



Sizing optimization of truss structures using flower pollination algorithm



Gebrail Bekdaş^a, Sinan Melih Nigdeli^{a,*}, Xin-She Yang^b

^a Department of Civil Engineering, Istanbul University, 34320 Avcılar, Istanbul, Turkey

^b Design Engineering and Mathematics, Middlesex University London, The Burroughs, London, UK

ARTICLE INFO

Article history:

Received 6 May 2015

Received in revised form 19 August 2015

Accepted 20 August 2015

Available online 2 September 2015

Keywords:

Flower pollination algorithm

Truss structures

Sizing optimization

Iterative constraint handling strategy

ABSTRACT

The recently developed flower pollination algorithm is used to minimize the weight of truss structures, including sizing design variables. The new algorithm can efficiently combine local and global searches, inspired by cross-pollination and self-pollination of flowering plants, respectively. Furthermore, it implements an iterative constraint handling strategy where trial designs are accepted or rejected based on the allowed amount of constraint violation that is progressively reduced as the search process approaches the optimum. This strategy aims to obtain always feasible optimized designs. The new algorithm is tested using three classical sizing optimization problems of 2D and 3D truss structures. Optimization results show that the proposed method is competitive with other state-of-the-art metaheuristic algorithms presented in the literature.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The main objectives in structural design are to ensure the safety of structures and find a design with the maximum gain. Generally speaking, safety measures are defined as design constraints, while objective functions depending on design variables are defined as the maximum gain. In recent years, nature-inspired metaheuristic algorithms have been commonly used in engineering optimization. These iterative algorithms are very effective to find precise optimum values of challenging engineering problems with multiple variables and constraints. In addition, metaheuristic algorithms allow to account for design limitations by combining optimization process with accurate engineering analysis. Metaheuristic algorithms can be grouped as either trajectory-based algorithms or population-based algorithms. Simulated Annealing (SA) method developed by Kirkpatrick et al. [1] is a trajectory-based algorithm, while Harmony Search (HS) [2], Genetic Algorithm (GA) [3], Cuckoo Search [4], Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6] are all population-based algorithms.

In addition, new metaheuristic algorithms are also being developed in order to improve the optimization capability and convergence behavior. For example, the Flower Pollination Algorithm (FPA) is a population-based metaheuristic method recently developed by Yang [7], which imitates the nature of flower pollination.

In the development of the algorithm, the main characteristics of flower pollination were idealized into four rules. This study will apply FPA to solve sizing optimization problems of 3D and 2D truss structures.

Truss structures have been optimized using several approaches. For example, Adeli and Kamal optimized space trusses with a dual simplex algorithm to find a local optimum of the approximate problem, while the original problem was iteratively solved [8]. Rajeev and Krishnamoorthy used discrete variables and GA with a penalty parameter depending on constraint violation [9]. Cao also employed GA for the optimum design of frame structures [10]. Erbatır et al. employed GA for the optimum design of planar and space truss structures with continuous and discrete variables [11]. Schutte and Groenwold used PSO for the sizing and layout optimization of truss structures [12]. Camp and Bichon employed ACO to minimize the total weight of the structure subject to stress and deflection constraints [13]. Lee and Geem optimally designed trusses under multiple loading conditions by using HS algorithm and continuous design variables [14]. Big bang–big crunch (BB–BC) algorithm developed by Erol and Eksin [15] was employed in the optimum design methodology of space trusses by Camp [16]. Li et al. developed a heuristic particle swarm optimizer based on the particle swarm optimizer with passive congregation and a HS scheme; this method was successfully applied to the optimum design of planar and spatial truss structures [17].

In addition, Perez and Behdinan optimized truss structures with PSO [18]. Togan and Daloglu improved GA with an initial population strategy and self-adaptive member grouping [19]. Lamberti

* Corresponding author. Tel.: +90 2124737070x17949; fax: +90 2128891887.
E-mail address: melihnig@istanbul.edu.tr (S.M. Nigdeli).

presented an efficient SA algorithm for sizing and layout optimization of truss structures [20]. A hybrid BB-BC/PSO algorithm, including Sub-Optimization Mechanism (SOM) was used for sizing optimization of space trusses [21]. Kaveh and Talatahari proposed a hybrid optimization method combining PSO, ACO and HS algorithm for truss structures with discrete and continuous variables [22,23]. Sonmez proposed an optimization methodology including the Artificial Bee Colony and Adaptive Penalty function approach (ABC-AP) in order to minimize the weight of truss structures [24]. Degertekin applied two different improved HS algorithms called the efficient HS algorithm and self-adaptive HS algorithm in order to optimize the size of truss structures [25]. Teaching Learning Based Optimization (TLBO) was applied to truss sizing optimization problems by Degertekin and Hayalioglu [26]. Also, Camp and Farshchin employed a modified TLBO for optimum design of truss [27]. Kaveh et al. developed hybrid particle swallow swarm optimization and the developed algorithm was tested with truss weight minimization problems [28].

Furthermore, chaotic swarming of particles which is the combination of swarm intelligence and chaos theory was developed for optimization of truss structures [29]. Dede and Ayvaz developed a methodology for sizing and layout of trusses using TLBO [30]. Kaveh et al. also used an improved magnetic charged system search in order to solve truss optimization problems with continuous and discrete variables [31]. Colliding bodies optimization (CBO), developed by Kaveh and Mahdavi [32], reproduces the mechanisms of the collisions of moving bodies [33]: the algorithm has successfully been utilized in truss optimization. Kaveh and Ilchi Ghazaan later developed an enhanced colliding bodies optimization algorithm (ECBO) which stores some best solutions into the memory in order to optimize truss structures with continuous and discrete variables [34]. Another efficient algorithm for sizing and layout optimization of truss structures is ray optimization (RO) [35].

In the present study, the newly developed metaheuristic Flower Pollination Algorithm (FPA) is applied to structural optimization problems of planar and space trusses. In order to reach the global optimum, an iterative or adaptive constraint handling strategy is included in the search engine. The efficiency of the proposed approach is demonstrated by solving three classical weight minimization problems including sizing variables. Optimization results indicate that FPA is very competitive with other metaheuristic algorithms and can always find efficient designs within the predefined constraint tolerance.

2. Optimum design of truss structures

Trusses are structural systems, consisting of N bars joined by nodes. The system is subject to the external forces applied at the joints. The aim of structural optimization of truss systems is to minimize the total weight of the structure.

In the proposed methodology, the optimization process is encoded together with the structural analysis of the truss. The latter is carried out by using the stiffness method, and nodal displacements are calculated according to

$$\Delta = K^{-1}P. \quad (1)$$

In Eq. (1), Δ , K and P are the nodal displacement vector, system stiffness matrix and external load vector, respectively. The system stiffness matrix is constructed by merging the element stiffness matrices in global coordinates and erasing row and columns which correspond to zero displacements according to the boundary

conditions. The stiffness matrix of a bar element with three degrees of freedom at each node is given by

$$K_i = EA_i \begin{bmatrix} l^2 & lm & nl & -l^2 & -lm & -nl \\ lm & m^2 & mn & -lm & -m^2 & -mn \\ nl & mn & n^2 & -nl & -mn & -n^2 \\ -l^2 & -lm & -nl & l^2 & lm & nl \\ -lm & -m^2 & -mn & lm & m^2 & mn \\ -nl & -mn & n^2 & nl & mn & n^2 \end{bmatrix} \quad (2)$$

where

$$l = \frac{L_{xi}}{L_i}, \quad m = \frac{L_{yi}}{L_i} \quad \text{and} \quad n = \frac{L_{zi}}{L_i}. \quad (3)$$

In Eqs. (2) and (3), the total length of the bars (L_i) and the dimensions of the length in x , y and z coordinates (L_{xi} , L_{yi} and L_{zi}) are calculated using the coordinates of the nodes and bounds of the elements which are determined as the design constants. Also, the elasticity modulus (E) and density (γ) of the material of bars are defined as design constants. The areas of bars (A_i) (from $i = 1$ to N) are the design variables (X) of the optimization problem. The aim of the optimization is to minimize the total structural weight. That is

$$\min W = \sum_{i=1}^N \gamma L_i A_i, \quad (A_i \in \mathbb{R}) \quad (4)$$

for the design variables:

$$X^T = \{A_1, A_2, \dots, A_N\} \quad (5)$$

within the ranges of

$$A^L \leq A_i \leq A^U \quad i = 1, N \quad (5')$$

subject to the stress ($g_1(X) \leq 0$) and displacement ($g_2(X) \leq 0$) constraints

$$\begin{aligned} g_1(X) : \sigma^L \leq \sigma_i \leq \sigma^U \quad i = 1, N \\ \text{and} \quad g_2(X) : \delta^L \leq \delta_j \leq \delta^U \quad j = 1, N_j. \end{aligned} \quad (6)$$

A^L and A^U are the lower and upper bounds of the solution ranges of the design variables. δ^L and δ^U are the displacement limits which are generally equal in absolute values, but with the opposite signs. The displacement of nodes (δ_j) from $j = 1$ to N_j (for a system with j nodes) are the components of the displacement vector:

$$\Delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{N_j-1} \\ \delta_{N_j} \end{bmatrix} \quad (\delta_{1,N_j} \in \mathbb{R}). \quad (7)$$

σ^L and σ^U are two different types of stress limits which are for compression (σ^L in $-$ sign) and tension (σ^U in $+$ sign). The stresses of a bar (σ_i^G) in global coordinate is calculated by

$$\sigma_i^G = \frac{K_i \Delta_i}{A_i}, \quad i = 1, N \quad (8)$$

where Δ_i is the vector of the nodal displacements of i th bar. Axial stress on a bar (σ_i) is calculated by multiplying global stresses by directional cosines.

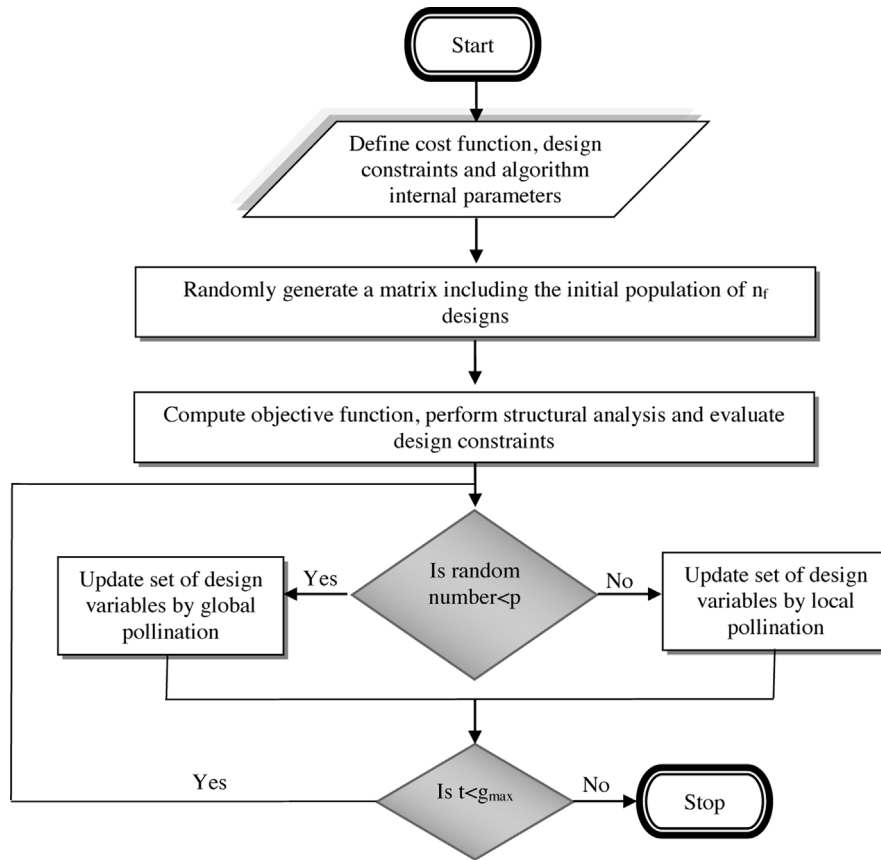


Fig. 1. Flowchart of the proposed optimization algorithm.

In the generation of the system stiffness matrix (K), different design variables are found in the same rows. The optimization problem is coupled since multiple design variables are found in the equilibrium equations found in each row. Without assuming these design variables, it is not possible to obtain nodal displacements according to Eq. (1) and stresses according to Eq. (8). These values may be taken as the limits of design constraints as defined by $g_1(X)$ and $g_2(X)$, but the optimum solution never corresponds to the situation where all nodal displacements and bar stresses are maximized. Also, when all displacements are at the limits, bar stresses may exceed the limit. The structures may be excited by different loads for different time spans. Because of the interaction of bars, the maximum stresses change when design variables are changed according to different loading cases. The number of design variables may be too large for big structural systems. All such reasons may prevent us to apply mathematical optimization methods for structural design problems. In this situation, metaheuristic methods using randomly generated design variables are good alternatives and they are suitable for solving such tasks. Thus, the coupled equilibrium equation can be solved. By conducting iterative analyses, optimum design variables can be found.

3. Flower Pollination Algorithm (FPA)

The Flower Pollination Algorithm (FPA) developed by Yang [7] is a recent metaheuristic algorithm which imitates the flow pollination process of flowering plants. This algorithm was also applied to multi-objective optimization problems by Yang et al. [36]. In FPA, four different rules about the characteristic of the pollination process, pollination behavior and flower constancy are considered.

First rule: Biotic pollination is the cross pollination process in which pollen is carried by a pollinator. The pollinators may be an

insect or animal and about 90% of pollinations of flower plants belong to this type. This situation is a global pollination process and can be realized in the imitation of global search. Additionally, the movement of pollinators obeys Lévy flights.

Second rule: Abiotic or self-pollination is the process done by the same plant or flower without pollinators. It is fertilization of one flower like peach flowers. The carrying process of pollen is generally done with the help of environmental factors such as wind and diffusion in the water. Since the pollen carrying distance is usually shorter than biotic pollination, this process is known as local pollination. In FPA, this process is the source of local search.

Third rule: Pollinators can develop flower constancy which is the tendency of pollinators to specific flower species. Flower constancy is equivalent to a reproduction probability. This probability is proportional to the similarity of the involved flowers.

Fourth rule: A switch probability; $p \in [0, 1]$ is used to control the type of pollination (local or global).

According to these rules, two different search techniques (local and global) can be used. By applying local search, the best solutions around existing ones are found. In addition to that, global pollination essentially prevents to be trapped at a local optimum solution.

The methodology of truss optimization problem can be explained in the following three steps.

First step: In this step, the design constant, ranges of design variables and limits of design constraints that are mentioned in the second section for the truss structure optimization problem are defined. Additionally, the algorithm constants such as the number of flowers (n_r), switch probability (p) and maximum number of generations (g_{max}) are defined.

Second step: Before the local and global search steps, design variables are randomly initialized according to the number of

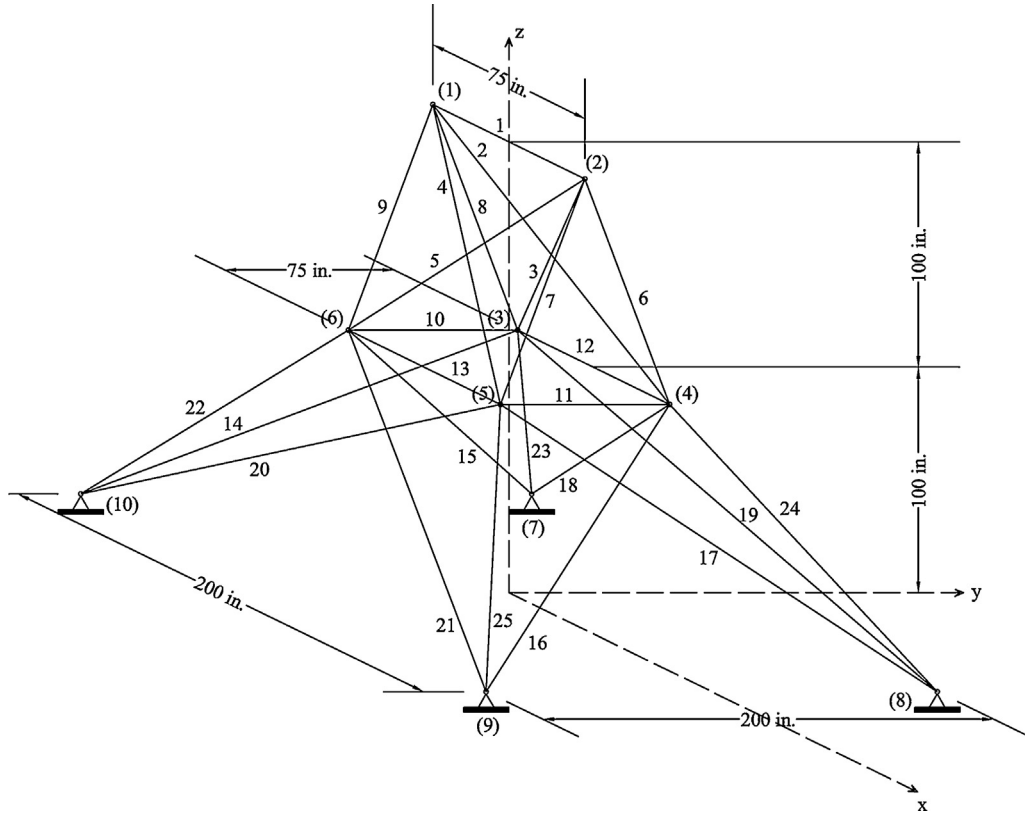


Fig. 2. Schematic of the spatial 25-bar truss structure [26].

flowers and these variables are stored in a solution matrix (*SM*). Each column of this matrix is a solution set of design variables within the defined range. Each variable can be taken as a pollen vector of a flower. This means that we have an $n_f \times N$ solution matrix:

$$SM = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n_f} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n_f} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n-1,1} & A_{n-1,2} & \dots & A_{n-1,n_f} \\ A_{n,1} & A_{n,2} & \dots & A_{n,n_f} \end{bmatrix} \quad (9)$$

For example in $A_{a,b}$, a and b represent the number a th bar (pollen) and b th random solution set (flower). After the design variables are defined, structural analyses are done and the objective function given in Eq. (4) is calculated together with the stress and displacement values in order to consider design constraints. The iterative constraint handling strategy described in Section 3.3 is applied in the proposed method.

Third step: In this step, the solution matrix is iteratively updated according to the local and global search steps. The updating process for local and global search (pollination) are described in Sections 3.1 and 3.2, respectively. For each iteration, a random number between [0,1] is generated and if this random number is smaller than p , then the global search is carried out. Otherwise, local search is performed. During this search process, a new set of design variable values is generated and corresponding analyses are also carried out. The updating of the solution matrix is done according to constraint violation or the objective function. If the design constraints are violated for any set of variables, the updating process is handled by the constraint handling strategy. This strategy is explained in Section 3.3. If the newly generated solutions are better in terms

of the objective function, the solution matrix is updated. This comparison is done for each set of variables. If the design constraints are violated for any set of variables, the updating process is handled by the constraint handling strategy. The iterative process in the third step is repeated until the iteration number reaches g_{max} .

After these three steps, as they continue in an iterative manner, it is expected that the optimum solution with the minimum objective function value can be found. The flowchart of the method is given in Fig. 1.

3.1. Global search (pollination)

The first and third rules for flower pollination are used together in the global search process. In this process, the set of solution for f th flower in the next iteration ($A_{1-N,f}^{t+1}$) is found according to a set of variable of the previous iteration ($A_{1-N,f}^t$) and the current best set of variables (g^*) with the smallest objective function value. Also, the modifications of the set of variables are done according to a Lévy distribution and the Lévy distribution of the f th flower (L_{Df}) is known as the strength of the pollination. The updating of the set of variables for f th flower is formulized as

$$A_{1-N,f}^{t+1} = A_{1-N,f}^t + L_{Df}(g^* - A_{1-N,f}^t). \quad (10)$$

The Lévy distribution (L_{Df}) with an exponent of 1.5 is approximated by

$$L_{Df} = \frac{1}{\sqrt{2\pi}} e^{-1.5} e^{\frac{-1}{2\pi}} \quad (11)$$

where L_{Df} is a random number between [0,1].

Table 1

Member grouping and corresponding stress limits for the spatial 25-bar truss structure.

Element group	Members	Compression (ksi)	Tension (ksi)
1	1	35.092	35
2	2–5	11.590	35
3	6–9	17.305	35
4	10,11	35.092	35
5	12,13	35.092	35
6	14–17	6.759	35
7	18–21	6.959	35
8	22–25	11.082	35

Table 2

Multiple loading conditions for the 25-bar truss.

Case	Node	P_x (kip)	P_y (kip)	P_z (kip)
1	1	1.0	10.0	–5.0
	2	0.0	10.0	–5.0
	3	0.5	0.0	0.0
	6	0.0	0.0	0.0
2	1	0.0	20.0	–5.0
	2	0.0	–20.0	–5.0

3.2. Local search (pollination)

In this type of pollination, the second rule of flower pollination is used together with the flower constancy (the third rule). The solution set of next iteration is updated according to the solutions of two flowers in the previous iteration. If $A_{1-N,m}^t$ and $A_{1-N,k}^t$ are the set of variables for m th and k th flowers, respectively, then the solution can be modified by

$$A_{1-N,f}^{t+1} = A_{1-N,f}^t + \varepsilon(A_{1-N,m}^t - A_{1-N,k}^t). \quad (12)$$

for the f th set of solution. ε is a random number between [0,1] in order to apply linear distribution in self-pollination. The m th and k th flowers are randomly chosen in the optimization process with no specific selection rule. Further research will analyze the effect of using elitist or other selection rules on convergence behavior.

3.3. Iterative strategy for constraint handling

In order to reach a precise optimum solution, the region around values which may slightly violate the design constraints must also be scanned. In addition to that, the set of design variables must not violate any design constraint or limit. Therefore, a special violation handling strategy is used in the proposed method. The constraint handling strategy is an iterative stage. First, a certain amount of constraint violation is allowed and then the value of this violation is getting smaller. At the end of the optimization process, the value of constraint violation should be zero. The constraint handling process can be explained in two stages.

Stage 1: Initially, a certain percentage of constraint violation ($a\%$) is allowed and it is linearly decreased for each iteration until t_1 th iteration. In the updating process of variables, if the constraint violation is more than the allowed value, the set of variables with the minimum constraint violation is accepted. If the constraint violation is less than the allowed, the best solution with the best objective is updated. However, the new design may be closer to the constraint domain boundary and special cares have to be taken in order not to end up in the infeasible region in the next iterations.

Stage 2: In order to prevent constraint violation in the search of an effective solution region, a slight violation percentage ($b\%$) is allowed after t_1 th iteration until t_2 th iteration and the allowed violation is iteratively decreased. The comparison of design variables is done by the same procedure of the first stage. After the

Table 3
Optimization results and comparison with literature for the 25-bar truss problem.

Group	GA [10]	ACO [13]	HPSO [17]	BB-BC [16]	CMLPSA [20]	HBB-BC [21]	ABC-AP [24]	TLBO [27]	HPSSO [28]	CBO [33]	Present study
1	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0110	0.0100	0.0100	0.0100	0.0100
2	2.0119	2.0000	1.9700	2.0920	1.9870	1.9930	1.9790	1.9878	1.9907	2.1297	1.8308
3	2.9493	2.9660	3.0160	2.9640	2.9935	3.0560	3.0030	2.9914	2.9881	2.8865	3.1834
4	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0102	0.0100	0.0100	0.0100
5	0.0295	0.0120	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
6	0.6838	0.6890	0.6940	0.6890	0.6840	0.6650	0.6900	0.6828	0.6824	0.6792	0.7017
7	1.6798	1.6790	1.6810	1.6010	1.6769	1.6420	1.6790	1.6775	1.6764	1.6077	1.7266
8	2.6759	2.6680	2.6430	2.6860	2.6621	2.6790	2.6520	2.6640	2.6656	2.6927	2.5713
Best weight (lb)	545.800	545.530	545.190	545.380	545.163	545.160	545.190	545.175	545.1640	544.31	545.159
Average weight (lb)	–	546.340	–	545.780	–	545.660	–	545.483	–	545.256	545.730
Standard deviation on optimized weight (lb)	–	0.94	–	0.491	–	0.367	–	0.306	–	0.294	0.59
Number of structural analyses	–	16,500	125,000	20,566	400	12,500	500,000	12,199	13,326	9090	8149

Table 4
Sensitivity of the optimized weight of the 25-bar truss structure to population size and limit number of iterations.

l/n	5	10	15	20	25	30
100	638.6077	604.8591	576.8526	585.4082	596.4658	588.3232
250	592.091	559.9247	560.0212	555.998	569.8168	567.2785
500	572.9288	559.9247	559.7946	554.8575	558.9415	556.6689
1000	553.2085	554.469	556.7446	553.0837	554.9815	556.3888
1500	553.2085	550.8044	556.7446	553.0837	547.3405	551.1369
2000	553.2085	550.8044	551.7404	551.1642	547.3405	548.4778

Table 5
Sensitivity of the optimized weight of the 25-bar truss structure to constraint tolerance parameters a and b , and numbers of iterations t_1 and t_2 .

Case	a	b	t_1	t_2	Optimum weight
1	1%	0.1%	1000	9000	546.7112
2	0.1%	0%	1000	9000	667.6402
3	0.5%	0.01%	500	5000	547.1570

Table 6
Multiple load cases for 72-bar truss.

Case	Node	P_x (kip)	P_y (kip)	P_z (kip)
1	17–20	–5.0	–5.0	–5.0
2	17	5.0	5.0	–5.0

t_2 th iteration, constraint violation is prevented. Also, the violation of constraints can be linearly decreased to zero for t_2 th iteration.

4. Test problems and optimization results

The proposed optimization algorithm was tested against three classical weight minimization problems of a spatial 25-bar truss, a spatial 72-bar truss and a planar 200-bar truss, respectively. Each structure must be designed against multiple loading conditions. The last test case was selected in order to show the efficiency of the present approach for fairly large structural systems. Optimization results were compared with results available in the literature. Internal parameters were set as follows: 30, 0.5, 10,000 (100,000 for the third problem), 0.5%, 0.01%, 1000 and 9000 (90,000 for the third problem) for n , p , g_{max} , a , b , t_1 and t_2 , respectively. The statistical results were obtained after 50 independent runs of the optimization process were carried out.

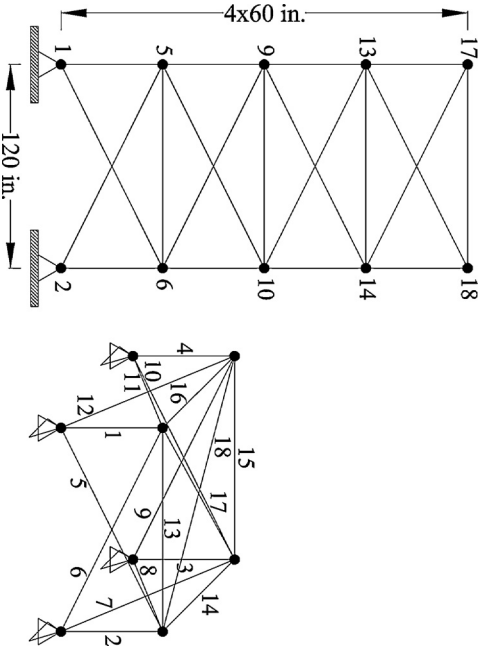


Fig. 3. Schematic of the spatial 72-bar truss structure [27].

Table 7
Optimization results and comparison with literature for the 72-bar truss problem.

Element group	Members	GGP [8]	GA [10]	GA [11]	ACO [13]	PSO [18]	BB-BC [16]	BB-BC [21]	SAHS [25]	TLBO [27]	CSP [29]	CBO [33]	ECBO [34]	RO [35]	Present study
1	1–4	2.0259	1.8562	1.7550	1.9480	1.7430	1.8577	1.9042	1.8600	1.8807	1.9446	1.90280	1.8519	1.8365	1.8758
2	5–12	0.5332	0.4933	0.5050	0.5080	0.5180	0.5059	0.5162	0.5210	0.5142	0.5026	0.51800	0.5141	0.5021	0.5160
3	13–16	0.1000	0.1000	0.1050	0.1010	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.10010	0.1000	0.1000	0.1000
4	17–18	0.1000	0.1000	0.1550	0.1020	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.10030	0.1000	0.1004	0.1000
5	19–22	1.1567	1.2830	1.1550	1.3030	1.3080	1.2476	1.2582	1.2710	1.2711	1.2676	1.27870	1.2819	1.2522	1.2993
6	23–30	0.5689	0.5028	0.5850	0.5110	0.5190	0.5269	0.5035	0.5090	0.5151	0.5099	0.50740	0.5091	0.5033	0.5246
7	31–34	0.1000	0.1000	0.1000	0.1010	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.10030	0.1000	0.1002	0.1001
8	35–36	0.1000	0.1000	0.1000	0.1000	0.1000	0.1012	0.1000	0.1000	0.1000	0.1000	0.10030	0.1000	0.1002	0.1000
9	37–40	0.5137	0.5177	0.4600	0.5610	0.5140	0.5209	0.5178	0.4850	0.5317	0.5067	0.52400	0.5312	0.5730	0.4971
10	41–48	0.4791	0.5227	0.5300	0.4920	0.5460	0.5172	0.5214	0.5010	0.5134	0.5165	0.51500	0.5173	0.5499	0.5089
11	49–52	0.1000	0.1000	0.1200	0.1000	0.1000	0.1004	0.1000	0.1000	0.1000	0.1075	0.10020	0.1000	0.1004	0.1000
12	53–54	0.1000	0.1049	0.1650	0.1070	0.1090	0.1005	0.1007	0.1000	0.1000	0.1000	0.10150	0.1000	0.1001	0.1000
13	55–58	0.1579	0.1557	0.1550	0.1560	0.1610	0.1565	0.1566	0.1680	0.1565	0.1562	0.15640	0.1560	0.1576	0.1575
14	59–66	0.5501	0.5501	0.5350	0.5500	0.5090	0.5507	0.5421	0.5840	0.5429	0.5402	0.54940	0.5572	0.5222	0.5329
15	67–70	0.3449	0.3981	0.4800	0.3900	0.4970	0.3922	0.4132	0.4330	0.4081	0.4223	0.40290	0.4259	0.4356	0.4089
16	71–72	0.4984	0.6749	0.5200	0.5920	0.5620	0.5922	0.5756	0.5200	0.5733	0.5794	0.55040	0.5271	0.5972	0.5731
Best weight (lb)		379.310	380.320	383.120	380.240	381.910	379.850	379.660	380.620	379.632	379.970	379.6943	379.77	380.458	379.095
Average weight (lb)		–	–	–	383.160	–	382.080	381.850	–	379.759	381.560	379.8961	380.39	382.5538	379.534
Standard deviation on optimized weight (lb)		–	–	–	3.66	–	1.912	1.201	–	0.149	1.803	0.0791	0.8099	1.2211	0.272
Number of structural analyses		–	15,000	–	18,500	8000	19,679	13,200	13,742	21,542	10,500	15,600	–	19,084	9029

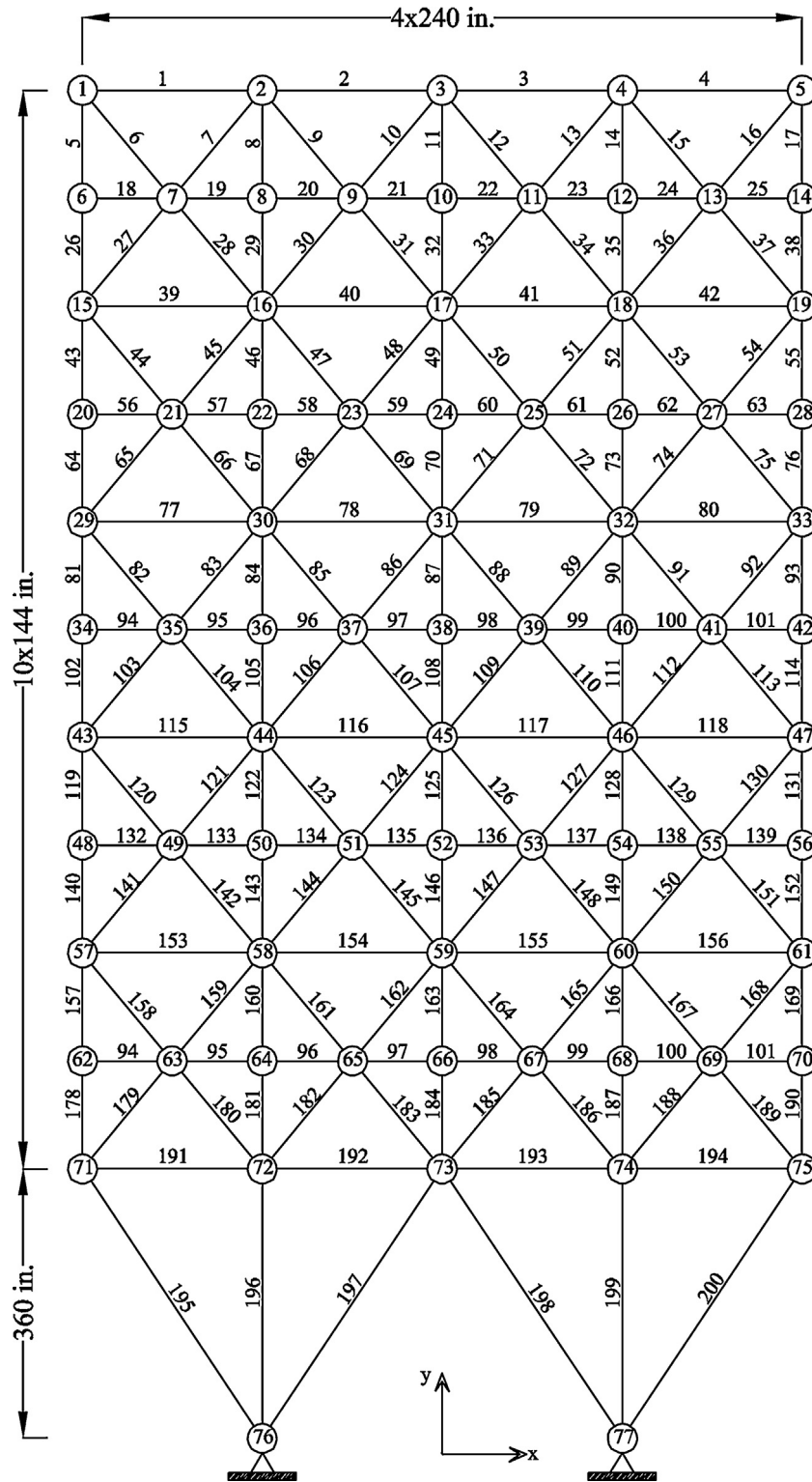


Fig. 4. Schematic of the planar 200-bar truss structure [26].

4.1. Spatial 25-bar truss

The first test case concerns the spatial 25-bar truss structure shown in Fig. 2. Elements are categorized in eight groups (see Table 1); hence, this problem includes eight sizing variables. The elasticity modulus of the material is 10 Msi and the density of the material is 0.1 lb/in.³. The cross sectional areas of bars can vary

between 0.01 and 3.4 in.². The structure must be designed for the two independent loading conditions listed in Table 2. The allowable displacement for each node in all coordinate directions is ± 0.35 in. Stress limits for compression and tension are given in Table 1 for all member groups. The optimized design of FPA weighs 545.159 lb, which is nearly equal to the best result quoted in the literature (see Table 3).

Table 8

Member grouping for the planar 200-bar truss structure.

Element group	Members	Element group	Members
1	1, 2, 3, 4	16	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
2	5, 8, 11, 14, 17	17	115, 116, 117, 118
3	19, 20, 21, 22, 23, 24	18	119, 122, 125, 128, 131
4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177	19	133, 134, 135, 136, 137, 138
5	26, 29, 32, 35, 38	20	140, 143, 146, 149, 152
6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34, 36, 37	21	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151
7	39, 40, 41, 42	22	153, 154, 155, 156
8	43, 46, 49, 52, 55	23	157, 160, 163, 166, 169
9	57, 58, 59, 60, 61, 62	24	171, 172, 173, 174, 175, 176
10	64, 67, 70, 73, 76	25	178, 181, 184, 187, 190
11	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75	26	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
12	77, 78, 79, 80	27	191, 192, 193, 194
13	81, 84, 87, 90, 93	28	195, 17, 198, 200
14	95, 96, 97, 98, 99, 100	29	196, 199
15	102, 105, 108, 111, 114		

Table 4 shows the trend of variation of the optimized weight for the spatial 25-bar truss when different combinations of population sizes and the maximum numbers of iterations are selected. It can be seen that increasing population has a positive effect after 1000th iteration. Using a small population size (e.g. 5 flowers) does not yield good results, especially for fewer iterations.

Sensitivity of FPA to constraint tolerance parameters α and b and numbers of iterations t_1 and t_2 was also investigated. For that

purpose, three combinations of these parameters were selected (see Table 5). While feasible designs could always be obtained by FPA, structural weights were pretty far from the optimum value quoted in Table 3. In the first case, the allowed constraint violation is smaller than the limit indicated in Section 4. Since constraint violation must be reduced to zero, lightweight feasible designs are eliminated from the search process. This behavior became more evident in the second case as no constraint violation is allowed

Table 9

Optimization results and comparison with literature for the 200-bar truss problem.

Element group	HS [14]	GA [19]	CMLPSA [20]	HPSACO [23]	SAHS [25]	TLBO [26]	CSP [29]	HPSSO [28]	TLBO [30]	Present study
1	0.1253	0.3469	0.1468	0.1033	0.1540	0.1460	0.1480	0.1213	0.1135	0.1425
2	1.0157	1.0810	0.9400	0.9184	0.9410	0.9410	0.9460	0.9426	0.9484	0.9637
3	0.1069	0.1000	0.1000	0.1202	0.1000	0.1000	0.1010	0.1220	0.1078	0.1005
4	0.1096	0.1000	0.1000	0.1009	0.1000	0.1010	0.1010	0.1000	0.1000	0.1000
5	1.9369	2.1421	1.9400	1.8664	1.9420	1.9410	1.9461	2.0143	1.9345	1.9514
6	0.2686	0.3470	0.2962	0.2826	0.3010	0.2960	0.2979	0.2800	0.2889	0.2957
7	0.1042	0.1000	0.1000	0.1000	0.1000	0.1000	0.1010	0.1589	0.2116	0.1156
8	2.9731	3.5650	3.1042	2.9683	3.1080	3.1210	3.1072	3.0666	3.0903	3.1133
9	0.1309	0.3470	0.1000	0.1000	0.1000	0.1000	0.1010	0.1002	0.1031	0.1006
10	4.1831	4.8050	4.1042	3.9456	4.1060	4.1730	4.1062	4.0418	4.0903	4.1100
11	0.3967	0.4400	0.4034	0.3742	0.4090	0.4010	0.4049	0.4142	0.4502	0.4165
12	0.4416	0.4400	0.1912	0.4501	0.1910	0.1810	0.1944	0.4852	0.1007	0.1843
13	5.1873	5.9520	5.4284	4.9603	5.4280	5.4230	5.4299	5.4196	5.4798	5.4567
14	0.1912	0.3470	0.1000	1.0738	0.1000	0.1000	0.1010	0.1000	0.1011	0.1000
15	6.2410	6.5720	6.4284	5.9785	6.4270	6.4220	6.4299	6.3749	6.4798	6.4559
16	0.6994	0.9540	0.5734	0.7863	0.5810	0.5710	0.5755	0.6813	0.5329	0.5800
17	0.1158	0.3470	0.1327	0.7374	0.1510	0.1560	0.1349	0.1576	0.1325	0.1547
18	7.7643	8.5250	7.9717	7.3809	7.9730	7.9580	7.9747	8.1447	7.9445	8.0132
19	0.1000	0.1000	0.1000	0.6674	0.1000	0.1000	0.1010	0.1000	0.1005	0.1000
20	8.8279	9.3000	8.9717	8.3000	8.9740	8.9580	8.9747	9.0920	8.9444	9.0135
21	0.6986	0.9540	0.7049	1.1967	0.7190	0.7200	0.7065	0.7462	0.7011	0.7391
22	1.5563	1.7639	0.4196	1.0000	0.4220	0.4780	0.4225	0.2114	1.3777	0.7870
23	10.9806	13.3006	10.8636	10.8262	10.8920	10.8970	10.8685	10.9587	11.2394	11.1795
24	0.1317	0.3470	0.1000	0.1000	0.1000	0.1000	0.1010	0.1000	0.2287	0.1462
25	12.1492	13.3006	11.8606	11.6976	11.8870	11.8970	11.8684	11.9832	12.2394	12.1799
26	1.6373	2.1421	1.0339	1.3880	1.0400	1.0800	1.0360	0.9241	1.6849	1.3424
27	5.0032	4.8050	6.6818	4.9523	6.6460	6.4620	6.6859	6.7676	4.9136	5.4844
28	9.3545	9.3000	10.8113	8.8000	10.8040	10.7990	10.8111	10.9639	9.7190	10.1372
29	15.0919	17.1740	13.8404	14.6645	13.8700	13.9220	13.8465	13.8186	15.0219	14.5262
Best weight (lb)	25,447.1	28,544.0	25,445.63	25,156.5	25,491.9	25,488.2	25,467.90	25,698.85	25,664.00	25,521.81
Average weight (lb)	–	–	–	–	–	–	25,547.60	28,386.72	–	25,543.51
Standard deviation on optimized weight (lb)	–	–	–	–	–	–	135.09	2403	–	18.13
Number of structural analyses	48,000	–	9650	9875	19,670	28,059	31,700	14,406	–	10,685

after 1000th iteration. In the last case, reducing the number of iterations t_1 and t_2 also was ineffective as it led to miss designs with small constraint violations. In summary, the present FPA algorithm is very sensitive to constraint tolerance and number of iteration parameters which must be carefully selected in order to optimize convergence behavior.

4.2. Spatial 72-bar truss

The second test problem is the spatial 72-bar truss structure shown in Fig. 3. The elasticity modulus and density of the material are the same as those in the first example. The maximum displacement of joints is ± 0.25 in for all coordinate directions. The same stress limit in tension and compression is set for all elements, ± 25 ksi. Cross sectional areas of elements can vary between 0.1 and 3.0 in.². The structure must be designed for the two independent loads as listed in Table 6.

The truss system is divided into 16 sizing variables. The corresponding group numbers for structural members are shown in Table 7, together with the optimization results of FPA and other state-of-the-art metaheuristic optimization algorithms. Overall, the present algorithm found the best design within the prescribed constraint tolerance.

4.3. Planar 200-bar truss

The last test problem regards the weight minimization of a fairly large planar truss structure including 200 elements and 77 nodes. The model of the structure is schematized in Fig. 4. The elasticity modulus and density of the material are taken as 30 Msi and 0.283 lb/in³, respectively. Only stress constraints are considered for this problem: the limit stress is the same in tension and compression for all members (± 25 ksi). Cross sectional areas of elements can vary between 0.1 and 20 in.². Member grouping leads to have 29 sizing variables (see Table 8). The structure is subject to three independent load cases:

- (i) +1 kip load in X-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62 and 71;
- (ii) –10 kip load in Y-direction at nodes 1–6, 8, 10, 12, 14–20, 22, 24, 26, 28–34, 36, 38, 40, 42–48, 50, 52, 54, 56–62, 64, 66, 68, 70–75;
- (iii) combination of first and second loading conditions.

Table 9 shows that the present approach is very competitive in finding optimum designs. Furthermore, the average weight is very close to the best weight and a small standard deviation on optimized weight (lb) is observed.

5. Conclusions

In this study, a structural optimization approach based on the flower pollination algorithm has been formulated, and it has been tested using three classical weight minimization problems of 3D and 2D truss structures. The largest problem consists of 29 sizing variables. The proposed algorithm proved itself to be competitive with other state-of-the-art metaheuristic optimization methods.

Similar to other metaheuristic algorithms, the flower pollination algorithm combines local and global optimization. However, FPA runs these processes in a fairly different way. In global search, the best sets of variables are used in the generation of new variables while the best sets of variables are generally considered in the local optimization part of the algorithm. The global optimization is sustained by using Lévy distribution. Since the best sets of variables are used, the convergence of the FPA to the optimum values is effective

compared with the algorithms using a simply randomization process for global optimization. In local optimization, the values of two sets of design variables are used. These sets are randomly chosen and this procedure allows not to be entrapped in a local optimum.

According to the sensitivity analysis carried out for the 25-bar truss structure, increasing population has a positive effect after 1000th iteration and FPA does not yield good results for small number of flowers, which is also true for many population-based algorithms because a small population does not provide sufficient diversity. Therefore, a moderate population size should be used. Additionally, the computational cost of the algorithm seems suitable. The number of structural analyses required in the optimization process are 1618, 422 and 2088, respectively, for examples 1–3, in order to find a feasible trial design only 1% heavier than the target optimum.

The FPA algorithm developed in this study is also combined with an iterative constraint handling strategy with the purpose of finding optimum designs that do not violate any constraint. By scanning trial designs with a small amount of constraint violation, it may be possible to generate new feasible solutions as the iterations proceed. This makes the proposed approach well suited for finding optimum cross-sectional areas of structural members.

Although the present results are encouraging, there is still room for improvement. Future studies must focus on the detailed parametric study so as to find the best parameter setting for the algorithms so that they can solve a wider range of structural optimization problems. In addition, even large-scale problems with more design variables and more complex constraints can be attempted. Furthermore, it would be also very useful to use a multiobjective approach to structural optimization in real-world applications.

References

- [1] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [2] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation* 76 (2001) 60–68.
- [3] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [4] X.S. Yang, S. Deb, Cuckoo search: recent advances and applications, *Neural Comput. Appl.* 24 (1) (2014) 169–174.
- [5] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks No. IV*, Perth, Australia, November 27–December 1, 1995, pp. 1942–1948.
- [6] M. Dorigo, V. Maniezzo, A. Colomi, The ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern.* 26 (1996) 29–41.
- [7] X.S. Yang, Flower pollination algorithm for global optimization, *Unconven. Comput. Nat. Comput.* (2012) 240–249.
- [8] H. Adeli, O. Kamal, Efficient optimization of space trusses, *Comput. Struct.* 24 (3) (1986) 501–511.
- [9] S. Rajeev, C.S. Krishnamoorthy, Discrete optimization of structures using genetic algorithms, *J. Struct. Eng.* 118 (5) (1992) 1233–1250.
- [10] G. Cao, *Optimized design of framed structures using a genetic algorithm*, Ph.D. thesis, The University of Memphis, 1996.
- [11] F. Erbatur, O. Hasançebi, I. Tütüncü, H. Kiliç, Optimal design of planar and space structures with genetic algorithms, *Comput. Struct.* 75 (2000) 209–224.
- [12] J.J. Schutte, A.A. Groenwold, Sizing design of truss structures using particle swarms, *Struct. Multidiscip. Optim.* 25 (2003) 261–269.
- [13] C.V. Camp, B.J. Bichon, Design of space trusses using ant colony optimization, *J. Struct. Eng.* 130 (5) (2004) 741–751.
- [14] K.S. Lee, Z.W. Geem, A new structural optimization method based on the harmony search algorithm, *Comput. Struct.* 82 (2004) 781–798.
- [15] O.K. Erol, I.I. Eksin, A new optimization method: big bang big crunch, *Adv. Eng. Softw.* 37 (2006) 106–111.
- [16] C.V. Camp, Design of space trusses using big bang–big crunch optimization, *J. Struct. Eng.* 133 (7) (2007) 999–1008.
- [17] L.J. Li, Z.B. Huang, F. Liu, Q.H. Wu, A heuristic particle swarm optimizer for optimization of pin connected structures, *Comput. Struct.* 85 (7–8) (2007) 340–349.
- [18] R.E. Perez, K. Behdinan, Particle swarm approach for structural design optimization, *Comput. Struct.* 85 (2007) 1579–1588.
- [19] V. Togan, A.T. Daloglu, An improved genetic algorithm with initial population strategy and self-adaptive member grouping, *Comput. Struct.* 86 (2008) 1204–1218.

- [20] L. Lamberti, An efficient simulated annealing algorithm for design optimization of truss structures, *Comput. Struct.* 86 (2008) 1936–1953.
- [21] A. Kaveh, S. Talatahari, Size optimization of space trusses using big bang–big crunch algorithm, *Comput. Struct.* 87 (17–18) (2009) 1129–1140.
- [22] A. Kaveh, S. Talatahari, A particle swarm ant colony optimization for truss structures with discrete variables, *J. Constr. Steel Res.* 65 (2009) 1558–1568.
- [23] A. Kaveh, S. Talatahari, Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Comput. Struct.* 87 (5–6) (2009) 267–283.
- [24] M. Sonmez, Artificial bee colony algorithm for optimization of truss structures, *Appl. Soft Comput.* 11 (2) (2011) 2406–2418.
- [25] S.O. Degertekin, Improved harmony search algorithms for sizing optimization of truss structures, *Comput. Struct.* 92–93 (2012) 229–241.
- [26] S.O. Degertekin, M.S. Hayalioğlu, Sizing truss structures using teaching–learning-based optimization, *Comput. Struct.* 119 (2013) 177–188.
- [27] C.V. Camp, M. Farshchin, Design of space trusses using modified teaching–learning based optimization, *Eng. Struct.* 62–63 (2014) 87–97.
- [28] A. Kaveh, T. Bakhshpoori, E. Afshari, An efficient hybrid Particle Swarm and Swallow Swarm Optimization algorithm, *Comput. Struct.* 143 (2014) 40–59.
- [29] A. Kaveh, R. Sheikholeslami, S. Talatahari, M. Keshvari-Ilkhichi, Chaotic swarming of particles: a new method for size optimization of truss structures, *Adv. Eng. Softw.* 67 (2014) 136–147.
- [30] T. Dede, Y. Ayvaz, Combined size and shape optimization of structures with a new meta-heuristic algorithm, *Appl. Soft Comput.* 28 (2015) 250–258.
- [31] A. Kaveh, B. Mirzaei, A. Jafarvand, An improved magnetic charged system search for optimization of truss structures with continuous and discrete variables, *Appl. Soft Comput.* 28 (2015) 400–410.
- [32] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, *Comput. Struct.* 139 (2014) 18–27.
- [33] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization method for optimum design of truss structures with continuous variables, *Adv. Eng. Softw.* 70 (2014) 1–12.
- [34] A. Kaveh, M. Ilchi Ghazaan, Enhanced colliding bodies optimization for design problems with continuous and discrete variables, *Adv. Eng. Softw.* 77 (2014) 66–75.
- [35] A. Kaveh, M. Khayatizad, Ray optimization for size and shape optimization of truss structures, *Comput. Struct.* 117 (2013) 82–94.
- [36] X.S. Yang, M. Karamanoglu, X. He, Flower pollination algorithm: a novel approach for multi objective optimization, *Eng. Optim.* 46 (9) (2014) 1222–1237.