INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2012; 2(4):443-458

PARTICLE SWARM-GROUP SEARCH ALGORITHM AND ITS APPLICATION TO SPATIAL STRUCTURAL DESIGN WITH DISCRETE VARIABLES

S.K. Zeng and L.J. Li[*,†]
School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China

ABSTRACT

Based on introducing two optimization algorithms, group search optimization (GSO) algorithm and particle swarm optimization (PSO) algorithm, a new hybrid optimization algorithm which named particle swarm-group search optimization (PS-GSO) algorithm is presented and its application to optimal structural design is analyzed. The PS-GSO is used to investigate the spatial truss structures with discrete variables and is tested by truss optimization problems. The optimization results are compared with that of the HPSO and GSO algorithm. The results show that the PS-GSO is able to accelerate the convergence rate effectively and has the fastest convergence rate among these three algorithms. The research shows the proposed PS-GSO algorithm can be effectively applied to optimal design of spatial structures with discrete variables.

Received: 5 August 2012; Accepted: 10 September 2012

KEYWORDS: particle swarm optimization; group search optimization; hybrid optimization algorithm; discrete variables; spatial structures

1. INTRODUCTION

Bionic optimization algorithms, notably Evolutionary Algorithms (EAs) had been widely used to solve various scientific and engineering problems and have been extensively used in structural optimization problems recently. Among them, ant colony optimizer (ACO),

[*] Corresponding author: L.J. Li, School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China
[†]E-mail address: lilj@gdut.edu.cn

particle swarm optimizer and group search optimization inspired by Dorigo [1], Kenndy and Eberthart [2], Barnard and He [3, 4] respectively, are three typical representatives. The 'individual behavior' of group is mainly considered by ACO and PSO, in which evolution behavior is based on the evolutionary theory. These two algorithms belong to 'evolutionary strategies' areas in a way. ACO is good at solving discrete variable tasks and combination optimization problems but shows a low evolutionary velocity [5]. PSO suits for continuous and discrete variables optimization problems but is easy to entrap into local minima [6-9]. Also they are time consuming in optimizing complex structures. As we know, gregariousness is a common phenomenon in animality, 'information communication' and 'mutual cooperation' are two important aspects of group behavior. Group search optimizer (GSO) is such an optimization algorithm which is based upon this group specialty and also has been successfully used in structural optimal design with continuous variables [10-12]. However, as the bars, the areas of cross-sections of practical engineering structures are produced according to specifications, structural optimization with discrete variables is of value of practical application and shows more obvious significance.

To improve the efficiency of PSO for structural optimization, a new hybrid optimization algorithm named Particle Swarm-Group Search Optimization (PS-GSO) is presented in this paper, which is based on the combination of the particle swarm optimizer (PSO) and group search optimizer (GSO). Compared with the heuristic particle warm algorithm [7], the basic GSO [13] algorithm, the genetic algorithm [14], the hybrid genetic algorithm [15], and the harmony search heuristic algorithm [16, 17], this hybrid PS-GSO algorithm has preferable convergence rate and accuracy.

2. MATHEMATIC MODEL OF OPTIMIZATION DESIGN

The mathematic model of optimization design can be express as follows:

$$\min f(X)$$
$$\text{subject to } g_i(X) \le 0,\ i = 1, 2, ..., m$$

where $f(X)$ is the objective function, and $g_i(X)$ is the inequality constraints. The variables vector X represents a set of the design variables, and m is the number of constraints.

3. GROUP SEARCH OPTIMIZER (GSO) ALGORITHM

GSO [10] is inspired by the food searching behavior and group living theory of social animals, such as birds, fish and lions. The foraging strategies of these animals mainly include: producing, e.g., searching for food; and joining (scrounging), e.g., joining resources uncovered by others. GSO also employs 'rangers' which perform random walks to avoid entrapment in local minima. Therefore, in GSO, a group consists of three kinds of members:

producers, scroungers and rangers. At each iteration, a group member, located in the most promising area, conferring the best fitness value, is chosen as the producer. It locates in the most promising area and stays still. The other group members are selected as scroungers or rangers by random. Then, each scrounger makes a random walk towards the producer, and the rangers make a random walk in arbitrary direction. It is also assumed that the producer, scroungers and rangers do not differ in their relevant phenotypic characteristics. Therefore, they can switch among the three roles. The GSO behaves as follows [18]:

In an n-dimensional search space, the i_{th} member at the k_{th} searching bout (iteration) has a current position $X_i^k \in \mathbf{R}^n$, a head angle $\varphi_i^k = \left(\varphi_{i1}^k, ..., \varphi_{i(n-1)}^k\right) \in \mathbf{R}^{n-1}$ and a head direction $D_i^k\left(\varphi_i^k\right) = \left(d_{i1}^k, ..., d_{in}^k\right) \in \mathbf{R}^n$ which can be calculated from φ_i^k via a Polar to Cartesian coordinate transformation:

$$d_{i1}^k = \prod_{p=1}^{n-1} \cos(\varphi_{ip}^k) \tag{1}$$

$$d_{ij}^k = \sin(\varphi_{i(j-1)}^k) \cdot \prod_{p=i}^{n-1} \cos(\varphi_{ip}^k) \tag{2}$$

$$d_{in}^k = \sin(\varphi_{i(n-1)}^k) \tag{3}$$

In the GSO algorithm, the group consists of three individuals: producer, scroungers and rangers. At the k_{th} iteration the producer X_p behaves as follows:

(1) The producer will scan at zero degree and then scan laterally by randomly sampling three points in the scanning field: one point at zero degree:

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \tag{4}$$

One point in the left hand side hypercube:

$$X_l = X_p^k + r_1 l_{\max} D_p^k(\varphi^k - r_2 \theta_{\max} / 2) \tag{5}$$

And one point in the right hand side hypercube:

$$X_r = X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2 \theta_{\max} / 2) \tag{6}$$

where, $r_1 \in \mathbf{R}^n$ is a normally distributed random number with mean 0 and standard deviation 1 and $r_2 \in \mathbf{R}^{n-1}$ is a random sequence in the range (0, 1). $\theta_{\max} \in R^{n-1}$ is maximum pursuit

angle. The maximum pursuit distance $l_{\max}$ is calculated from:

$$l_{\max} = \| U_i - L_i \| = \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2}$$

where, L_i and U_i are the lower and upper bounds for the i_{th} dimension.

(2) The producer will then find the best point with the best resource (fitness value). If the best point has a better resource than its current position, then it will fly to this point. Or it will stay in its current position and turn its head to a new angle:

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{\max} \tag{7}$$

where, $\alpha_{\max}$ is the maximum turning angle.

(3) If the producer cannot find a better area after a iterations, it will turn its head back to zero degree:

$$\varphi^{k+a} = \varphi^k \tag{8}$$

where, a is a constant.

At the k_{th} iteration, the area copying behavior of the i_{th} scrounger can be modelled as a random walk towards the producer:

$$X_i^{k+1} = X_i^k + r_3 (X_p^k - X_i^k) \tag{9}$$

where, $r_3 \in \mathbf{R}^n$ is a uniform random sequence in the range (0, 1).

Besides the producer and the scroungers, a small number of rangers have been also introduced into our GSO algorithm. Random walks, which are thought to be the most efficient searching method for randomly distributed resources, are employed by rangers. If the i_{th} group member is selected as a ranger, at the k_{th} iteration, firstly, it generates a random head angle φ_i:

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{\max} \tag{10}$$

where, $\alpha_{\max}$ is the maximum turning angle; and secondly, it chooses a random distance:

$$l_i = a \cdot r_1 l_{\max} \tag{11}$$

and move to the new point:

$$X_i^{k+1} = X_i^k + l_i D_i^k (\varphi^{k+1}) \tag{12}$$

4. PARTICLE SWARM OPTIMIZER (PSO) ALGORITHM

The PSO has been inspired by the social behavior of animals such as fish schooling, insects swarming and birds flocking Kennedy and Eberhart [19]. It involves a number of particles, which 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 particles fly through the search space and their positions are updated based on the best positions of individual particles for each iteration. The objective function is evaluated for each particle and the fitness values of particles are obtained to determine which position in the search space is the best Bergh and Engelbrecht [20] Using neighborhood with the guaranteed convergence PSO.

In each iteration, the swarm is updated using the following equations:

$$V_i^{(k+1)} = \omega V_i^{(k)} + c_1 r_1 \left(P_i^{(k)} - X_i^{(k)} \right) + c_2 r_2 \left(P_g^{(k)} - X_i^{(k)} \right) \tag{13}$$

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k)} \tag{14}$$

where X_i and V_i represent the current position and the velocity of the i_{th} particle, respectively; P_i is the best previous position of the i_{th} particle (called P_{best}) and P_g is the best global position among all the particles in the swarm (called G_{best}); r_1 and r_2 are two uniform random sequences generated from $U(0,1)$; and ω is the inertia weight used to discount the previous velocity of the particle persevered [21].

5. PARTICLE SWARM HYBRID GROUP SEARCH OPTIMIZER (PS-GSO)

The PS-GSO algorithm continues the classification model of PSO and GSO. Random search, angle search and step search are used at the same time. The PS-GSO is an improved PSO or GSO. The improvements are as follows: 1) Use the step search mechanism of PSO when the algorithm does not move forward. 2) Adopt angle search mechanism of GSO and step search mechanism PSO at the same time. 3) Deal with the boundary problem with the harmony search (HS) algorithm [22].

The realization of this hybrid optimization algorithm is as follows:

In an n-dimensional search space, the i_{th} member at the k_{th} searching bout (iteration) has a current position $X_i^k \in \mathbf{R}^n$, the position of each member are initialized by random value before the start of the iterative search.

At the k_{th} searching iteration, calculate the fitness of each member, take the position of the best members as the producer and denoted it as X_p^k, and scan at zero degree first time, then search for a new position in front, left and right directions, respectively, by Eqs. (4-6)

to seek a better position. The remaining members are selected as the scroungers by certain probability randomly, which will participate in search with random steps and follow the producer by Eq. (9). The remaining group members are selected as the rangers, which will select search angle and search distance randomly by Eq. (10) and Eq. (11), and then move to the new position by Eq. (12).

At the end of k_{th} searching iteration, the algorithm will use PSO search mechanism if the producer does not update, meaning satisfy Eq. (15). Meanwhile, the producer will be considered as the best position P_g^k of PSO algorithm. If the algorithm enters the PSO algorithm for the first time, consider the scroungers and rangers as personal best position P_i^k. Next time the personal best position P_g^k will be selected by Eq. (17), which means the personal best position P^{k-1} at previous iteration will be replaced by the better position X^k at current iteration, then receive the latest personal best position P_i^k.

$$f\left(X_p^k\right) \geq f\left(X_p^{k-1}\right) \tag{15}$$

$$P_g^k = X_p^k \tag{16}$$

$$P_i^k = M(X_i^k, P_i^{k-1}) \tag{17}$$

In the PSO search mechanism, the selection of particle inherits the group of GSO. The position and the velocity of each particle will be calculated by Eqs. (13 and 14). At the end of k_{th} searching iteration, it will calculate the latest global best position P_g^k. If $f\left(P_g^k\right) < f\left(X_p^k\right)$, at next iteration, the producer will be replaced by the latest global best position P_g^k, as follow equation,

$$X_p^{k+1} = P_g^k \tag{18}$$

After the $(k+1)_{th}$ iterations, if $X_i^d < X_i^{d(L)}(LowerBound)$ or $X_i^d > X_i^{d(U)}(UpperBound)$, the scalar X_i^d is regenerated by selecting the corresponding component of the vector from p_{best} swarm randomly, which can be described as follows:

$$X_i^d = (P_i)_t^d, \quad t = \text{int}(rand(1,n)) \tag{19}$$

where $(P_i)_t^d$ denotes the d_{th} dimension scalar of p_{best} swarm of the t_{th} particle, t denotes a random integer number and n denotes the swarm sizes. The pseudo code for structural optimization by PS-GSO is listed in Table 1.

Table 1. Pseudo code for structural optimization by PS-GSO algorithm

Set k = 0;
Randomly initialize positions X_i , the velocity V_i and head angles φ_i of all members;
FOR (each member i in the group)
WHILE (the constraints are violated)
Randomly re-generate the current member X_i
END WHILE
END FOR
WHILE (the termination conditions are not met)
FOR (each members i in the group)
Calculate fitness: Calculate the fitness value of current member: $f(X_i)$
Choose producer: Find the producer X_p of the group;
The producer, scroungers and rangers update their position by the equations of GSO algorithm.
Calculate the fitness of each individual; update the producer and the search angle.
Check whether the algorithm move forward. If not, enter the PSO algorithm. If it does, skip the PSO algorithm.
PSO algorithm: Update the best position P^k_g and the personal best position P^k_i by Eq. (13) and Eq. (14); Update the position and velocity via equations of PSO algorithm.
Check whether each member of the group violates the variable boundary. If it does, calculated by Eq. (19).
Calculate the fitness value of current member, and then update the best position P^k_g (the producer X_p) and the personal best position P^k_i.
END FOR
Set k = k + 1;
END WHILE

6. PS-GSO ALGORITHM WITH DISCRETE VARIABLES

When using the PS-GSO algorithm in optimizing with discrete variables, because of the areas of cross-sections aren't continuum, before optimization there will be a mapping function which makes the discrete section areas correspond to the continuum integer from small to large. Suppose a discrete set A_n with n discrete variables, after arranging from small to large:

$$A_n = \left\{X_1, X_2, ..., X_j, ...X_n\right\}, 1 \le j \le n$$

Employ a mapping function to replace the discrete values of A_n with its serial numbers like:

$$h(j) = X_j$$

The serial numbers replace the discrete values to keep searching with continuum values and avoid efficiency of search declining. Suppose that there are p members the search space

with D dimension. And the position of the i_{th} member is denoted with vector x_i like:

$$x_i = \left(x_i^1, x_i^2, ...x_i^d, ...x_i^D\right), 1 \le d \le D, i = 1, 2, ..., p$$

in which, $x_i^d \in \{1, 2, ..., j, ..., n\}$ corresponds to the discrete variables $\{X_1, X_2, ..., X_j, ...X_n\}$ by mapping function $h(j)$. After that, all of the members search in the continuum space which is the integer space, as each component of vector x_i is integer. Accordingly, the producer, scroungers, rangers and particles become:

$$X_i = \text{floor}(X_i) \tag{20}$$

in which floor is a function rounding to negative infinity. After that, there is no change with the objective function and constraints, which just before being substitution into, the iterated integer are turn into areas of cross-sections correspondingly by the mapping function.

7. NUMERICAL EXAMPLES

In this section, two spatial truss structures commonly used in literature are selected as benchmark structures to test the PS-GSO. A double layer reticulated shell structure is chosen to see the efficiency of the hybrid PS-GSO algorithm. For all the three algorithms, the population size is set to at 50. The inertia weight ω decrease linearly from 0.9 to 0.4; and the value of acceleration constants c_1 and c_2 are set to be the same and equal to 0.8. For the GSO algorithm, 20% of the population is selected as rangers; the initial head angle φ_0 of each individual is set to be $\pi/4$. The constant a is given by round $\left(\sqrt{n+1}\right)$. The maximum pursuit angle $\theta_{\max}$ is π/a^2. The maximum turning angle $\alpha_{\max}$ is set to be $\pi/2a^2$. For the new hybrid algorithm (PS-GSO), the parameters are set to be the same as GSO and HPSO, except that the inertia weight ω is set to be 0.9; and the value of acceleration constants c_1 and c_2 are set to be the same and equal to 2. Different iteration numbers are used for different optimization structures, with smaller iteration number for smaller variable number structures and larger one for large variable number structures.

7.1. Example 1: The 25-bar spatial truss structure

The 25-bar spatial truss structure is shown in Figure 1. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10,000 ksi. The stress limits of the members are ±40,000 psi. All nodes in all directions are subjected to the displacement limits of ±0.35 in. Load case is listed in Table 2. There are 25 members, which are divided into 8 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}. The discrete variables are selected from the set D = {0.01, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0, 4.4, 4.8, 5.2, 5.6, 6.0} (in^2). The maximum number of iterations is 500.

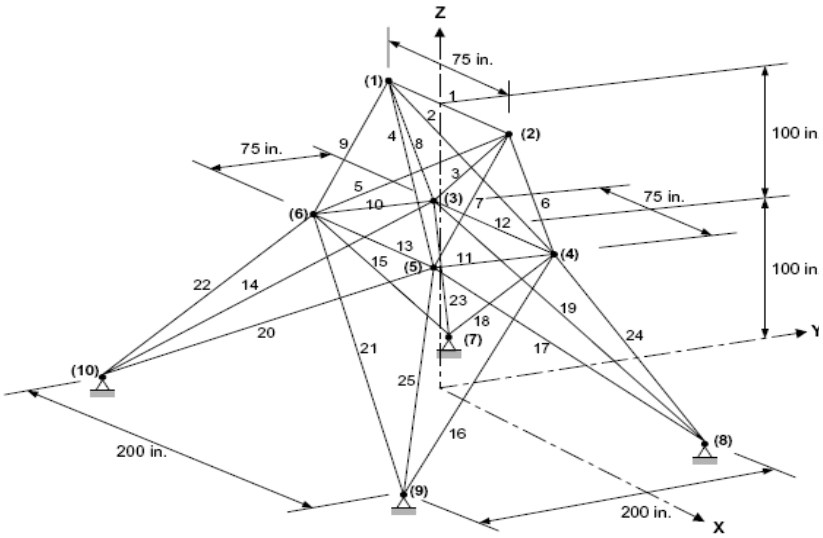

Figure 1. The 25-bar spatial truss structure

Table 2. Load cases for the 25-bar spatial truss structure

Node	Load case		
	P_X(kips)	P_Y(kips)	P_Z(kips)
1	0.0	20.0	−5.0
2	0.0	−20.0	−5.0
3	0.0	0.0	0.0
6	0.0	0.0	0.0

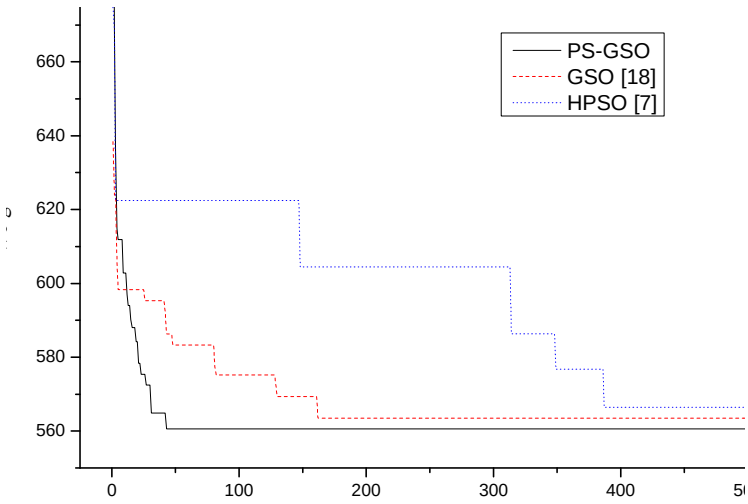

Figure 2. Comparison of convergence rates for the 25-bar truss structure

Table 3 shows the optimization solutions of 25-bar truss structure. Figure 2 gives the convergence rate of the three algorithms. The three algorithms can achieve the optimal solution after 500 iterations. But it can be seen from Figure 2 that it takes about 160 and 400 iterations for the GSO and HPSO algorithms to converge respectively, while the PS-GSO algorithm only needs less than 50 iterations.

Table 3. Design results for the 25-bar truss structure

Variables		Optimal cross-sectional areas (in^2)		
		PS-GSO	**GSO [18]**	**HPSO [7]**
1	A_1	0.01	0.40	0.01
2	A_2~A_5	2.00	2.00	2.00
3	A_6~A_9	3.60	3.60	3.60
4	A_{10}~A_{11}	0.01	0.01	0.01
5	A_{12}~A_{13}	0.01	0.01	0.40
6	A_{14}~A_{17}	0.80	0.80	0.80
7	A_{18}~A_{21}	1.60	1.60	1.60
8	A_{22}~A_{25}	2.40	2.40	2.40
Weight (lb)		560.59	563.52	566.44

7.2. Example 2: The 72-bar spatial truss structure

The 72-bar spatial truss structure is shown in Figure 3. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10^7psi. The stress limits of all the members are ±25ksi. All nodes in all directions are subjected to the displacement limits of ± 0.25 in. The load cases are listed in Table 4. There are 72 bars, which are divided into 16 groups, as follows: (1) A_1~A_4, (2) A_5~A_{12}, (3) A_{13}~A_{16}, (4) A_{17}~A_{18}, (5) A_{19}~A_{22}, (6) A_{23}~A_{30} (7) A_{31}~A_{34}, (8) A_{35}~A_{36}, (9) A_{37}~A_{40}, (10) A_{41}~A_{48}, (11) A_{49}~A_{52}, (12) A_{53}~A_{54}, (13) A_{55}~A_{58}, (14) A_{59}~A_{66}, (15) A_{67}~A_{70} ,(16) A_{71}~A_{72}. The optional discrete variables are: D={0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2}. The maximum number of iterations is 1000.

Table 4. Load cases for the 72-bar spatial truss structure

Node	Load cases		
	P_X(kips)	**P_Y(kips)**	**P_Z(kips)**
17	5.0	5.0	-5.0
18	0.0	0.0	0.0
19	0.0	0.0	0.0
20	0.0	0.0	0.0

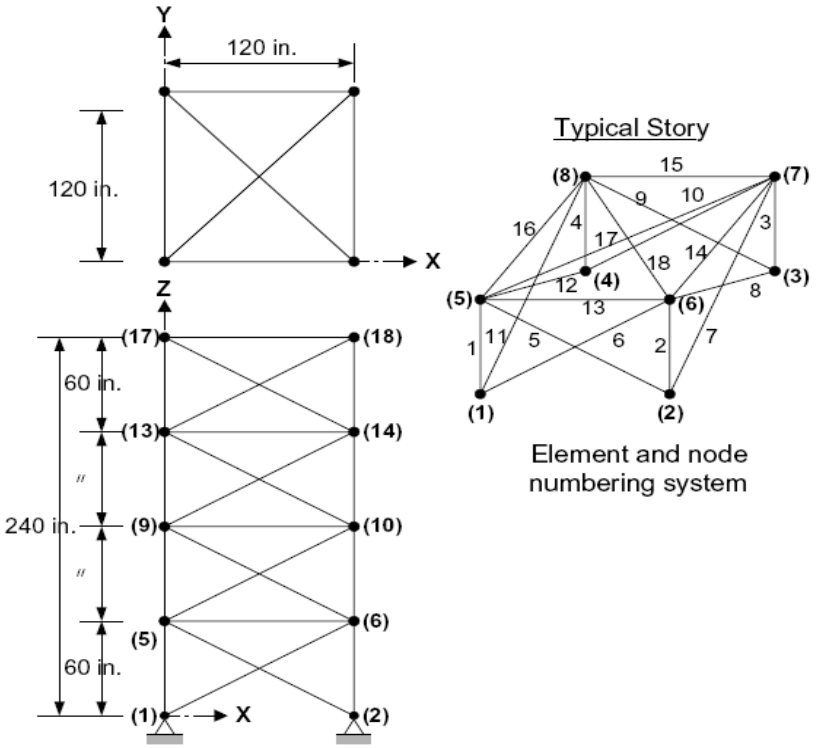

Figure 3. The 72-bar spatial truss structure

It can be seen from Table 5 that the PS-GSO algorithms achieve the best solution after 1000 iterations, its result is statistically better than that of GSO and HPSO. Moreover, it can be seen from Figure 4 that the PS-GSO algorithm provides best convergence rates than that of the GSO and HPSO.

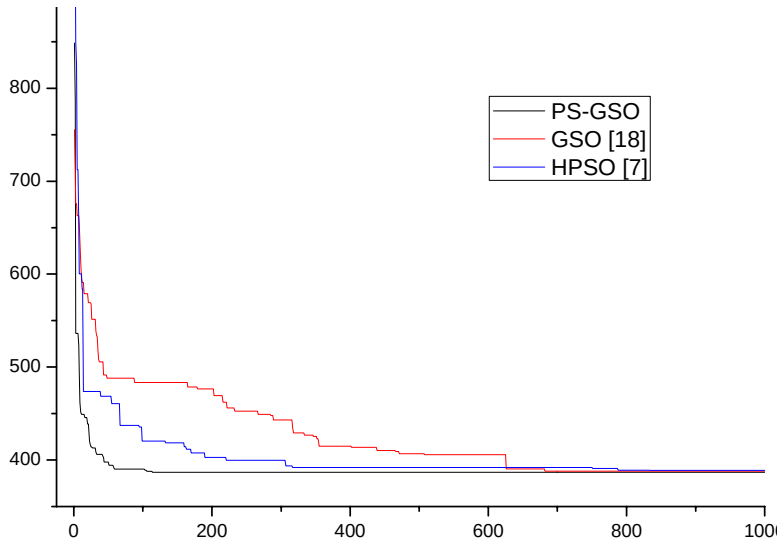

Figure 4. Comparison of convergence rates for the 72-bar truss structure

Table 5. Design results for the 72-bar truss structure

Variables		Optimal cross-sectional areas (in^2)		
		PS-GSO	**GSO [18]**	**HPSO [7]**
1	A_1~A_4	2.1	1.7	2.1
2	A_5~A_{12}	0.5	0.6	0.6
3	A_{13}~A_{16}	0.1	0.1	0.1
4	A_{17}~A_{18}	0.1	0.1	0.1
5	A_{19}~A_{22}	1.5	1.3	1.4
6	A_{23}~A_{30}	0.5	0.5	0.5
7	A_{31}~A_{34}	0.1	0.1	0.1
8	A_{35}~A_{36}	0.1	0.1	0.1
9	A_{37}~A_{40}	0.5	0.6	0.5
10	A_{41}~A_{48}	0.5	0.5	0.5
11	A_{49}~A_{52}	0.1	0.1	0.1
12	A_{53}~A_{54}	0.1	0.1	0.1
13	A_{55}~A_{58}	0.2	0.2	0.2
14	A_{59}~A_{66}	0.5	0.6	0.5
15	A_{67}~A_{70}	0.5	0.4	0.3
16	A_{71}~A_{72}	0.6	0.5	0.7
Weight (lb)		386.81	388.08	388.94

7.2. Example 3: A double layer spherical shell

There are 6761 nodes and 1834 bars in the double layer shell structure and it shows in Figure 5. All bars are divided into three groups which are upper chords, lower chords and web members. The span of the shell is 83.6 meters long and the rise is 14 meters high. The layer highness between the upper and the lower chord is 1.5 meters. The shell structure is made of steel. And the elasticity modulus is 210 GPa, the density 7850kg/m^3. The periphery nodes of the lower chord layer are joined with hinges.

The target is the weight of the structure. The areas of cross-sections of bar members are selected as the design variables. Furthermore, all of the members are thin-walled circular bar which are selected from national standard GB8162-87 of China with 379 kinds of cross-sections. The structure bears a single load case. Each node of upper chord layer bears a

vertical downward load of 50kN. All the nodes in all directions are subjected to the displacement limits of ±0.209m which is the span of 1/400. Every bar is subjected to the stress limits of ±215MPa. Meanwhile, the restriction of stability of bar members is considered in accordance with criterion. The maximum allowable slenderness ratio of pressure and pull bar is 180 and 300 respectively. The maximum number of iteration is limited to 200.

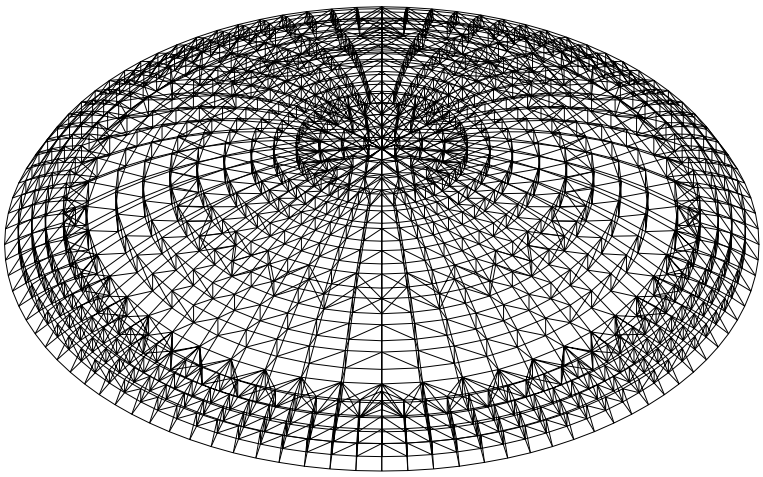

Figure 5. A double layer spherical shell

Table 6. Comparison of optimal solutions for a double layer spherical shell

Optimal algorithms	Optimal areas of cross-sections			Optimal weight (kg)
	Upper chords	Lower chords	Web members	
PS-GSO	φ89×4	φ73×4.5	Φ54×3.5	116898.01
GSO [18]	φ108×4	φ95×3.5	φ95×4	163954.70
HPSO [7]	φ108×4	φ83×3.5	φ89×3.5	148811.71

Table 6 shows the optimization solutions of double layer spherical shell. Figure 6 gives the convergence rate of the three algorithms. The three algorithms can achieve the optimal solution after 200 iterations. But it can be seen from Figure 6 that it takes about 130 and 100 iterations for the GSO and HPSO algorithms to converge respectively, while the PS-GSO algorithm only needs less than 15 iterations. It can be seen from Table 6 that the PS-GSO algorithms achieve the lightest structure, its result is statistically better than that of GSO and HPSO. Moreover, it can be seen from Figure 6 that the PS-GSO algorithm provides best convergence rates than those of the GSO and HPSO.

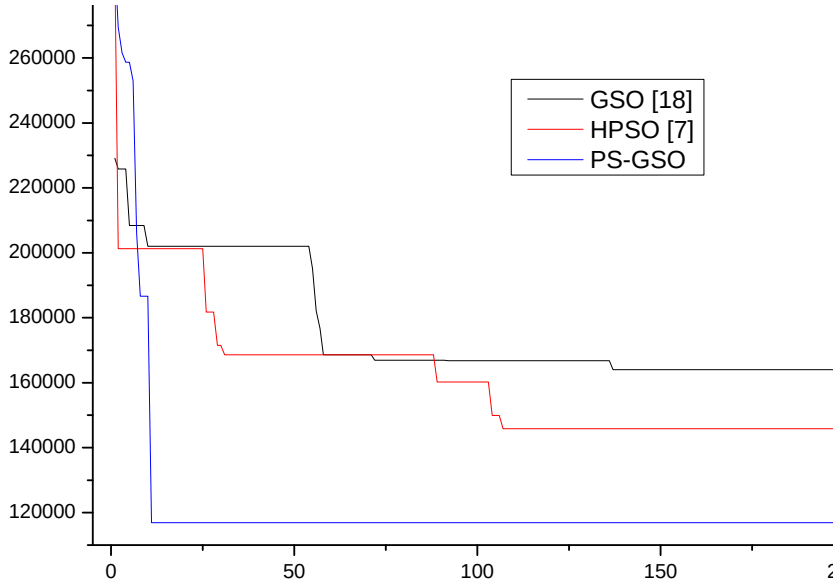

Figure 6. Comparison of convergence rates for the double layer spherical shell

8. CONCLUSIONS

In this paper, a new hybrid optimization algorithm named particle swarm-group search optimization (PS-GSO) handling discrete variables is presented based on the GSO, PSO and harmony search method. The PS-GSO algorithm for discrete variables has all the advantages that belong to the PS-GSO algorithm for continuous variables, and has faster convergence rate than the GSO and HPSO algorithms for discrete variables. It is the most efficient one of these three algorithms. The PS-GSO algorithm is tested by three spatial truss structures optimization problems. All the results show that the PS-GSO algorithm converges much quickly than the HPSO and the GSO. It is proved that the PS-GSO can be used for optimum problems with discrete variables efficiently. Compared with GSO and HPSO algorithm, the PS-GSO algorithm achieves convergent results during the early iterations, which illustrates the rapidly converging to the optimal solution by using the PS-GSO algorithm.

Acknowledgements: The Corresponding author, Prof. Lijuan Li would like to thank the financially supporting by the National Natural Science Foundation of China (project numbers: 10772052, 51178121) and the Natural Science Foundation of Guangdong Province. (Project numbers: 8151009001000042, 9151009001000059).

REFERENCES

1. Dorigo M, Di Caro G, Gambardella L. Ant algorithms for discrete optimization, *Artificial Life*, 1999; **5**: 137–72.

2. Kenndy J, Eberhart RC. Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, 1995, pp. 1942-1948.
3. Barnard CJ, Sibly RM. Producers and scroungers: a general model and its application to captive flocks of house sparrows, *Anim Behav*, 1981; **29**: 543–50.
4. He S, Wu QH, Saunder JR. A novel group search optimizer inspired by animal behaviour ecology, *Proceeding of the 2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, 2006, pp, 4415-4421.
5. Kaveh A, Shojaee S. Optimal design of scissor-link foldable structures using ant colony optimization algorithm, *Comput-Aided Civ Inf,* 2007; **22**: 56–64.
6. Li LJ, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures, *Comput Struct,* 2007; **85**: 340–49.
7. Li LJ, Huang ZB, Liu F: A heuristic particle swarm optimization method for truss structures with discrete variables, *Comput Struct,* 2009; **87**: 435–43.
8. He S. Prempain, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems, *Eng Optimiz,* 2004; **36**: 585–605.
9. Perez RE, Behdinan K. Particle swarm approach for structural design optimization, *Comput Struct,* 2007; **85**: 1579–88.
10. He S, Wu QH, Saunders JR. Group search optimizer: an optimization algorithm inspired by animal searching behavior, *IEEE Trans Evol Comput,* 2009; **13**: 973–9.
11. Liu F, Xu XT, Li LJ. The group search optimizer and its application on truss Structure design, *The 4th International Conference on Natural Computation*, Jinan, China, 2008, pp. 688–692.
12. Liu F, Xu XT, Li LJ. Group search optimization for design of space trusses, *The 10th International Symposium on Structural Engineering for Young Experts*, Changsha, China, 2008, pp. 854–858.
13. Li LJ, Xu XT, Liu F. The group search optimizer and its application to truss structure design, *Adv Struct Eng,* 2010; **13**: 43–51.
14. Wu SJ, Chow PT. Intergrated discrete and configuration optimization of trusses using genetic algorithm, *Comput Struct,* 1995; **55**: 695–702
15. Ali Falakian, Seyed Yaser Mousavi, Hybrid genetic algorithm for structural design optimization, *J Basic Appl Sci Res,* 2011; **2**: 256–61.
16. Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search heuristic algorithm for discrete structural optimization, *Eng Optimiz,* 2005; **37**: 663–84.
17. Geem, Zong Woo. (Ed.) Harmony search algorithms for structural design optimization. Berlin, Springer Verlag, 2009.
18. Li Lijuan, Liu Feng. Group search optimization for applications in structural design, Springer Berlin / Heidelberg, 2011.
19. Kennedy J, Eberhart RC. *Swarm intelligence*, Morgan Kaufman Publishers, 2001.
20. Van den Bergh, Engelbrecht A. Using neighborhood with the guaranteed convergence PSO, *Proceeding in 2003 IEEE Swarm Intelligence Symposium*, USA, 2003, pp. 235–242.
21. Shi Y, Eberhart RC. A modified particle swarm optimizer. *Proceeding of the IEEE International Conference on Evolutionary Computation,* Piscataway, NJ, IEEE Press,

1998, pp. 69–73.
22. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm, *Comput Struct,* 2004; **82**: 781–98.