

Metaheuristic Optimization in Structural Engineering

S.O. Degertekin and Zong Woo Geem

Abstract Metaheuristic search methods have been extensively used for optimization of the structures over the past two decades. Genetic algorithms (GA), ant colony optimization (ACO), particle swarm optimization (PSO), harmony search (HS), big bang-big crunch (BB-BC), artificial bee colony algorithm (ABC) and teaching-learning-based optimization (TLBO) are the most popular metaheuristic optimization methods. The basic principle of these methods is that they make an analogy between the natural phenomena and the optimization problems. In this chapter, recently developed metaheuristic optimization methods such as self-adaptive harmony search and teaching-learning-based optimization are reviewed and the performance of these methods in the field of structural engineering are compared with each other and the other metaheuristic methods.

Keywords Metaheuristic optimization · Structural engineering · Truss structures

1 Introduction

A family of metaheuristic optimization methods based on swarm intelligence have been developed in the past two decades. These methods simulate the behaviour of different groups/swarms/colonies of animals and insects. Ant colony optimization (ACO), harmony search (HS), particle swarm optimization (PSO), big bang-big crunch optimization (BB-BC) and artificial bee colony optimization (ABC), teaching-learning-based optimization (TLBO) are few examples of recent metaheuristic

S.O. Degertekin
Department of Civil Engineering, Dicle University,
21280 Diyarbakir, Turkey

Z.W. Geem (✉)
Department of Energy IT, Gachon University,
Seongnam 461-701, South Korea
e-mail: zwgeem@gmail.com

algorithms. They also are classified as population-based or nature-inspired optimization methods. The main philosophy of all metaheuristic optimization methods is the mimicking of the natural phenomenon. Design optimization of skeletal structures using metaheuristic search methods is an important field of engineering under continuous development. The state-of-the-art utilization of metaheuristic algorithms in weight or cost optimization of skeletal structures have been recently reviewed by Lamberti and Pappalettere [1] and Saka [2].

The ACO was originally proposed by Dorigo et al. [3] for optimization problems. The method simulates the foraging behaviour of real-life ant colonies. The ACO attempts to model some of the fundamental capabilities observed in the behaviour of ants as a method stochastic combinatorial optimization [4]. In addition to its different applications, the method has also been used for design optimization of structural systems. ACO was used for optimization of truss structures by Camp and Bichon [5], Capriles et al. [6], Serra and Venini [7], and Hasancebi et al. [8], and frame structures by Camp et al. [4], Kaveh and Shojaee [9], Hasancebi et al. [10], Kaveh and Talatahari [11].

HS was first developed by Geem et al. [12] for solving combinatorial optimization problems. The method bases on the analogy between the musical process of searching for a perfect state of harmony and searching for solutions to optimization problems. HS has been used for a variety of structural optimization problems including optimum design of truss structures [8, 13, 14], geodesic domes [15], grillage systems [16] and steel frames [17–19].

In recent years, improved/modified HS algorithms have been developed in order to increase the efficiency of the method. Saka and Hasancebi [20] developed an adaptive harmony search algorithm for design code optimization of steel structures. Hasancebi et al. [21] proposed an adaptive harmony search method for structural optimization. Lamberti and Pappalettere [22] proposed an improved harmony-search algorithm, where trial designs are generated including information on the gradients of cost function for truss structure optimization. Two improved harmony search algorithms called efficient harmony-search (EHS) algorithm and self-adaptive harmony-search (SAHS) algorithm were proposed by Degertekin [23] for sizing the optimization of truss structures.

The PSO method was first developed by Kennedy and Eberhart [24]. It is based on the premise that social sharing of information among members of a species offers and evolutionary advantage [25]. PSO has been used in optimization of skeletal structures [26–29]. Researchers introduced new features in the standard implementation of PSO. Li et al. [28, 29] developed a heuristic particle swarm optimizer (HPSO), which combines the PSO scheme and the HS scheme, for sizing optimization of truss structures. Kaveh and Talatahari [30, 31] introduced a particle swarmer, ant colony optimization and harmony search scheme for truss structures with both discrete [30] and continuous variables [31].

The BB-BC proposed by Erol and Eksin [32] simulates the theories of the evolution of the universe. According to this theory, in the big bang phase energy dissipation produces disorder, and randomness is the main feature of this phase; whereas, in the big crunch phase, randomly distributed particles are drawn into an

order [33]. BB-BC algorithm was applied for sizing optimization of truss structures [34]. In order to improve the convergence capability of standard BB-BC algorithm, Kaveh and Talatahari [33, 35] developed hybrid BB-BC (HBB-BC) algorithm to optimize space trusses and ribbed domes.

The ABC method was first developed by Karaboga [36] for numerical function optimization. The ABC is an optimization method based on the intelligent behaviour of honey bee swarm. The ABC has successfully been applied to the size optimization of truss structures with both continuous [37] and discrete variables [38].

Another metaheuristic method called ‘teaching-learning-based optimization (TLBO)’ has been proposed by Rao et al. [39] for constrained mechanical design optimization problems. The method bases on the effect of influence of a teacher on learners and the effect of learners with each other. Rao et al. [40] developed the TLBO method for large-scale nonlinear optimization problems for finding global solutions. TLBO was employed for optimum design of planar steel frames [41] and sizing optimization of truss structures Degertekin and Hayalioglu [42].

In this chapter, the robustness of the SAHS [23] and TLBO [42] will be investigated in the optimization of truss type structures. Three benchmark truss structures existing in the current literature are presented to test the efficiency of the methods. The results obtained from these methods will be compared with those of other metaheuristic optimization algorithms recently presented in the literatures like particle swarm optimization (PSO), heuristic particle swarm optimizer (HPSO), hybrid particle swarm optimization (HPSO), big bang-big crunch optimization (BB-BC), heuristic particle swarm ant colony optimization (HPSACO), hybrid big bang-big crunch optimization (HBB-BC), artificial bee colony optimization (ABC-AP) and improved harmony search algorithm (IHS).

The rest of this study is organized as follows. The formulation of optimum design problem is given in Sect. 2. SAHS and TLBO methods are explained in Sects. 3 and 4. The results obtained from the design examples are presented and compared with other metaheuristic optimization methods in Sect. 5. Finally, conclusions are presented in Sect. 6.

2 Formulation of Optimum Design Problem

The minimum weight design problem for a truss structure can be formulated as

Find $X = [x_1, x_2, \dots, x_{ng}]$ to minimize

$$W(X) = \sum_{k=1}^{ng} x_k \sum_{i=1}^{mk} \rho_i L_i \quad (1)$$

subject to the following normalized constraints

$$g_{nl}^s(X) = \frac{|\sigma_{nl}|}{|\sigma_{nu}|} - 1 \leq 0, \quad 1 \leq n \leq nm, \quad 1 \leq l \leq nl \quad (2)$$

$$g_{nl}^b(X) = \frac{|\sigma_{cl}|}{|\sigma_{cu}|} - 1 \leq 0, \quad 1 \leq n \leq ncm, \quad 1 \leq l \leq nl \quad (3)$$

$$g_{jl}^d(X) = \frac{|d_{jl}|}{|d_{ju}|} - 1 \leq 0, \quad 1 \leq j \leq nn, \quad 1 \leq l \leq nl \quad (4)$$

$$x_{\min} \leq x_k \leq x_{\max}, \quad k = 1, 2, \dots, ng \quad (5)$$

where X is the vector containing the design variables, $W(X)$ is the weight of the truss structure, ng is the total number of member groups (i.e. design variables), x_k is the cross-sectional area of the members belonging to the group k , mk is the total number of members in the group k , ρ_i is the density of member i , L_i is the length of member i , $g_{nl}^s(X)$, $g_{nl}^b(X)$ and $g_{jl}^d(X)$ are the constraint violations for member stress, member buckling stress and joint displacements of the structure. σ_{nl} and σ_{cl} are the member stress and the member buckling stress of the n th member due to loading condition l , σ_{nu} and σ_{cu} are their upper limits. d_{jl} is the nodal displacement of the j th translational degree of freedom due to loading condition l , d_{ju} is its upper limit. nl is the number of load conditions, nn is the number of nodes, max and min are the upper and lower limits for cross-sectional area.

The optimum design of truss structures must satisfy optimization constraints stated by Eqs. (2)–(5). In this study, the constraints are handled using a modified feasible-based mechanism [30]. The efficiency of the method was previously verified for optimization of truss structures [23, 30]. The method consists of the following four rules [30]:

Rule 1: Any feasible design is preferred to any infeasible design.

Rule 2: Infeasible designs containing slight violation of the constraints (from 0.01 in the first search to 0.001 in the last search) are considered as feasible designs.

Rule 3: Between two feasible designs, the one having the better objective function value is preferred.

Rule 4: Between two infeasible designs, the one having the smaller constraint violation is preferred.

3 Self-Adaptive Harmony Search Algorithm (SAHS)

Harmony-search (HS) algorithm for sizing optimization of the truss structures could be explained in the following steps [23]:

Step 1. Initializing the harmony search parameters

HS parameters are assigned in this step. The number of design vectors in harmony memory (HMS), harmony memory consideration rate (HMCR), pitch adjusting rate (PAR) and the stopping criterion are selected in this step.

Step 2. Initializing harmony memory

All design vectors are stored in the harmony memory (HM). The HM matrix given in Eq. (6) is filled with randomly generated design vectors as the size of the harmony memory (HMS) in this step.

$$\text{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{ng-1}^1 & x_{ng}^1 \\ x_1^2 & x_2^2 & \dots & x_{ng-1}^2 & x_{ng}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{\text{HMS}-1} & x_2^{\text{HMS}-1} & \dots & x_{ng-1}^{\text{HMS}-1} & x_{ng}^{\text{HMS}-1} \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \dots & x_{ng-1}^{\text{HMS}} & x_{ng}^{\text{HMS}} \end{bmatrix} \begin{matrix} \rightarrow W(X^1) \\ \rightarrow W(X^2) \\ \rightarrow \vdots \\ \rightarrow \vdots \\ \rightarrow W(X^{\text{HMS}-1}) \\ \rightarrow W(X^{\text{HMS}}) \end{matrix} \quad (6)$$

In the HM, each row represents a truss design. $X^1, X^2, \dots, X^{\text{HMS}-1}, X^{\text{HMS}}$ and $W(X^1), W(X^2), \dots, W(X^{\text{HMS}-1}), W(X^{\text{HMS}})$ are designs and the corresponding objective function values, respectively. The truss designs in the HM are sorted by their objective function values ($W(X^1) \leq W(X^2) \leq \dots \leq W(X^{\text{HMS}-1}) \leq W(X^{\text{HMS}})$) which are calculated using Eq. (1).

Step 3. Improvising a new harmony

A new harmony (i.e. new truss design) $X^{\text{new}} = (x_1^{\text{new}}, x_2^{\text{new}}, \dots, x_{ng}^{\text{new}})$ is generated using three rules: (i) HM consideration, (ii) pitch adjustment and (iii) random generation. Generating a new harmony is called ‘improvisation’ [13].

In the HM consideration, the value of the first design variable x_1^{new} for the new harmony is chosen from the HM, (i.e. $\{x_1^1, x_1^2, \dots, x_1^{\text{HMS}-1}, x_1^{\text{HMS}}\}$) or from the possible range of values. The other design variables of new harmony ($x_2^{\text{new}}, \dots, x_{ng-1}^{\text{new}}, x_{ng}^{\text{new}}$) are chosen by the same consideration. HMCR is applied as follows:

$$\begin{cases} x_i^{\text{new}} \in \{x_i^1, x_i^2, \dots, x_i^{\text{HMS}-1}, x_i^{\text{HMS}}\} & \text{with probability HMCR} \\ x_i^{\text{new}} \in X_s & \text{with probability } (1 - \text{HMCR}) \end{cases} \quad (7)$$

where X_s is the set of the possible range of values for each design variable ($x_{\min} \leq X_s \leq x_{\max}$). The HMCR, which varies between 0 and 1, is the rate of

choosing one value from historical values stored in the HM, while $(1-\text{HMCR})$ is the rate of randomly selecting one value from the possible range of values [14]. For example, a HMCR value of 0.95 indicates that HS algorithm will choose the design variable from historically stored values in the HM with a 95 % probability and from the entire possible range with a 5 % probability [13].

Any design variable of the new harmony obtained by the memory consideration is examined to determine whether it is pitch-adjusted or not. This is performed by the pitch adjusting rate (PAR). PAR investigates a better design in the neighbouring of the current design and is applied as follows:

Pitch adjusting decision for

$$x_i^{\text{new}} \leftarrow \begin{cases} \text{yes with probability} & \text{PAR} \\ \text{no with probability} & (1 - \text{PAR}) \end{cases} \quad (8)$$

The value of $(1-\text{PAR})$ sets the rate of doing nothing, whereas the value of PAR indicates that x_i^{new} is replaced as follows:

$$x_i^{\text{new}} \leftarrow x_i^{\text{new}} + \text{bw} \times u(-1, 1) \quad (9)$$

where bw is the arbitrary distance bandwidth for continuous variable and $u(-1, 1)$ is a uniform distribution between -1 and 1 . For example, a PAR of 0.1 indicates that the algorithm will choose a neighbouring value with $10 \% \times \text{HMCR}$ probability [13]. HMCR and PAR parameters are introduced to allow the solution to escape from the local optima and to improve the global optimum prediction of HS algorithm [13].

Step 4. Updating the harmony memory

If the new harmony $X^{\text{new}} = (x_1^{\text{new}}, x_2^{\text{new}}, \dots, x_{\text{ng}}^{\text{new}})$ is better than the worst design in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the worst harmony is excluded from the HM. In this process, the HM is sorted again by objective function values.

Step 5. Terminating the process

Steps 3 and 4 are repeated until the termination criterion is satisfied.

The proposed SAHS algorithm differs from the standard HS algorithm as indicated in the following aspects:

SAHS algorithm presented in this study dynamically updates PAR during the search process as follows [23]:

$$\text{PAR}(ns) = \text{PAR}_{\max} - \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{NI} \times ns \quad (10)$$

In SAHS algorithm, bw is completely removed and Eq. (9) is replaced as follows:

$$x_i^{new} = x_i^{new} + [\max(\text{HM})_i - x_i^{new}] \times u(0, 1) \quad \text{if } u(0, 1) \leq 0.5 \quad (11)$$

$$x_i^{new} = x_i^{new} - [x_i^{new} - \min(\text{HM})_i] \times u(0, 1) \quad \text{if } u(0, 1) > 0.5 \quad (12)$$

where $\min(\text{HM})_i$ and $\max(\text{HM})_i$ are the lowest and highest values of the i th design variable in the HM. $u(0, 1)$ is a uniform random number in the $[0,1]$ range. Since $\min(\text{HM})_i$ and $\max(\text{HM})_i$ approach optimum gradually, SAHS algorithm produces finer adjustments to the harmony.

4 Teaching–Learning-Based Optimization

The TLBO method presents a mathematical model for optimization problems based on the simple teaching process. In the TLBO, the learners in a class are considered as the population. The teacher is accepted as the well-versed person in his/her profession. Hence, the learner with the highest mark in a class is mimicked as a teacher.

An analogy between the TLBO and the optimization of truss structures is established in the following way: a class is considered as a population which contains truss designs, a learner in a class denotes a truss design in the population, a design variable represents a subject taught to student, the grade of a student denotes the weight of the truss design, the teacher is the truss design with the lowest weight in the population.

Optimization of truss structures using the TLBO method consists of following steps [42]:

Step 1. Initializing the TLBO

In this step, the class is filled with randomly generated learners (truss designs) as the size of the population (ps).

$$ps = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_{ng-1}^1 & x_{ng}^1 \\ x_1^2 & x_2^2 & \dots & x_{ng-1}^2 & x_{ng}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{ps-1} & x_2^{ps-1} & \dots & x_{ng-1}^{ps-1} & x_{ng}^{ps-1} \\ x_1^{ps} & x_2^{ps} & \dots & x_{ng-1}^{ps} & x_{ng}^{ps} \end{bmatrix} \begin{matrix} \rightarrow W(X^1) \\ \rightarrow W(X^2) \\ \rightarrow \vdots \\ \rightarrow \vdots \\ \rightarrow W(X^{ps-1}) \\ \rightarrow W(X^{ps}) \end{matrix} \quad (13)$$

In the class, each row represents a truss design. $X^1, X^2, \dots, X^{ps-1}, X^{ps}$ and $W(X^1), W(X^2), \dots, W(X^{ps-1}), W(X^{ps})$ are truss designs and the corresponding weight values, respectively. It should be noted that the designs in the class are sorted by their weight values $(W(X^1) \leq W(X^2) \leq \dots \leq W(X^{ps-1}) \leq W(X^{ps}))$ which are calculated using Eq. (1).

Step 2. Teaching phase

The truss design with the lowest objective function ($W(X^1)$) is assigned as the teacher ($X^{\text{teacher}} = X^1$). The aim of the teacher is to put effort to move the mean of the class (X^{mean}). Therefore, i th design ($i \neq 1$) is modified using the following expression:

$$X^{\text{new},i} = X^i + r(X^1 - T_F X^{\text{mean}}) \quad (14)$$

in which X^i and $X^{\text{new},i}$ are the current and new designs, respectively. r is the random number uniformly distributed in the range of $[0,1]$, T_F is the teaching factor which is either 0 or 1 [39] and X^{mean} is the mean of the designs calculated as the following way:

$$X^{\text{mean}} = \left[m \left(\text{class} \left(\sum_{i=1}^{\text{ps}} x_1^i \right) \right), m \left(\text{class} \left(\sum_{i=1}^{\text{ps}} x_2^i \right) \right), \dots, m \left(\text{class} \left(\sum_{i=1}^{\text{ps}} x_{\text{ng}}^i \right) \right) \right] \quad (15)$$

where $m(\cdot)$ is the mean of the design variable. If the new design ($X^{\text{new},i}$) is better than the current design (X^i) (i.e. $W(X^{\text{new},i}) < W(X^i)$), the new design is replaced with the current design, $X^i = X^{\text{new},i}$.

Step 3. Learning phase

In addition to the teacher's effort to improve the mean of the class, the learners also interact with each other to improve themselves. A design in the population is randomly interacted with other designs to improve its quality. The learning phase is applied to learn new information between the design i and j ($i \neq j$) in the population and can be expressed as [41]

$$X^{\text{new},i} = X^i + r(X^i - X^j) \quad \text{if } W(X^i) < W(X^j) \quad (16a)$$

$$X^{\text{new},i} = X^i + r(X^j - X^i) \quad \text{if } W(X^j) < W(X^i) \quad (16b)$$

in which X^j is the randomly determined design which has to be different from X^i . If the value of $W(X^{\text{new},i})$ is better than $W(X^i)$ (i.e. $W(X^{\text{new},i}) < W(X^i)$), the new design is replaced with the current design $X^i = X^{\text{new},i}$.

Step 4. Terminating the search process

The steps 2 and 3 are repeated until the lightest truss design does not improve during a predetermined number of structural analyses.

5 Design Examples

The effectiveness and robustness of the SAHS [23] and TLBO [42] are tested using four truss structures. The results obtained by the methods are compared with those of HS [13], IHS [22], PSO [25], PSO, PSOPC and HPSO [28], HPSACO [31], HBB-BC [33], BB-BC [34] and ABC-AP [37].

SAHS algorithm produces the minimum weight design for the values of 20 for HMS, 0.90 for HMCR, 0.20 and 0.80 for PARmin and PARmax, 0.001 and 0.01 [23].

Two tuning parameters are employed in the TLBO: the population size (ps) and the number of designs generated in the learning phase (ndlp). The best combination of them obtained after sensitivity analysis and the minimum weight design for the TLBO is obtained by setting ps = 30, ndlp = 4 [42].

Twenty independent runs are made for each design example involving 20 different initial designs because of the stochastic nature of the SAHS and TLBO. The best designs obtained by the methods, the number of structural analyses required to the optimum solutions, the average weight and the standard deviation of 20 independent runs are given in the tables. The SAHS and TLBO are coded in FORTRAN language and executed on a Intel Pentium Core 2 Duo 2.2 GHz machine.

5.1 Ten-Bar Plane Truss

The planar ten-bar plane truss, shown in Fig. 1 is the first design example. The Young's modulus and density of truss members are 10^4 ksi (1 ksi = 6.895 MPa) and 0.1 lb/in^3 , respectively. The allowable stress for all members is specified as 25 ksi both in tension and compression. The maximum displacements of all free nodes in the X and Y directions are limited to ± 2 . Each member is considered as a design variable with the minimum gauge of 0.1 in^2 .

The results obtained by the SAHS [23], TLBO [42] and the other optimization methods are reported in Table 1. The presented methods in this chapter found the

Fig. 1 Ten-bar plane truss
 (1 in. = 2.54 cm,
 1 kip = 4.448 kN)

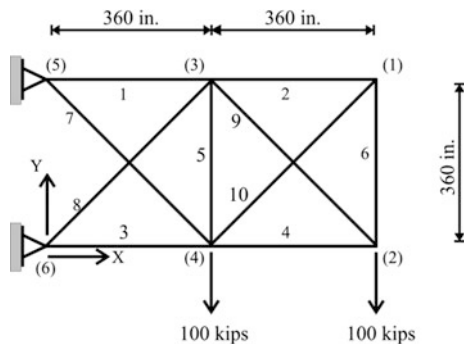


Table 1 The pseudocode of the HS algorithm

Design variables A_i (in. ²)	[13]	[25]	[28]			[31]	[22]	[37]	Proposed [23, 42]	
	HS	PSO	PSO	PSOPC	HPSO	HPSACO	IHS	ABC-AP	SAHS	TBLO
A_1	30.15	33.500	33.469	30.569	30.704	30.307	30.5222	30.548	30.394	30.4286
A_2	0.102	0.100	0.110	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
A_3	22.71	22.766	23.177	22.974	23.167	23.434	23.2005	23.180	23.098	23.2436
A_4	15.27	14.417	15.475	15.148	15.183	15.505	15.2232	15.218	15.491	15.3677
A_5	0.102	0.100	3.649	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
A_6	0.544	0.100	0.116	0.547	0.551	0.5241	0.5513	0.551	0.529	0.5751
A_7	7.541	7.534	8.328	7.493	7.460	7.4365	7.4572	7.463	7.488	7.4404
A_8	21.56	20.467	23.340	21.159	20.978	21.079	21.0367	21.058	21.189	20.9665
A_9	21.45	20.392	23.014	21.156	21.508	21.229	21.5288	21.501	21.342	21.5330
A_{10}	0.100	0.100	0.190	0.100	0.100	0.100	0.1000	0.100	0.100	0.1000
Weight (lb) ^a	5057.88	5024.21	5529.50	5061.00	5060.92	5056.56	5060.82	5060.880	5061.42	5060.96
Average weight (lb)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	5061.95	5062.08
Standard dev. (lb)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.71	0.79
Constraint violation (%)	0.091	1.95	None	None	None	0.099	None	None	None	None
N. struct. analyses	20,000	N/A	150,000	150000	125,000	10650	1350	500×10^3	7081	16872

^a1 lb = 0.4536 kg

better designs than those of the HS [13], PSO [25] and HPSACO [31] since the lighter designs obtained by classical HS [13], PSO [25] and HPSACO [31] violated the design constraints while the designs obtained by the SAHS [23] and TLBO [42]

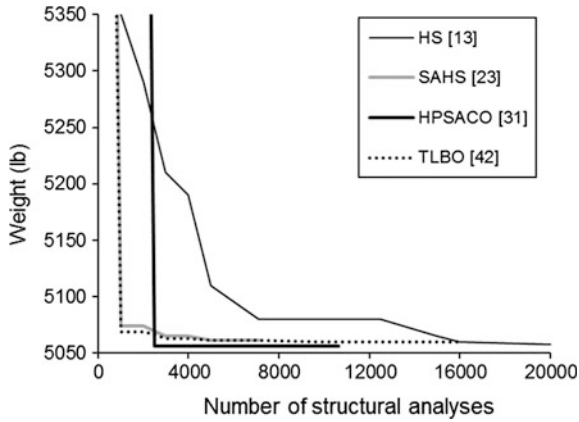


Fig. 2 Convergence histories for the ten-bar plane truss

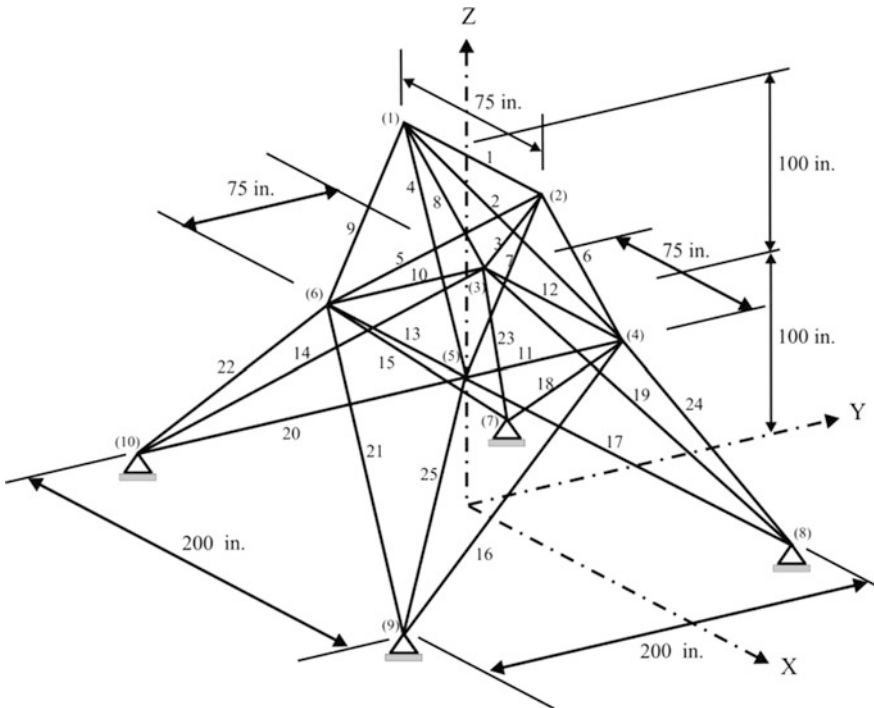


Fig. 3 Twenty-five-bar space truss

are feasible. Although IHS [22] required the lowest number of structural analyses, it should be noted that the number of structural analyses required by IHS [22] is not a significant basis of comparison to evaluate the efficiency of SAHS [23] and TLBO [42] since the IHS [22] included gradient information that allowed the number of function evaluations to be drastically reduced as gradient information, inherently speed up the optimization process [23]. Convergence histories (i.e. structural weight versus number of structural analyses) are illustrated in Fig. 2.

5.2 Twenty-Five-Bar Space Truss

The twenty-five-bar space truss, shown in Fig. 3 is one of the most popular design examples used in the literature for comparing different optimization methods. The Young's modulus and the density of truss members are 10^4 ksi and 0.1 lb/in^3 , respectively. The structure is subject to the two loading conditions given in Table 2. The design variables of the structure and the allowable stress values for all groups are listed in Table 3. The displacement of nodes in all directions is restricted to be less than ± 0.35 in. The minimum cross-sectional area for each group of elements is 0.01 in^2 .

Table 2 Loading conditions for the twenty-five-bar space truss

Node	Condition 1			Condition 2		
	F_x	F_y	F_z	F_x	F_y	F_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

Note Loads are in kips

Table 3 Allowable stress values for the twenty-five-bar space truss

Design variables A_t (in^2)	Allowable compressive stress (ksi)	Allowable tension stress (ksi)
A_1	35.092	40.0
A_2 - A_5	11.590	40.0
A_6 - A_9	17.305	40.0
A_{10} - A_{11}	35.092	40.0
A_{12} - A_{13}	35.092	40.0
A_{14} - A_{17}	6.7590	40.0
A_{18} - A_{21}	6.9590	40.0
A_{22} - A_{25}	11.082	40.0

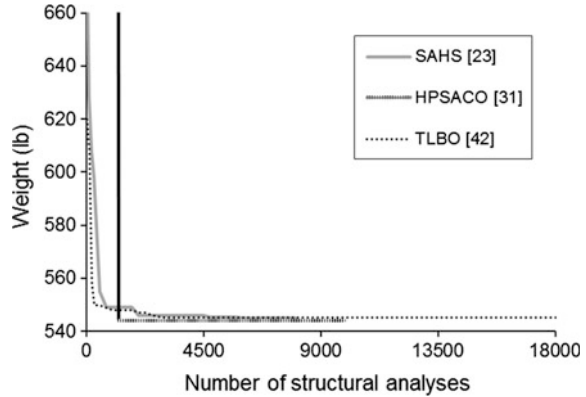
Table 4 Optimization results obtained for the twenty-five-bar space truss

A _i (in. ²)	[13]	[28]			[34]	[31]	[33]	[22]	Proposed [23, 42]	
	HS	PSO	PSOPC	HPSO	BB-BC	HPSACO	HBB-BC	IHS	SAHS	TLBO
A ₁	0.047	9.863	0.010	0.010	0.010	0.010	2.6622	0.0100	0.010	0.0100
A ₂ -A ₅	2.022	1.798	1.979	1.970	2.092	2.054	1.993	1.9871	2.074	2.0712
A ₆ -A ₉	2.950	3.654	3.011	3.016	2.964	3.008	3.056	2.9935	2.961	2.9570
A ₁₀ -A ₁₁	0.010	0.100	0.100	0.010	0.010	0.010	0.010	0.0100	0.010	0.0100
A ₁₂ -A ₁₃	0.014	0.100	0.100	0.010	0.010	0.010	0.010	0.0100	0.010	0.0100
A ₁₄ -A ₁₇	0.688	0.596	0.657	0.694	0.689	0.679	0.665	0.6839	0.691	0.6891
A ₁₈ -A ₂₁	1.657	1.659	1.678	1.681	1.601	1.611	1.642	1.6769	1.617	1.6209
A ₂₂ -A ₂₅	2.663	2.612	2.693	2.643	2.686	2.678	2.679	2.6622	2.674	2.6768
Weight (lb)	544.38	627.08	545.27	545.19	545.38	544.99	545.16	545.15	545.12 [545.38] ^a	545.09 [545.38] ^b
Average weight (lb)	N/A	N/A	N/A	N/A	545.78	545.52	545.66	N/A	545.94	545.41
Std. dev. (lb)	N/A	N/A	N/A	N/A	0.491	0.315	0.367	N/A	0.91	0.42
Constraint tolerance (%)	0.206	None	None	None	None	3.52	2.06	None	None	None
Num. struct. analyses	15,000	150,000	150,000	125,000	20,566	9875	12,500	1050	9051	15,318

^aSAHS obtained a weight of 545.38 lb after 6941 analyses (the result of BB-BC)

^bTLBO obtained a weight of 545.38 lb after 6665 analyses (the result of BB-BC)

Fig. 4 Convergence history for the twenty-five-bar space truss



The results obtained by the SAHS [23], TLBO [42] and the other optimization methods existing in the literature are reported in Table 4. It is clear from Table 4 that the TLBO [42] developed the best design overall since the lighter design obtained by HS [13] and HPSACO [31] violated the design constraints while the design obtained by the TLBO [42] is feasible. Moreover, it should be noted that although the TLBO [42] required more structural analyses than the SAHS [23] to find the optimum design, the TLBO [42] developed a design with a weight of 545.38 lb after 6665 structural analyses while the SAHS [23] required 6941 structural analyses to find the same weight. The convergence histories are illustrated in Fig. 4.

5.3 Seventy-Two-Bar Space Truss

The third example deals with the design of the seventy-two-bar space truss shown in Fig. 5. The structure is subject to the loading conditions given in Table 5. The Young's modulus and density of the material are 10^4 ksi and 0.1 lb/in^3 , respectively. The member cross-sectional areas are treated as design variables, and are divided into 16 groups. The allowable stress for all members is specified as 25 ksi both in tension and compression. The maximum displacements of all free nodes are limited to ± 0.25 in. The minimum cross-sectional areas are specified as 0.1 in^2 .

Table 6 shows the results obtained by the SAHS [23], TLBO [42] and the other optimization methods reported in current literature [13, 25, 28, 33, 34, 37]. The TLBO [42] founded the lightest design overall because the lighter design obtained by the HS [13] violates the design constraints. It is seen from Table 6 that the BB-BC [34] found a minimum weight of 379.85 lb after 19621 structural analyses for case 1 while the TLBO [42] developed the same weight after 8422 structural analyses. Figure 6 shows the convergence histories of the SAHS [23] and TLBO [42].

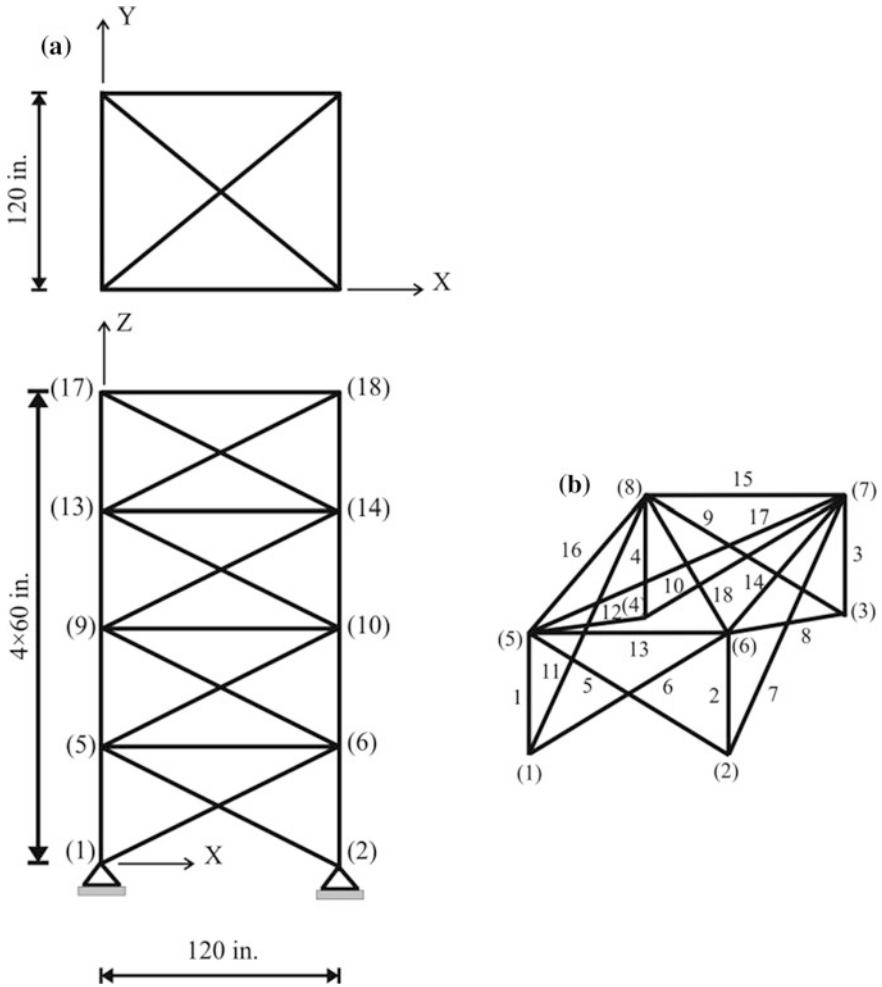


Fig. 5 Seventy-two-bar space truss: **a** dimensions **b** element and node numbering patterns for the first storey

Table 5 Loading conditions for the seventy-two-bar space truss

Node	Condition 1			Condition 2		
	F_x	F_y	F_z	F_x	F_y	F_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

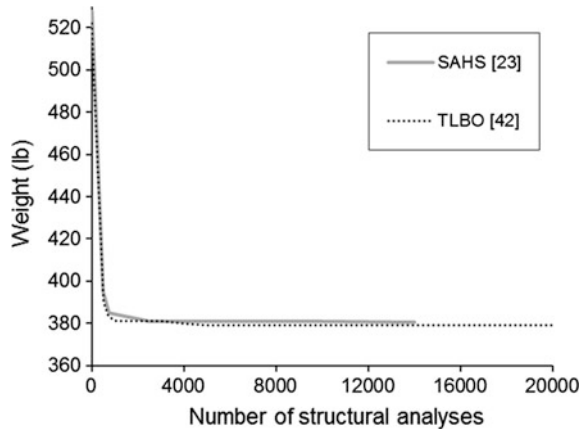
Note Loads are in kips

Table 6 Optimization results obtained for the seventy-two-bar space truss

Design variables A_i (in. ²)	[13]	[25]		[28]		[34]		[33]	Proposed [23, 42]	
	HS	PSO	PSO	PSO	PSOPC	HPSO	BB-BC	HBB-BC	SAHS	TLBO
A_1 - A_4	1.790	1.7427	41.794	1.855	1.857	1.8577	1.9042	1.860	1.9064	
A_5 - A_{12}	0.521	0.5185	0.195	0.504	0.505	0.5059	0.5162	0.521	0.50612	
A_{13} - A_{16}	0.100	0.1000	10.797	0.100	0.100	0.1000	0.1000	0.100	0.100	
A_{17} - A_{18}	0.100	0.1000	6.861	0.100	0.100	0.1000	0.1000	0.100	0.100	
A_{19} - A_{22}	1.229	1.3079	0.438	1.253	1.255	1.2476	1.2582	1.271	1.2617	
A_{23} - A_{30}	0.522	0.5193	0.286	0.505	0.503	0.5269	0.5035	0.509	0.5111	
A_{31} - A_{34}	0.100	0.1000	18.309	0.100	0.100	0.1000	0.1000	0.100	0.100	
A_{35} - A_{36}	0.100	0.1000	1.220	0.100	0.100	0.1012	0.1000	0.100	0.100	
A_{37} - A_{40}	0.517	0.5142	5.933	0.497	0.496	0.5209	0.5178	0.485	0.5317	
A_{41} - A_{48}	0.504	0.5464	19.545	0.508	0.506	0.5172	0.5214	0.501	0.51591	
A_{49} - A_{52}	0.100	0.1000	0.159	0.100	0.100	0.1004	0.1000	0.100	0.100	
A_{53} - A_{54}	0.101	0.1095	0.151	0.100	0.100	0.1005	0.1007	0.100	0.100	
A_{55} - A_{58}	0.156	0.1615	10.127	0.100	0.100	0.1565	0.1566	0.168	0.1562	
A_{59} - A_{66}	0.547	0.5092	7.320	0.525	0.524	0.5507	0.5421	0.584	0.54927	
A_{67} - A_{70}	0.442	0.4967	3.812	0.394	0.400	0.3922	0.4132	0.433	0.40966	
A_{71} - A_{72}	0.590	0.5619	18.196	0.535	0.534	0.5922	0.5756	0.520	0.56976	
Weight (lb)	379.27	381.91	6818.67	369.65	369.65	379.85	379.66	380.62	379.63	
Average weight	N/A	N/A	N/A	N/A	N/A	382.08	381.85	382.42	380.20	
Std. dev.	N/A	N/A	N/A	N/A	N/A	1.912	1.201	1.38	0.41	
Constraint tolerance (%)	0.218	None	None	39.051	39.075	None	None	None	None	
Num. struct. analyses	20,000	N/A	150,000	125,000	125,000	19,621	13,200	13,742	19,709	

^aTLBO obtained a weight of 379.85 lb after 8422 analyses (the result of BB-BC)

Fig. 6 Convergence histories for the seventy-two-bar space truss



6 Conclusions

The SAHS and TLBO obtained results as good as or better than other metaheuristic optimization methods in terms of both the optimum solutions and the convergence capability. It appeared that although the TLBO developed slightly heavier designs than the PSOPC, HPSO and ABC-AP in a few cases, it required significantly less structural analyses than the PSOPC, HPSO and ABC-AP in all design examples. It should be noted that standard deviation of optimized weights obtained over 20 independent runs was quite small, which is $<1.0\%$ in all design examples, compared with average optimized weight. This points out that the SAHS and TLBO are able to find a nearly global optimum design.

References

1. Lamberti, L., Pappalettere, C.: Metaheuristic design optimization of skeletal structures: a review. *Comput. Technol. Rev.* **4**, 1–32 (2011)
2. Saka, M.P.: Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. In: Topping, B.H.V. (ed.) *Civil engineering computations: tools and techniques*, pp. 105–147. Saxe-Coburg Publications, Stirlingshire (2007)
3. Dorigo, M., Maniezzo, V., Colomi, V.: An investigation of some properties of an ant algorithm. In: *Proceedings of the 1992 Parallel Problem Solving from Nature Conference*, Elsevier, Amsterdam, pp. 509–520 (1992)
4. Camp, C.V., Bichon, B.J., Stovall, S.P.: Design of steel frames ant colony optimization. *J. Struct. Eng. ASCE*. **131**, 369–379 (2005)
5. Camp, C.V., Bichon, B.J.: Design of space trusses using ant colony optimization. *J. Struct. Eng. ASCE*. **130**, 741–751 (2004)
6. Capriles, P.V.S., Fonseca, L.G., Barbosa, H.J.C., Lemonge, A.C.C.: Rank-based ant colony algorithms for truss weight minimization with discrete variables. *Commun. Numer. Meth. En.* **23**, 553–575 (2007)

7. Serra, M., Venini, P.: On some applications of ant colony to plane truss optimization. *Struct. Multidiscip. O.* **32**, 499–506 (2006)
8. Hasancebi, O., Carbas, S., Dogan, E., Erdal, F., Saka, M.P.: Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Comput. Struct.* **87**, 284–302 (2009)
9. Kaveh, A., Shojaee, S.: Optimal design of skeletal structures using ant colony optimization. *Int. J. Numer. Meth. Eng.* **70**, 563–581 (2007)
10. Hasancebi, O., Carbas, S., Dogan, E., Erdal, F., Saka, M.P.: Comparison of nondeterministic search techniques in the optimum design of real size steel frames. *Comput. Struct.* **88**, 1033–1048 (2010)
11. Kaveh, A., Talatahari, S.: An improved ant colony optimization for the design of planar steel frames. *Eng. Struct.* **32**, 864–873 (2010)
12. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**, 60–68 (2001)
13. Lee, K.S., Geem, Z.W.: A new structural optimization method based on the harmony search algorithm. *Comput. Struct.* **82**, 781–798 (2004)
14. Lee, K.S., Geem, Z.W., Lee, S.H., Bae, K.W.: The harmony search heuristic algorithm for discrete structural optimization. *Eng. Optimiz.* **37**, 663–684 (2005)
15. Saka, M.P.: Optimum geometry design of geodesic domes using harmony search algorithm. *Adv. Struct. Eng.* **10**, 595–606 (2007)
16. Saka, M.P., Erdal, F.: Harmony search based algorithm for the optimum design of grillage systems to AISC-LRFD. *Struct. Multidiscip. O.* **38**, 25–41 (2009)
17. Degertekin, S.O.: Harmony search algorithm for optimum design of steel frame structures: a comparative study with other optimization methods. *Struct. Eng. Mech.* **29**, 391–410 (2008)
18. Degertekin, S.O.: Optimum design of steel frames using harmony search algorithm. *Struct. Multidiscip. O.* **36**, 393–401 (2008)
19. Saka, M.P.: Optimum design of steel sway frames to BS 5950 using harmony search algorithm. *J. Constr. Steel Res.* **65**, 36–43 (2009)
20. Saka, M.P., Hasancebi, O.: Adaptive harmony search algorithm for design code optimization of steel structures. *Harmony search algorithms for structural design optimization*. In: Geem Z. W. (ed.) *Studies in computational intelligence 239*, Springer-Verlag, Berlin Heidelberg, pp. 79–120 (2009)
21. Hasancebi, O., Erdal, F., Saka, M.P.: Adaptive harmony search method for structural optimization. *J. Struct. Eng. ASCE.* **136**, 419–431 (2010)
22. Lamberti, L., Pappalettere, C.: An improved harmony-search algorithm for truss structure optimization. In: Topping B.H.V., Neves L.F.C., Barros, R.C. (eds.) *Proceedings of the Twelfth International Conference Civil, Structural and Environmental Engineering Computing Civil-Comp Press, Stirlingshire, Scotland* (2009)
23. Degertekin, S.O.: An improved harmony search algorithms for sizing optimization of truss structures. *Comput. Struct.* **92–93**, 229–241 (2012)
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *IEEE International Conference on Neural Networks*, pp. 1942–1948 (1995)
25. Perez, R.E., Behdinan, K.: Particle swarm approach for structural design optimization. *Comput. Struct.* **85**, 1579–1588 (2007)
26. Fourie, P., Groenwold, A.: The particle swarm optimization algorithm in size and shape optimization. *Struct. Multidiscip. O.* **23**, 259–267 (2002)
27. Schutte, J.F., Groenwold, A.: Sizing design of truss structures using particle swarms. *Struct. Multidiscip. O.* **25**, 261–269 (2009)
28. Li, L.J., Huang, Z.B., Liu, F., Wu, Q.H.: A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput. Struct.* **85**, 340–349 (2007)
29. Li, L.J., Huang, Z.B., Liu, F.: A heuristic particle swarm optimization method for truss structures with discrete variables. *Comput. Struct.* **87**, 435–443 (2009)
30. Kaveh, A., Talatahari, S.: A particle swarm ant colony optimization for truss structures with discrete variables. *J. Constr. Steel Res.* **65**, 1558–1568 (2009)

31. Kaveh, A., Talatahari, S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Comput. Struct.* **87**, 267–283 (2009)
32. Erol, O.K., Eksin, I.: New optimization method: big bang-big crunch. *Adv. Eng. Softw.* **37**, 106–111 (2006)
33. Kaveh, A., Talatahari, S.: Size optimization of space trusses using big-bang big-crunch algorithm. *Comput. Struct.* **87**, 1129–1140 (2009)
34. Camp, C.V.: Design of space trusses using big bang-big crunch optimization. *J. Struct. Eng. ASCE.* **133**, 999–1008 (2007)
35. Kaveh, A., Talatahari, S.: Optimal design of Schwedler and ribbed domes via hybrid big bang-big crunch algorithm. *J. Constr. Steel Res.* **66**, 412–419 (2010)
36. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
37. Sonmez, M.: Artificial bee colony algorithm for optimization of truss optimization. *Appl. Soft Comput.* **11**, 2406–2418 (2011)
38. Sonmez, M.: Discrete optimum design of truss structures using artificial bee colony algorithm. *Struct. Multidiscip. O.* **43**, 85–97 (2011)
39. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comp. Aid. Des.* **43**, 303–315 (2011)
40. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inf. Sci.* **183**, 1–15 (2012)
41. Toğan, V.: Design of steel frames using teaching-learning based optimization. *Eng. Struct.* **34**, 225–232 (2012)
42. Degertekin, S.O., Hayalioglu, M.S.: Sizing optimization of truss structures using teaching-learning-based optimization. *Comput. Struct.* **119**, 177–188 (2013)