top view c) side view

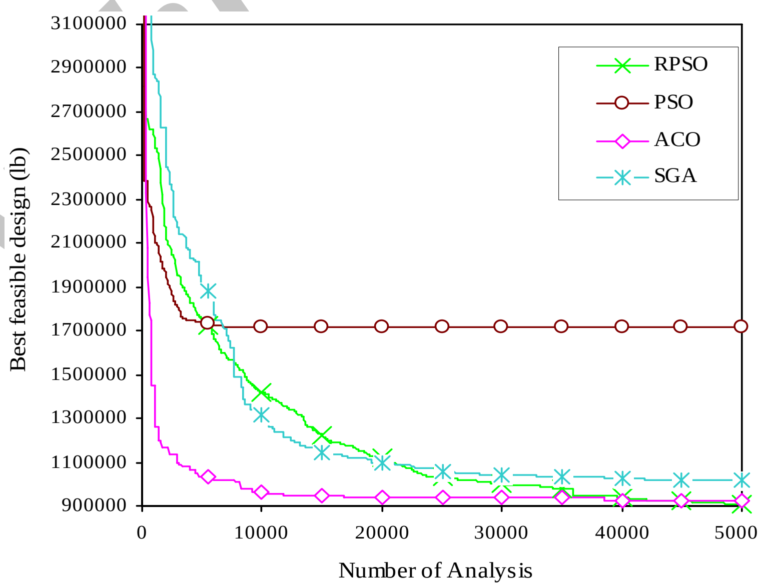
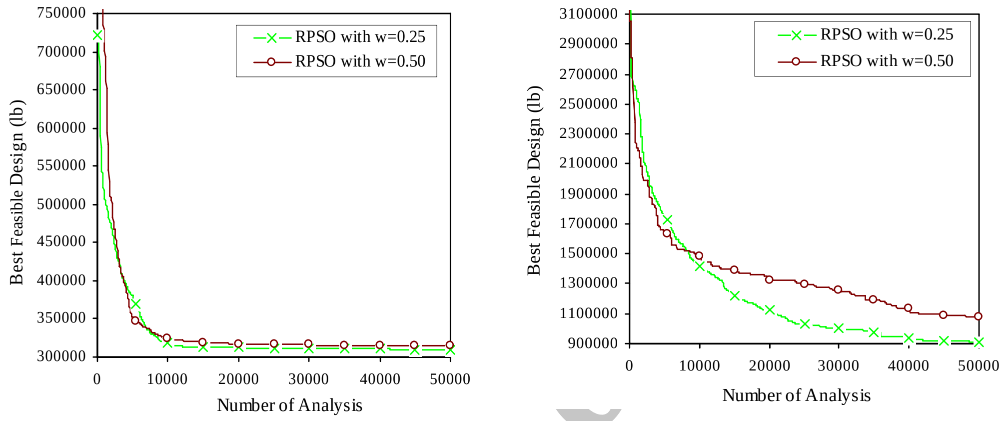


Figure 5. The design history graph obtained with meta-heuristic techniques for 564-member space truss tower

Two examples are also designed with different inertia weight parameters and it is observed that it is highly effective in finding the optimum weight. Two static inertia parameters are selected as 0.25 and 0.5. It is observed that the best design obtained with (w=0.5) is 314719.90lb which is 1.57% heavier than the one with (w=0.25). The second example namely 564 bar truss tower is also designed with both inertia parameters. Similar to the first example, lower one shows better performance that is the optimum weight with (w=0.5) is 1078196.72lb while the one attained with (w=0.25) is 910354.08lb which is 15.56% lighter. Results obtained during these trials are demonstrated Figure 6.



a) 230-bar truss bridge   b) 564-bar truss tower

Figure 6. The design history graph obtained with different inertia weights.

Table 4. The optimum design obtained with RPSO for 564-member space truss tower

Size Variable	Ready Section	Area, cm^2 (in^2)	Size Variable	Ready Section	Area, cm^2 (in^2)
1	W8X31	58.6 (9.08)	17	W8X31	58.6 (9.08)
2	W12X65	123 (19.07)	18	W8X24	45.8 (7.1)
3	W12X65	123 (19.07)	19	W8X21	40 (6.2)
4	W8X28	53.1 (8.23)	20	W8X31	58.6 (9.08)
5	W12X72	136 (21.08)	21	W8X48	91.1 (14.12)
6	W12X65	123 (19.07)	22	W10X49	92.8 (14.38)
7	W8X24	45.8 (7.1)	23	W8X21	40 (6.2)
8	W12X65	123 (19.07)	24	W8X31	129.03 (20.0)
9	W12X65	123 (19.07)	25	W8X48	85.81 (13.3)
10	W8X21	40 (6.2)	26	W8X21	40 (6.2)
11	W12X65	123 (19.07)	27	W8X31	58.6 (9.08)
12	W12X65	123 (19.07)	28	W8X21	40 (6.2)
13	W8X21	40 (6.2)	29	W8X21	40 (6.2)
14	W12X65	123 (19.07)	30	W8X21	40 (6.2)
15	W12X65	123 (19.07)	31	W8X21	40 (6.2)
16	W10X49	92.8 (14.38)			
		Weight **412900 kg** (910354.08lb)			

6. CONCLUSIONS

In this paper the particle swarm optimizer is refined so that it can effectively be used in the optimum design of large size pin jointed structures. It is shown that the refined design algorithm is mathematically quite simple but effective in finding the solutions of large size structural optimization problems. The optimum design algorithm presented selects optimum W-sections from American steel sections table for the members of trusses such that design constraints described in ASD-AISC are satisfied and the structure has the minimum weight. The results obtained from the optimum design of 230-member and 564-member trusses revealed the fact that the refined particle swarm optimizer performs better than ant colony optimizer and simple genetic algorithm. The minimum weights attained by the refined particle swarm algorithm in both cases are less than the ones obtained from ant colony and simple genetic algorithm. The tests on different inertia parameters imply that this parameter has important role in finding the optimum.

REFERENCES

1. Kochenberger GA, Glover F. *Handbook of Meta-Heuristics*, Kluwer Academic Publishers, The Netherlands, 2003.
2. Dreo J, Petrowski A, Siarry P, Taillard E. *Meta-Heuristics for Hard Optimization*, Springer-Verlag, Berlin, Heidelberg, 2006.
3. Horst R, Pardolos PM. (Editors), *Handbook of Global Optimization*, Kluwer Academic Publishers, The Netherlands, 1995.
4. Arora, JS. *Methods for Discrete Variable Structural Optimization, Recent Advances in Optimum Structural Design*, Ed. S.A. Burns, ASCE, 1-40, USA, 2002.
5. Saka, M.P., Optimum design of steel frames using stochastic search techniques based on natural phenomena: A Review, Civil Engineering Computations: Tools and Techniques, Ed. B.H.V. Topping, Saxe-Coburgh Publications, 2007, pp. 105-47.
6. Kennedy J, Eberhart R, Shi Y. *Swarm Intelligence*, Morgan Kaufmann Publishers, USA, 2001.
7. *Manual of Steel Construction, Allowable Stress Design*, 9th edition, AISC, American Institutes of Steel Construction, Inc, Chicago, Illinois, USA, 1989.
8. Kennedy J, Eberhart R. Particle Swarm Optimization, IEEE International Conference on Neural Networks, IEEE Press, Vol. 4, 1995, pp. 1942-8.
9. Tasgetiren MF, Liang YC, Sevkli M, Gencyilmaz G. Particle Swarm Optimization Algorithm for Makespan and Maximum lateness minimization in Permutation Flowshop Sequencing problem, In: Proceedings of the Fourth International Symposium on Intelligent Manufacturing Systems, Turkey: Sakarya; 2004, pp. 431-41.
10. He S, Prempain E, Wu QH. An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems, Engineering Optimization, No. 5, **36**(2004)585-605.
11. Arumugam MS, Rao MVC, Chandramohan A. A new and improved version of particle swarm optimization algorithm with global–local best parameters, Knowledge and Information Systems. **16**(3):2008:331-57.

12. Kennedy J, Eberhart RC. A Discrete Binary Version of the Particle Swarm Algorithm, In Proceedings of the World Multi-Conference on Systemic, Cybernetics and Informatics, pp. 4104-49, NJ: Piscatawary; 1997.
13. Liu H, Shichang S, Ajith A. Particle swarm approach to scheduling work-flow applications in distributed data-intensive computing environments, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06) 0-7695-2528-8/06.
14. Fourie P, Groenwold A. The particle swarm optimization algorithm in size and shape optimization, *Structural and Multidisciplinary Optimization*, No. 4, **23**(2002)259-67.
15. Parsopoulos KE, Vrahatis MN. Particle Swarm Optimization Method for Constrained Optimization Problems, P Sincak, J Vascak, V Kvasnicka, J Pospichal (eds.), Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies, pp. 214-20, IOS Press (Frontiers in Artificial Intelligence and Applications series, Vol. 76), 2002, ISBN: 1-58603-256-9.
16. Michalewicz Z. A survey of constraint handling techniques in evolutionary computations methods, 4th Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, 1995, pp. 135-55.
17. Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering* **191**(2002)1245-87.
18. Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, USA, 1989.
19. Mitchell M., *An Introduction to Genetic Algorithms*, The MIT Press, 1998.
20. Dorigo M, Stützle T. *Ant Colony Optimization*, A Bradford Book, Massachusetts Institute of Technology, USA, 2004.
21. Camp CV, Bichon JB, Stovall SP. Design of steel frames using ant colony optimization, *Journal of Structural Engineering, ASCE*, No. 3, **131**(2004)369-79.