INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2013; 3(1):57-83

A HYBRID ALGORITHM FOR SIZING AND LAYOUT OPTIMIZATION OF TRUSS STRUCTURES COMBINING DISCRETE PSO AND CONVEX APPROXIMATION

S. Shojaee [*, †] M. Arjomand and M. Khatibinia
Department of Civil Engineering, Shahid Bahonar University, Kerman, Iran

ABSTRACT

An efficient method for size and layout optimization of the truss structures is presented in this paper. In order to this, an efficient method by combining an improved discrete particle swarm optimization (IDPSO) and method of moving asymptotes (MMA) is proposed. In the hybrid of IDPSO and MMA, the nodal coordinates defining the layout of the structure are optimized with MMA, and afterwards the results of MMA are used in IDPSO to optimize the cross-section areas. The results show that the hybrid of IDPSO and MMA can effectively accelerate the convergence rate and can quickly reach the optimum design.

Received: 17 April 201; Accepted: 12 December 2012

KEY WORDS: size optimization, layout optimization, improved discrete particle swarm optimization, method of moving asymptotes.

1. INTRODUCTION

Recently, meta-heuristic optimization methods such as genetic algorithms (GAs), evolutionary programming (EP), ant colony optimization (ACO), particle swarm optimization (PSO) and Big Bang and Big Crunch algorithm (BB-BC) have become more attractive [1]. These methods do not need conventional mathematical assumptions and thus increase chance of locating the global optimum than the conventional optimization algorithms. In particular, PSO [2] has relatively few internal parameters and a convenient floating point treatment of design variables. The number of capabilities that attractively

[*]Corresponding author: S. Shojaee, Department of Civil Engineering, Shahid Bahonar University, Kerman, Iran
[†]E-mail address: saeed.shojaee@uk.ac.ir (S. Shojaee)

increase this evolutionary algorithm is its lower number of parameter necessary to set before and its floating point treatment for the design variables. Successful applications of PSO to solve structural engineering problems are documented in [3-11]. Fourie and Groenwold [12] used PSO for sizing and shape optimization of skeletal and continuum structures. While the continuous PSO is in general highly suited for structural optimization, discrete PSO (DPSO) may be trapped in local minima if the optimization problem includes discrete variables. This explains why there are rather few examples of application of discrete PSO compared to continuous optimization problems. The applications of DPSO in discrete search space are relatively much less than those in continuous problem spaces.

The method of moving asymptotes (MMA) is an iterative mathematical programming technique which builds strictly convex approximations of the optimization problem [13, 14]. MMA uses both function values and first or second derivative values computed at the current design point found in the previous iterations to solve a sequence of linearized, convex and separable sub-problems. MMA was proven to be very efficient in solving large-scale topology optimization problems with multiple constraints [15, 16].

This paper describes an efficient hybrid algorithm for simultaneous sizing and layout optimization of truss structures subject to displacement and stress constraints. The proposed algorithm combines improved discrete PSO (IDPSO) and MMA. In IDPSO, a new scheme based on positive integer numbers is utilized to update the position of each particle. IDPSO and MMA are hybridized to form a two-stage optimization strategy for combined sizing and layout design of truss structures. MMA is built with respect to nodal coordinates as these layout variables are continuous while cross sectional areas (i.e. sizing variables) can take only discrete variables in IDPSO and hence must be fixed in MMA. In the first stage, cross-sectional areas of elements are assigned and the nodal coordinates defining the layout of the structure are optimized with MMA. In the second stage, the new configuration thus determined is used to update the cross-sectional areas of elements of the particles included in the swarm. Particles are evaluated with the approximate model, and the new velocities and positions are determined by means of the IDPSO scheme.

Although the single-stage approach involving simultaneous sizing-layout optimization is inherently more efficient it may be much more expensive computationally compared to the proposed two-stage approach which treats with design spaces including less design variables. Further complications may arise from the mixed coding scheme that has to be implemented to deal with discrete or continuous variables [17].

The new hybrid algorithm proposed in this research is tested in four benchmark problems of mixed sizing-layout weight minimization problems of truss structures. Results fully demonstrate the efficiency of the proposed algorithm.

2. PROBLEM FORMULATION

A structural optimization problem can be formulated in the following form,

$$\begin{aligned} &\textit{Minimize} \quad f(\mathrm{X}) \\ &\textit{Subject to} \quad g_i(\mathrm{X}) \le 0 \qquad i = 1,2,...,m \\ &\qquad \mathrm{X} = \{ x_1, x_2, ..., x_j, ..., x_n \} \in R^d \end{aligned} \tag{1}$$

where $f(\boldsymbol{X})$ represents objective function, $g(\boldsymbol{X})$ is the behavioral constraint, m and n are the number of constraints and design variables, respectively. A given set of values is expressed by R^d and design variables x_j can take values only from this set.

In the size and layout optimization of trusses, the aim is usually to minimize the weight of the truss structures under design constraints. The design variables are chosen to be cross-section areas of the elements and coordinates of nodes in the structure. The cross-section areas of the elements and coordinates of nodes are usually selected from a set of discrete and continuous available values, respectively. Therefore, the optimization problem can be reformulated in the following form,

$$\begin{aligned} &\text{Minimize} \quad f(\boldsymbol{X}) = W(A_i, X_j) = \sum_{i=1}^{Ne} \rho_i A_i L_i \\ &\text{Subject to} \quad g_{Si} = \frac{\sigma_i(A_i, X_j)}{\sigma_{all}} - 1 \le 0 \qquad i = 1, 2, ..., N_e \\ &\qquad g_{Di} = \frac{\Delta_j(A_i, X_j)}{\Delta_{all}} - 1 \le 0 \qquad j = 1, 2, ..., N_n \\ &\qquad A_i \in \boldsymbol{A}_e = \{ A_{e1}, A_{e2}, ..., A_{ep} \} \\ &\qquad X_j^L \le X_j \le X_j^U \end{aligned} \tag{2}$$

where W is the structural weight, ρ_i, A_i and L_i are the material density, cross-section area and length of the ith element; σ_i and σ_{all} are the stress of the ith element and the allowable axial stress; Δ_j and Δ_{all} are the displacement of the jth node and the allowable displacement; N_e and N_n are the number of elements and nodes in the structure; $\boldsymbol{A}_e$ is the available profile list; X_j^L and X_j^U are the lower and upper bounds of the node coordinate, respectively.

A number of constraint-handling techniques have been proposed to solve constrained optimization problems. In this study, the penalty function is used to deal with constrained search spaces as,

$$\hat{f}(X) = \begin{cases} f(\boldsymbol{X}) & \textit{if} \quad \boldsymbol{X} \in R^d \\ f(\boldsymbol{X}) + \sum_i \max(g_i(\boldsymbol{X}), 0.0) & \textit{otherwise} \end{cases} \tag{3}$$

where $\hat{f}(X)$ is modified function. Also, R^d denotes the feasible search space.

3. PARTICLE SWARM OPTIMIZATION

The Particle swarm optimization (PSO) was inspired by the social behavior of animals such as fish schooling, insects swarming and birds flocking. PSO was introduced by Kennedy and Eberhart in the mid 1990s, to simulate the graceful motion of bird swarms as a part of a socio-cognitive study. It involves a number of particles that are initialized randomly in the search space of an objective function. These particles are referred to as swarm. Each particle of the swarm represents a potential solution of the optimization problem. The *i*th particle in *t*th iteration is associated with a position vector, $\boldsymbol{X}_i^t$, and a velocity vector, $\boldsymbol{V}_i^t$, that shown as following,

$$\begin{aligned} \boldsymbol{X}_i^t &= \{ x_{i1}^t, x_{i2}^t, ..., x_{iD}^t \} \\ \boldsymbol{V}_i^t &= \{ v_{i1}^t, v_{i2}^t, ..., v_{iD}^t \} \end{aligned} \tag{4}$$

where *D* is dimension of the solution space.

The particle fly through the solution space and its position is updated based on its velocity, the best position particle (***pbest***) and the global best position (***gbest***) that swarm has visited since the first iteration as,

$$\boldsymbol{V}_i^{t+1} = \omega^t \boldsymbol{V}_i^t + c_1 r_1 (\boldsymbol{pbest}_i^t - \boldsymbol{X}_i^t) + c_2 r_2 (\boldsymbol{gbest}^t - \boldsymbol{X}_i^t) \tag{5}$$

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \boldsymbol{V}_i^{t+1} \tag{6}$$

where r_1 and r_2 are two uniform random sequences generated from interval [0, 1]; c_1 and c_2 are the cognitive and social scaling parameters, respectively and ω^t is the inertia weight that controls the influence of the previous velocity.

Shi and Eberhart (1998) proposed that the cognitive and social scaling parameters c_1 and c_2 should be selected as c_1=c_2=2 to allow the product c_1r_1 or c_2r_2 to have a mean of 1. The performance of PSO is very sensitive to the inertia weight (ω) parameter which may decrease with the number of iteration (Shi and Eberhart 1998) as follows:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} . t \tag{7}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of ω, respectively; and t_{max} is the limit numbers of optimization iteration.

3.1. Discrete PSO

The Particle swarm optimization (PSO) was inspired by the social behavior of animals such as fish schooling, insects swarming and birds flocking. PSO was introduced by Kennedy and

Eberhart [2] in the mid 1990s, to simulate the graceful motion of bird swarms as a part of a socio-cognitive study. It involves a number of particles that are initialized randomly in the search space of an objective function. These particles are referred to as swarm. Each particle of the swarm represents a potential solution of the optimization problem. The *i*th particle in *t*th iteration is associated with a position vector, $\boldsymbol{X}_i^t$, and a velocity vector, $\boldsymbol{V}_i^t$, that is shown as follows,

$$\begin{aligned} \boldsymbol{X}_i^t &= \{ x_{i1}^t, x_{i2}^t, ..., x_{iD}^t \} \\ \boldsymbol{V}_i^t &= \{ v_{i1}^t, v_{i2}^t, ..., v_{iD}^t \} \end{aligned} \tag{4}$$

where *D* is the dimension of the solution space.

The particle fly through the solution space and its position is updated based on its velocity, the best position particle (***pbest***) and the global best position (***gbest***) that swarm has visited since the first iteration as,

$$\boldsymbol{V}_i^{t+1} = \omega^t \boldsymbol{V}_i^t + c_1 r_1 (\boldsymbol{pbest}_i^t - \boldsymbol{X}_i^t) + c_2 r_2 (\boldsymbol{gbest}^t - \boldsymbol{X}_i^t) \tag{5}$$

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \boldsymbol{V}_i^{t+1} \tag{6}$$

where r_1 and r_2 are two uniform random sequences generated from interval [0, 1]; c_1 and c_2 are the cognitive and social scaling parameters, respectively and ω^t is the inertia weight that controls the influence of the previous velocity.

Shi and Eberhart [18] proposed that the cognitive and social scaling parameters c_1 and c_2 should be selected as c_1=c_2=2 to allow the product c_1r_1 or c_2r_2 to have a mean of 1. The performance of PSO is very sensitive to the inertia weight (ω) parameter which may decrease with the number of iteration [18] as follows:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} . t \tag{7}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of ω, respectively; and t_{max} is the maximum number of optimization iteration.

3.2. Discrete PSO

Kennedy and Eberhart also developed the discrete particle swarm optimization (DPSO) algorithm to solve problems with binary-valued design variables [19]. Since positions of particles are defined by 1 or 0, any particle can move in a binary design space including only values 0 and 1. New definitions for velocity, distance and movement path must be presented in terms of the probability that bits have to be in a specific position. The velocity of each particle is related to this probability. In DPSO, Eq. (5) can be used without any changes for

updating the velocity which has to be converted in the interval [0,1] by a logic function. The particle position update Eq. (6) must instead be redefined as:

$$x_{ij}^{t+1} = \begin{cases} 1 & \text{for } rand(.) < S(v_{ij}^{t+1}) \\ 0 & \text{for } rand(.) \geq S(v_{ij}^{t+1}) \end{cases} \tag{8}$$

where $rand(.)$ is a random number selected from a uniform distribution in the interval [0,1], and $S(v_{ij}^{t+1})$ is a converting sigmoid function expressed as [19]:

$$S(v_{ij}^{t+1}) = \text{sigmode}(v_{ij}^{t+1}) = \frac{1}{1+\exp(-v_{ij}^{t+1})} \tag{9}$$

To avoid $S(v_{ij}^{t+1})$ approaching 0 or 1, a constant V_{max} is used to limit the range of velocity. $\boldsymbol{V}_{max}$ is often limited to 4 so that $v_{ij}^{t+1} \in [-4, 4]$ and

$$v_{ij}^{t+1} = \begin{cases} V_{\max} & \text{if } v_{ij}^{t+1} > V_{\max} \\ -V_{\max} & \text{if } v_{ij}^{t+1} < -V_{\max} \\ v_{ij}^{t+1} & \text{otherwise} \end{cases} \tag{10}$$

As DPSO approaches the optimum design, the probability of being 1 or 0 becomes 0.5 regardless of previous values of velocity. Since the particle position updating probability cannot tend to 0, DPSO slowly converges to the global optimum. In order to solve this problem, a modified sigmoid function was introduced by Rostami and Nezamabadipour [20]:

$$\bar{S}(v_{ij}^{t+1}) = 2\left| sigmoid(v_{ij}^{t+1}) - \frac{1}{2} \right| \tag{11}$$

where $\bar{S}(\cdot)$ is a modified sigmoid function.

3.2 The improved DPSO

Storage of particle positions in DPSO requires large computer memory. In addition, coding and encoding of particle positions is computationally expensive. In order to overcome these limitations, different discrete versions of PSO were proposed in literature [21-29].

This study presents an improved DPSO (IDPSO) algorithm where each particle is coded in terms of positive integer numbers. Therefore, the scalar $x_{ij}^{t} \in \{1, 2, ..., p\}$ corresponds to one of the discrete values included in the set $\{A_1, A_2, ..., A_p\}$ of available cross-sections. The definition of velocity remains the same as in the standard DPSO but the position of particles

is now updated as follows:

$$x_{ij}^{t+1} = \begin{cases} INT(x_{ij}^{t} + v_{ij}^{t+1}) & \text{for } rand(.) < \bar{S}(v_{ij}^{t+1}) \\ x_{ij}^{l} & \text{for } rand(.) \geq \bar{S}(v_{ij}^{t+1}) \end{cases} \tag{12}$$

In order to avoid stagnation of particle positions, the following update rule is utilized:

$$INT(x_{ij}^{t} + v_{ij}^{t+1}) = \begin{cases} pbest_{ij}^{t} & t < t_{gbest} \\ gbest_{j}^{t} & t \geq t_{gbest} \\ INT(x_{ij}^{t} + v_{ij}^{t+1}) & otherwise \end{cases} \tag{13}$$

where $pbest_{ij}^{t}$ and $gbest_{j}^{t}$ are the *j*th component of best position particle (pbest) in the *t*th iteration and the *j*th component of the global best position (gbest), respectively; and t_{gbest} is the optimization iteration at which the global best was achieved

4. METHOD OF MOVING ASYMPTOTES

The method of moving asymptotes (MMA) developed by Svanberg [13] is a generalization of the convex linearization method (CONLIN) [30] without global convergence. The optimization problem (1) is reformulated by including artificial variables $\boldsymbol{Y} = (y_1, \ldots, y_m)^T$:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{X}) + a_0 z + \sum_{i=1}^{m} (c_i y_i + \frac{1}{2} d_i y_i^2) \\ \text{subject to} \quad & g_i(\boldsymbol{X}) - a_i z - y_i \leq 0 \qquad i = 1,...,m \\ & x_j^{\min} \leq x_j \leq x_j^{\max} \qquad j = 1,...,n \\ & z \geq 0, y_i \geq 0 \qquad i = 1,...,m \end{aligned} \tag{14}$$

where $g_0, g_1, \ldots, g_m$ are given, twice differentiable and real-valued functions, while a_0, a_i, c_i and d_i are given real numbers so that $a_0 > 0, a_i > 0, c_i > 0, d_i > 0$ and $c_i + d_i > 0$ for $i = 1, 2, \ldots, m$.

MMA can be applied to structural optimization problems where cost function and constraint gradient evaluations are very time consuming. The algorithm includes outer and inner iterations denoted by indices *k* and *l*, respectively. A sub-problem is generated and solved with respect to design variables $\boldsymbol{X}^{(k)}$ and artificial variables $\boldsymbol{Y}^{(k)}$. The sub-problem is obtained from the original problem (2) by replacing cost function and non-linear constraints with separable strictly convex approximating functions $\tilde{f}^{(k,l)}$ and $\tilde{g}_i^{(k,l)}$. That is:

$$\begin{aligned} \text{minimize} \quad & \tilde{f}^{(k,l)}(\boldsymbol{X}) + a_0 z + \sum_{i=1}^{m}(c_i y_i + \frac{1}{2} d_i y_i^2) \\ \text{subject to} \quad & \tilde{g}_i^{(k,l)}(\boldsymbol{X}) - a_i z - y_i \le 0, \qquad i = 1,...,m \\ & \alpha_j^{(k)} \le x_j \le \beta_j^k, \qquad j = 1,...,n \\ & z \ge 0, y_i \ge 0, \qquad i = 1,...,m \end{aligned} \tag{15}$$

where

$$\alpha_j^{(k)} = \max\{ x_j^{\min}, 0.9 l_j^{(k)} + 0.1 x_j^{(k)}\}, \quad \beta_j^{(k)} = \max\{ x_j^{\max}, 0.9 u_j^{(k)} + 0.1 x_j^{(k)}\} \tag{16}$$

and $\tilde{f}_i^{(k,l)}$ is the approximating functions. The parameters a_o, a_i and z (Eqs. 14 and 15) are set and updated according to Ref [14]. The rules for updating the lower asymptotes $l_j^{(k)}$ and the upper $u_j^{(k)}$ asymptotes have been presented by Svanberg [14].

5. FORMULATION OF THE HYBRID IDPSO-MMA

In sizing-layout optimization of truss structures cross-sectional areas of elements and nodal coordinates can vary in discrete and continuous search spaces, respectively. The hybrid optimization algorithm developed in this research combines the search capabilities of IDPSO and MMA in the two distinct design sub-spaces. A two-stage optimization strategy is followed. After having initialized the cross-sectional areas of elements, the layout of the structure is optimized with MMA by including only the nodal coordinates as design variables. The optimized layout is hence included in the design vectors corresponding to particles. In the second stage of the optimization process, IDPSO is utilized to update particle positions with respect to sizing variables and to find the global/local best design points with respect to which convex approximation is built in the subsequent layout optimization cycle. It should be noted that the best layout of truss structure (i.e. the best particle) that each particle has visited since the first iteration is selected as the base point necessary to construct the convex approximation for MMA in the subsequent layout optimization cycle. This approach allows the convergence to optimum design to be speeded up.

The basic steps of the hybrid optimization algorithm can be outlined as follows:

Step 1: Set internal parameters of IDPSO (c_1, c_2, ω) and population size of IDPSO (N_{PSO}). Also set the limit number of optimization iterations (l_{max}).

Step 2: Randomly initialize positions $(\boldsymbol{X}_i^0)$ as initial cross-sectional areas of elements with the initial configuration of truss structure and velocities $(\boldsymbol{V}_i^0)$ for $i = 1,2,...,N_{PSO}$.

Step 3: Optimize the layout of the structure with MMA for each particle.

Step 4: Update $\hat{f}_{i,pbest}^t$ for each particle and $\hat{f}_{gbest}^t = \min(\hat{f}_i^{t+1})$.

Step 5: Evaluate structural response and weight for each particle by including the

optimized layout variables.

Step 6: If $\hat{f}_i^{t+1} < \hat{f}_{i,pbest}^{t}$ then $\hat{f}_{i,pbest}^{t} = \hat{f}_i^{t+1}$, $\boldsymbol{pbest}_i^{t+1} = \boldsymbol{X}_i^{t+1}$ for $i = 1, 2, \ldots, N_{PSO}$.

Step 7: If $\min(\hat{f}_i^{t+1}) < \hat{f}_{gbest}^{t}$ then $\hat{f}_{gbest}^{t} = \min(\hat{f}_i^{t+1})$, $\boldsymbol{gbest}^{t+1} = \boldsymbol{X}_{i,\min}^{t+1}$.

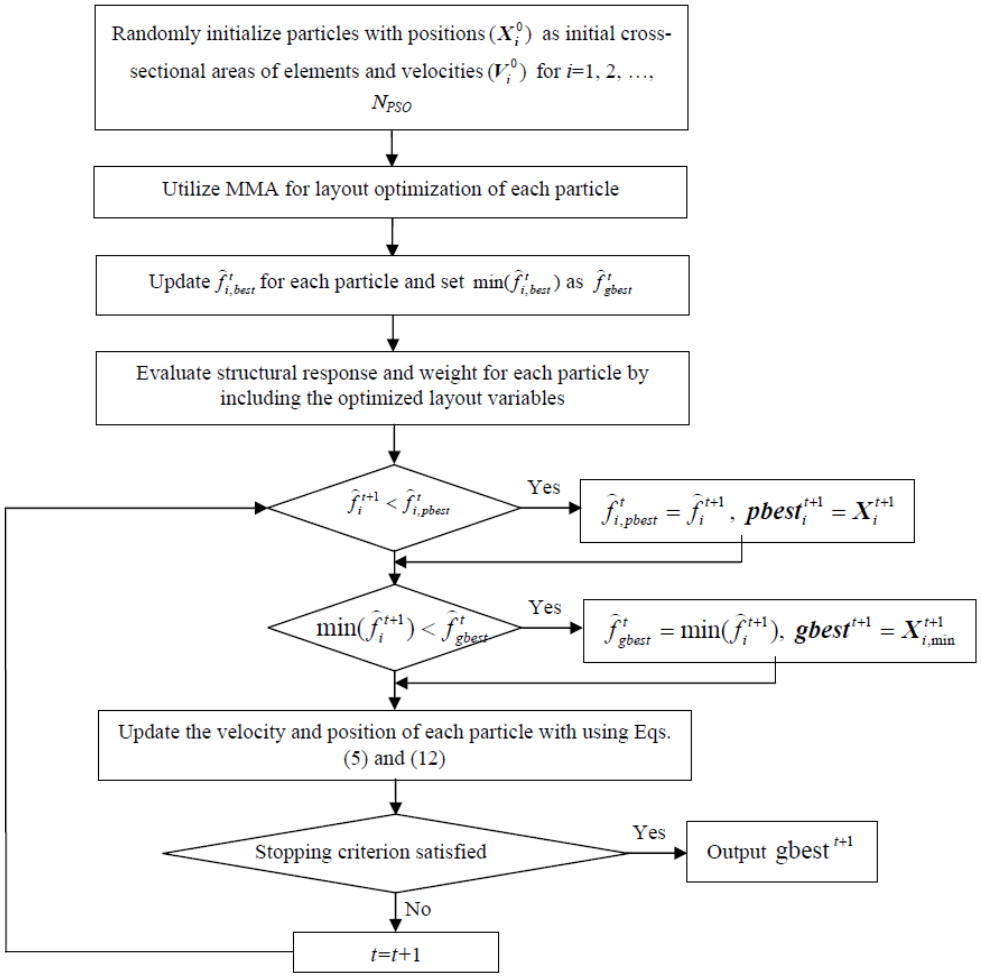

Fig. 1. Flow chart for IDPSO-MMA algorithm

Step 8: Update particle velocity $\boldsymbol{V}_j^{t+1}$ and particle position $\boldsymbol{X}_j^{t+1}$ using Eqs. (5) and (12), respectively.

Step 9: If $t < t_{\max}$, then stop; otherwise go to step 3.

Figure 1 shows the flow-chart of the hybrid IDPSO-MMA algorithm developed in this study. To assess the efficiency of the proposed IDPSO-MMA, the optimization is performed by using DPSO-MMA and IDPSO-MMA methods at first, and then the optimal solutions of IDPSO-MMA is compared with those of DPSO-MMA. The DPSO-MMA algorithm is the same IDPSO-MMA algorithm, but Eq. (8) is used in stead of Eq. (12) for updating of particle position in step 8, which is mentioned earlier.

6. TEST CASES AND RESULTS

Four classical weight minimization problems of truss structures were solved in order to check the numerical efficiency of the hybrid IDPSO-MMA algorithm developed in this research. Optimization results were compared with data reported in literature. Furthermore, the new algorithm was compared to another hybrid algorithm combining standard DPSO and MMA. Internal parameters set for the particle swarm optimizer are summarized in Table 1.

Table 1: Values of internal parameters set for IDPSO

Parameter	Value
Swarm size	50
Cognitive parameter	2.5
Social parameter	2.5
Minimum of inertia weight	0.01
Maximum of inertia weight	0.90
Maximum number of iterations	200

The values of internal parameters for DPSO are the same as these of IDPSO. In order to consider the stochastic nature of the optimization process, ten independent optimization runs are performed for DPSO-MMA and IDPSO-MMA methods and the best solutions are reported. The methods are coded in MATLAB and structures are analyzed using the direct stiffness method.

6.1. Test case 1: Fifteen-bar truss structure

The fifteen-bar truss shown in Fig. 2 is subjected to a concentrated load of 44.537 kN acting downward at node 8. The Young modulus of the material is 68.95 GPa while mass density is 2720 kg/m^3. Design variables are listed in Table 2: this test case included 15 sizing variables and 8 layout variables. The optimization constraints with allowable stress limits and side limits are presented in Table 3 which also includes the list of available cross-sectional area values.

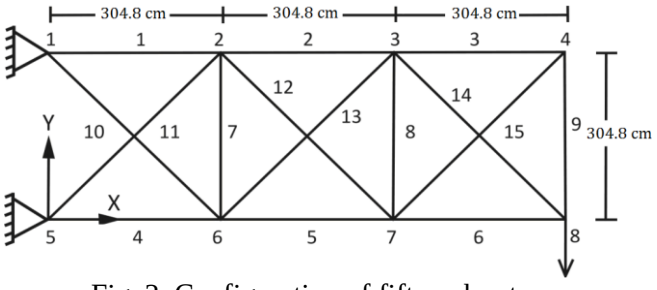

Fig. 2. Configuration of fifteen-bar truss

Table 2: Optimization variables for the fifteen-bar truss problem

A_i; i=1, 2, 3, … ,15	Sizing
$x_2 = x_6;\ x_3 = x_7; y_2; y_3; y_4; y_6; y_7; y_8$	Layout

Table 3: Optimization constraints for the fifteen-bar truss problem

$\lvert\sigma_i\rvert \le 172.4\ MPa\ ,\ i = 1,2,...,15$	Stress limit	Constraint data
$254cm \le x_2 \le 355.6cm$ $558.8cm \le x_3 \le 660.4cm$ $254cm \le y_2 \le 355.6cm$ $254cm \le y_3 \le 355.6cm$ $127cm \le y_4 \le 228.6cm$ $-50.8cm \le y_6 \le 50.8cm$ $-50.8cm \le y_7 \le 50.8cm$ $50.8cm \le y_8 \le 152.4cm$	Side limit on layout variables	
$A_i \in S =$ {0.716,0.910,1.123,1.419,1.742,1.852,2.239,2.839,3.477,6.155,6.974,7.574,8.600, 9.600, 11.381, 13.819, 17.400, 18.064, 20.200, 23.00,24.6, 31.0, 38.4, 42.4, 46.4, 55.0, 60.0, 70.0, 86.0, 92.193, 110.774, 123.742} cm^2	List of the available profiles	

Table 4 represents a comparison between the cross-sectional areas and node coordinates obtained by DPSO-MMA and different researchers together with the corresponding weight for the fifteen-bar truss. It can be seen that IDPSO-MMA performed better than DPSO-MMA and other optimization techniques and found lighter structures while satisfying all the constraints. The optimized layout of the structure is shown in Fig. 3.

Figure 4 compares the convergence curves recorded for IDPSO-MMA and DPSO-MMA. It can be seen that the proposed algorithm implementing the new discrete PSO particle update scheme was considerably faster.

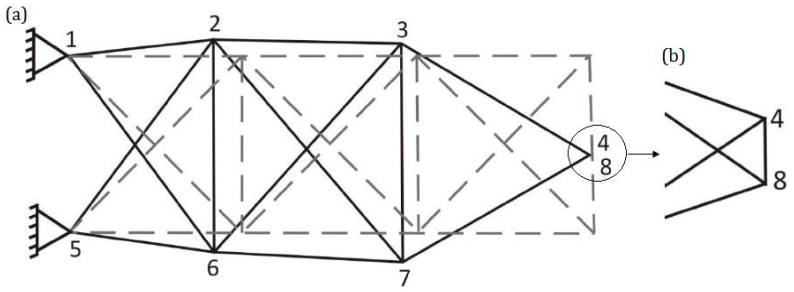

Fig. 3. (a) Optimal layout of the fifteen-bar truss, (b) The layout of the nodes 4 and 8.

Table 4: Optimization results for the fifteen-bar truss problem

IDPSO-MMA	DPSO-MMA	Rahami *et al.* [34]	Wenyan [33]	Hwang and He [32]	Wu and Chow [31]	Design variables

7.574	7.574	6.974	6.155	0.954	7.574	A_1	Sizing variable (cm^2)
3.477	3.477	3.477	6.974	1.081	6.155	A_2	
0.910	1.852	1.852	2.839	0.440	2.839	A_3	
6.155	6.155	6.155	7.574	1.174	8.600	A_4	
3.477	3.477	3.477	9.600	1.488	6.155	A_5	
1.852	1.852	0.910	1.742	0.270	1.123	A_6	
0.716	0.716	0.716	1.742	0.270	2.839	A_7	
0.716	0.716	0.716	2.239	0.347	2.839	A_8	
0.910	2.239	3.477	1.419	0.220	6.974	A_9	
2.239	2.839	2.839	2.839	0.440	8.600	A_{10}	
2.839	3.477	3.477	1.419	0.220	1.123	A_{11}	
1.852	1.852	1.742	2.839	0.440	1.123	A_{12}	
1.852	1.419	1.419	2.239	0.347	2.239	A_{13}	
1.852	1.852	0.910	1.750	0.270	2.239	A_{14}	
0.910	1.852	1.852	1.419	0.220	2.839	A_{15}	
254.00	254.00	258.007	339.375	300.600	312.900	x_2	Layout variable (cm)
583.330	660.400	578.894	596.270	572.031	588.251	x_3	
332.635	347.350	342.388	255.141	302.377	272.260	y_2	
324.489	285.846	325.680	266.035	266.918	302.704	y_3	
133.516	127.00	139.352	187.356	160.973	153.574	y_4	
-32.177	-23.071	-41.779	-25.570	-50.800	-42.489	y_6	
-50.800	1.879	-33.784	-3.401	-50.800	-39.535	y_7	
133.116	126.620	139.337	128.021	146.614	93.078	y_8	
34.33	37.04	34.77	36.21	47.43	54.67	Weight (kg)	

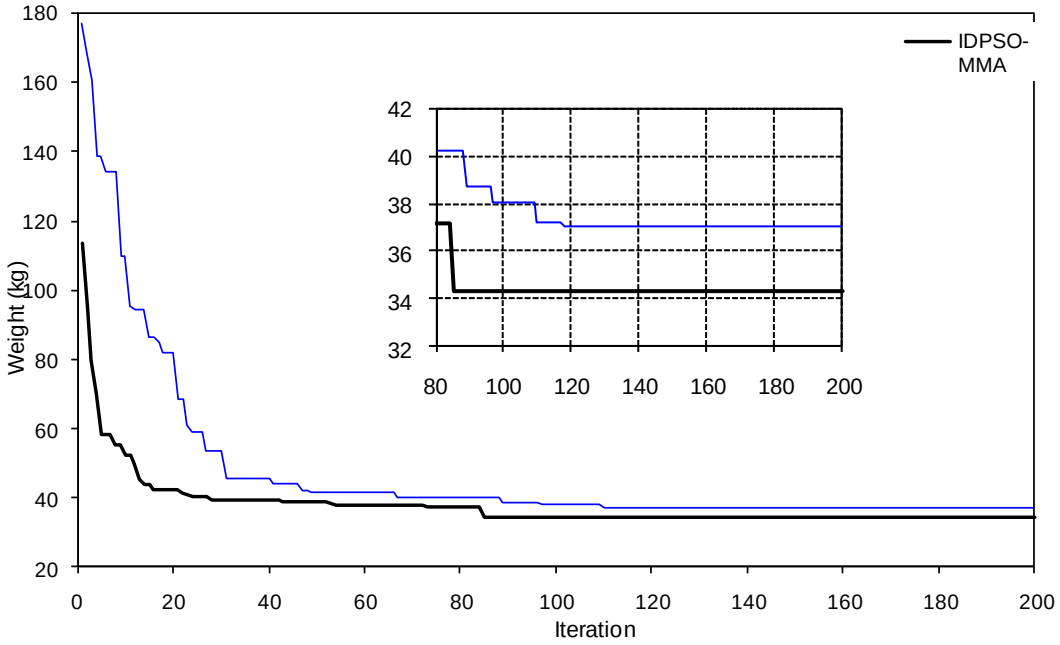

Fig. 4. Comparison of the convergence rates between the IDPSO-MMA and the DPSO-MMA

6.2. Test case 2: Michell arch

The Michell semi-circular arch shown in Fig. 5 must be designed for minimum weight. The arch is subject to a load of 200 kN acting downward at node 1. The Young modulus of the material is 210 GPa while mass density is 7800 kg/m3. This optimization problem has an analytical solution if allowable stresses in tension and compression are equal [35]:

$$W = \frac{12}{\sigma^+} LP\rho \tan(\frac{\pi}{12})$$

where L is the length of half span, which in this example is equal to 1 m; and σ^+ is the allowable stress in tension.

During the optimization process of the arch, nodes 3 and 7 are shifted in horizontal direction and nodes 4, 5 and 6 in vertical direction, respectively. The structural symmetry is maintained. Therefore, only the coordinate of two nodes are variable in the optimization process. The members of truss are classified into seven groups to represent seven variables of cross-section areas that shown in Table 5. The design variables are given in Table 6. The constraint data and cross-section areas of elements are listed in Table 7.

Table 8 represents a comparison between the cross-sectional areas and node coordinates obtained by DPSO-MMA together with the corresponding weight for the fifteen-bar truss. It can be seen that IDPSO-MMA performed better than DPSO-MMA and found lighter structures while satisfying all the constraints. The optimized layout of the structure is shown in Fig. 6.

Fig. 7 compares the convergence rate between IDSPO-MMA and DSPO-MMA for this example. It can be seen that the proposed algorithm implementing the new discrete PSO particle update scheme was considerably faster.

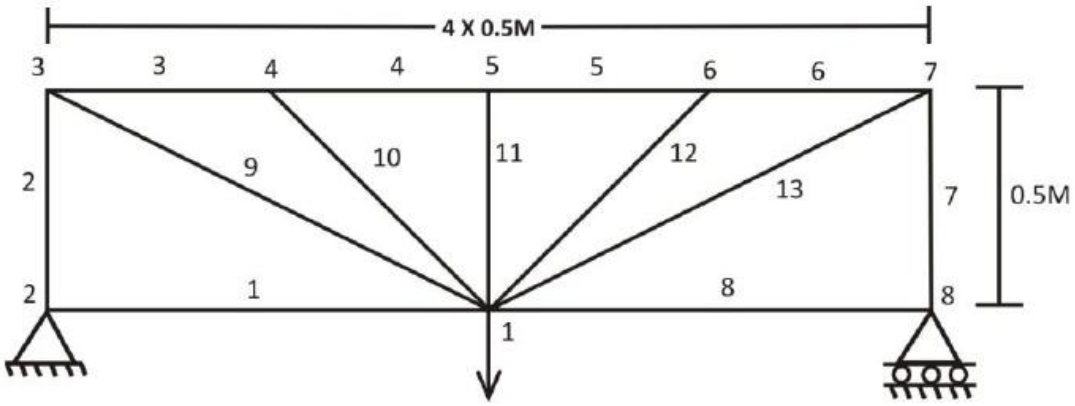

Fig. 5. Configuration of Michell arch

Table 5: Element group linking for the Michell arch problem

Members (end nodes)	Group
1 (1, 2), 8(1, 8)	A_1
2 (2,3) ,7 (7,8)	A_2
3 (3,4) ,6 (6,7)	A_3
4 (4,5) ,5 (5,6)	A_4
9 (1,3) ,13 (1,7)	A_5
10 (1,4) ,12 (1,6)	A_6
11 (1,5)	A_7

Table 6: Optimization variables for the Michell arch problem

A_i ; i=1, 2, … ,7	Sizing
$x_3 = -x_7$, $y_4 = y_6, y_5$	Layout

Table 7: Optimization constraints for the Michell arch problem

$\lvert\sigma_i\rvert \le 240\ MPa$; $i = 1,2,...,13$	Stress limit	Constraint data
$\lvert\Delta_{y,1}\rvert \le 3.8\,mm$	Displacement limit	
$0 \le x_3 \le 1(m)$ $0 \le y_4 \le 1(m)$ $0 \le y_5 \le 1(m)$	Side constraints on layout variables	
$A_i \in S = \{1.01, 1.02, 1.03, 1.04, ..., 5\}(cm^2)$		List of the available profiles

Table 8: Optimization results for the Michell arch problem

IDPSO-MMA	DPSO-MMA	Wang *et al.*[35]	Exact solution	Design variables	
1	1.16	1.132	1.116	A_1	Sizing variable (m^2)
4.28	4.51	4.318	4.314	A_2	
4.42	4.36	4.315	4.314	A_3	
4.62	4.95	4.311	4.314	A_4	
2.83	1.16	2.201	2.233	A_5	
2.02	2.63	2.262	2.233	A_6	
2.13	2.96	2.209	2.233	A_7	
0.926	0.871	1.000	1.000	y_5	Layout variable (m)
0.808	0.834	0.867	0.866	y_6	
0.882	0.123	0.864	0.866	x_3	
21.07	22.57	20.90	20.9	Weight (kg)	

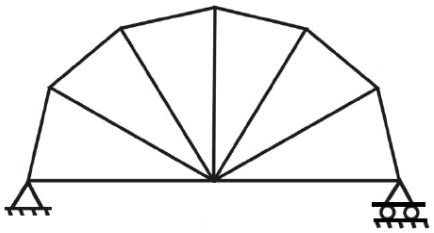

Fig. 6. Optimal layout of the Michell arch

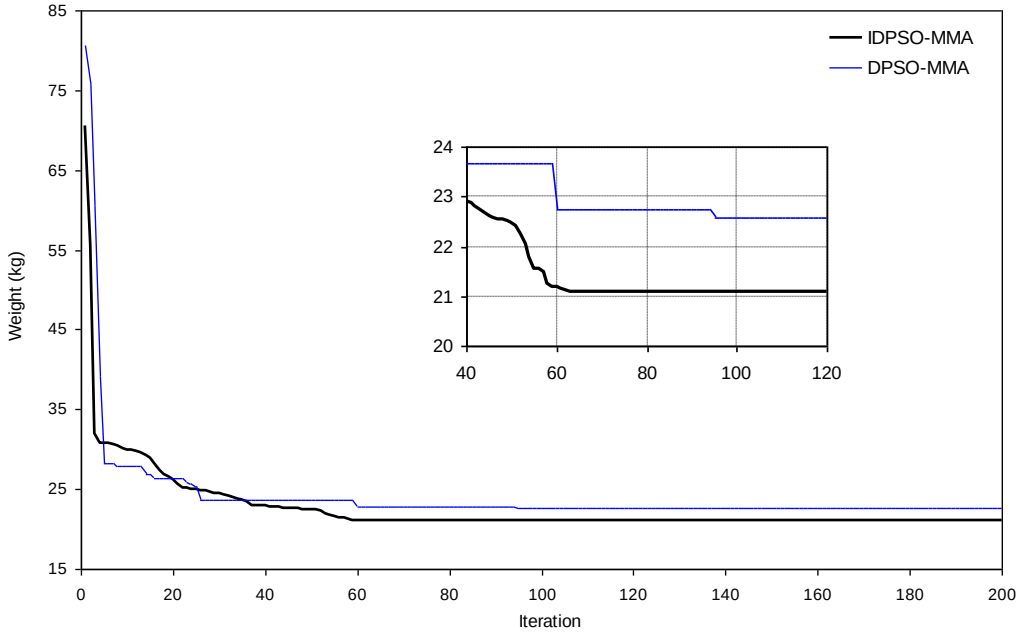

Fig. 7. Comparison of the convergence rates between the IDPSO-MMA and the DPSO-MMA

6.3. Test case 3: Twenty five-bar truss structure

A twenty five-bar truss is considered as shown in Fig. 8. The coordinates of the nodes are listed in Table 9. The members of truss are classified into eight groups to represent eight variables of cross-section areas. The member grouping is shown in Table 10. Table 11 lists the values and directions of the loads applied to the truss. The Young modulus of the material is 68.95 GPa while mass density is 2720 kg/m^3. The constraint data and the design variables are presented in Tables 12 and 13. The cross-section areas of elements are selected from Table 13.

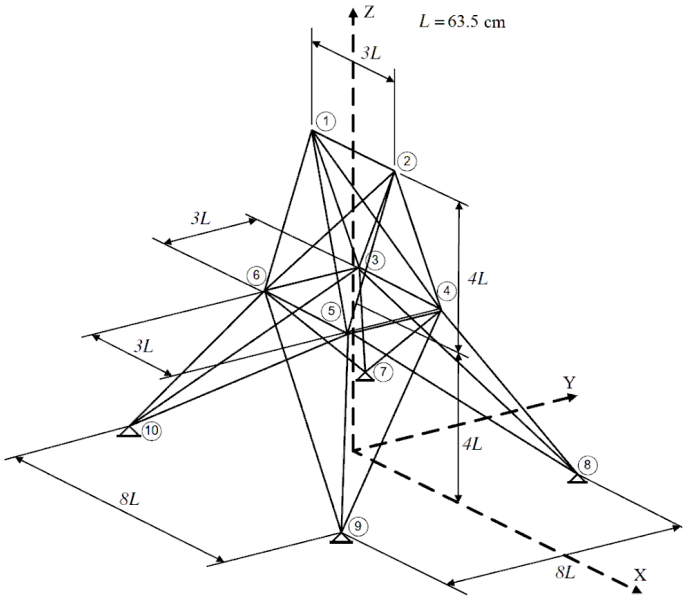

Fig. 8. Configuration of twenty five-bar space truss

The comparison of the results with those of the other references is provided in Table 14. It can be seen that IDPSO-MMA performed better than DPSO-MMA and other optimization techniques and found lighter structures while satisfying all the constraints.

Tbale 9: Nodal coordinates of the initial configuration considered for the spatial twenty five-bar truss problem

z (cm)	*y* (cm)	*x* (cm)	Node
508	0.0	-95.25	1
508	0.0	95.25	2
254	95.25	-95.25	3
254	95.25	95.25	4
254	-95.25	95.25	5
254	-95.25	-95.25	6
0.0	254	-254	7
0.0	254	254	8
0.0	-254	254	9
0.0	-254	-254	10

Table 10: Element group linking for the spatial twenty five-bar truss problem

Members (end nodes)	Group
1 (1, 2)	A_1
2 (1,4) ,3 (2,3) ,4 (1,5) ,5 (2,6)	A_2
6 (2,5) ,7 (2,4) ,8 (1,3) ,9 (1,6)	A_3
10 (3,6) , 11 (4,5)	A_4
12 (3,4) , 13 (5,6)	A_5
14 (3,10) , 15 (6,7) , 16 (4 ,9) , 17 (5,8)	A_6
18 (3,8) , 19 (4,7) , 20 (6,9) , 21 (5,10)	A_7
22 (3,7) , 23 (4,8) , 24 (5,9) , 25 (6,10)	A_8

Table 11: Loading acting on the twenty five-bar truss problem

F_z kN	F_y kN	F_x kN	Node
-44.537	-44.537	4.454	1
-44.537	-44.537	0.0	2
0.0	0.0	2.22	3
0.0	0.0	2.672	6

Table 12: Optimization variables of the spatial twenty five-bar truss problem

A_1; A_2; A_3; A_4; A_5; A_6; A_7; A_8	Sizing
$x_4 = x_5 = -x_3 = -x_6$; $x_8 = x_9 = -x_7 = -x_{10}$ $y_3 = y_4 = -y_5 = -y_6$; $y_7 = y_8 = -y_9 = -y_{10}$; $z_3 = z_4 = z_5 = z_6$	Layout

Table 13: Optimization constraints for the spatial twenty five-bar truss problem

$\lvert\sigma_i\rvert \le 275.8\ MPa$; $i = 1,2,\ldots,25$	Stress limit	Constraint data
$\lvert\Delta_i\rvert \le 0.89\text{cm}$; $i = 1,\ldots,6$	Displacement limit (all directions coordinate)	
$50.8cm \le x_4 \le 152.4cm$ $101.6cm \le y_4 \le 203.2cm$ $228.6cm \le z_4 \le 330.2cm$ $101.6cm \le x_8 \le 203.2cm$ $254cm \le y_8 \le 355.6cm$	Side constraints on layout variables	
$A_i \in S = \{0.645I\ (I = 1,\ldots,26), 18.064, 19.355, 20.645, 21.935\}\ cm^2$		List of the available profiles

Table 14: Optimization results for the spatial twenty five-bar truss problem

IDPSO-MMA	DPSO-MMA	Rahami *et al* [34]	Kaveh and Kalatjari [36]	Wu and Chow [31]	Design variables	
0.645	0.1	0.645	0.645	0.645	A_1	Sizing variable (cm^2)
0.645	0.1	0.645	0.645	1.29	A_2	
6.4516	1.1	7.097	7.097	7.097	A_3	
0.645	0.1	0.645	0.645	1.29	A_4	
0.645	0.2	0.645	0.645	1.935	A_5	
0.645	0.1	0.645	0.645	0.645	A_6	
0.645	0.1	1.29	0.645	1.29	A_7	
5.806	0.9	5.16	6.452	5.806	A_8	
95.008	37.205	83.9436	92.024	104.318	x_4	Layout variable (cm)
140.694	60.598	136.0584	148.742	135.814	y_4	
330.2 00	121.862	329.9694	293.599	316.484	z_4	
138.200	52.690	111.2078	118.008	129.032	x_8	
355.600	140	347.569	324.993	333.959	y_8	
53.35	56.03	54.53	56.29	61.83	Weight (kg)	

Optimal layout for the truss is shown in Fig. 9. Fig. 10 compares the convergence rate between IDSPO-MMA and DSPO-MMA for this example. It can be seen that the proposed algorithm implementing the new discrete PSO particle update scheme was considerably faster.

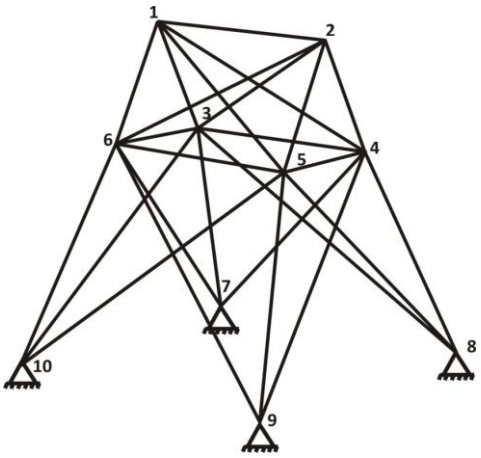

Fig. 9. Optimal layout of the twenty five-bar space truss

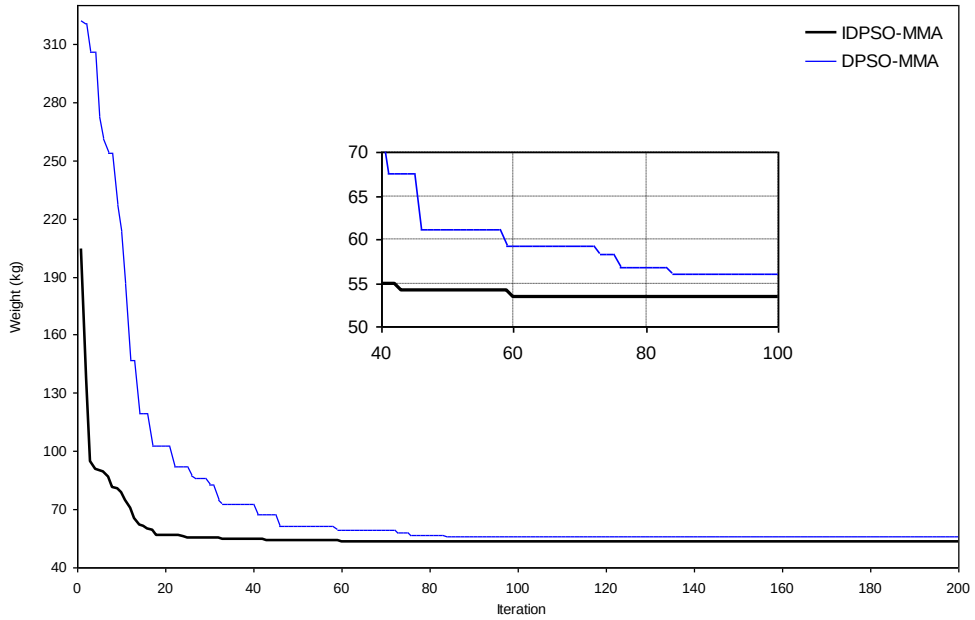

Fig. 10. Comparison of the convergence rates between the IDPSO-MMA and the DPSO-MMA

6.4. Test case 4: Thirty nine-bar tower

A triangular tower with the given configuration is shown in Fig. 11. The bottom and top nodes are fixed while all the intermediate node positions will be redesigned. The coordinates of three bottom nodes 1, 2, 3 and three top nodes 13, 14 and 15 are given in Table 15. The members of truss are classified into five groups to represent five variables of cross-section areas. The member grouping is shown in Table 16. Table 17 lists the values and directions of the loads applied to the tower. During the optimization process, the structural symmetry is remained, which implies that there are only 6 shape variables: y_4, z_4, y_7, z_7, y_{10}, z_{10}. The list of variables is presented in Table 18. The displacement of node 13 in the y-direction must not exceed a given limit of 4 mm. The constraint data and the design variables are presented in Tables 19. The cross-sectional areas of elements are selected from Table 19.

Table 15: Nodal coordinates of the fixed nodes (top and bottom sides) of the thirty nine-bar truss problem

Top nodes				Bottom nodes			
z (*m*)	*y* (*m*)	*x* (*m*)	Number	*z* (*m*)	*y* (*m*)	*x* (*m*)	Number
4	0.28	0	13	0	1	0	1
4	-0.14	$-\frac{0.42}{\sqrt{3}}$	14	0	-0.5	$-\frac{\sqrt{3}}{2}$	2

4	-0.14	$\frac{0.42}{\sqrt{3}}$	15	0	-0.5	$\frac{\sqrt{3}}{2}$	3

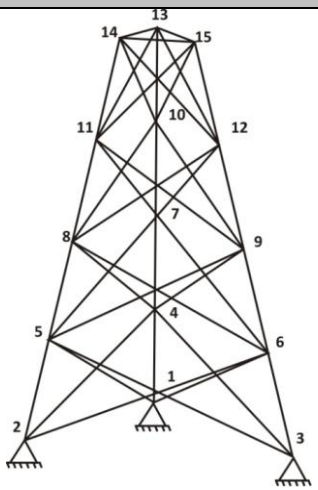

Fig. 11. Configuration of thirty nine-bar tower

Table 16: Element Group linking for the thirty nine-bar truss problem

End nodes	Group
(1,4), (2,5), (3,6)	A_1
(4,7), (5,8), (6,9)	A_2
(7,10),(8,11), (9,12)	A_3
(10,13),(11,14), (12,15)	A_4
Rest of the elements	A_5

Table 17: Loading acting on the thirty nine-bar truss problem

F_z kN	F_y kN	F_x kN	Node
0	10	0	13,14,15

Table 18: Optimization variables for the thirty nine-bar truss problem

$A_1;\ A_2;\ A_3;\ A_4;\ A_5$	Sizing
$y_4, z_4, y_7, z_7, y_{10}, z_{10}$	Layout

Table 19: Optimization constraint for the thirty nine-bar truss problem

$\lvert\sigma_i\rvert \le 240\ MPa \quad ;\ i = 1,2,...,39$	Stress limit	Constraint data
$\lvert\Delta_{y,13}\rvert \le 4\ mm$	Displacement limit	
$0(m) \le z_4 \le 2(m)$	Side constraints on	

$1(m) \le z_7 \le 3(m)$ $2(m) \le z_{10} \le 4(m)$ $0.28(m) \le y_4 \le 1(m)$ $0.28(m) \le y_7 \le 1(m)$ $0.28(m) \le y_{10} \le 1(m)$	layout variables	
$A_i \in S = \{0.1, 0.2, 0.3, 0.4, ..., 13\}(cm^2)$		List of the available profiles

The comparison of the results with those of Wang et al. [35] is provided in Table 20. Optimum weight obtained by IDPSO-MMA is about 16% and 3.5% lighter than those in Ref. [35] and DPSO-MMA while satisfying all the constraints.

Table 20: Optimization results for the thirty nine-bar truss problem

IDPSO-MMA	DPSO-MMA	Wang *et al.*[35]	Design variables	
13	10.12	11.01	A_1	Sizing variable (cm^2)
9.6	9.91	8.63	A_2	
8.2	8.56	6.69	A_3	
3	3.92	4.11	A_4	
13	3.44	4.37	A_5	
9.6	10.12	11.01	A_6	
0.786	0.6683	0.805	y_4	Layout variable (m)
1.330	1.9	1.186	z_4	
0.417	0.4732	0.654	y_7	
2.920	2.8734	2.204	z_7	
0.360	0.3002	0.466	y_{10}	
3.193	3.4415	3.092	z_{10}	
170.607	176.834	203.18	Weight (kg)	

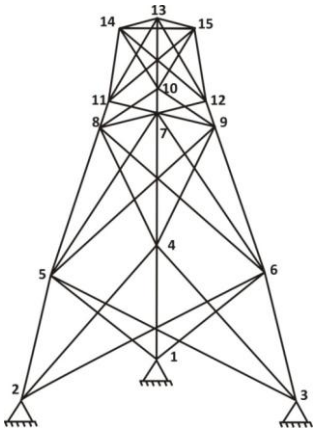

Fig. 12. Optimal layout of thirty nine-bar tower

Optimal layout for the truss is shown in Fig. 12 and Fig. 13 compares the convergence rate between IDSPO-MMA and DSPO-MMA for this example. It can be seen that the proposed IDPSO-MMA implementing the new discrete PSO particle update scheme was considerably faster.

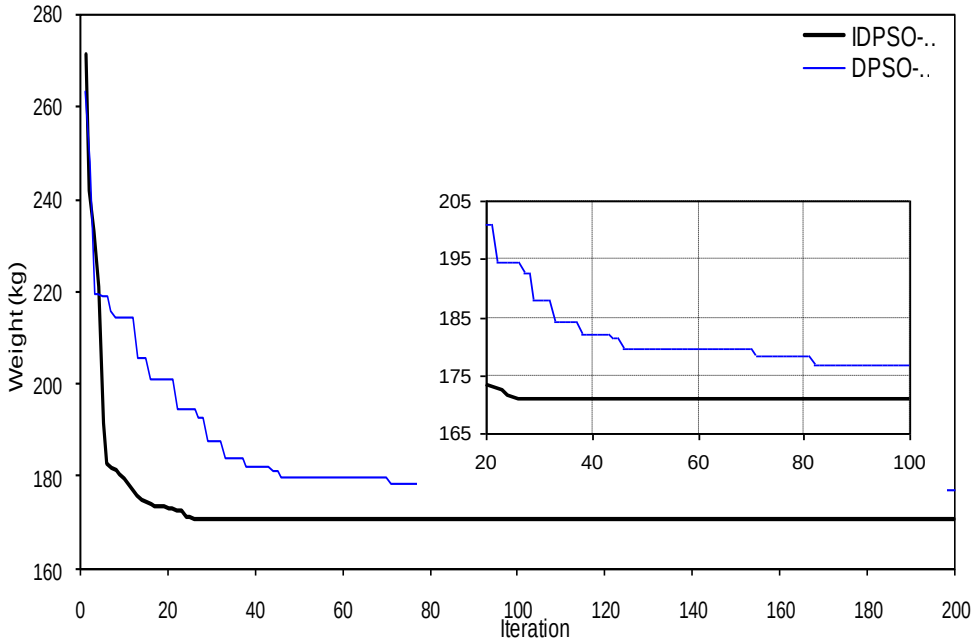

Fig. 13. Comparison of the convergence rates between the IDPSO-MMA and the DPSO-MMA

6.5. Test case 5: Eighteen-bar truss structure

In the example the aim is to optimize the cross-section size and layout of an eighteen-bar truss shown in Fig. 14 subject to stress and Euler buckling stress constraints. Optimization variables of the spatial structure are listed in Table 21. The constraint data and the design variables are presented in Tables 22. The Young modulus of the material is 68.95 GPa while

mass density is 2720 kg/m^3. The single loading condition is a set of vertical loads acting on the upper joints of the truss. The lower joints 3, 5, 7 and 9 are allowed to move in any direction in the x-y plane. The sections are taken from a profile list S of 80 sections starting with an area of 12.903 cm^2 increasing in the steps of 1.613 cm^2 to 140.322 cm^2. For layout variables, a precision of 2.54 cm is considered.

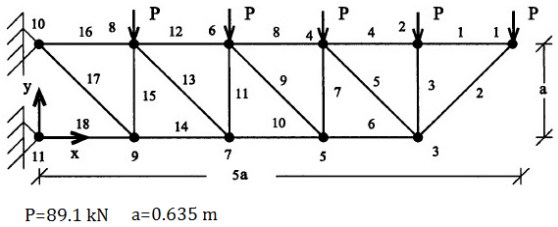

Fig. 14. Configuration of eighteen-bar truss

The results of the size/layout optimization using the IDSPO-MMA method are compared to those of other references in Table 23. It can be seen that IDPSO-MMA performed better than other optimization techniques and found lighter structures while the stress constraint is violated about 0.47%. Optimal layout for the truss is shown in Fig. 15.

Table 21: Optimization variables for the eighteen-bar truss problem

$A_1 = A_4 = A_8 = A_{12} = A_{16};\ A_2 = A_6 = A_{14} = A_{18};$ $A_3 = A_7 = A_{11} = A_{15};\ A_5 = A_9 = A_{13} = A_{17}$	Sizing
$x_3; y_3; x_5; y_5; x_7; y_7; x_9; y_9$	Layout

Table 22: Optimization constraint for the eighteen-bar truss problem

$\lvert\sigma_i\rvert \le 137.9\ MPa \quad ;\ i = 1,2,\ldots,18$	Stress limit	Constraint data
$\lvert\sigma_{c,i}\rvert \le \alpha E A_i / L_i^2\ ,\alpha = 4\ ;\ i = 1,2,\ldots,18$	Euler buckling stress limit	
$-5.715(m) \le y_3, y_5, y_7, y_9 \le 6.223(m)$ $19.685(m) \le x_3 \le 31.115(m)$ $13.335(m) \le x_5 \le 24.765(m)$ $6.985(m) \le x_7 \le 18.415(m)$ $0.635(m) \le x_9 \le 12.065(m)$ $0.28(m) \le y_{10} \le 1(m)$	Side constraints on layout variables	
$A_i \in S = \{12.903, 14.516, \ldots, 138.709, 140.322\}(cm^2)$		List of the available profiles

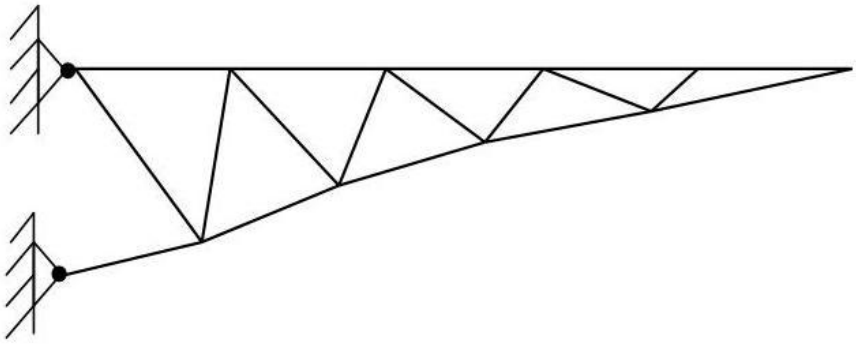

Fig. 15. Optimal layout of eighteen-bar truss

Table 23: Optimization results for the Eighteen-bar truss structure

IDPSO-MMA	Hasancebi and Erbatuer [37]	Kaveh and Kalatjari [36]	Rahami *et al.* [34]	Design variables	
80.645	80.645	79.032	82.258	A_1	Sizing variable (cm^2)
119.356	117.742	116.128	119.356	A_2	
30.645	35.484	33.871	30.645	A_3	
22.581	24.194	27.419	20.967	A_5	
23.432	23.698	23.190	23.303	x_2	Layout variable (m)
4.994	4.775	4.744	4.922	y_2	
16.740	16.713	16.510	16.619	x_5	
4.021	3.759	3.822	4.062	y_5	
10.841	10.718	10.637	10.782	x_7	
2.707	2.540	2.474	2.758	y_7	
5.316	5.207	5.201	5.295	x_9	
0.883	0.813	0.678	0.955	y_9	
2035.85	2038.82	2046.55	2058.43	Weight (kg)	

At the present work, Max stress = 138.55 MPa

7. CONCLUSIONS

In this paper, hybrid of an improved discrete particle swarm optimization (IDPSO) and method of moving asymptotes (MMA) is presented to find the optimal size and layout of the truss structures subject to displacement and stress constraints. The optimal size and layout of the truss structures simultaneously deal with mixed discrete and continuous variables. Due to this fact, the hybrid of IDPSO and MMA consisting of two stages for the size and geometry optimization is used. The IDPSO-MMA can increase the probability of finding the near global optimum. The main idea behind the proposed IDPSO-MMA is to combine the advantages of IDPSO and MMA methods in the two search spaces. In the first stage, the layout of truss structure is optimized with MMA by including only the nodal coordinates as design variables. In the second stage of the optimization process, IDPSO is utilized to update sizing variables and to find the global/local best design points with respect to which convex approximation is built in the subsequent layout optimization cycle. IDPSO is introduced to overcome the limitations of the standard discrete PSO. The truss structures are used in the related literature as benchmarks are considered to verify the efficiency of the IDPSO-MMA method. The results are shown that the proposed method can effectively accelerate the convergence rate and can more quickly reach the optimum design. The comparisons of the numerical results performed using the IDPSO-MMA method demonstrate the robustness of the proposed method.

REFERENCES

1. Kaveh A, Talatahari S. A general model for meta-heuristic algorithms using the concept of fields of forces, *Acta Mech*, 2011; 221: 99-118.
2. Kennedy J, Eberhart RC, Swarm intelligence, Morgan Kaufman Publishers, 2001.
3. He S, Prempain E, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems, *Eng Optim*, 2004; 36:585-605.
4. Perez RE, Behdinan K. Particle swarm approach for structural design optimization, *Comput Struct*, 2007;85:1579-88.
5. Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimizer (HPSO) for optimization of pin connected structures, *Comput Struct*, 2007; 85: 340-9.
6. Salajegheh E, Gholizadeh S, Khatibinia M. Optimal design of structures for earthquake loads by a hybrid RBF-BPSO method, *Earthq Eng Eng*, 2008; 7: 14-24.
7. Salajegheh E, Salajegheh J, Seyedpoor SM, Khatibinia M. Optimal design of geometrically nonlinear space trusses using adaptive neuro-fuzzy inference system, *Sci Iran*, 2009; 6: 403-14.
8. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Comput Struct*, 2009; 87: 267-83.
9. Gomes HM. Truss optimization with dynamic constraints using a particle swarm algorithm, *Expert Syst Appl*, 2011; 38: 957-68.

10. Jansen PW, Perez RE. Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach, *Comput Struct*, 2011; 89: 1352-66.
11. Fontan M, Ndiaye A, Breysse D, Bos F, Fernandez C. Soil-structure interaction: parameters identification using particle swarm optimization, *Comput Struct*, 2011; 89: 1602-14.
12. Fourie PC, Groenwold AA. The particle swarm optimization algorithm in size and shape optimization, *Struct Multidisc Optim*, 2002; 23: 259-67.
13. Svanberg K. The method of moving asymptotes: a new method for structural optimization, *Int J Numer Meth Eng*, 1987; 24: 359-73.
14. Svanberg K, A class of globally convergent optimization methods based on conservative convex separable approximations, *SIAM J Optimiz*, 2002; 12: 555-73.
15. Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima, *Struct Multidisc Optim*, 1998;16: 68-75.
16. Luo Z, Chen LP, Yang JZ. Compliant mechanism design using multi-objective topology optimization scheme of continuum structures, *Struct Multidisc Optim*, 2005;30: 142-54.
17. Luh GC, Lin CY. Optimal design of truss structures using ant algorithm, *Struct Multidisc Optim*, 2008;36: 365-79.
18. Shi Y, Eberhart R. A modified particle swarm optimizer, in: IEEE international Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ 1998; 69-73.
19. Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm, in: *IEEE Conference on Systems, Man, and Cybernetics*, 1997; 4104-9.
20. Rostami M, Nezamabadipour H. A new method for particle swarm optimization method, 14th Conference Electronic Engineering, Amirkabir University of Iran 2006.
21. Wang C, Zhang J, Yang J, Hu C, Liu J. A modified particle swarm optimization algorithm and its applications for solving travelling salesman problem, in: Proceedings of the International Conference on Neural Networks and Brain, ICNN 2005; 689-94.
22. Zeng X, Zhu Y, Nan L, Hu K, Niu B, He X. Solving weapon-target assignment problem using discrete particle swarm optimization, in: Proceedings of the IEEE World Congress on Intelligent Control and Automation 2006; pp. 3562-3565.
23. Pan Q, Tasgetiren F, Liang Y. A discrete particle swarm optimization algorithm for single machine total earliness and tardiness problem with a common due date, in: Proceedings of the IEEE Congress on Evolutionary Computation 2006; pp. 3281-3288.
24. Kong X, Sun J, Xu W. Permutation-based particle swarm algorithm for tasks scheduling in heterogeneous systems with communication delays, IJCR. 2008; 4: 61-70.
25. Xue Y, Yang Q, Feng J. Improved particle swarm optimization algorithm for optimum steelmaking charge plan based on the pseudo TSP solution, in: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics 2005; pp. 5452-5457.
26. Neethling M, Engelbrecht A. Determining RNA secondary structure using set based particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2006; pp. 1670-1677.
27. Moraglio A, Chio CD, Togelius J, Poli R. Geometric particle swarm optimization,

JAEA 2008; 1-14.
28. El-Abd M, Hassan H, Anis M, Kamel MS, Elmasry M. Discrete cooperative particle swarm optimization for FPGA placement, *Appl Soft Comput*, 2010;10:284-95.
29. Datta D, Figueira JR. A real-integer-discrete-coded particle swarm optimization for design problems, *Appl Soft Comput*, 2011; 11:3625-33.
30. Fleury C, Braibant V. Structural optimization: a new dual method using mixed variables, *Int J Numer Methods Eng*, 1986; 23: 409-28.
31. Wu SJ, Chow PT. Integrated discrete and configuration optimization of trusses using genetic algorithms, *Comput Struct*, 1995;55:695-702.
32. Hwang SF, He RS. A hybrid real-parameter genetic algorithm for function optimization, *Adv Eng Infor*, 2006;20:7-21.
33. Tang W, Tong L, Gu Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *Int J Numer Methods Eng*, 2005; 62:1737-62.
34. Rahami H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm, *Eng Struct*, 2008; 30:2360-9.
35. Wang D, Zhang WH, Jiang JS. Combined shape and sizing optimization of truss structures, *Comput Mech*, 2002; 29:307-12.
36. Kaveh A, Kalatjari V. Size-geometry optimization of trusses by the force method and genetic algorithm, *Z Angew Math Mech*, 2004; 84:347-57.
37. Hasancebi O, Erbatur F. Layout optimization of trusses using improved GA methodologies. *Acta Mech*, 2001; 146:87–107.