ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 9, NO. 4 (2008)
PAGES 329-348

# A HYBRID PARTICLE SWARM AND ANT COLONY OPTIMIZATION FOR DESIGN OF TRUSS STRUCTURES

A. Kaveh[*a] and S. Talatahari[b]
[a]*Department of Civil Engineering, Iran University of Science and Technology, Tehran-16, Iran*
[b]*Department of Civil Engineering, University of Tabriz, Tabriz, Iran*

## Abstract

This paper presents a particle swarm ant colony optimization for design of truss structures. The algorithm is based on the particle swarm optimizer with passive congregation and ant colony optimization. The particle swarm ant colony optimization applies the particle swarm optimizer with passive congregation for global optimization and ant colony approach is employed to update positions of particles to attain rapidly the feasible solution space. Ant colony optimization works as a local search, wherein, ants apply pheromone-guided mechanism to update the positions found by the particles in the earlier stage. A new relation is defined for the inertia weight, and the terminating criterion is changed in the way that after decreasing the movements of particles, the search process stops. With these changes, the number of iterations does not increase. The proposed method is tested on several benchmark trusses from literature. The result comparisons with particle swarm optimizer, particle swarm optimizer with passive congregation and other optimization algorithms demonstrate the effectiveness of the presented method.

**Keywords:** Truss; optimization; particle swarm; ant colony; hybrid

## 1. Introduction

Since the material cost is one of the major factors in the construction of a building, it is preferable to reduce it by minimizing the weight or volume of the structural system. All of the methods used for minimizing the volume or weight intend to achieve an optimum design having a set of design variables under certain design criteria [1].

Size optimization of truss structures involves determining optimum values for member cross-sectional areas $A_i$ that minimize the structural weight $W$. This minimum design also has to satisfy inequality constraints that limit design variable sizes and structural responses [2]. Thus, the optimal design of a truss is formulated as

$$\text{Minimize} \quad W(\{x\}) = \sum_{i=1}^{n} \gamma_i \cdot A_i \cdot L_i \tag{1}$$

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

$$\text{subject to}: \quad \begin{aligned} &\delta_{\min} \le \delta_{\mathrm{i}} \le \delta_{\max} && i = 1,2,....,m \\ &\sigma_{\min} \le \sigma_{\mathrm{i}} \le \sigma_{\max} && i = 1,2,....,n \\ &\sigma_i^b \le \sigma_{\mathrm{i}} \le 0 && i = 1,2,....,nc \\ &A_{\min} \le A_i \le A_{\max} && i = 1,2,....,ng \end{aligned}$$

where $W(\{x\})$=weight of the structure; $n$=number of members making up the structure; $m$=number of nodes; $nc$=number of compression elements; $ng$= number of groups (number of design variables); $\gamma_i$ =material density of member $i$; $L_i$= length of member $i$; $A_i$=cross-sectional area of member $i$ chosen between $A_{min}$ and $A_{max}$; $min$ = lower bound and $max$= upper bound; $\sigma_i$ and $\delta_i$ = the stress and nodal deflection, respectively; $\sigma_i^b$ = allowable buckling stress in member $i$ when it is in compression.

Traditional mathematical programming methods such as the Lagrange multiplier methods [3] usually require the derivative information of the objective function and constraints. Besides, the obtained solution often tends to be a local optimum unless the search space is convex. In recent years, evolutionary algorithms (EAs) have attracted much attention for a variety of optimization problems due to their superior advantages. EAs do not require the objective function to be derivable or even continuous, and EAs perform as global optimization techniques due to the appropriate balance between the exploration and exploitation of the whole search space.

Genetic algorithm is one of the EA types initially suggested by Holland, and developed and extended by some of his students, Goldberg and Ann Arbor. These algorithms simulate natural genetics mechanism for synthetic systems based on operators that are duplicates of natural ones. In the last decade, GA is used in the optimum structural design. One of the first applications was the weight minimization of a 10-bar truss by Goldberg and Samtani [4]. Hajela [5], and Kaveh and Kalatjari [6], among many others, used genetic search in design of various structures in which the search space was non-convex or discrete.

Two new evolutionary algorithms are Particle Swarm Optimizer (PSO) and Ant Colony Optimization (ACO) that are used in structural optimization problems. He et al. [7,8] and Li et al. [9,10] had applied PSO and Kaveh and colleagues [11-13], and Camp et al. [14] had applied ACO in structural design optimization. The PSO is simple and effective where ACO appears a robust approach.

It is known that the PSO may perform better than the EAs in the early iterations, but it does not appear competitive when the number of iterations increases [15]. To improve this character of PSO, one of the methods is hybridizing PSO with other approaches such as ACO. The resulted method, called Particle Swarm Ant Colony Optimization (PSACO), was initially introduced by Shelokar et al. [16] for solving the continuous unconstrained problems and by Mozafari et al. [17] for reactive power market simulation. PSACO utilized PSO as a global search and the idea of ant colony approach worked as a local search and updated the positions of the particles by applied pheromone-guided mechanism. The proposed method in this paper is basically similar to that algorithm but with some differences. We have applied PSOPC (a hybrid PSO with passive congregation [7]) instead of PSO to improve the performance of the new method. The relation of standard deviation in

ACO stage is different with Ref. [16] and the inertia weight is changed in PSOPC stage. New terminating criterion is employed to increase the probability of obtaining an optimum solution in minimum number of iterations.

There are some constraints in truss optimization problems that should be carefully handled. So far, a number of approaches have been proposed by incorporating constraint-handling techniques into EAs to solve constrained optimization problems [18]. To our knowledge, the penalty function method has been the most popular constraint-handling technique due to its simple principle and ease of implementation. The main difficulty of the penalty function method lies in that the appropriate values of penalty factors are problem-dependent and a considerable effort is needed for fine-tuning of penalty factors. Therefore several novel techniques have been incorporated into EAs to handle constraints. Koziel and Michalewicz [19] proposed a homomorphous mapping (HM); Runarsson and Yao [20] proposed Stochastic Ranking (SR); Coello and Montes [21] presented a dominance-based selection scheme to handle constraints in a GA; Coello and Becerra [22] incorporated a cultural algorithm that used domain knowledge to improve the performance of an evolutionary programming technique. Fly-back mechanism, a new technique handling the constraints, has been introduced by He et al. [8]. Compared with other constraint-handling techniques, this method is relatively simple and easy to implement [10]. Therefore, in this paper the constraints are handled by using fly-back mechanism.

## 2. Introduction to PSO and ACO

### *2.1 Particle swarm optimization*

The application of swarm intelligence in optimization was first developed by Eberhart and Kennedy under the name of Particle Swarm Optimization (PSO) [23]. The strength of PSO is underpinned by the fact that decentralized (without central supervision) biological creatures can often accomplish complex goals by cooperation. A standard PSO algorithm is initialized with a population (swarm) of random potential solutions (particles). Each particle iteratively moves across the search space and is attracted to the position of the best fitness (evaluation of the objective function) historically achieved by the particle itself (local best) and by the best among the neighbors of the particle (global best). In essence, each particle continuously focuses and refocuses the effort of its search according to both local and global best. This behavior mimics the cultural adaptation of a biological agent in a swarm: it evaluates its own position based on certain fitness criteria, compares with others, and imitates the best in the entire swarm [24].

The update moves a particle by adding a change velocity $V_i^{k+1}$ to the current position $X_i^k$ as follows:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{2}$$

The velocity is a combination of three contributing factors: (1) previous velocity $V_i^k$, (2) movement in the direction of the local best $P_i^k$, and (3) movement in the direction of the global best $P_g^k$. The mathematical formulation is expressed as

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \tag{3}$$

where $\omega$ is an inertia weight to control the influence of the previous velocity; $r_1$ and $r_2$ are two random numbers uniformly distributed in the range of (0, 1); $c_1$ and $c_2$ are two acceleration constants [25]; $P_i^k$ is the best position of the $i$ th particle up to iteration $k$ and $P_g^k$ is the best position among all particles in the swarm up to iteration $k$.

Adding the passive congregation model to the PSO may increase its performance. He et al. proposed a hybrid PSO with passive congregation (PSOPC) [7]. In this method the velocity is defined as

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (R_i^k - X_i^k) \tag{4}$$

where $R_i$ is a particle selected randomly from the swarm, $c_3$ is the passive congregation coefficient, and $r_3$ is a uniform random sequence in the range (0, 1).

Several benchmark functions have been tested in Ref. [7]. The results show that the PSOPC has a better convergence rate and a higher accuracy than the PSO.

### *2.2 Ant colony optimization*

Ant Colony Optimization (ACO) was first proposed by Dorigo [26] as a multi-agent approach to solve difficult combinatorial optimization problems. ACO was inspired by the observation of real ant colonies. Ants are social insects whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. An important behavior of ant colonies is their foraging behavior and, in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest-mates. When more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back [27]. In fact, ACO simulates the optimization of ant foraging behavior.

## 3. Fly-Back Mechanism

Fly-back mechanism has been introduced by He et al. [8]. For most of the structural optimization problems, the global minimum locates on or close to the boundary of a feasible design space. The particles are initialized in the feasible region. When the particles fly in the feasible space to search the solution, if any one of them flies into the infeasible region, it will be forced to fly back to the previous position to guarantee a feasible solution. The particle which flies back to the previous position may be closer to the boundary at the next iteration. This makes the particles to fly to the global minimum in a great probability. Some experimental results have shown that it can find a better solution with fewer iterations than other techniques [8].

## 4. Particle Swarm Ant Colony Optimization (PSACO) for Truss Design

The implementation of PSACO algorithm consists of two stages [18]. In the first stage, it applies PSOPC, while ACO is implemented in the second stage. ACO works as a local search, wherein, ants apply pheromone-guided mechanism to refine the positions found by particles in the PSOPC stage. In PSACO, a simple pheromone-guided mechanism of ACO is proposed to be applied for the local search. The proposed ACO algorithm handles *P* ants equal to the number of particles in PSOPC.

In ACO stage, each ant generates a solution around $P_g^k$ which can be written as

$$Z_i^k = N(P_g^k, \sigma) \tag{5}$$

In Eq. (5), $Z_i^k$ is the solution constructed by ant *i* in the stage *k*; $N(P_g^k, \sigma)$ denotes a random number normally distributed with mean value $P_g^k$ and variance $\sigma$, where $\eta$ is used to control the step size.

$$\sigma = (A_{\max} - A_{\min}) \times \eta \tag{6}$$

In the proposed method, objective function value, $f(Z_i^k)$, is computed and the current position of ant *i*, $Z_i^k$, is replaced with the position $X_i^k$, the current position of particle *i* in the swarm, if $f(X_i^k) > f(Z_i^k)$ and current ant is in the feasible space. This simple pheromone-guided mechanism considers, there is highest density of trails (single pheromone spot) at the global best solution $P_g^k$ of the swarm at any iteration *k+1* in each stage of ACO implementation and all ants search for better solutions in the neighborhood of the global best solution [16]. The pseudo-code for the PSACO algorithm is listed in Table 1.

Table 1. The pseudo-code for the PSACO

```
    Set k=0
    Randomly initialize positions and velocities of all particles
FOR(each particle i in the initial population)
    WHILE(the constraints are violated)
        Randomly re-generate the current particle X_i
    END WHILE
    Generate local best: Set P_i^k = X_i^k
    Generate global best: Find min f(X_i^k), P_g^k is set to the position of X_min^k
END FOR
    WHILE(the terminating criterion is not met)
        FOR(each particle (ant) i in the swarm(colony))
            Generate the velocity and update the position of the current particle (vector) X_i^k
            Constraint-handling: Check whether the current particle violates the problem
                    constraints or not. If it does, reset it to the previous position X_i^{k-1}
            Calculate the fitness value f(X_i^k) of the current particle
            Generate the position of the current ant Z_i^k = N(P_g^k, σ)
            Constraint-handling: Check whether the current ant violates the problem
                    constraints or not. If it does, reset it to the current particle X_i^k
            Calculate the fitness value f(Z_i^k) of the current ant
            Update current particle position: Compare the fitness value of current ant with
                    current particle. If the f(Z_i^k) is better than the fitness value of f(X_i^k),
                    set f(X_i^k) = f(Z_i^k) and X_i^k = Z_i^k
            Update local best: Compare the fitness value of f(P_i^k) with f(X_i^k).
                    If the f(X_i^k) is better than the fitness value of f(P_i^k),
                    set P_i^k to the current position X_i^k
        END FOR
        Update global best: Find the global best position in the swarm. If the f(X_i^k) is
                better than the fitness value of f(P_g^k), P_g^k is set to the position
                of the current particle X_i^k
        Set k =k+1
    END WHILE
```

## 5. Terminating Criterion

The maximum number of the iterations is the most usual terminating criterion in PSO literature. If it is selected great, the number of analyses and as a result, the time of optimization will increase; vice versa, if it is selected less, the probability of finding a

desirable solution will decrease. Thus, the necessity for an exact definition of the terminating criterion is really felt. This paper defines a new terminating criterion to fulfill this goal.

In truss optimization, a discrete solution is better than continuous one. If $A_{min}$, $A_{max}$, $A^*$ are the minimum cross-sectional area, the maximum cross-sectional area, and the amount of increase in cross-sectional areas for a given truss respectively, discrete allowable series of allowable cross-sectional areas will be [13]:

$$A_{\min}\ ,\ A_{\min}+A^*, A_{\min}+2A^*, \ldots, A_{\max} \tag{7}$$

$A^*$ controls the exactitude of the solutions with a reverse relation; as $A^*$ gets more, exactitude of the solutions decreases and searching process must be stopped earlier and if the amount of $A^*$ gets less, searching process must be continued until reaching an exact result.

In PSACO algorithm, the current position of each particle equals the previous position of that particle added to the velocity vector. Components of the velocity vector and the largeness of the search space decrease as optimization continues.

Considering these facts, the terminating criterion is redefined:

Searching continues until the absolute value of every component of the velocity vector is greater than $A^*$, and as soon as the maximum absolute value of Component of the velocity vector gets less than $A^*$, searching stops. This can be summarized as

$$\text{Terminating criterion: } \max(|V_{ij}^k|) < A^* \begin{cases} \forall i = 1,2,\ldots, P \\ \forall j = 1,2,\ldots, ng \end{cases} \tag{8}$$

With this criterion, the extra iterations are eliminated and optimum solution is reached earlier.

The value of the inertia weight ($\omega(k)$) is related with the number of the iterations. For example, in truss design, $\omega(k)$ decreases linearly from 0.9 in first iteration to 0.4 in 3000$^{th}$ iteration [10]. Since the convergence rate of PSACO is higher than PSOPC, the solution is reached in less iterations. Therefore, $\omega(k)$ is redefined as

$$\omega(k) = 0.9 - 0.001 \times k \geq 0.4 \tag{9}$$

where $k$ =the iteration number. With this new equation, the inertia weight decreases from 0.9 to 0.4 in 500 iterations then the amount of it (0.4) remains fixed. In this way, the balance between $\omega(k)$ and the fast rate of convergence is saved; consequently, the performance improves.

## 6. Numerical Examples

In this section, common truss optimization examples as benchmark problems are optimized with this method; then, the final results are compared with solutions of other methods to demonstrate the effectiveness of this work.

For the proposed algorithm, a population of 50 individuals is used for both particles and ants; the value of constants $c_1$ and $c_2$ are set 0.8 and the passive congregation coefficient $c_3$ is given 0.6. The algorithms are coded in Matlab and structures are analyzed using the direct stiffness method.

### *6.1 The ten-bar planar truss*

The 10-bar truss problem shown in Figure 1, has become a common problem in the field of structural design to test and verify the efficiency of many different optimization methods. The material density is 0.1 lb/in$^3$ (2767.990 kg/m$^3$) and the modulus of elasticity is 10,000 ksi (68,950 MPa). The members are subjected to the stress limits of ±25 ksi (172.375 MPa) and all nodes in both vertical and horizontal directions are subjected to the displacement limits of ±2.0 in (5.08 cm). There are 10 design variables in this example and a set of pseudo variables ranging from 0.1 to 35.0 in$^2$ (from 0.6452 cm$^2$ to 225.806 cm$^2$). Two cases are considered: Case 1, $P_1$=100 kips (444.8 kN) and $P_2$=0; and Case 2, $P_1$= 150 kips (667.2 kN) and $P_2$=50 kips (222.4 kN).

For both load cases, the PSO and PSOPC algorithms achieve the best solutions after 3000 iterations [10]. However, the PSACO algorithm finds the best solution after about 619 and 650 iterations respectively for Case 1 and Case 2. In first iteration, the PSACO algorithm achieves 7426 lb while the PSO and PSOPC algorithms do not reach it until nearly 1780 and 900 iterations respectively. Figure 2 provides a comparison of the convergence rates of the three algorithms. The best weights of PSACO are 5057.36 lb for Case 1 and 4676.05 lb for Case 2 while the best results of PSO and PSOPC are 5061.00 lb, 5529.50 lb for Case 1 and 4679.47 lb, 4677.70 lb for Case 2 respectively. Table 2 and Table 3 compare the obtained results in this research with the existing results.

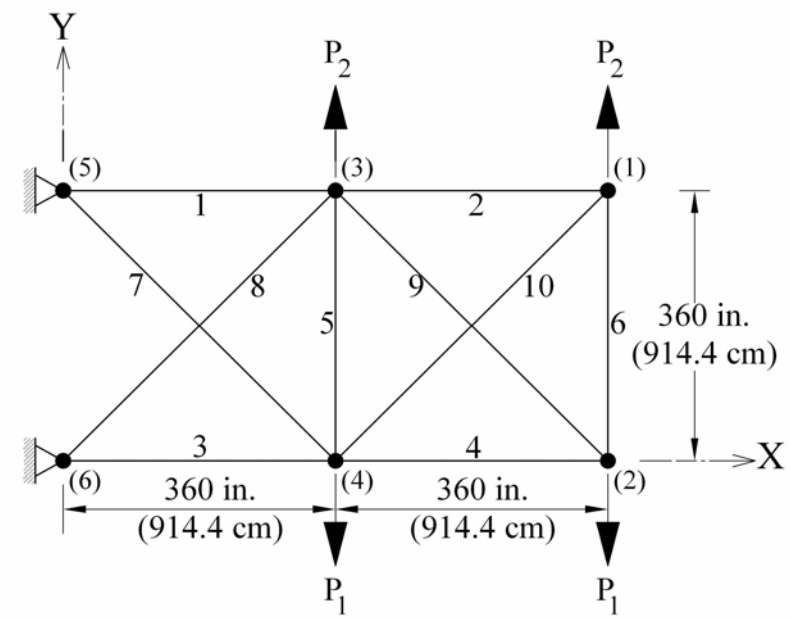
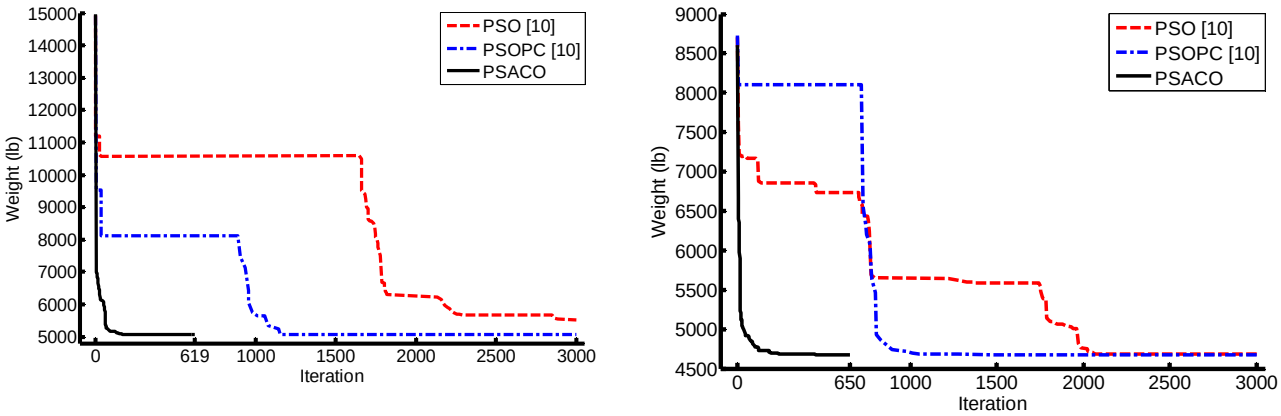



Figure 1. Ten-bar planar truss




a) Case 1 b) Case 2

Figure 2. Comparison of the convergence rates between the three algorithms for the 10-bar planar truss structure

Table 2. Optimal design comparison for the 10-bar planner truss (Case 1)

| Optimal cross-sectional areas ($in^2$) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Element group** | | **Schmit & Miura [28]** | | **Lee & Geem [2]** | **Li et al. [10]** | | **PSACO** | |
| | | **NEWSUMT** | **CONMIN** | | **PSO** | **PSOPC** | **$in^2$** | **$cm^2$** |
| 1 | $A_1$ | 30.67 | 30.57 | 30.15 | 33.469 | 30.569 | 30.068 | (193.99) |
| 2 | $A_2$ | 0.100 | 0.369 | 0.102 | 0.110 | 0.100 | 0.100 | (0.6452) |
| 3 | $A_3$ | 23.76 | 23.97 | 22.71 | 23.177 | 22.974 | 23.207 | (149.72) |
| 4 | $A_4$ | 14.59 | 14.73 | 15.27 | 15.475 | 15.148 | 15.168 | (97.858) |
| 5 | $A_5$ | 0.100 | 0.100 | 0.102 | 3.649 | 0.100 | 0.100 | (0.6452) |
| 6 | $A_6$ | 0.100 | 0.364 | 0.544 | 0.116 | 0.547 | 0.536 | (3.4581) |
| 7 | $A_7$ | 8.578 | 8.547 | 7.541 | 8.328 | 7.493 | 7.462 | (48.142) |
| 8 | $A_8$ | 21.07 | 21.11 | 21.56 | 23.340 | 21.159 | 21.228 | (136.95) |
| 9 | $A_9$ | 20.96 | 20.77 | 21.45 | 23.014 | 21.556 | 21.630 | (139.55) |
| 10 | $A_{10}$ | 0.100 | 0.320 | 0.100 | 0.190 | 0.100 | 0.100 | (0.6452) |
| *Weight(lb)* | | 5076.9 | 5107.3 | 5057.88 | 5529.50 | 5061.00 | 5057.36 | 22505N |

Table 3. Optimal design comparison for the 10-bar planner truss (Case 2)

| Optimal cross-sectional areas ($in^2$) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Element group | | Schmit & Miura [28] | | Lee & Geem [2] | Li et al. [10] | | PSACO | |
| | | NEWSUMT | CONMIN | | PSO | PSOPC | $in^2$ | $cm^2$ |
| 1 | $A_1$ | 23.55 | 23.55 | 23.25 | 22.935 | 23.473 | 23.052 | (148.72) |
| 2 | $A_2$ | 0.100 | 0.176 | 0.102 | 0.113 | 0.101 | 0.100 | (0.6452) |
| 3 | $A_3$ | 25.29 | 25.20 | 25.73 | 25.355 | 25.287 | 25.601 | (165.17) |
| 4 | $A_4$ | 14.36 | 14.39 | 14.51 | 14.373 | 14.413 | 15.139 | (97.671) |
| 5 | $A_5$ | 0.100 | 0.100 | 0.100 | 0.100 | 0.100 | 0.100 | (0.6452) |
| 6 | $A_6$ | 1.97 | 1.967 | 1.977 | 1.990 | 1.969 | 1.969 | (12.703) |
| 7 | $A_7$ | 12.39 | 12.40 | 12.21 | 12.346 | 12.362 | 12.206 | (78.748) |
| 8 | $A_8$ | 12.81 | 12.86 | 12.61 | 12.923 | 12.694 | 12.568 | (81.084) |
| 9 | $A_9$ | 20.34 | 20.41 | 20.36 | 20.678 | 20.323 | 20.330 | (131.16) |
| 10 | $A_{10}$ | 0.100 | 0.100 | 0.100 | 0.100 | 0.103 | 0.100 | (0.6452) |
| Weight (lb) | | 4677 | 4684.1 | 4668.81 | 4679.47 | 4677.70 | 4676.05 | 20808N |

### *6.2 Twenty five-bar spatial truss*

Figure 3 shows the topology of a 25-bar spatial truss. The material density is 0.1 lb/in$^3$ (2767.990 kg/m$^3$) and the modulus of elasticity is 10,000 ksi (68,950 MPa).

25 members are categorized into eight groups, as follows: (1) $A_1$, (2) $A_2$–$A_5$, (3) $A_6$–$A_9$, (4) $A_{10}$–$A_{11}$, (5) $A_{12}$–$A_{13}$, (6) $A_{14}$–$A_{17}$, (7) $A_{18}$–$A_{21}$, and (8) $A_{22}$–$A_{25}$. This spatial truss was subjected to the two loading conditions shown in Table 4. Maximum displacement limitations of ±0.35 in (8.89 mm) were imposed on every node in every direction and the axial stress constraints vary for each group shown in Table 5. The range of cross-sectional areas varies from 0.01 to 3.4 in$^2$ (from 0.6452 cm$^2$ to 21.94 cm$^2$).

For this spatial truss structure, it takes about 1000 and 3000 iterations for the PSOPC and the PSO algorithms to converge, respectively. However the PSACO algorithm takes 577 iterations to converge. Indeed, in this example, the PSO algorithm did not fully converge when the maximum number of iterations is reached [10]. In first iteration, the PSACO algorithm achieves 652.38 lb while the PSO and PSOPC algorithms do not reach it until

nearly 2600 and 550 iterations respectively. Figure 4 compares the convergence rate of the three algorithms. Table 6 lists the optimal values of the eight size variables obtained by this research, and compares them with other results.

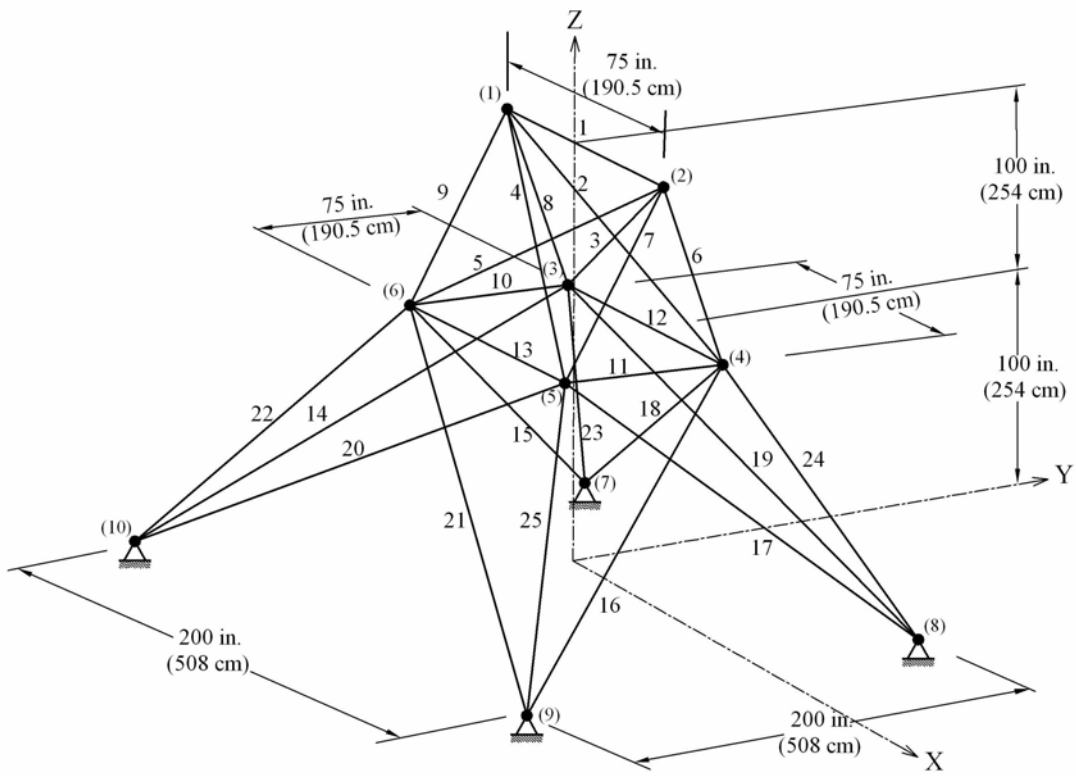



Figure 3. Twenty five-bar spatial truss

Table 4. Loading conditions for the 25-bar spatial truss

| Node | Case 1 | | | Case 2 | | |
|---|---|---|---|---|---|---|
| | $P_X$ kips(kN) | $P_Y$ kips(kN) | $P_Z$ kips(kN) | $P_X$ kips(kN) | $P_Y$ kips(kN) | $P_Z$ kips(kN) |
| 1 | 0.0 | 20.0 (89) | –5.0 (22.25) | 1.0 (4.45) | 10.0 (44.5) | –5.0 (22.25) |
| 2 | 0.0 | –20.0 (89) | –5.0 (22.25) | 0.0 | 10.0 (44.5) | –5.0 (22.25) |
| 3 | 0.0 | 0.0 | 0.0 | 0.5 (2.22) | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.5 (2.22) | 0.0 | 0.0 |

Table 5. Member stress limitation for the 25-bar spatial truss

| Element group | | Compressive stress limitations ksi (MPa) | Tensile stress limitations ksi (MPa) |
|---|---|---|---|
| 1 | $A_1$ | 35.092 (241.96) | 40.0 (275.80) |
| 2 | $A_2 \sim A_5$ | 11.590 (79.913) | 40.0 (275.80) |
| 3 | $A_6 \sim A_9$ | 17.305 (119.31) | 40.0 (275.80) |
| 4 | $A_{10} \sim A_{11}$ | 35.092 (241.96) | 40.0 (275.80) |
| 5 | $A_{12} \sim A_{13}$ | 35.092 (241.96) | 40.0 (275.80) |
| 6 | $A_{14} \sim A_{17}$ | 6.759 (46.603) | 40.0 (275.80) |
| 7 | $A_{18} \sim A_{21}$ | 6.959 (47.982) | 40.0 (275.80) |
| 8 | $A_{22} \sim A_{25}$ | 11.082 (76.410) | 40.0 (275.80) |

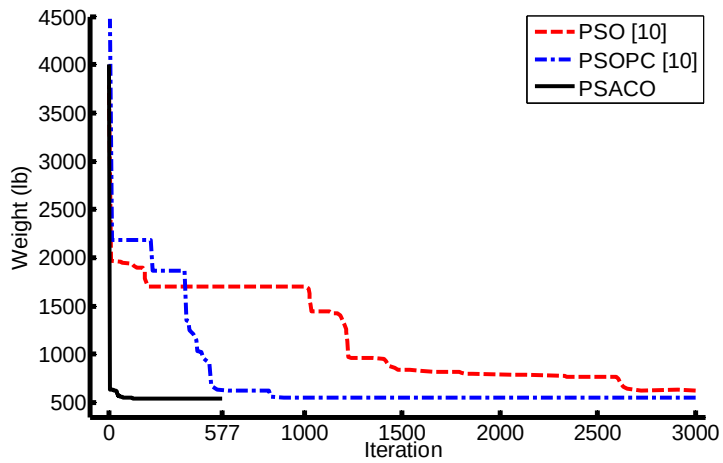



Figure 4. Convergence rate comparison for the three algorithms for the 25-bar spatial truss

### *6.3 One hundred twenty-bar dome truss*

120-bar dome truss, shown in Figure 5, was first analyzed by Soh and Yang [30] to obtain the optimal sizing and configuration variables. In the example considered in this study similar to Lee and Geem [2] and Keleşoğlu and Ülker [31], only sizing variables to minimize the structural weight are considered. In addition, the allowable tensile and compressive stresses are used according to the AISC ASD (1989) [32] code, as follows

$$\begin{cases} \sigma_i^+ = 0.6F_y & for\ \sigma_i \geq 0 \\ \sigma_i^- & for\ \sigma_i < 0 \end{cases} \tag{10}$$

where $\sigma_i^-$ is calculated according to the slenderness ratio

Table 6. Optimal design comparison for the 25-bar spatial truss

| Optimal cross-sectional areas (in$^2$) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Element group | | Saka [29] | Camp et al. [14] | Lee & Geem [2] | Li et al. [10] | | PSACO | |
| | | | | | PSO | PSOPC | in$^2$ | cm$^2$ |
| 1 | $A_1$ | 0.010 | 0.010 | 0.047 | 9.863 | 0.010 | 0.010 | 0.065 |
| 2 | $A_2 \sim A_5$ | 2.085 | 2.000 | 2.022 | 1.798 | 1.979 | 2.052 | 13.24 |
| 3 | $A_6 \sim A_9$ | 2.988 | 2.966 | 2.950 | 3.654 | 3.011 | 3.001 | 19.36 |
| 4 | $A_{10} \sim A_{11}$ | 0.010 | 0.010 | 0.010 | 0.100 | 0.100 | 0.010 | 0.065 |
| 5 | $A_{12} \sim A_{13}$ | 0.010 | 0.012 | 0.014 | 0.100 | 0.100 | 0.010 | 0.065 |
| 6 | $A_{14} \sim A_{17}$ | 0.696 | 0.689 | 0.688 | 0.596 | 0.657 | 0.684 | 4.413 |
| 7 | $A_{18} \sim A_{21}$ | 1.67 | 1.679 | 1.657 | 1.659 | 1.678 | 1.616 | 10.42 |
| 8 | $A_{22} \sim A_{25}$ | 2.592 | 2.668 | 2.663 | 2.612 | 2.693 | 2.673 | 17.24 |
| Weight (lb) | | 545.23 | 545.53 | 544.38 | 627.08 | 545.27 | 545.04 | 2425.4N |

$$\sigma_i^- = \begin{cases} \left[\left(1-\dfrac{\lambda_i^2}{2C_C^2}\right)Fy\right] \Big/ \left(\dfrac{5}{3}+\dfrac{3\lambda_i}{8C_C}-\dfrac{\lambda_i^3}{8C_C^3}\right) & for\ \lambda_i < C_C \\ \dfrac{12\pi^2 E}{23\lambda_i^2} & for\ \lambda_i \geq C_C \end{cases} \tag{11}$$

where $E$=the modulus of elasticity; $F_y$ =the yield stress of steel; $C_c$ =the slenderness ratio ($\lambda_i$) dividing the elastic and inelastic buckling regions ($C_C = \sqrt{2\pi^2 E/F_y}$); $\lambda_i$ = the slenderness ratio ($\lambda_i = kL_i/r_i$); $k$ =the effective length factor; $L_i$ =the member length; and $r_i$ =the radius of gyration.

The modulus of elasticity is 30,450 ksi (209,952 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 (405 MPa). On the other hand, the radius of gyration ($r_i$) can be expressed in terms of cross-sectional areas, i.e., $r_i = aA_i^b$ [29]. Here, $a$ and $b$ are the constants depending on the types of sections adopted for the members such as pipes, angles, and tees. In this example, pipe sections ($a$ = 0.4993 and $b$ = 0.6777) were adopted for bars. All members of the dome are linked into seven groups, as shown in Figure 5. The dome is considered to be subjected to vertical loading at all the unsupported joints. These were 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 rest of the nodes. The minimum cross-

sectional area of all members is 0.775 in$^2$ (2 cm$^2$). In this example, four cases of constraints are considered: with stress constraints and no displacement constraints (Case 1), with stress constraints and displacement limitations of $\pm 0.1969$ in (5 mm) imposed on all nodes in x- and y-directions (Case 2), no stress constraints but displacement limitations of $\pm 0.1969$ in (5 mm) imposed on all nodes in z-directions (Case 3), and all constraints explained above (Case 4). For Case 1 and Case 2, the maximum cross-sectional area is 5.0 in$^2$ (32.26 cm$^2$) and for Case 3 and Case 4 is 20.0 in$^2$ (129.03 cm$^2$).

Table 7 gives the best solution vectors and the corresponding weights for all cases. Figure 6 shows the convergence for all cases.

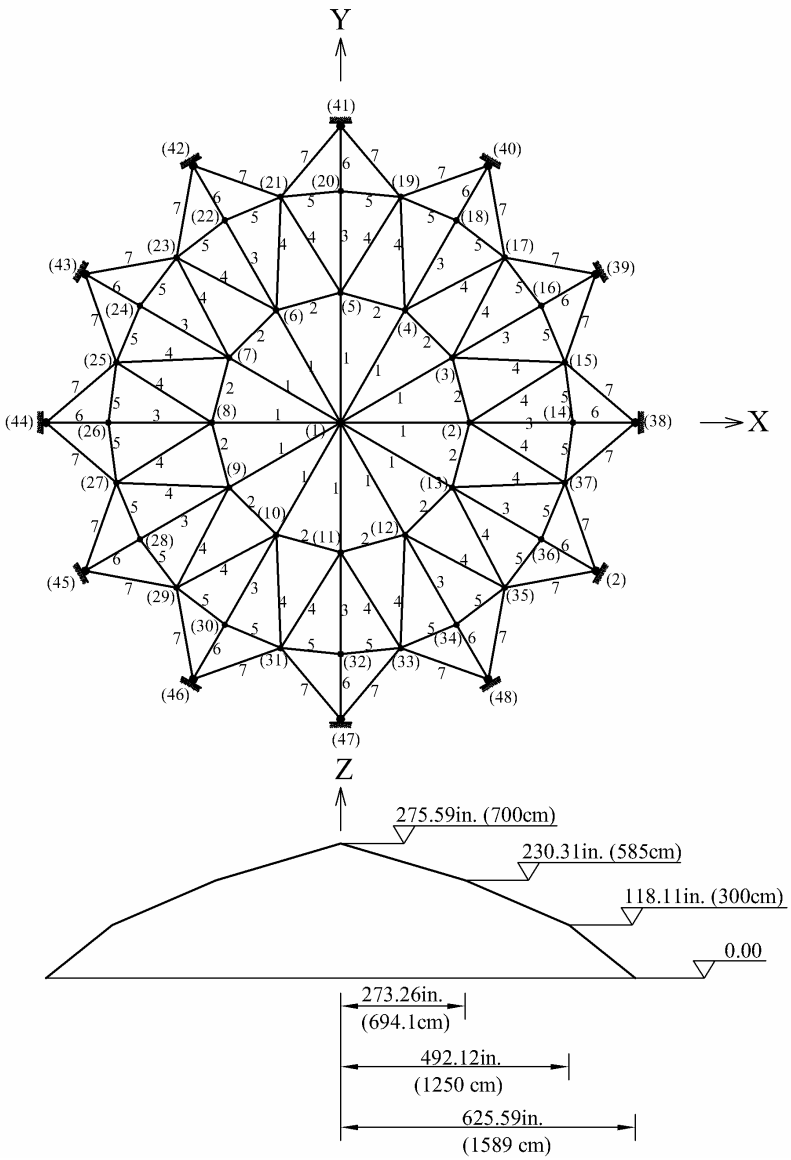



Figure 5. One hundred twenty-bar dome truss

Table 7. Optimal design comparison for the 120-bar dome truss (four Cases)

| Optimal cross-sectional areas ($in^2$) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Element group | Case 1 | | | | | Case 2 | | | | |
| | Lee & Geem [2] | PSO | PSOPC | PSACO | | Lee & Geem [2] | PSO | PSOPC | PSACO | |
| | | | | $in^2$ | $cm^2$ | | | | $in^2$ | $cm^2$ |
| 1 | 3.295 | 3.147 | 3.235 | 3.125 | (20.16) | 3.296 | 15.978 | 3.083 | 3.091 | (19.94) |
| 2 | 3.396 | 6.376 | 3.370 | 3.355 | (21.65) | 2.789 | 9.599 | 3.639 | 3.354 | (21.64) |
| 3 | 3.874 | 5.957 | 4.116 | 4.125 | (26.62) | 3.872 | 7.467 | 4.095 | 4.095 | (26.42) |
| 4 | 2.571 | 4.806 | 2.784 | 2.783 | (17.96) | 2.570 | 2.790 | 2.765 | 2.764 | (17.83) |
| 5 | 1.150 | 0.775 | 0.777 | 0.775 | (5.00) | 1.149 | 4.324 | 1.776 | 1.592 | (10.27) |
| 6 | 3.331 | 13.798 | 3.343 | 3.354 | (21.64) | 3.331 | 3.294 | 3.779 | 3.317 | (21.40) |
| 7 | 2.784 | 2.452 | 2.454 | 2.447 | (15.79) | 2.781 | 2.479 | 2.438 | 2.435 | (15.70) |
| Weight (lb) | 19707.77 | 32432.9 | 19618.7 | 19504.6 | 86795N | 19893.34 | 41052.7 | 20681.7 | 20095.0 | 89423N |
| Element group | Case 3 | | | | | Case 4 | | | | |
| | Keleşoğlu & Ülker [31] | PSO | PSOPC | PSACO | | Lee & Geem [2] | PSO | PSOPC | PSACO | |
| | | | | $in^2$ | $cm^2$ | | | | $in^2$ | $cm^2$ |
| 1 | (19.52) | 5.606 | 1.773 | 2.098 | 1.970 | (12.71) | 12.802 | 3.040 | 3.026 | (19.52) |
| 2 | (98.21) | 7.750 | 17.635 | 16.444 | 15.420 | (99.48) | 11.765 | 13.149 | 15.222 | (98.21) |
| 3 | (31.64) | 4.311 | 7.406 | 5.613 | 5.850 | (37.74) | 5.654 | 5.646 | 4.904 | (31.64) |
| 4 | (20.15) | 5.424 | 2.153 | 2.312 | 2.204 | (14.22) | 6.333 | 3.143 | 3.123 | (20.15) |
| 5 | (53.81) | 4.402 | 15.232 | 8.793 | 9.371 | (60.46) | 6.963 | 8.759 | 8.341 | (53.81) |
| 6 | (22.05) | 6.223 | 19.544 | 3.629 | 3.676 | (23.71) | 6.492 | 3.758 | 3.418 | (22.05) |
| 7 | (16.11) | 5.405 | 0.800 | 1.954 | 1.938 | (12.51) | 4.988 | 2.502 | 2.498 | (16.11) |
| Weight (lb) | 148024N | 38237.83 | 46893.5 | 31776.2 | 31671.5 | 140938N | 51986.2 | 33481.2 | 33263.9 | 148024N |

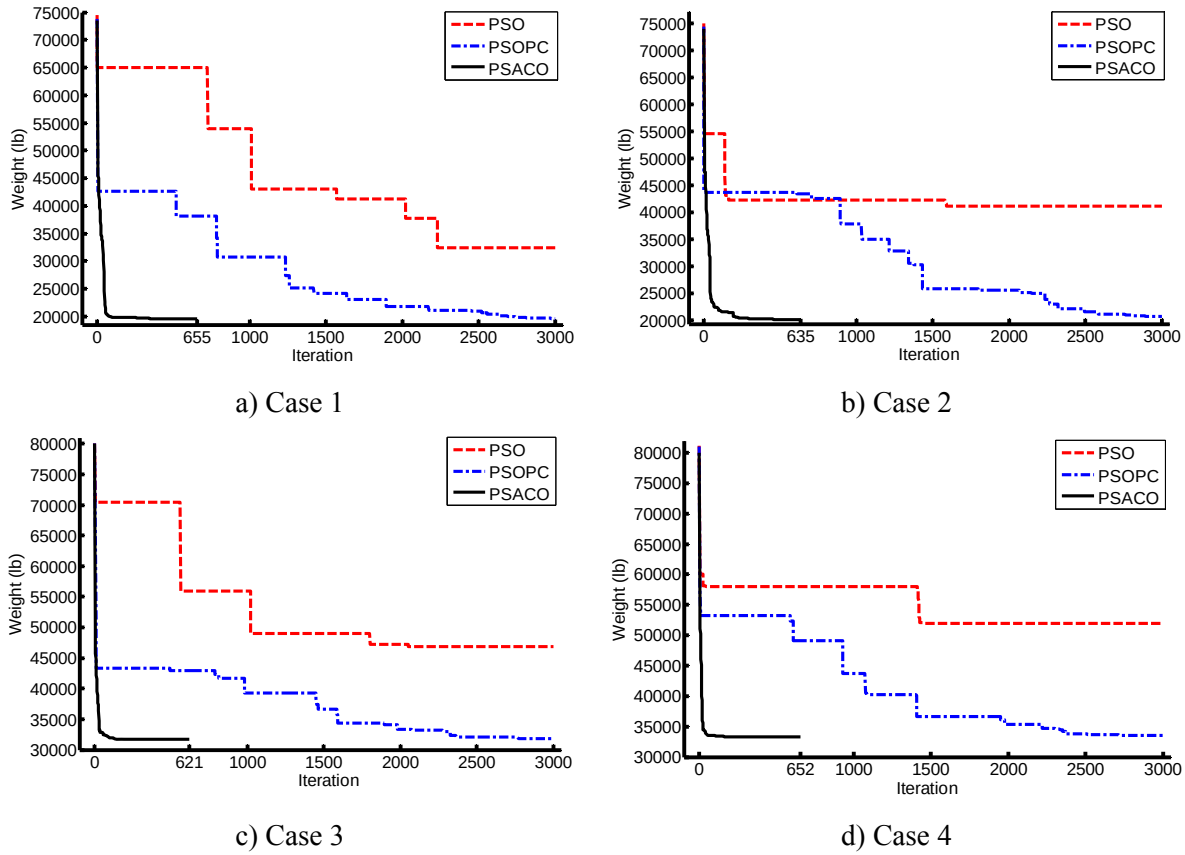



Figure 6. Comparison of the convergence rates between the three algorithms for the 120-bar dome truss structure

## 7. Discussion

### *7.1 Efficiency of PSACO*

Figures of the PSACO convergence in all examples are similar and in almost all trusses, an optimum solution is reached after nearly 600 iterations and global search is completed after 200 iterations. In that condition, the average weight is nearly 0.5 percent more than the last result. The convergence rate in the global search stage is very high, but decreases in the local search stage (after nearly 200 iterations); however, it is more acceptable than the convergence rates in the PSO and the PSOPC.

The difference between the best and the worst results of the 10-bar truss (Case 1) in 50 tests is only 3.2lb (0.06%) and the standard deviation is 1.46lb (see Table 8). This fact shows that the proposed method is able to achieve an optimum solution and is never trapped in local optimums.

Applying the developed method has led to a significant improvement in PSOPC. Perhaps, this can not be formulated as an exact mathematical relation but the major reasons of this

improvement can be summarized as following:

a) Heuristic methods utilize two factors: the random search factor and the information collected from the search space during optimization process. In early iterations, the random search factor has more power than the collective information factor, but the increase in the number of iterations gradually abates the power of the random search factor and increases the power of the collective information factor. In PSACO, ACO stage actually plays an auxiliary role in rapidly increasing the collective information factor; consequently, the convergence rate increases highly.

b) In truss optimization, usually there are some local optimums in the neighborhood of a desirable solution. So, the probability of finding a desirable optimum increases with more search around the local optimums. PSACO does extra search around the local optimums, $P_g^k$, (by using Eq. 5) and obtains the desirable solution with more probability in less iterations.

### *7.2 Efficiency of the new terminating criteria*

Figure 7 shows the average and a typical of $\max(|V_{ij}^k|)$ in 50 tests for the 10-bar truss (Case 1). This figure shows that from beginning of searching to nearly 580$^{th}$ iteration, the velocity of particles decreases quickly; and in result, the search space in the neighborhood of the optimum solution gets less rapidly. After 580$^{th}$ iteration, velocity of particles is very small and after few numbers of iterations, the search process stops.

The movement of particles and the rapid decrease in the search space demonstrate the good performance of the new terminating criterion. In this criterion, when the velocity of particles becomes neglect able and the probability of weight reduction gets less, the search process stops; consequently, the time of optimization decreases.

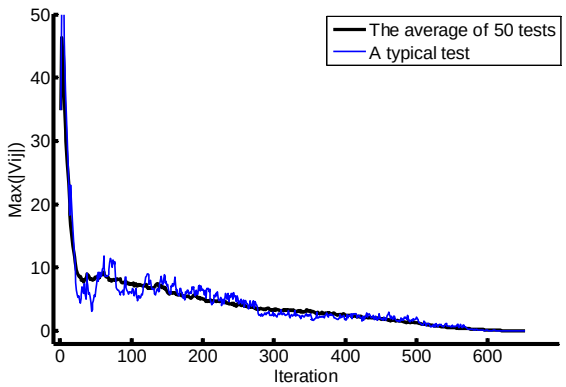



Figure 7. The average and a typical of $\max(|V_{ij}^k|)$ in 50 tests for 10-bar truss (Case 1)

### *7.3 Effect of step size in PSACO*

In ACO stage, the amount of step size, $\eta$, highly influences the results. Table 8 compares the minimum, maximum and average of the optimum weight and the required iterations for the 10-bar truss (Case 1) in 50 runs. If $\eta$ is too small, the velocity of particles will decrease very

fast and the search process will stop in early iterations; thus the obtained results stay far away from an optimum solution and amount of the standard deviation is too great; on the contrary, if $\eta$ is selected too great, the PSACO algorithm will perform similar to the PSOPC algorithm and the effect of ACO stage will be eliminated, and a desirable solution can not be obtained in less iterations. Since the standard deviation becomes minimum and a best solution is achieved in the acceptable number of iterations, a step size of 0.01 is recommended.

Table 8. Effect of step size in PSACO algorithm for 10-bar truss (Case 1) in 50 runs

| $\eta$ | Minimum iterations | Maximum iterations | Average iterations | Best weight (lb) | Worst weight (lb) | Average weight (lb) | Standard deviation (lb) |
|---|---|---|---|---|---|---|---|
| $10^{-6}$ | 65 | 90 | 72.6 | 5241.2 | 7260 | 5879.3 | 557.2 |
| $10^{-5}$ | 95 | 456 | 217.3 | 5175.7 | 7149.6 | 5807.6 | 538.2 |
| $10^{-4}$ | 404 | 628 | 462.6 | 5065.9 | 6498.2 | 5324.8 | 398.3 |
| $10^{-3}$ | 598 | 638 | 605.4 | 5060.5 | 5076.6 | 5064.3 | 6.35 |
| $10^{-2}$ | 619 | 655 | 635.2 | 5057.4 | 5060.6 | 5058.2 | 1.46 |
| $10^{-1}$ | 654 | 702 | 667.4 | 5060.9 | 5057.4 | 5063.1 | 3.31 |
| $10^{0}$ | 718 | 1000 | 810.5 | 5069.6 | 5267.8 | 5110.0 | 58.86 |

## 8. Conclusions

In this paper PSACO, based on PSOPC and ACO, is employed for optimizing truss structures. ACO plays a helping role for PSOPC process in creating an optimum solution and rapidly attaining the feasible solution space.

In order to make the particles remain in the feasible space, fly-back mechanism is utilized which shows better performance than other constraint-handling methods and does not posses the disadvantages of the penalty function approach.

Since the convergence rate of the proposed method is greater than PSO and PSOPC in comparison, a new relation is defined for the inertia weight. The terminating criterion is changed in the way that after decreasing the movements of particles, the search process stops. With these changes, the number of iterations and time of optimization do not increase. The comparisons based on several well-studied benchmark trusses demonstrate the robustness of the proposed method.

**Acknowledgement:** The first author is grateful to the Iran National Science Foundation for the support.

## References

1. Toğan V, Daloğlu AT. Optimization of 3d trusses with adaptive approach in genetic algorithms, *Engineering Structures*, **28**(2006)1019-27.
2. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, **82**(2004)781-98.
3. Bertsekas DP. *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
4. Goldberg DE, Samtani MP. Engineering optimization via genetic algorithm, in Will KM. (Eds), *Proceeding of the 9th Conference on Electronic Computation*, ASCE, Feb 1986, 471-82.
5. Hajela P. Genetic search - an approach to the non-convex optimization problem, *AIAA Journal*, No. 7, **28**(1990)1205-10.
6. Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm force method and graph theory, *International Journal for Numerical Methods in Engineering*, **58**(2003)771-91.
7. He S, Wu QH, Wen JY, Saunders JR, Paton RC. A particle swarm optimizer with passive congregation, *Biosystem*, **78**(2004)135-47.
8. He S, Prempain E, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, No. 5, **36**(2004)585-605.
9. Li LI, Ren FM, Liu F, Wu QH. An improved particle swarm optimization method and its application in civil engineering. *Proceedings of the Fifth International Conference on Engineering Computational Technology*, United Kingdom: Civil-Comp Press, Stirlingshire, 2006.
10. Li LI, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers and Structures*, **85**(2007)340-9.
11. Kaveh A, Shojaee S. Optimal design of skeletal structures using ant colony optimization, *International Journal of Numerical Methods for Engineering*, No. 70, **5**(2007)563-81.
12. Kaveh A, Shahrouzi M. A hybrid ant strategy and genetic algorithm to tune the population size for efficient structural optimization, *Engineering Computations*, No. 24, **3**(2007) 237-54.
13. Kaveh A. Farhmand Azar B, Talatahari S. *Ant colony optimization for design of space trusses, submitted for publication*, 2008.
14. Camp C, Bichon J. Design of space trusses using ant colony optimization, *Journal of Structural Engineering*, *ASCE*, No. 5, **130**(2004)741-51.
15. Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference, *Proceeding of the Evolutionary Programming Conference*, San Diego, USA, 1998.
16. Shelokar PS, Siarry P, Jayaraman VK, Kulkarni BD. Particle swarm and ant colony

algorithms hybridized for improved continuous optimization, *Applied Mathematic Computation*, **188**(2007)129-42.
17. Mozafari B, Ranjbar T AM , Amraee M, Mirjafari AR, Shiranib. A hybrid of particle swarm and ant colony optimization algorithms for reactive power market simulation. *Journal of Intelligent and Fuzzy Systems*, **17**(2006)557-74.
18. Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Journal of Computer Methods in Applied Mechanics and Engineering*, **191**(2002)1245-87.
19. Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, *Evolutionary Computation*, No. 1, **7**(1999)19-44.
20. Runarsson TP, Yao X, Stochastic ranking for constrained evolutionary optimization, IEEE Trans, *Evolutionary Computation*, No. 3, **4**(2000)284-94.
21. Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, *Advances in Engineering.* Information, **16**(2002) 193-203.
22. Coello CAC, Becerra RL. Efficient evolutionary optimization through the use of a cultural algorithm, *Engineering Optimization*, No. 2, **36**(2004)219-36.
23. 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.
24. Kennedy J, Eberhart RC, Shi Y. *Swarm intelligence*, San Francisco (CA), Morgan Kaufman Publishers, 2001.
25. Shi Y, Eberhart RC. A modified particle swarm optimizer, *Proceedings of IEEE international Conference on Evolutionary Computation*, 1997, pp. 303-8.
26. Dorigo M. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, IT, 1992.
27. Dorigo M, Caro DG, Gambardella LM. Ant Algorithms for Discrete Optimization, *Artificial Life*, No. 3, **5**(1999)137-72.
28. Schmit Jr LA, Miura H. Approximation concepts for efficient structural synthesis, NASA CR-2552, Washington, DC, 1976.
29. Saka MP. Optimum design of pin-jointed steel structures with practical applications. *Journal of Structural Engineering, ASCE*, No. 10, **116**(1990)2599-620.
30. Soh CK, Yang J. Fuzzy controlled genetic algorithms search for shape optimization, *Journal of Computing in Civil Engineering, ASCE*, No. 2, **10**(1996)143-50.
31. Keleşoğlu O, Ülker M. Fuzzy optimization geometrical nonlinear space truss design, *Turkish Journal of Engineering and Environmental Science*, **29**(2005)321-9.
32. American Institute of Steel Construction (AISC), Manual of steel construction–allowable stress design, 9th ed., Chicago, IL, 1989.