Paper 67

©Civil-Comp Press, 2012
Proceedings of the Eleventh International Conference
on Computational Structures Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Particle Swarm Optimization for Non-Convex Problems of Size and Shape Optimization of Trusses

Y.J. Xu[1], M. Domaszewski[2,3], D. Chamoret[2], W.H. Zhang[1] and J.G. Korvink[3,4]
[1] The Key Laboratory of Contemporary Design and Manufacturing Technology
Northwestern Polytechnical University, Xi'an Shaanxi, China
[2] M3M Laboratory
University of Technology of Belfort-Montbéliard, Belfort, France
[3] Department of Microsystems Engineering (IMTEK)
[4] Freiburg Institute for Advanced Studies (FRIAS)
University of Freiburg, Germany

Abstract

The simultaneous size and shape optimization of truss structures can represent the non-convex problems. In this paper, an improved particle swarm optimization algorithm is developed and applied to find global optimum solutions. The optimisation problem is concerned with minimizing the structural weight subject to the constraints on nodal displacements, stresses and buckling in the bars. The optimisation variables are the nodal coordinates and the cross-sectional areas. Also the lower and upper bounds are imposed on these two types of optimisation variables. The classical PSO algorithm is modified to satisfy that all the particles fly inside the variable boundaries. A method derived from the harmony search algorithm is used to deal with the particles which fly outside the variables boundaries. The multi-stage penalty function method is adopted within PSO to satisfy the constraints of the optimisation problem and obtain the feasible optimal solutions. The classical PSO algorithm has good exploration abilities but weak exploitation of local optima. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity of each particle. Thus this parameter regulates the trade-off between global and local exploration ability of the swarm. A general rule is to set the initial inertia weight to a large value in order to make better global exploration of the search space and then gradually decrease it to get more refined solutions. Thus a dynamic variation of inertia weight is used in the paper by applying a fraction multiplier. Some benchmark problems of size and shape optimisation of truss structures are tested and the obtained results are compared with the known results from the literature.

Keywords: particle swarm optimisation, truss structural optimisation, constraint handling, size and shape optimisation.

1 Introduction

Over the past decade a number of optimisation algorithms have been used extensively in truss structural optimisation, from the traditional mathematical optimisation algorithms to the metaheuristic algorithms. The metaheuristic algorithms do not require conventional mathematical assumptions and possess better global search abilities than the conventional optimisation algorithms, have become more attractive in the structural optimisation applications.
Recently, particle swarm optimisation (PSO) which is also an evolutionary algorithm was developed [1]. In PSO, each solution to the optimisation problem is regarded as a 'particle' in the search space, which adjusts its position according to its own flying experience and the flying experience of other particles. The advantage of PSO over the other evolutionary algorithms such as the genetic algorithms is that PSO does not need complicated encoding and decoding and special genetic operators such as mutation and crossover but can work directly with real numbers. It is therefore quite straightforward to implement a PSO algorithm.
The PSO has increasingly gained popularity in the area of structural optimisation. Applications in the size optimisation and shape optimisation of truss structures had been implemented by some researchers [1-8]. This paper focuses on the implementation and application of PSO for the simultaneous size and shape optimisation of truss structures. The paper is organized as follows. We introduce firstly the basic mathematical formulation of the PSO followed by description of the constraint handling method. Then numerical results are obtained and compared with the reference results for the validation. Finally, concluding remarks are proposed.

2 Presentation of simultaneous size and shape optimisation problem of truss structures

In this paper, simultaneous size and shape optimisation problems for truss structures with continuous design variables are studied. The simultaneous size and shape optimisation is concerned with determining the cross-sectional areas of truss members and the nodal coordinates simultaneously. The objective function is a weight of the truss, which is subjected to the stress and displacements constraints. In general, these truss structural optimisation problems can be formulated as the follows

$$\begin{aligned} &\mathit{Minimize}\ f(X) \\ &\mathit{Subject\ to}: g_i(X) \le 0,\ \ i = 1, 2, ..., m \end{aligned} \tag{1}$$

where $f(X)$ is the weight of the truss, which is a scalar function, and $g_i(X)$ is the inequality constraints. The variables vector X represents a set of the design variables (the areas of cross-sections of the truss members and the coordinates of the nodes). It can be denoted as

$$X = \left(A_1, A_2, ..., A_n, R_1, R_2, ..., R_p \right)$$
$$A_i^l < A_i < A_i^u, \; i = 1, 2, ..., n \tag{2}$$
$$R_i^l < R_i < R_i^u, \; i = 1, 2, ..., p$$

where that A_i^l and A_i^u are the lower and the upper bounds of the ith truss member cross-sectional area, while R_i^l and R_i^u are the lower and the upper bounds of the ith nodal coordinates.

3 Particle Swarm Optimisation

3.1 Basis of Particle Swarm Optimisation algorithm

The PSO algorithm is a global optimisation algorithm and described as sociologically inspired. In PSO, each individual of the swarm is considered as a particle in a multi-dimensional space that has a position and a velocity. These particles fly through hyperspace and remember the best position that they have seen. Members of a swarm remember the location where they had their best success and communicate good positions to each other, then adjust their own position and velocity based on these good positions. Updating the position and velocity is done as follows

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

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{4}$$

where V_i and X_i represent the current velocity and the position of the ith particle respectively (Note that the subscripts k and k+1 refer to the recent and the next iterations respectively); P_i is the best previous position (*pbest*) of the ith particle and P_g is the best global position (*gbest*) among all the particles in the swarm; c_1 and c_2 represent “trust” parameters indicating how much confidence the current particle has in itself and how much confidence it has in the swarm; r_1 and r_2 are two random numbers between 0 and 1; and ω is the inertia weight.

The acceleration constants c_1 and c_2 indicate the stochastic acceleration terms which pull each particle towards the best position attained by the particle or the best position attained by the swarm. Low values of c_1 and c_2 allow the particles to wander far away from the optimum regions before being tugged back, while the high values pull the particles toward the optimum or make the particles to pass through the optimum abruptly. In reference [3], the constants c_1 and c_2 are chosen equal to 2 corresponding to the optimal value for the studied problem. In the same reference, it is mentioned that the choice of these constants is problem dependent. In this work, $c_1 = 2$ and $c_2 = 2$ are chosen.

The role of the inertia weight ω is considered important for the convergence behavior of PSO algorithm. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. Thus, the parameter ω regulates the trade off between the global (wide ranging) and the local (nearby) exploration abilities of the swarm. A proper value for the inertia weight ω provides balance between the global and local exploration ability of the swarm, and, thus results in better solutions. Experimental results imply that it is preferable to initially set the inertia to a large value, to promote global exploration of the search space, and gradually decrease it to obtain refined solutions [1]. Thus, a dynamic variation of inertia weight proposed in reference [2] is used in this paper. ω is decreased dynamically based on a fraction multiplier k_ω as is shown below

$$\omega_{k+1} = k_\omega \omega_k \tag{5}$$

The basic procedure of the PSO algorithm is constructed as follows:
(1) Initialize a set of particles positions and velocities randomly distributed throughout the design space bounded by specified limits.
(2) Evaluate the objective function values using the design space positions.
(3) Update the optimum particle position and global optimum particle position at current iteration.
(4) Update the current velocity vector and modify the position of each particle in the swarm using Eqs. (3) and (4).
(5) Repeat from step (2) until the stop criterion is achieved. The stopping criterion is usually defined as the number of iterations the PSO algorithm executes.

3.2 Constraints handling strategy

Most structural optimisation problems include the problem-specific constraints and the variable limits. For the present truss structural optimisation, the problem-specific constraints consist of the stress, buckling and displacement constraints, and the variable limits are the bounds of the truss member's cross-sectional areas and the nodal coordinates. If a particle flies out of the variable boundaries, the solution cannot be used even if the problem-specific constraints are satisfied, so it is essential to make sure that all of the particles fly inside the variable boundaries, and then to check whether they violate the problem-specific constraints.

Harmony search scheme*: handling the variable limits*

A method introduced by Li and Huang [7] dealing with the particles that fly outside the variables boundary is used in the present study.

This method is derived from the harmony search (HS) algorithm [9]. In the HS algorithm, the harmony memory (HM) stores the feasible vectors, which are all in the feasible space. The harmony memory size determines how many vectors can be stored. A new vector is generated by selecting the components of different vectors randomly in the harmony memory. Undoubtedly, the new vector does not violate the variables boundaries, but it is not certain if it violates the problem-specific constraints. When it is generated, the harmony memory will be updated by accepting this new vector if it gets a better solution and deleting the worst vector. Similarly, the PSO algorithm stores the feasible and "good" vectors (particles) in the *pbest* swarm, like does the harmony memory in the HS algorithm. Hence, the vector (particle) violating the variables boundaries can be generated randomly again by such a strategy: if any component of the current particle violates its corresponding boundary, then it will be replaced by selecting the corresponding component of the particle from *pbest* swarm randomly. To highlight the presentation, a schematic diagram is given in Fig. 1 to illustrate this strategy.

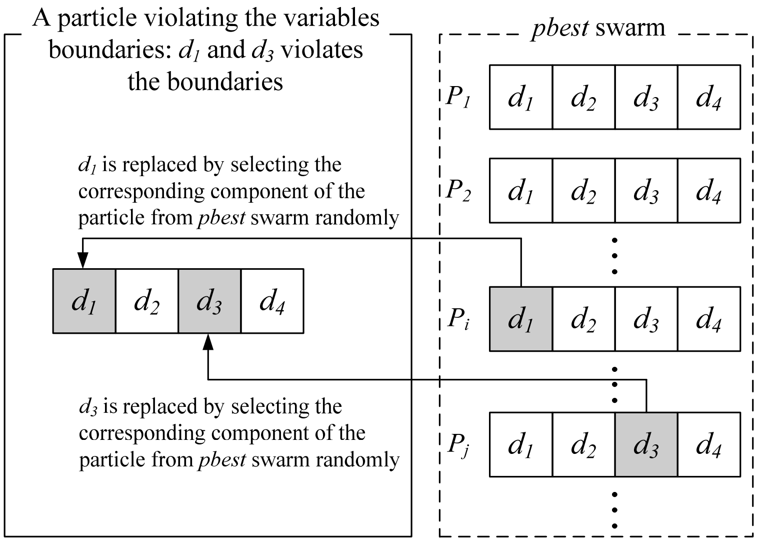

Figure 1: Illustration of the variable limits handling strategy

***Penalty functions method**: handling the problem-specific constraints*

The most common method to handle the constraints is the use of a penalty function. The constrained problem is transformed to an unconstrained one, by penalizing the constraints and building a single objective function. The optimisation problem reduces to minimize the objective function with the penalty function together.

In this paper, a non-stationary, multi-stage penalty function method implemented by Parsopoulos and Vrahatis in [10] is adopted for constraints handling in PSO. The penalty function used in [10] is

$$F(X) = f(X) + h(k)H(X), \quad X \in S \subset R^n \tag{6}$$

where $f(X)$ is the original objective function to be optimized, $h(k)$ is a penalty value which is modified according to the algorithm's current iteration number k and usually set to $h(k) = \sqrt{k}$. $H(X)$ is a penalty factor defined as

$$H(X) = \sum_{i=1}^{m} \theta(q_i(X)) q_i(X)^{\gamma(q_i(X))} \tag{7}$$

where $q_i(X)$ is a relative violated function of the constraints, which is defined as $q_i(X) = \max\{0, g_i(X)\}$ (Note that $g_i(X)$ is the constraint); $\theta(q_i(X))$ is an assignment function, and $\gamma(q_i(X))$ is the power of the penalty function.

Regarding the [10], the following values are used for the penalty function:
If $q_i(X)<1$, then $\gamma(q_i(X))=1$;
otherwise $\gamma(q_i(X))=2$;
If $q_i(X)<0.001$, then $\theta(q_i(X))=10$;
else if $q_i(X)\leq 0.1$, then $\theta(q_i(X))=20$;
else if $q_i(X)\leq 1$, then $\theta(q_i(X))=100$;
otherwise $\theta(q_i(X))=300$

3.3 Algorithm description

Based on the basic PSO algorithm and constraints handling technology as explained above, the pseudo-code for the present algorithm is listed in Table 1.

```
Set k = 0;
Randomly initialize positions and velocities of all particles distributed throughout
the design space
Calculate the objective function f(X_i^k) and penalty function F(X_i^k) of the
initialized particle
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.
WHILE (the maximum number of iterations are not met)
    FOR (each particle i in the swarm)
        Generate the velocity and update the position of the current particle X_i^k.
        Variable limit handing: Check whether each component of the current
        particle violates its corresponding boundary or not. If it does, select the
        corresponding component of the vector from pbest swarm randomly.
        Calculate the objective function f(X_i^k) and penalty function F(X_i^k) of the
        current particle.
        Update pbest: Compare the penalty function value of pbest with F(X_i^k). If
        the F(X_i^k) is better than the penalty function value of pbest, set pbest to the
        current position X_i^k.
        Update gbest: Find the global best position in the swarm. If the F(X_i^k) is
        better than the penalty function value of gbest, gbest is set to the position of
        the current particle X_i^k
    END FOR
    Set k = k +1
END WHILE
```

Table 1: The pseudo-code for the present PSO algorithm

4 Numerical examples

In this section, three truss structures commonly used in literature are selected as benchmark problems to test the present PSO algorithm. The optimized solutions are then compared to those of the existing results. The examples given in the simulation studies include:

- Simultaneous size and shape optimisation of a 15-bar planar truss structure illustrated in Fig. 2.
- Simultaneous size and shape optimisation of a 18-bar planar truss structure illustrated in Fig. 5.
- Simple size optimisation and simultaneous size and shape optimisation of a 25-bar space truss structure illustrated in Fig. 8.

For all these problems, a population of 50 individuals is used. The maximum number of iterations is limited to 1000.

4.1 15-bar planar truss structure

The weight of the 15-bar planar truss shown in Fig. 2 is firstly to be minimized with stress constraints. A tip load of 10*kips* is applied to the truss. The stress limit is 25*ksi* in both tension and compression for all truss members. The modulus of elasticity is specified as $1.0\times10^4 ksi$, and the material density as $0.1 lb/in^3$. The x and y coordinates of nodes 2, 3, 6, and 7 are allowed to vary, nodes 6 and 7 being constrained to have the same x coordinates as nodes 2 and 3 respectively. Nodes 4 and 8 are allowed to move only in the y direction. Hence, there are 23 independent design variables which include 15 sizing variables (cross-sectional areas of the truss members) and 8 shape variables ($x_2 = x_6$, $x_3 = x_7$, y_2, y_3, y_4, y_6, y_7, y_8). The bounds on the member cross-sectional areas are $0.1in^2$ and $20.0in^2$. Side constraints for the shape variables are $100in \le x_2 \le 140in$, $220in \le x_3 \le 260in$, $100in \le y_2 \le 140in$, $100in \le y_3 \le 140in$, $50in \le y_4 \le 90in$, $-20in \le y_6 \le 20in$, $-20in \le y_7 \le 20in$, $20in \le y_8 \le 60in$.

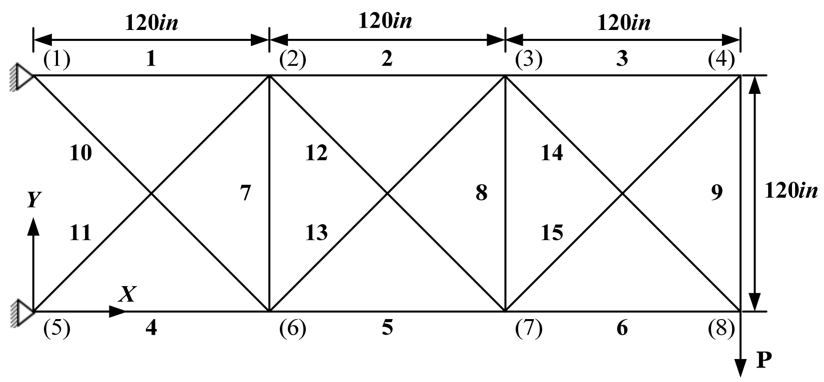

Figure 2: 15-bar planar truss

Variables	Optimal cross-sectional areas A (in^2) and coordinates (in)				
	Wu [11]	Hwang [12]	Tang [13]	Rahami [14]	Present study
A_1	1.174	0.954	1.081	1.081	1.021
A_2	0.954	1.081	0.539	0.539	0.508
A_3	0.440	0.440	0.287	0.287	0.282
A_4	1.333	1.174	0.954	0.954	1.081
A_5	0.954	1.488	0.954	0.539	0.685
A_6	0.174	0.270	0.220	0.141	0.130
A_7	0.440	0.270	0.111	0.111	0.126
A_8	0.440	0.347	0.111	0.111	0.134
A_9	1.081	0.220	0.287	0.539	0.551
A_{10}	1.333	0.440	0.220	0.440	0.388
A_{11}	0.174	0.220	0.440	0.539	0.413
A_{12}	0.174	0.440	0.440	0.270	0.317
A_{13}	0.347	0.347	0.111	0.220	0.273
A_{14}	0.347	0.270	0.220	0.141	0.194
A_{15}	0.440	0.220	0.347	0.287	0.287
x_2	123.189	118.346	133.612	101.5775	105.3433
x_3	231.595	225.209	234.752	227.9112	247.8316
y_2	107.189	119.046	100.449	134.7986	115.8325
y_3	119.175	105.086	104.738	128.2206	104.4993
y_4	60.462	63.375	73.762	54.8630	56.3801
y_6	-16.728	-20.0	-10.067	-16.4484	-18.6139
y_7	15.565	-20.0	-1.339	-13.3007	-12.7236
y_8	36.645	57.722	50.402	54.8572	49.9081
Weight (lb)	120.528	104.573	79.820	76.6854	78.739

Table 2: Comparison of optimal solutions of 15-bar planar truss with results reported in the literature

Fig. 3 gives a convergence rate of the optimisation procedure. It can be seen that the algorithm achieves the best solutions after 1000 iterations and it is close to the best solution after about 100 iterations.

The best vector found using the PSO approach is listed in Table 2 and compared with the results obtained using other mathematical methods. The corresponding objective function value (weight of the structure) is 78.739*lb*. The optimal configuration is illustrated in Fig. 4.

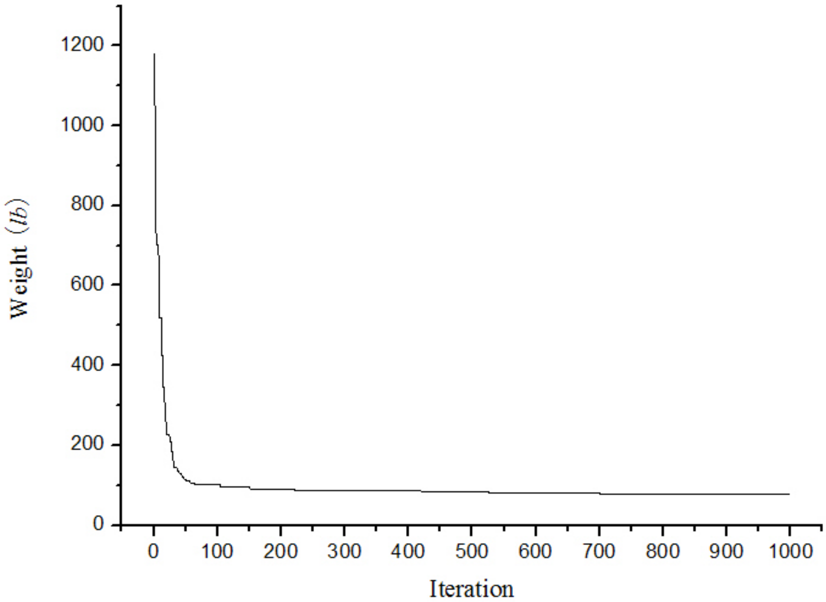

Figure 3: Convergence rate for the size and shape optimisation of 15-bar planar truss

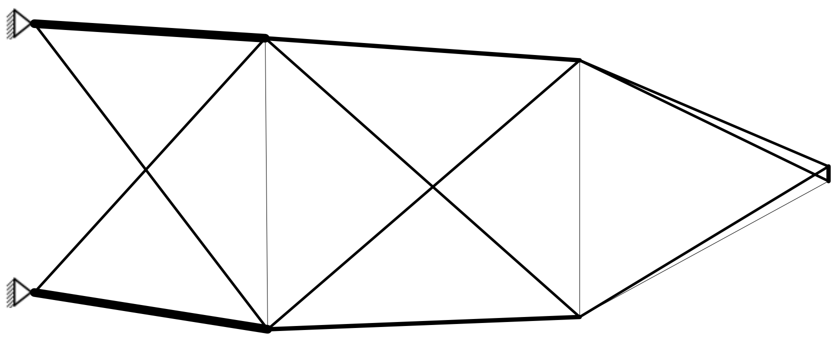

Figure 4: Optimized configuration of 15-bar planar truss

4.2 18-bar planar truss structure

The second example considers the 18-bar planar truss shown in Fig. 5. The optimisation objective is the minimization of structural weight where the design variables are specified as the cross-sectional area for the truss members and the coordinates of the nodes 3, 5, 7 and 9 (bottom part of the structure). The cross-sectional areas of the members were categorized into four groups: (1): $A_1 = A_4 = A_8 = A_{12} = A_{16}$, (2): $A_2 = A_6 = A_{10} = A_{14} = A_{18}$, (3): $A_3 = A_7 = A_{11} = A_{15}$, and (4): $A_5 = A_9 = A_{13} = A_{17}$. Thus, there are a total of 12 independent design variables that included four sizing and eight coordinate variables. The bounds on the member cross-sectional areas are $3.5in^2$ and $18.0in^2$. Side constraints for the coordinate variables are $775in \le x_3 \le 1225in$, $525in \le x_5 \le 975in$, $275in \le x_7 \le 725in$, $25in \le x_9 \le 475in$, $-225in \le y_3, y_5, y_7, y_9 \le 245in$.

The material density is $0.1 lb/in^3$ and the modulus of elasticity is $1.0\times10^4 ksi$. The single loading condition considered in this study is a set of vertical loads, P = $20kips$, acting on the upper nodal points of the truss, as illustrated in Fig. 5.
The allowable stresses and the buckling constraints are imposed. In this problem, the allowable tensile and compressive stresses are $20ksi$. The Euler buckling compressive stress limit for truss member used for the buckling constraints is computed as:

$$\left|(\sigma_c)_i\right| \le \left|{}_b\sigma_i\right| = \left|-\alpha E A_i / L_i^2\right|$$

where α is a constant determined from the cross-sectional geometry and is taken to be $\alpha = 4$ in this study. E is the modulus of elasticity of the material, and L_i is the member length.

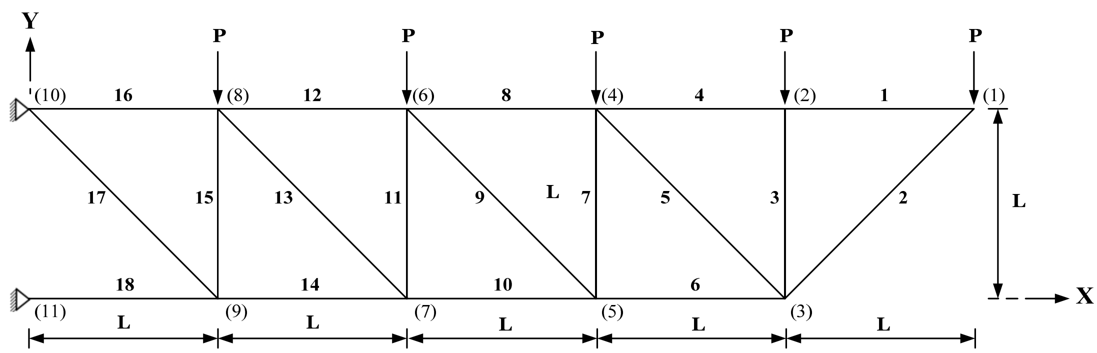

Figure 5: 18-bar planar truss

Fig. 6 provides a convergence rate of the optimisation procedure. It can be seen that the algorithm achieve the best solutions after 1000 iterations and it is close to the best solution after about 500 iterations.

The optimized solution found using the presented approach is listed in Table 3, and the corresponding objective function value (weight of the structure) is $4599.4lb$. A comparison to other references with respect to the cross-sectional area of each truss member, the nodal coordinates and the final weight reached for the 18-bar planar truss is shown in the Table 3. The optimal configuration is illustrated in Fig. 7.

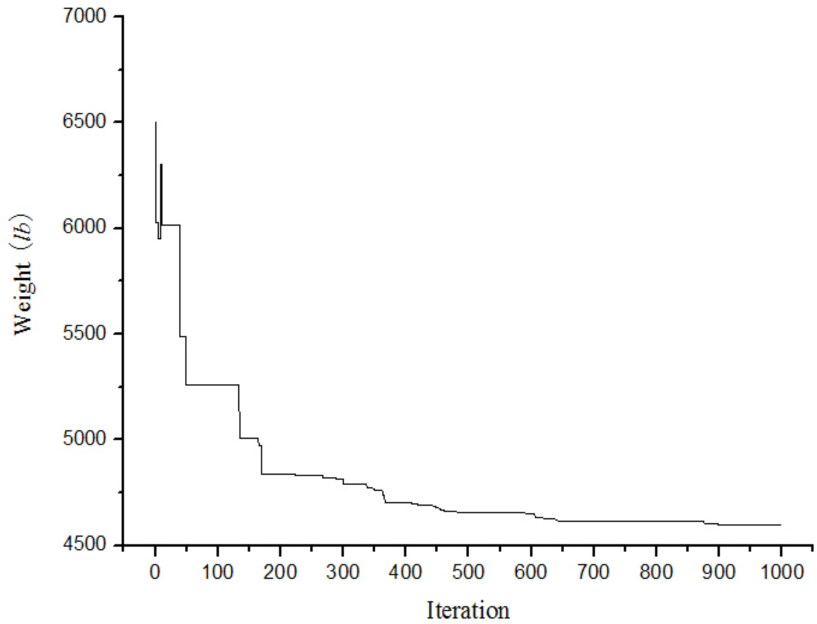

Figure 6: Convergence rate for the size and shape optimisation of 18-bar planar truss

Variables	Optimal cross-sectional areas A (in.^{2}) and coordinates (in.)					
	Felix [15]	Rajeev [16]	Yang [17]	Soh [18]	Yang [19]	Present study
A_1	11.34	12.50	12.61	12.59	12.33	12.08
A_2	19.28	16.25	18.10	17.91	17.97	17.16
A_3	10.97	8.000	5.470	5.500	5.600	7.015
A_4	5.300	4.000	3.540	3.550	3.660	4.492
x_3	994.6	891.9	914.5	909.8	907.2	912.9
y_3	162.3	145.3	183.0	184.5	184.2	178.9
x_5	747.4	610.6	647.0	640.3	643.3	641.5
y_5	102.9	118.2	147.4	147.8	149.2	127.2
x_7	482.9	385.4	414.2	410.0	413.9	406.7
y_7	33.00	72.50	100.4	97.00	102.2	79.80
x_9	221.7	184.4	200.0	200.9	202.1	196.5
y_9	17.10	23.40	31.90	32.00	30.90	20.95
Weight (*lb*)	5713.0	4616.8	4552.8	4531.9	4520.0	4599.4

Table 3: Comparison of optimal solutions of 18-bar planar truss with results reported in the literature

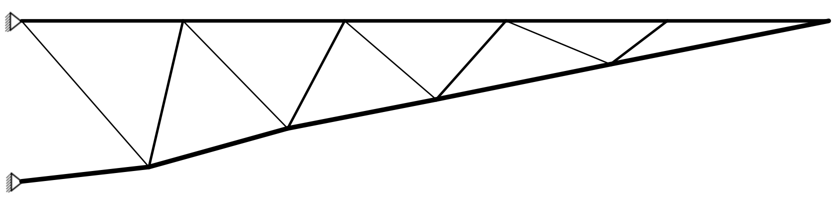

Figure 7: Optimized configuration of 18-bar planar truss

4.3 25-bar space truss structure

The third example concerns the weight minimization of o 25-bar transmission tower shown in Fig. 8. The design variables are the cross-sectional areas for the truss members, which are divided into eight member groups, as follows: (1): A1, (2): A2-A5, (3): A6-A9, (4): A10-A11, (5): A12-A13, (6): A14-A17, (7): A18-A21 and (8): A22-A25. The range of cross-sectional areas varies from 0.01 to 3.4 in^2. The material density is $0.1 lb/in^3$ and the modulus of elasticity is $1.0\times10^4 ksi$. The stress limits of the truss members are listed in Table 4. All nodes in all directions are subjected to the displacement limits of $\pm0.35in$. The load case listed in Table 5 is considered.

Variables	Compressive stress limitations (*ksi*)	Tensile stress limitations (*ksi*)
A_1	35.092	40.0
A_2	11.590	40.0
A_3	17.305	40.0
A_4	35.092	40.0
A_5	35.092	40.0
A_6	6.759	40.0
A_7	6.959	40.0
A_8	11.082	40.0

Table 4: Member stress limitations for 25-bar truss

Node	Concentrated loads		
	P_X (kips)	P_Y (kips)	P_Z (kips)
1	1.0	10.0	-5.0
2	0.0	10.0	-5.0
3	0.5	0.0	0.0
6	0.5	0.0	0.0

Table 5: Loads for 25-bar truss

First, the weight of this truss is minimized considering the cross-sectional areas of bars only. The optimal solution of size optimisation is given in Table 6 and compared with the results using other methods. It is illustrated in Fig. 9. The convergence rate of PSO algorithm is given in Fig. 10. It can be seen that the algorithm achieve the best solutions after 1000 iterations and it is close to the best solution after about 100 iterations.

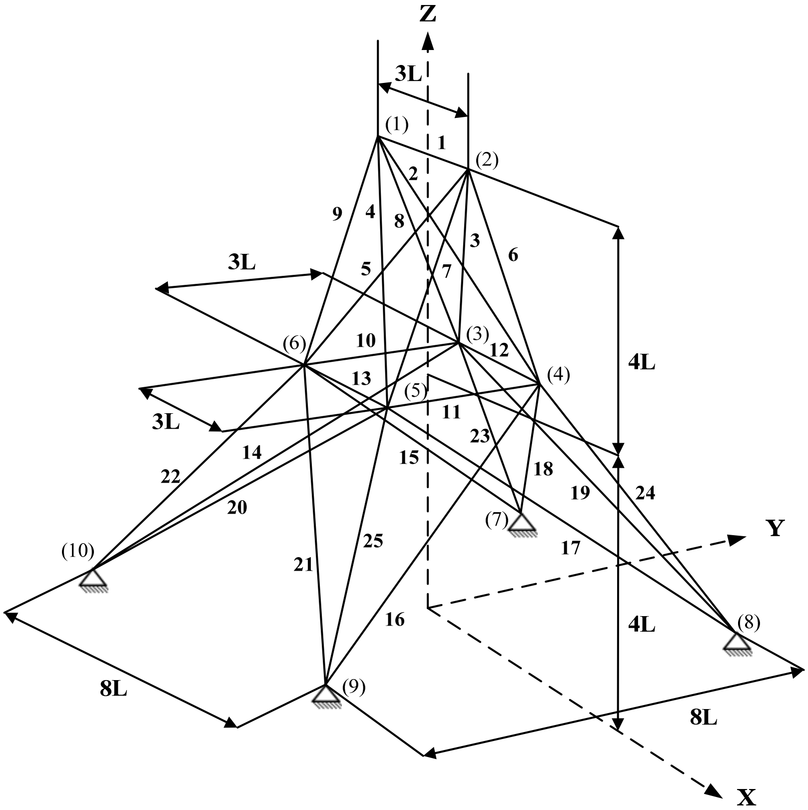

Figure 8: 25-bar space truss

Methods	Optimal cross-sectional areas								Weight
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	(lb)
Zhu [20]	0.01	1.90	2.60	0.01	0.01	0.80	2.10	0.60	562.93
Erbatur [21]	0.01	1.20	3.20	0.01	1.10	0.90	0.40	3.40	493.80
Wu and Chow [22]	0.01	0.50	3.40	0.01	1.50	0.90	0.60	3.40	486.29
Perez [8]	0.01	1.02	3.40	0.01	0.01	0.64	2.04	3.40	485.33
Present study	0.01	0.53	3.40	0.01	2.13	0.99	0.30	3.40	479.60

Table 6: Comparison of optimal solution of 25-bar truss with results reported in the literature

Now, the same 25-bar truss is simultaneously optimized for size and shape. The design variables are concerned with eight cross-sectional area groups and twelve nodal coordinates (x,y,z) of the four nodes: 3, 4, 5, 6. The range of cross-sectional areas varies from 0.01 in^2 to 3.4 in^2. The bounds for the coordinates are as follows

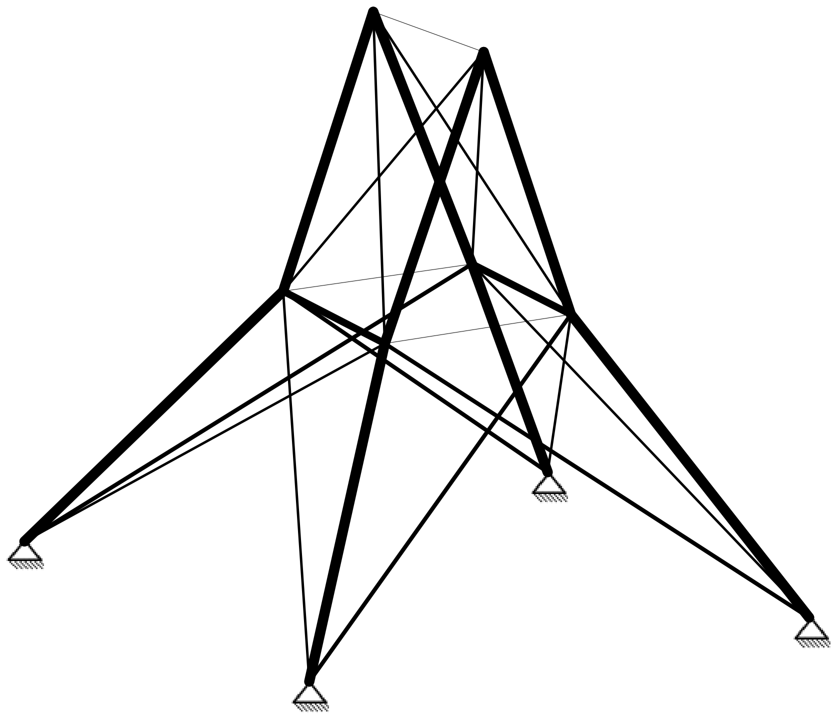

Figure 9: Size optimisation of 25-bar space truss

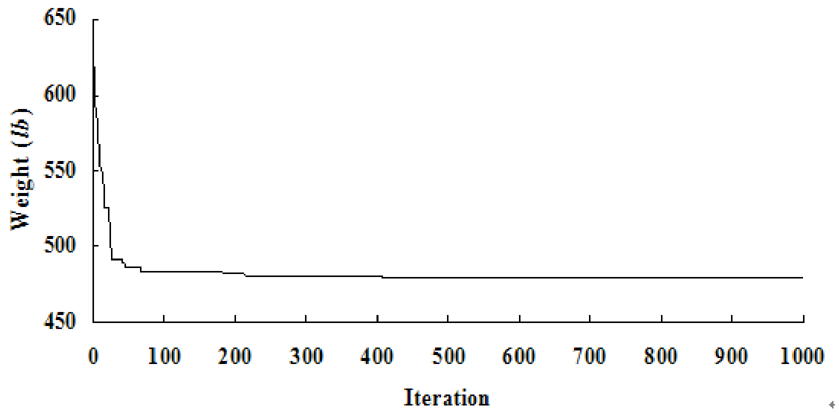

Figure 10: Convergence rate for the size optimisation of 25-bar space truss

$$-2.5L \le x(3) \le -0.5L,\ 0.5L \le y(3) \le 2.5L,\ 16L/3 \le z(3) \le 8L/3$$
$$2.5L \le x(4) \le 0.5L,\ 0.5L \le y(4) \le 2.5L,\ 16L/3 \le z(4) \le 8L/3$$
$$2.5L \le x(5) \le 0.5L,\ -2.5L \le y(5) \le -0.5L,\ 16L/3 \le z(5) \le 8L/3$$
$$-2.5L \le x(6) \le -0.5L,\ -2.5L \le y(6) \le -0.5L,\ 16L/3 \le z(6) \le 8L/3$$

The optimal solution obtained by the developed PSO algorithm is illustrated in Fig. 11. The minimum weight is 280.79 lb and represents 58% of the weight obtained by size optimisation (479.60 lb). This is evident that the simultaneous size and shape optimisation gives a weight which is smaller than a weight obtained by the simple size optimisation.

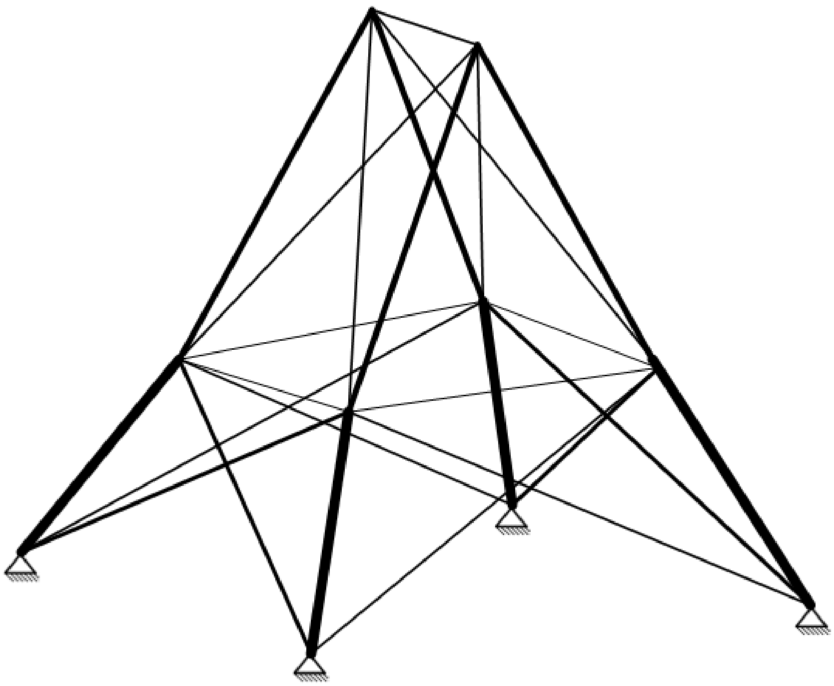

Fig. 11: Size and shape optimisation of 25-bar truss

5 Conclusions

The aim of this study is to test a new developed PSO algorithm in application to simultaneous size and shape optimisation of truss structures. The main difficulty of applying the PSO algorithm to structural optimisation is to satisfy the nonlinear constraints concerning the nodal displacements and element stresses. Also, the limit boundaries of cross-sectional areas and nodal coordinates should be taken into account. The classical PSO algorithm [1] has been modified to satisfy that all the particles fly inside the variable's boundaries. A method derived from the Harmony Search algorithm [9] was used to deal with the particles which fly outside the variables boundaries. The multi-stage penalty function method was adopted within PSO to satisfy the constraints of the optimisation problem.

The first numerical example is concerned with a simultaneous size and shape optimisation of a 15-bar planar truss structure. The obtained weight is 78.739 lb. When compared with other four results from the literature ([11],[12],[13],[14]) where the genetic algorithms were used for this example, only Rahami et al. [14] obtained a smaller weight (97% of our weight).

In the second example, a weight of a18-bar planar truss is minimized subject to the stress and buckling constraints. The design cross-sectional areas are grouped in four classes and there are eight variable coordinates of four bottom nodes. The comparison with other five results from the literature ([15],[16],[17],[18],[19]) based on genetic algorithms shows that the best result (98% of our weight) has been obtained by Yang and Soh [19] and the worst (124% of our weight) has been obtained by Feix [15].

The last example concerns the simple size and simultaneous size-shape optimisation of a 25-bar space transmission tower. The variable cross-sectional areas of truss members are classified into eight groups. In addition, there are twelve coordinates (x,y,z) of the nodes 3, 4, 5 and 6. A comparison with other four results from the literature [8, 20-22] for simple size optimisation shows that our result is the best. When comparing the simple size and simultaneous size-shape optimisation, this is evident that the simultaneous approach gives the better result (the optimal weight is reduced by 58%).

The developed modified PSO algorithm has been successfully applied to simultaneous size and shape optimisation of truss structures. In order to improve the efficiency of PSO for structural optimisation, the hybridization strategies combining other metaheuristic methods with gradient-based techniques and using rapid finite element reanalysis approach can be developed.

Acknowledgements

The second author gratefully acknowledges a support from the German Research Foundation.

References

[1] J. Kennedy and R.C. Eberhart, Particle swarm optimization, Proceedings of the IEEE international conference on neural networks, 4, 1942-1948, 1995.

[2] P. Fourie and A. Groenwold, The particle swarm optimization algorithm in size and shape optimization, Structural and Multidisciplinary Optimization, 23, 259-267, 2002.

[3] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters, 85, 317-325, 2003.

[4] G. Venter and J. Sobieszczanski-Sobieski, Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization, Structural and Multidisciplinary Optimization, 26, 121-131, 2004.

[5] S. He, E. Prempain and Q.H. Wu, An improved particle swarm optimizer for mechanical design optimization problems. Engineering Optimization, 36, 585-605, 2004.

[6] C. Elegbede, Structural reliability assessment based on particles swarm optimization, Structural Safety, 27, 171-186, 2005.

[7] L.J., Li Z.B. Huang, F. Liu and Q.H. Wu, A heuristic particle swarm optimizer for optimization of pin connected structures, Computers and Structures, 85,

340-349, 2007.
[8] R.E. Perez and K. Behdinan, Particle swarm approach for structural design optimization, Computers and Structures, 85, 1579-1588, 2007.
[9] K. Lee and Z. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering, 194, 3902-3933, 2005.
[10] K.E. Parsopoulos and M.N. Vrahatis, Particle swarm optimization method for constrained optimization problems, Proceedings of the 2nd Euro-International Symposium on Computational Intelligence, Kosice, Slovakia, 2002.
[11] S.J. Wu and P.T. Chow, Integrated discrete and configuration optimization of trusses using genetic algorithms, Computers and Structures, 55, 695-702, 1995.
[12] S.F. Hwang and R.S. He, A hybrid real-parameter genetic algorithm for function optimization, Advanced Engineering Informatics, 20, 7-21, 2006.
[13] W. Tang, L. Tong and Y. Gu, Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables, International Journal for Numerical Methods in Engineering, 62, 1737-1762, 2005.
[14] H. Rahami, A. Kaveh and Y. Gholipour, Sizing, geometry and topology optimization of trusses via force method and genetic algorithm, Engineering Structures, 30, 2360-2369, 2008.
[15] J.E. Felix, Shape optimization of trusses subjected to strength, displacement, and frequency constraints, Masters thesis, Naval Postgraduate School, 1981.
[16] S. Rajeev and C.S. Krishnamoorthy, Genetic algorithms based methodologies for design optimization of trusses. Journal of Structural Engineering, 123, 350-358, 1997.
[17] J.P. Yang, Development of genetic algorithm-based approach for structural optimization, Ph.D. thesis, Nanyang Technology University, Singapore, 1996.
[18] C.K. Soh and J.P. Yang, Fuzzy controlled genetic algorithm search for shape optimization, Journal of Computing in Civil Engineering, 10, 143-150, 1996.
[19] J.P. Yang and C.K. Soh, Structural optimization by genetic algorithms with tournament selection, Journal of Computing in Civil Engineering, 11, 195-200, 1997.
[20] D. Zhu, An improved Templemans algorithm for optimum design of trusses with discrete member sizes, Eng. Optimization, 9, 303-312, 1986.
[21] F. Erbatur, O. Hasançebi, I. Tütüncü and H. Kiliç, Optimal design of planar and space structures with genetic algorithms, Computers and Structures, 75, 209-224, 2000.
[22] S. Wu and P. Chow, Steady-state genetic algorithms for discrete optimization of trusses, Computers and Structures, 56, 979-991, 1995.