Available online at www.sciencedirect.com

ScienceDirect

Computers and Structures 85 (2007) 340–349

Computers & Structures

www.elsevier.com/locate/compstruc

A heuristic particle swarm optimizer for optimization of pin connected structures

L.J. Li [a,*], Z.B. Huang [a], F. Liu [a], Q.H. Wu [b]

[a] Department of Civil Engineering, Faculty of Construction Engineering, Guangdong University of Technology, University City, Guangzhou 510006, PR China

[b] Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool L69 3GJ, United Kingdom

Received 24 July 2006; accepted 23 November 2006
Available online 18 January 2007

Abstract

This paper presents a heuristic particle swarm optimizer (HPSO) for optimum design of pin connected structures. The algorithm is based on the particle swarm optimizer with passive congregation (PSOPC) and a harmony search (HS) scheme. The HPSO algorithm handles the problem-specific constraints using a 'fly-back mechanism' method, and the harmony search scheme deals with the variable constraints. The method is verified and compared with the PSO and PSOPC algorithms used for the designs of five planar and spatial truss structures. The results show that the HPSO algorithm can effectively accelerate the convergence rate and can more quickly reach the optimum design than the two other algorithms.

Keywords: Particle swarm optimization; Passive congregation; Harmony search; Continuous variables; Pin connected structures; Size optimization

1. Introduction

In the last 30 years, a great attention has been paid to structural optimization, since material consumption is one of the most important factors influencing building construction. Designers prefer to reduce the volume or weight of structures through optimization.

Many traditional mathematical optimization algorithms have been used in structural optimization problems. Recently, evolutionary algorithms (EAs) such as genetic algorithms (GAs), evolutionary programming (EP) and evolution strategies (ES) have become more attractive because they do not require conventional mathematical assumptions and thus possess better global search abilities than the conventional optimization algorithms [1]. For example, GAs have been applied for structural optimization problems [2–4]. A new evolutionary algorithm called particle swarm optimizer (PSO) was developed [5]. The PSO has fewer parameters and is easier to implement than the GAs. The PSO also shows a faster convergence rate than the other EAs for solving some optimization problems [6].

It is known that the PSO may perform better than the EAs in the early iterations, but it does not appear competitive when the number of iterations increases [7]. Recently, many investigations have been undertaken to improve the performance of the standard PSO (SPSO). For example, He et al. improved the standard PSO with passive congregation (PSOPC), which can improve the convergence rate and accuracy of the SPSO efficiently [8].

Most structural optimization problems include the problem-specific constraints, which are difficult to solve using the traditional mathematical optimization algorithms, and the GAs [9] Penalty functions have been commonly used to deal with constraints. However, the major disadvantage of using the penalty functions is that some tuning parameters are added in the algorithm and the penalty coefficients have to be tuned in order to balance the objective and penalty functions. If appropriate penalty coefficients cannot be provided, difficulties will be

* Corresponding author. Tel.: +86 20 39322513; fax: +86 20 39322511.
E-mail address: lilj@scnu.edu.cn (L.J. Li).

0045-7949/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2006.11.020

encountered in the solution of the optimization problems [10,11]. To avoid such difficulties, a new method, called 'fly-back mechanism', was developed [8]. Compared with the other constraint handling techniques, this method is relatively simple and easy to implement.

For most structural optimization problems, the computation time spent to reach for a satisfactory solution is one of the major concerns, in particular for large and complex structures. The PSO may be applied to solve structural optimization problems, if its convergence rate can be effectively improved [12]. This paper presents a heuristic particle swarm optimizer (HPSO), which is based on the PSOPC and the harmony search (HS) scheme. It handles the constraints by using a 'fly-back mechanism' method. It is able to accelerate the convergence rate and improve the accuracy effectively in comparison with the PSOPC and PSO, respectively.

2. The presentation of structural optimization problems

A structural design optimization problem can be formulated as the nonlinear programming problem (NLP). Areas of cross-sections of bar members are normally selected as the design variables in the optimization of a truss type of structure. The objective function is the weight of a truss, which is subjected to the stress and the displacement constraints. This can be expressed as follows:

$$\min f(X)$$

subject to:

$$g_i(X) \geqslant 0, \quad i = 1, 2, \ldots, m$$

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 members). It can be denoted as:

$$\begin{aligned} &X = [x_1, x_2, \ldots, x_n]^T \\ &x_i^l \leqslant x_i \leqslant x_i^u, \quad i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

where x_i^l and x_i^u are the lower and the upper bounds of the ith variable, respectively.

3. The particle swarm optimizer (PSO)

The PSO has been inspired by the social behavior of animals such as fish schooling, insects swarming and birds flocking [6]. 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 in 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 [13].

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

$$\begin{aligned} &V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \\ &X_i^{k+1} = X_i^k + V_i^{k+1} \end{aligned}$$

where X_i and V_i represent the current position and the velocity of the ith particle, respectively; P_i is the best previous position of the ith particle (called *pbest*) and P_{g} is the best global position among all the particles in the swarm (called *gbest*); 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 [14].

4. The particle swarm optimizer with passive congregation (PSOPC)

The PSO can be enhanced by one type of social behaviors such as bird flocking, fish schooling and insects swarming, which are considered as congregation. This behavior is concerned with grouping by social forces, which is the source of attraction. The congregation involves active congregation and passive congregation. The latter is an attraction of an individual to other group members but not a display of social behavior [8]. Fish schooling is one of the representative types of passive congregation and the PSO is inspired by it. Adding the passive congregation model to the SPSO may increase its performance. He et al. proposed a hybrid PSO with passive congregation (PSOPC) as follows [8]:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + c_3 r_3 (R_i^k - X_i^k) \tag{2}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{3}$$

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

Several benchmark functions have been tested in Ref. [8]. The results show that the PSOPC has a better convergence rate and a higher accuracy than the PSO.

5. Constraint handling method: fly-back mechanism

The PSO has already been applied to optimize constrained problems [8,9,12]. The most common method to handle the constraints is to use penalty functions. However, some experimental results indicate that such a technique will reduce the efficiency of the PSO, because it resets the infeasible particles to their previous best positions *pbest*, which sometimes prevents the search from reaching a global minimum [9]. A new technique handling the constraints, which is called 'fly-back mechanism', has been introduced by He et al. [9]. For most of the optimization problems containing constraints, the global minimum locates on or close to the boundary of a feasible design space. The particles are initialized in the feasible region. When the optimization process starts, the particles fly in the feasible space to search the solution. If any one of the particles flies into the infeasible region, it will be forced to fly back to the previous position to guarantee a feasible solution. The particle which flies back to the previous position may be closer to the boundary at the next iteration. This makes the particles to fly to the global minimum in a great probability. Therefore, such a 'fly-back mechanism' technique is suitable for handling the optimization problem containing the constraints. Some experimental results have shown that it can find a better solution with fewer iterations than the other techniques [9].

6. A heuristic particle swarm optimization (HPSO)

The heuristic particle swarm optimizer (HPSO) is based on the PSOPC and a harmony search (HS) scheme, and uses a 'fly-back mechanism' method to handle the constraints.

When a particle flies in the searching space, it may fly into infeasible regions. In this case, there are two possibilities. It may violate either the problem-specific constraints or the limits of the variables, as illustrated in Fig. 1. Because the 'fly-back mechanism' technique is used to handle the problem-specific constraints, the particle will be forced to fly back to its previous position no matter if it violates the problem-specific constraints or the variable boundaries. If it flies out of the variable boundaries, the solution cannot be used even if the problem-specific constraints are satisfied. In our experiments, particles violate the variables' boundary frequently for some simple structural optimization problems. If the structure becomes complicated, the number of occurrences of violating tends to rise. In other words, a large amount of particles' flying behaviors are wasted, due to searching outside the variables' boundary. Although minimizing the maximum of the velocity can make fewer particles violate the variable boundaries, it may also prevent the particles to cross the problem-specific constraints. Therefore, we hope that all of the particles fly inside the variable boundaries and then to check whether they violate the problem-specific constraints and get better solutions or not. The particles, which fly outside the variables' boundary, have to be regenerated in an alternative way. Here, we introduce a new method to handle these particles. It is derived from one of the ideas in a new meta-heuristic algorithm called harmony search (HS) algorithm [15].

Harmony search (HS) algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation [15]. The engineers seek for a global solution as determined by an objective function, just like the musicians seek to find musically pleasing harmony as determined by an aesthetic [16]. The harmony search (HS) algorithm includes a number of optimization operators, such as the harmony memory (HM), the harmony memory size (HMS), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR). In this paper, the harmony memory (HM) concept has been used in the PSO algorithm to avoid searching trapped in local solutions. The other operators have not been employed. How the HS algorithm generates a new vector from its harmony memory and how it is used to improve the PSO algorithm will be discussed as follows.

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 it stores. 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.

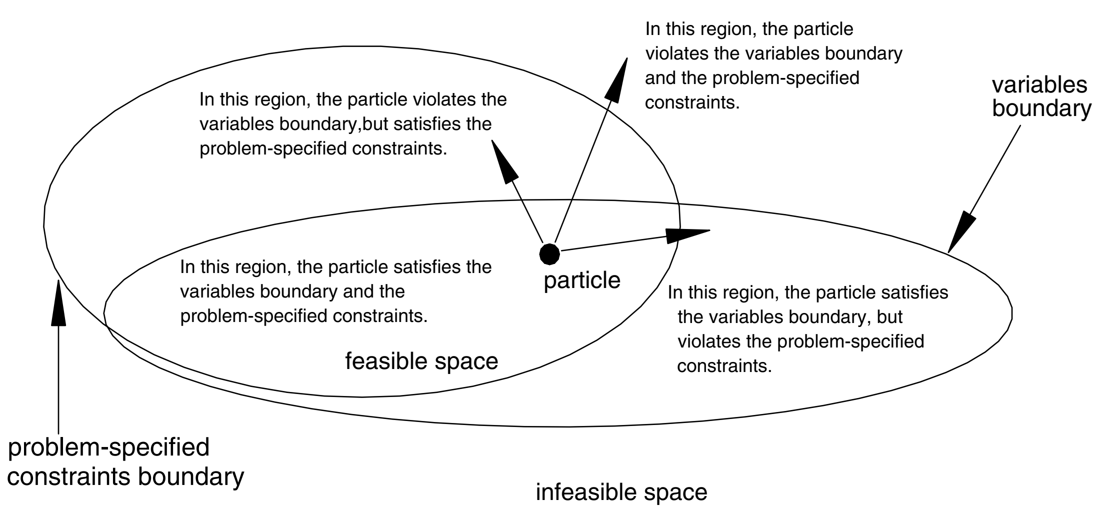

Fig. 1. The particle may violate the problem-specific constraints or the variables' boundary.

Table 1
The pseudo-code for the HPSO

```
Set k = 1;
Randomly initialize positions and velocities of all particles;
  FOR(each particle i in the initial population)
    WHILE(the constraints are violated)
      Randomly re-generate the current particle X_i
    END WHILE
  END FOR
WHILE(the termination conditions are not met)
  FOR(each particle i in the swarm)
    Generate the velocity and update the position of the current particle (vector) X_i
    Check feasibility stage I: Check whether each component of the current vector violates
        its corresponding boundary or not. If it does, select the
        corresponding component of the vector from pbest swarm
        randomly
    Check feasibility stage II: Check whether the current particle violates the problem
        specified constraints or not. If it does, reset it to the
        previous position X_i^{k-1}
    Calculate the fitness value f(X_i^k) of the current particle
    Update pbest: Compare the fitness value of pbest with f(X_i^k). If the f(X_i^k) is better than
        the fitness 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
        fitness value of gbest, gbest is set to the position of the current
    particle X_i^k
  END FOR
  Set k = k + 1
END WHILE
```

Similarly, the PSO stores the feasible and "good" vectors (particles) in the *pbest swarm*, as does the harmony memory in the HS algorithm. Hence, the vector (particle) violating the variables' boundaries can be generated randomly again by such a technique-selecting for the components of different vectors in the *pbest swarm*. There are two different ways to apply this technique to the PSO when any one of the components of the vector violates its corresponding variables' boundary. Firstly, all the components of this vector should be generated. Secondly, only this component of the vector should be generated again by such a technique. In our experiments, the results show that the former makes the particles moving to the local solution easily, and the latter can reach the global solution in relatively less number of iterations.

Therefore, applying such a technique to the PSOPC can improve its performance, although it already has a better convergence rate and accuracy than the PSO. The pseudo-code for the HPSO algorithm is listed in Table 1.

7. Numerical examples

In this section, five pin connected structures commonly used in literature are selected as benchmark problem to test the HPSO. The proposed algorithm is coded in Fortran language and executed on a Pentium 4, 2.93 GHz machine.

The examples given in the simulation studies include

- a 10-bar planar truss structure subjected to four concentrated loads as shown in Fig. 2;
- a 17-bar planar truss structure subjected to a single concentrated load at its free end as shown in Fig. 5;

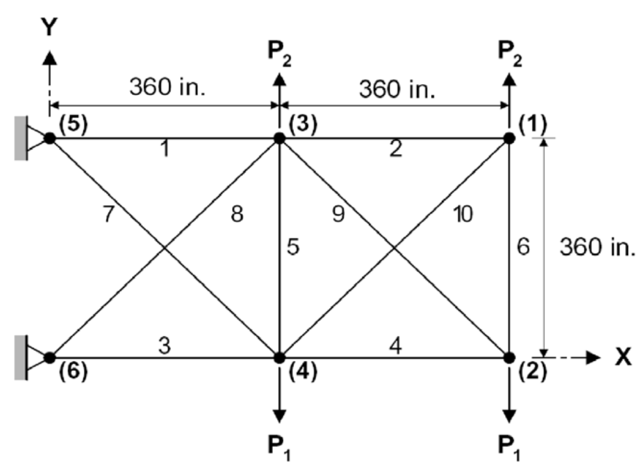

Fig. 2. A 10-bar planar truss structure.

- a 22-bar spatial truss structure subjected to three load cases;
- a 25-bar spatial truss structure subjected to two load cases; and
- a 72-bar spatial truss structure subjected to two load cases.

All these truss structures are analyzed by the finite element method (FEM).

The PSO, PSOPC and HPSO schemes are applied, respectively, to all these examples and the results are compared in order to evaluate the performance of the new algorithm. For all these algorithms, a population of 50 individuals is used; 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. The passive congregation coefficient c_3 is given as 0.6 for the PSOPC [8] and the HPSO algorithms. The maximum number of iterations is limited to 3000. The maximum velocity is set as the difference between the upper bound and the lower bound of variables, which ensures that the particles are able to fly into the problem-specific constraints' region.

7.1. The 10-bar planar truss structure

The 10-bar truss structure, shown in Fig. 2 [16], has previously been analyzed by many researchers, such as Schmit and Farshi [17], Rizzi [18], and Lee and Geem [16]. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10,000 ksi. The members are subjected to the stress limits of ±25 ksi. All nodes in both vertical and horizontal directions are subjected to the displacement limits of ±2.0 in. There are 10 design variables in this example and the minimum permitted cross-sectional area of each member is 0.1 in^2. Two cases are considered: Case 1, $P_1 = 100$ kips and $P_2 = 0$; and Case 2, $P_1 = 150$ kips and $P_2 = 50$ kips.

For both load cases, the PSOPC and the HPSO algorithms achieve the best solutions after 3000 iterations. However, the latter is closer to the best solution than the former after about 500 iterations. The HPSO algorithm displays a faster convergence rate than the PSOPC algorithm in this example. The performance of the PSO algorithm is the worst among the three. Tables 2 and 3 show the solutions. Figs. 3 and 4 provide a comparison of the convergence rates of the three algorithms.

7.2. The 17-bar planar truss structure

The 17-bar truss structure, shown in Fig. 5 [16], had been analyzed by Khot and Berke [19], Adeli and Kumar

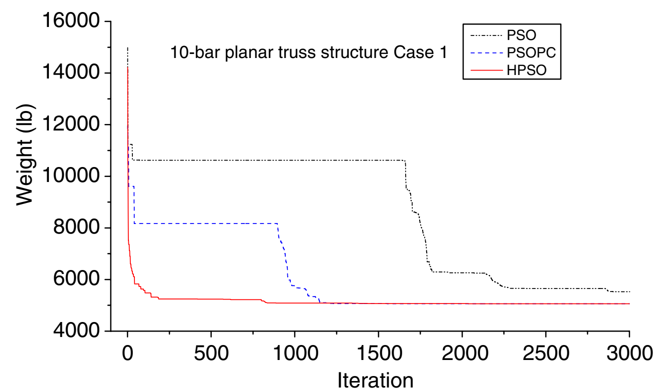

Fig. 3. Comparison of the convergence rates between the three algorithms for the 10-bar planar truss structure (Case 1).

Table 2
Comparison of optimal designs for the 10-bar planar truss (Case 1)

Variables		Optimal cross-sectional areas (in.^{2})					
		Schmit [17]	Rizzi [18]	Lee [16]	PSO	PSOPC	HPSO
1	A_1	33.43	30.73	30.15	33.469	30.569	30.704
2	A_2	0.100	0.100	0.102	0.110	0.100	0.100
3	A_3	24.26	23.93	22.71	23.177	22.974	23.167
4	A_4	14.26	14.73	15.27	15.475	15.148	15.183
5	A_5	0.100	0.100	0.102	3.649	0.100	0.100
6	A_6	0.100	0.100	0.544	0.116	0.547	0.551
7	A_7	8.388	8.542	7.541	8.328	7.493	7.460
8	A_8	20.74	20.95	21.56	23.340	21.159	20.978
9	A_9	19.69	21.84	21.45	23.014	21.556	21.508
10	A_{10}	0.100	0.100	0.100	0.190	0.100	0.100
Weight (lb)		5089.0	5076.66	5057.88	5529.50	5061.00	5060.92

Table 3
Comparison of optimal designs for the 10-bar planar truss (Case 2)

Variables		Optimal cross-sectional areas (in.^{2})					
		Schmit [17]	Rizzi [18]	Lee [16]	PSO	PSOPC	HPSO
1	A_1	24.29	23.53	23.25	22.935	23.743	23.353
2	A_2	0.100	0.100	0.102	0.113	0.101	0.100
3	A_3	23.35	25.29	25.73	25.355	25.287	25.502
4	A_4	13.66	14.37	14.51	14.373	14.413	14.250
5	A_5	0.100	0.100	0.100	0.100	0.100	0.100
6	A_6	1.969	1.970	1.977	1.990	1.969	1.972
7	A_7	12.67	12.39	12.21	12.346	12.362	12.363
8	A_8	12.54	12.83	12.61	12.923	12.694	12.894
9	A_9	21.97	20.33	20.36	20.678	20.323	20.356
10	A_{10}	0.100	0.100	0.100	0.100	0.103	0.101
Weight (lb)		4691.84	4676.92	4668.81	4679.47	4677.70	4677.29

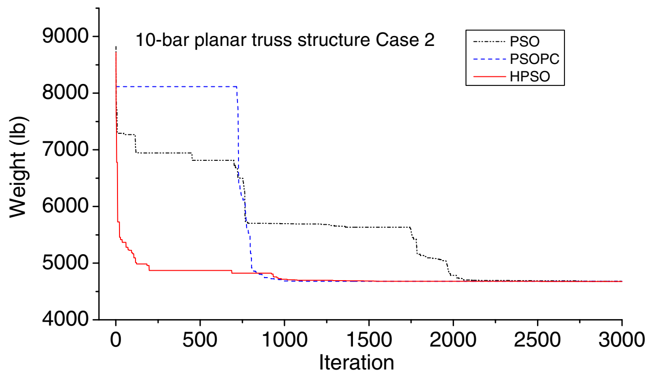

Fig. 4. Comparison of the convergence rates between the three algorithms for the 10-bar planar truss structure (Case 2).

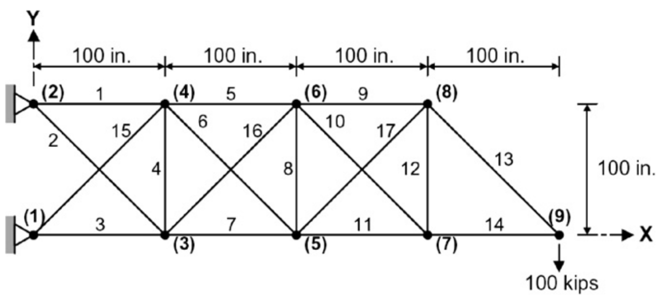

Fig. 5. A 17-bar planar truss structure.

[20] and Lee and Geem [16]. The material density is 0.268 lb/in.^{3} and the modulus of elasticity is 30,000 ksi. The members are subjected to the stress limits of $\pm$ 50 ksi. All nodes in both directions are subjected to the displacement limits of $\pm$ 2.0 in. There are 17 design variables in this example and the minimum permitted cross-sectional area of each member is 0.1 in.^{2}. A single vertical downward load of 100 kips at node 9 is considered.

Both the PSOPC and HPSO algorithms achieve a good solution after 3000 iterations, and the latter shows a better convergence rate than the former, especially at the early stage of iterations. In this case, the PSO algorithm is not fully converged when the maximum number of iterations is reached. Table 4 shows the solutions and Fig. 6 compares the convergence rates of the three algorithms.

7.3. The 22-bar spatial truss structure

The 22-bar spatial truss structure, shown in Fig. 7 [16], had been studied by Lee and Geem [16]. 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 listed in Table 5. All nodes in all three directions are subjected to the displacement limits of $\pm$2.0 in. Three load cases are listed in Table 6. There are 22 members, which fall into 7 groups, as follows: (1) A_1–A_4, (2) A_5–A_6, (3) A_7–A_8, (4) A_9–A_{10}, (5) A_{11}–A_{14}, (6) A_{15}–A_{18}, and (7) A_{19}–A_{22}. The minimum permitted cross-sectional area of each member is 0.1 in.^{2}.

In this example, the HPSO algorithm has converged after 50 iterations, while the PSOPC and PSO algorithms need more than 500 and 1000 iterations, respectively. The optimum results obtained by using the HPSO algorithm are significantly better than that obtained by the HS [16] and the PSO algorithms. Table 7 shows the optimal solutions of the four algorithms and Fig. 8 provides the convergence rates of three of the four algorithms.

7.4. The 25-bar spatial truss structure

The 25-bar spatial truss structure shown in Fig. 9 [16] had been studied by several researchers, such as Schmit and Farshi [17], Rizzi [18], and Lee and Geem [16]. The material density is 0.1 lb/in.^{3} and the modulus of elasticity

Table 4
Comparison of optimal designs for the 17-bar planar truss

Variables		Optimal cross-sectional areas (in.^{2})					
		Khot [19]	Adeli [20]	Lee [16]	PSO	PSOPC	HPSO
1	A_1	15.930	16.029	15.821	15.766	15.981	15.896
2	A_2	0.100	0.107	0.108	2.263	0.100	0.103
3	A_3	12.070	12.183	11.996	13.854	12.142	12.092
4	A_4	0.100	0.110	0.100	0.106	0.100	0.100
5	A_5	8.067	8.417	8.150	11.356	8.098	8.063
6	A_6	5.562	5.715	5.507	3.915	5.566	5.591
7	A_7	11.933	11.331	11.829	8.071	11.732	11.915
8	A_8	0.100	0.105	0.100	0.100	0.100	0.100
9	A_9	7.945	7.301	7.934	5.850	7.982	7.965
10	A_{10}	0.100	0.115	0.100	2.294	0.113	0.100
11	A_{11}	4.055	4.046	4.093	6.313	4.074	4.076
12	A_{12}	0.100	0.101	0.100	3.375	0.132	0.100
13	A_{13}	5.657	5.611	5.660	5.434	5.667	5.670
14	A_{14}	4.000	4.046	4.061	3.918	3.991	3.998
15	A_{15}	5.558	5.152	5.656	3.534	5.555	5.548
16	A_{16}	0.100	0.107	0.100	2.314	0.101	0.103
17	A_{17}	5.579	5.286	5.582	3.542	5.555	5.537
Weight (lb)		2581.89	2594.42	2580.81	2724.37	2582.85	2581.94

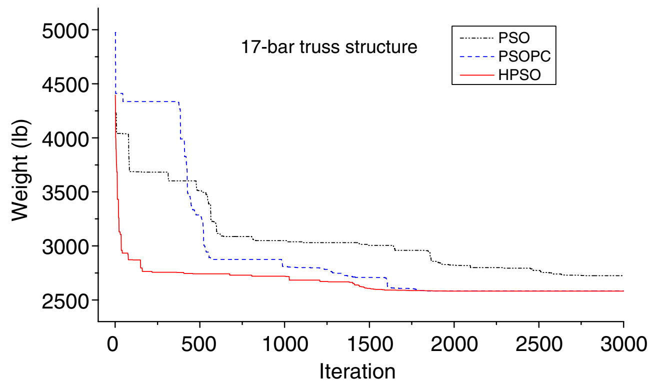

Fig. 6. Comparison of the convergence rates between the three algorithms for the 17-bar planar truss structure.

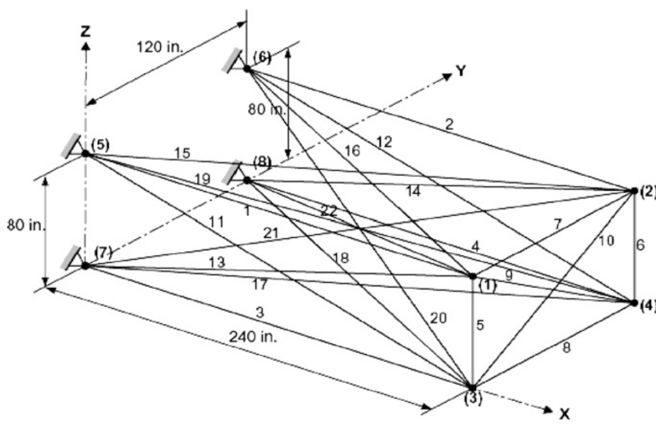

Fig. 7. A 22-bar spatial truss structure.

Table 5
Member stress limits for the 22-bar spatial truss structure

Variables		Compressive stress limits (ksi)	Tensile stress limits (ksi)
1	A_1	24.0	36.0
2	A_2	30.0	36.0
3	A_3	28.0	36.0
4	A_4	26.0	36.0
5	A_5	22.0	36.0
6	A_6	20.0	36.0
7	A_7	18.0	36.0

Table 6
Load cases for the 22-bar spatial truss structure

Node	Case 1 (Kips)			Case 2 (Kips)			Case 3 (Kips)		
	P_X	P_Y	P_Z	P_X	P_Y	P_Z	P_X	P_Y	P_Z
1	−20.0	0.0	−5.0	−20.0	−5.0	0.0	−20.0	0.0	35.0
2	−20.0	0.0	−5.0	−20.0	−50.0	0.0	−20.0	0.0	0.0
3	−20.0	0.0	−30.0	−20.0	−5.0	0.0	−20.0	0.0	0.0
4	−20.0	0.0	−30.0	−20.0	−50.0	0.0	−20.0	0.0	−35.0

is 10,000 ksi. The stress limits of the members are listed in Table 8. All nodes in all directions are subjected to the displacement limits of ±0.35 in. Two load cases listed in Table 9 are considered. 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)

Table 7
Comparison of optimal designs for the 22-bar spatial truss structure

Variables		Optimal cross-sectional areas (in.^{2})			
		Lee [16]	PSO	PSOPC	HPSO
1	A_1	2.588	1.657	3.041	3.157
2	A_2	1.083	0.716	1.191	1.269
3	A_3	0.363	0.919	0.985	0.980
4	A_4	0.422	0.175	0.105	0.100
5	A_5	2.827	4.576	3.430	3.280
6	A_6	2.055	3.224	1.543	1.402
7	A_7	2.044	0.450	1.138	1.301
Weight (lb)		1022.23	1057.14	977.80	977.81

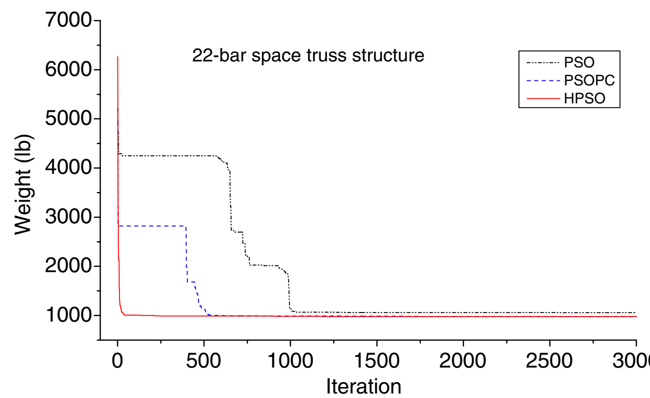

Fig. 8. Comparison of the convergence rates between the three algorithms for the 22-bar spatial truss structure.

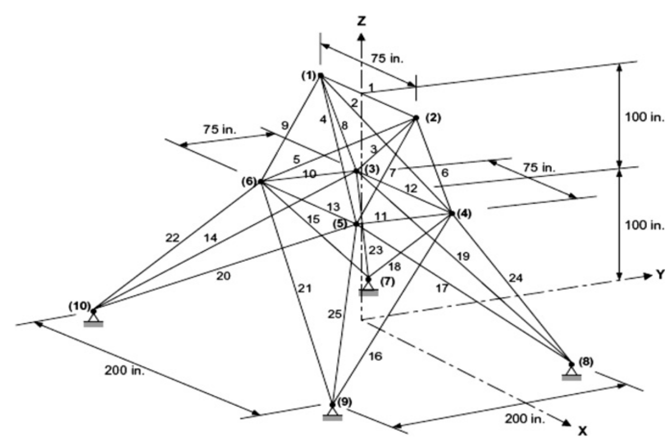

Fig. 9. A 25-bar spatial truss structure.

Table 8
Member stress limits for the 25-bar spatial truss structure

Variables		Compressive stress limits (ksi)	Tensile stress limits (ksi)
1	A_1	35.092	40.0
2	A_2	11.590	40.0
3	A_3	17.305	40.0
4	A_4	35.092	40.0
5	A_5	35.902	40.0
6	A_6	6.759	40.0
7	A_7	6.959	40.0
8	A_8	11.802	40.0

Table 9
Load cases for the 25-bar spatial truss structure

Node	Case 1 (Kips)			Case 2 (Kips)		
	P_X	P_Y	P_Z	P_X	P_Y	P_Z
1	0.0	20.0	−5.0	1.0	10.0	−5.0
2	0.0	−20.0	−5.0	0.0	10.0	−5.0
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0

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 minimum permitted cross-sectional area of each member is 0.01 in.^{2}.

For this spatial truss structure, it takes about 1000 and 3000 iterations for the PSOPC and the PSO algorithms to converge, respectively. However the HPSO algorithm takes only 50 iterations to converge. Indeed, in this example, the PSO algorithm did not fully converge when the maximum number of iterations is reached. Table 10 shows the solutions and Fig. 10 compares the convergence rate of the three algorithms.

7.5. The 72-bar spatial truss structure

The 72-bar spatial truss structure shown in Fig. 11 had also been studied by many researchers, such as Schmit and Farshi [17], Khot and Berke [19], Adeli and Kumar [20], Lee and Geem [16], Adeli and Park [21], and Sarma and Adeli [22]. The material density is 0.1 lb/in.^{3} and the modulus of elasticity is 10,000 ksi. The members are subjected to the stress limits of ±25 ksi. The uppermost nodes are subjected to the displacement limits of ±0.25 in. in both the x and y directions. Two load cases are listed in Table 11. There are 72 members classified into 16 groups: (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}. For case 1, the minimum permitted cross-sectional area of each member is 0.1 in^2. For case 2, the minimum permitted cross-sectional area of each member is 0.01 in.^{2}.

For both the loading cases, the PSOPC and the HPSO algorithms can achieve the optimal solution after 2500 iter-

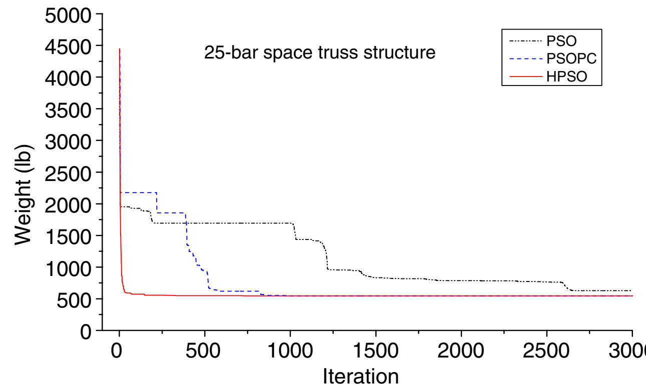

Fig. 10. Convergence rate comparison between the three algorithms for the 25-bar spatial truss structure.

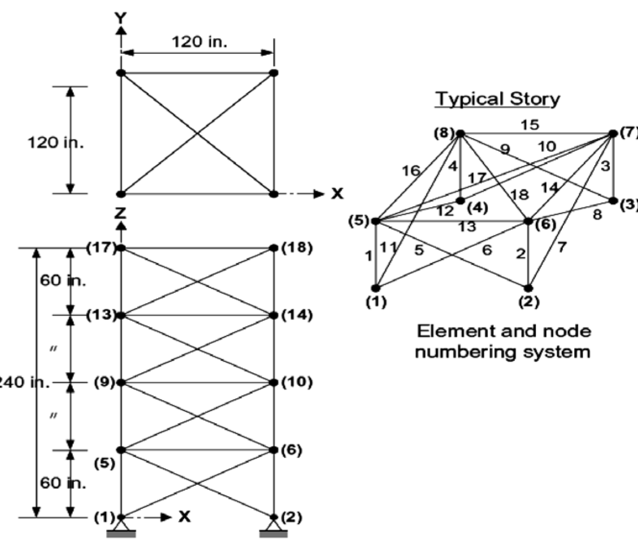

Fig. 11. A 72-bar spatial truss structure.

Table 11
Load cases for the 72-bar spatial truss structure

Node	Case 1 (Kips)			Case 2 (Kips)		
	P_X	P_Y	P_Z	P_X	P_Y	P_Z
17	5.0	5.0	−5.0	0.0	0.0	−5.0
18	0.0	0.0	0.0	0.0	0.0	−5.0
19	0.0	0.0	0.0	0.0	0.0	−5.0
20	0.0	0.0	0.0	0.0	0.0	−5.0

Table 10
Comparison of optimal designs for the 25-bar spatial truss structure

Variables		Optimal cross-sectional areas (in.^{2})					
		Schmit [17]	Rizzi [18]	Lee [16]	PSO	PSOPC	HIPSO
1	A_1	0.010	0.010	0.047	9.863	0.010	0.010
2	A_2–A_5	1.964	1.988	2.022	1.798	1.979	1.970
3	A_6–A_9	3.033	2.991	2.950	3.654	3.011	3.016
4	A_{10}–A_{11}	0.010	0.010	0.010	0.100	0.100	0.010
5	A_{12}–A_{13}	0.010	0.010	0.014	0.100	0.100	0.010
6	A_{14}–A_{17}	0.670	0.684	0.688	0.596	0.657	0.694
7	A_{18}–A_{21}	1.680	1.677	1.657	1.659	1.678	1.681
8	A_{22}–A_{25}	2.670	2.663	2.663	2.612	2.693	2.643
Weight (lb)		545.22	545.36	544.38	627.08	545.27	545.19

Table 12
Comparison of optimal designs for the 72-bar spatial truss structure (Case 1)

Variables		Optimal cross-sectional areas (in.^{2})						
		Schmit [17]	Adeli [20]	Khot [19]	Lee [16]	PSO	PSOPC	IPSO
1	A_1–A_4	2.078	2.026	1.893	1.7901	41.794	1.855	1.857
2	A_5–A_{12}	0.503	0.533	0.517	0.521	0.195	0.504	0.505
3	A_{13}–A_{16}	0.100	0.100	0.100	0.100	10.797	0.100	0.100
4	A_{17}–A_{18}	0.100	0.100	0.100	0.100	6.861	0.100	0.100
5	A_{19}–A_{22}	1.107	1.157	1.279	1.229	0.438	1.253	1.255
6	A_{23}–A_{30}	0.579	0.569	0.515	0.522	0.286	0.505	0.503
7	A_{31}–A_{34}	0.100	0.100	0.100	0.100	18.309	0.100	0.100
8	A_{35}–A_{36}	0.100	0.100	0.100	0.100	1.220	0.100	0.100
9	A_{37}–A_{40}	0.264	0.514	0.508	0.517	5.933	0.497	0.496
10	A_{41}–A_{48}	0.548	0.479	0.520	0.504	19.545	0.508	0.506
11	A_{49}–A_{52}	0.100	0.100	0.100	0.100	0.159	0.100	0.100
12	A_{53}–A_{54}	0.151	0.100	0.100	0.101	0.151	0.100	0.100
13	A_{55}–A_{58}	0.158	0.158	0.157	0.156	10.127	0.100	0.100
14	A_{59}–A_{66}	0.594	0.550	0.539	0.547	7.320	0.525	0.524
15	A_{67}–A_{70}	0.341	0.345	0.416	0.442	3.812	0.394	0.400
16	A_{71}–A_{72}	0.608	0.498	0.551	0.590	18.196	0.535	0.534
Weight (lb)		388.63	379.31	379.67	379.27	6818.67	369.65	369.65

Table 13
Comparison of optimal designs for the 72-bar spatial truss structure (Case 2)

Variables		Optimal cross-sectional areas (in.^{2})						
		Adeli [20]	Sarma [22]		Lee [16]	PSO	PSOPC	HPSO
			Simple GA	Fuzzy GA				
1	A_1–A_4	2.755	2.141	1.732	1.963	40.053	1.652	1.907
2	A_5–A_{12}	0.510	0.510	0.522	0.481	0.237	0.547	0.524
3	A_{13}–A_{16}	0.010	0.054	0.010	0.010	21.692	0.100	0.010
4	A_{17}–A_{18}	0.010	0.010	0.013	0.011	0.657	0.101	0.010
5	A_{19}–A_{22}	1.370	1.489	1.345	1.233	22.144	1.102	1.288
6	A_{23}–A_{30}	0.507	0.551	0.551	0.506	0.266	0.589	0.523
7	A_{31}–A_{34}	0.010	0.057	0.010	0.011	1.654	0.011	0.010
8	A_{35}–A_{36}	0.010	0.013	0.013	0.012	10.284	0.010	0.010
9	A_{37}–A_{40}	0.481	0.565	0.492	0.538	0.559	0.581	0.544
10	A_{41}–A_{48}	0.508	0.527	0.545	0.533	12.883	0.458	0.528
11	A_{49}–A_{52}	0.010	0.010	0.066	0.010	0.138	0.010	0.019
12	A_{53}–A_{54}	0.643	0.066	0.013	0.167	0.188	0.152	0.020
13	A_{55}–A_{58}	0.215	0.174	0.178	0.161	29.048	0.161	0.176
14	A_{59}–A_{66}	0.518	0.425	0.524	0.542	0.632	0.555	0.535
15	A_{67}–A_{70}	0.419	0.437	0.396	0.478	3.045	0.514	0.426
16	A_{71}–A_{72}	0.504	0.641	0.595	0.551	1.711	0.648	0.612
Weight (lb)		376.50	372.40	364.40	364.33	5417.02	368.45	364.86

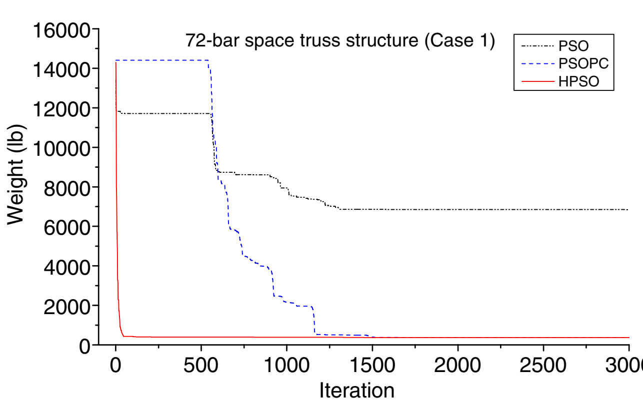

Fig. 12. Comparison of the convergence rates between the three algorithms for the 72-bar spatial truss structure (Case 1).

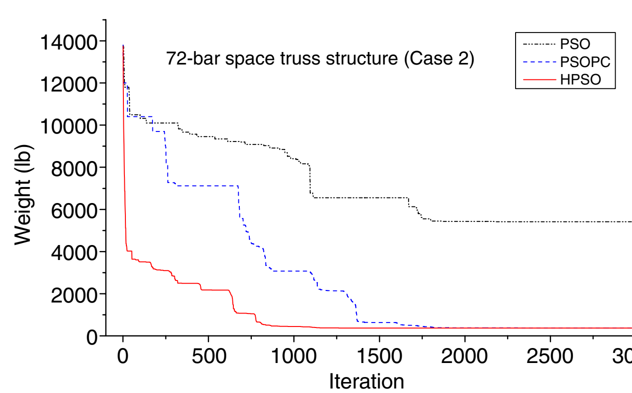

Fig. 13. Comparison of the convergence rates between the three algorithms for the 72-bar spatial truss structure (Case 2).

ations. However, the latter shows a faster convergence rate than the former, especially at the early stage of iterations. The PSO algorithm cannot reach the optimal solution after the maximum number of iterations. The solutions of the two loading cases are given in Tables 12 and 13, respectively. Figs. 12 and 13 compares the convergence rate of the three algorithms for the two loading cases.

8. Conclusions

In this paper, a heuristic particle swarm optimizer (HPSO), based on the particle swarm optimizer with passive congregation (PSOPC), and the harmony search (HS) algorithm are presented. The HPSO algorithm handles the constraints of variables using the harmony search scheme in corporation with the 'fly-back mechanism' method used to deal with the problem-specific constraints. Compared with the PSO and the PSOPC algorithms, the HPSO algorithm does not allow any particles to fly outside the boundary of the variables and makes a full use of algorithm flying behavior of each particle. Thus this algorithm performs most efficient than the others.

The efficiency of the HPSO algorithm presented in this paper is tested for optimum design of five planar and spatial pin connected structures. The results show that the HPSO algorithm converge more quickly than the PSO and the PSOPC algorithms. In particular, in the early iterations.

A drawback of this HPSO algorithm at present is that its convergence rate will slow down, when the number of the iterations increase. Further study is being conducted for improvement.

Acknowledgements

We would like to thank Guangdong Natural Science Foundation and Guangzhou Bureau of Science and Technology, China, for supporting this project (the Project numbers are 032489, 06104655 and 2003Z3-D0221, respectively).

References

[1] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Method Appl Mech Eng 2002;191(11–12):1245–87.
[2] Nanakorn Pruettha, Konlakarn, Meesomklin. An adaptive penalty function in genetic algorithms for structural design optimization. Comput Struct 2001;79(2001):2527–39.
[3] Kalyanmoy Deb, Surendra Gulati. Design of truss-structures for minimum weight using genetic algorithms. Finite Elem Anal Des 2001;37(2001):447–65.
[4] Ali Nicholas, Behdinan Kamran, Fawaz Zouheir. Applicability and viability of a GA based finite element analysis architecture for structural design optimization. Comput Struct 2003;81:2259–71.
[5] Kennedy J, Eberhart R. Particle swarm optimization. IEEE International Conference on Neural Networks, vol. 4. IEEE Press; 1995. 1942–8.
[6] Kennedy J, Eberhart RC. Swarm intelligence. Morgan Kaufman Publishers; 2001.
[7] Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. In: Proceeding of the Evolutionary Programming Conference, San Diego, USA, 1998.
[8] He S, Wu QH, Wen JY, Saunders JR, Paton RC. A particle swarm optimizer with passive congregation. Biosystem 2004;78:135–47.
[9] He S, Prempain E, Wu QH. An improved particle swarm optimizer for mechanical design optimization problems. Eng Optimiz 2004; 36(5):585–605.
[10] Davis L. Genetic algorithms and simulated annealing. London: Pitman; 1987.
[11] Le Riche RG, Knopf-Lenoir C, Haftka RT. A segregated genetic algorithm for constrained structural optimization. In: Sixth International Conference on Genetic Algorithms. University of Pittsburgh: Morgan Kaufman; 1995. p. 558–65.
[12] Li LJ, Ren FM, Liu F, Wu QH. An improved particle swarm optimization method and its application in civil engineering. In: Topping BHV, Montero G, Montenegro R, editors. Proceedings of the Fifth International Conference on Engineering Computational Technology. Stirlingshire, United Kingdom: Civil-Comp Press; 2006. Paper 42.
[13] Van den Bergh, Engelbrecht A. Using neighborhood with the guaranteed convergence PSO. In: 2003 IEEE Swarm Intelligence Symposium, USA, 2003, p. 235–42.
[14] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proc IEEE Inc Conf on Evolutionary Computation, 1997, p. 303–8.
[15] Geem ZW, Kim J-H, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76:60–8.
[16] Lee Kang Seok, Geem Zong Woo. A new structural optimization method based on the harmony search algorithm. Comput Struct 2004;82:781–98.
[17] Schmit Jr LA, Farshi B. Some approximiation concepts for structural synthesis. AIAA J 1974;12(5):692–9.
[18] Rizzi P. Optimization of multiconstrained structures based on optimality criteria. AIAA/ASME/SAE 17th Structures. In: Structural Dynamics and Materials Conference, King of Prussia, PA; 1976.
[19] Khot NS, Berke L. Structural optimization using optimality criteria methods. In: Atrek E, Gallagher RH, Ragsdell KM, Zienkiewicz OC, editors. New directions in optimum structural design. New York: - John Wiley; 1984.
[20] Adeli H, Kumar S. Distributed genetic algorithm for structural optimization. J Aerospace Eng, ASCE 1995;8(3):156–63.
[21] Adeli H, Park H-S. Neurocomputing for design automation. Boca Raton, FL: CRC Press; 1998.
[22] Sarma KC, Adeli H. Fuzzy genetic algorithm for optimization of steel structures. J Struct Eng, ASCE 2000;126(5):596–604.