Structural Engineering and Mechanics, *Vol. 47, No. 2 (2013) 209-225*
DOI: http://dx.doi.org/10.12989/sem.2013.47.2.209

Design of pin jointed structures using teaching-learning based optimization

Vedat Toğan*

Karadeniz Technical University, Department of Civil Engineering 61080 Trabzon, Turkey

(Received June 7, 2012, Revised June 5, 2013, Accepted July 17, 2013)

Abstract. A procedure employing a Teaching-Learning Based Optimization (TLBO) method is developed to design discrete pin jointed structures. TLBO process consists of two parts: the first part represents learning from teacher and the second part illustrates learning by interaction among the learners. The effectiveness of the TLBO method is demonstrated on the four design optimization problems. The results are compared with those obtained using other various evolutionary optimization methods considering the best solution, average solution, and computational effort. Consequently, the TLBO algorithm works effectively and demonstrates remarkable performance for the optimization of engineering design applications.

Keywords: optimization; algorithms; trusses; structural design; teaching learning based optimization

1. Introduction

The optimization techniques used for minimizing the volume or weight, intend to achieve an optimum design having a set of design variables under certain design criteria. They can be roughly categorized as gradient-based and gradient-free methods. In contrast to gradient-based ones, gradient-free techniques have been emerging as powerful tools for discrete structural optimization, in which the design variables are discrete and must be chosen from a predetermined set. The key feature of the gradient-free techniques is to simulate natural phenomena.

The most widely recognized optimization method among others is the Genetic Algorithm (GA), which is based on the concept of natural selection, and it has been used in the various structural optimization applications (Rajeev and Krishnamoorthy 1992, Toğan and Daloğlu 2009, Chen and Rajan 2004, Toğan and Daloğlu 2006). In order to further, improve the performance of the GAs, some enhancements which are based on the adaptive approaches haven been suggested (Nanakorn and Meesomklin 2001, Toğan and Daloğlu 2006, 2008). After noticing the rationale behind GA, different techniques inspired from the nature have been consecutively suggested and they have been widely used in fields varying from engineering to finance. Particle swarm optimization (PSO) (Kennedy and Eberhart 1995, Li *et al.* 2009), ant colony optimization (ACO) (Camp and Bichon 2004, Camp *et al.* 2005, Capriles *et al.* 2007), harmony search (HS) (Lee and Geem 2004, Saka 2007, Değertekin 2007), artificial bee colony (ABC) (Karaboğa 2005, Karaboğa and Baştürk 2007, Sönmez 2011) and charged system search (CSS) (Kaveh and Talatahari 2010) are the optimization

*Corresponding author, Ph.D., E-mail: togan@ktu.edu.tr

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

methods, which are recently developed and have been frequently used for the structural optimization applications by the researchers. To develop an algorithm simulating the nature has been of increasing popularity due to its ability of solving different optimization problems.

Rao *et al.* (2011) developed a new optimization method, the so-called Teaching-Learning Based Optimization (TLBO), as an innovative optimization algorithm inspiring the natural phenomena, which mimics teaching-learning process in a class between the teacher and the students (learners). In their model, the "Teaching" phase produces a random ordered state of points called learners within the search space. Then a point is considered as the teacher, who is highly knowledgeable person and shares his or her knowledge with the learners, thus the others receive significant information from the teacher. The learners also learn by interacting among them. After a number of sequential Teaching-Learning cycles, where the teacher conveys knowledge among the learners and elevates their knowledge close to her or his level, the distribution of the randomness within the search space becomes smaller and smaller reaching around a point adopted as the teacher. Convergence over a solution means that the knowledge level of the whole class shows smoothness. Rao *et al.* (2011) have shown that the TLBO algorithm is more effective and efficient than the other optimization methods mentioned above in solving the mechanical design optimization problems found in the literature. In addition, Toğan (2011) determined the optimum designation conditions of the cross-section areas of the planar steel frames by using TLBO.

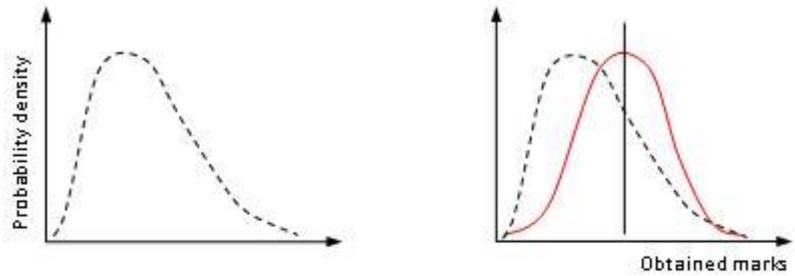
In this study, a procedure employing a Teaching-Learning Based Optimization (TLBO) method is developed for discrete design of trusses to show the performance of TLBO on the structural optimization problems encountered in the civil engineering field. For the purpose of fulfilling the performance evaluation criteria on the structural design optimization problems, the effectiveness of the TLBO method is demonstrated on the four different truss design optimization problems considering the best solution, average solution, and computational effort. The minimum weight of the truss structure, which is subjected to constraints in the form of stress and deflection limits, is considered as the objective function.

2. Teaching-Learning Based Optimization (TLBO)

The TLBO can be summarized as a procedure simulating the interactivity between the teacher and students in a class, also known as the Teaching-Learning process. The teacher, who is considered as the most knowledgeable person in the society, teaches a subject to the learners to increase information level of the whole class on a specific subject. He or she intends to enhance the knowledge level of the learners thus approximating it to his or her own level. The teacher evaluates the learning level of the learners through an exam. The marks obtained by the learners represent the knowledge level that has been reached by the teacher.

Before teaching a subject, the learners of the class is expected to exhibit irregular level of initial knowledge on that every topic. The final aim of the teacher, on the other hand, is to diminish this irregularity and provide uniformity in the class after the teaching-learning process.

It is also possible for the learners to learn through an interaction among them. If a normal distribution is assumed for the obtained marks by the learners, the corresponding distribution can be illustrated as in Fig. 1 before and after teaching-learning process, respectively. The quality of a teacher affects the outcome of the learners. Fig. 2 shows the effect of the influence of a teacher on the output of learners examined in terms of the marks in a class. It should be noted in Fig. 2 that the output of the learners increases as the quality of the teacher as well as the quality of the



(a) before teaching-learning process (b) after teaching-learning process

Fig. 1 The level of whole class: (a) before and (b) after teaching-learning process

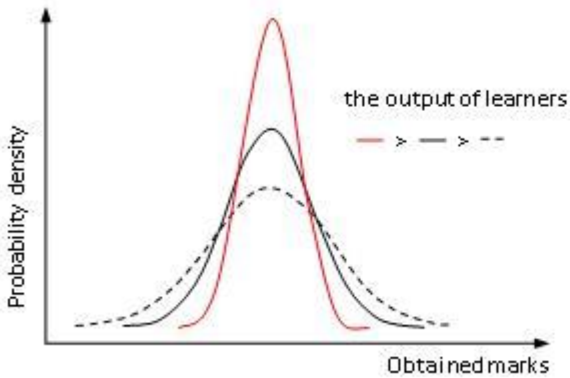


Fig. 2 The level of whole class depending on the teacher and learners qualities

students increase. In an actual practice, however, increasing the learners' level uniformly to a certain level or up to the level of the teacher may be not possible. It has skewness about to point considering as the mean of the results of the learners.

The TLBO method, extensively used in this paper, is summarized above. The method consists of two phases; a) *Teaching Phase*, where the candidate solutions are randomly distributed over the search space, the best solution is then determined among those candidates, and the information obtained from the best solution is shared with the others; and b) *Learning Phase*, where the solutions are shared their knowledge among themselves.

Teaching Phase, that is the initial part of TLBO, is the same with other nature-inspired methods. An initial population consisting of candidate solutions is generated randomly over the entire search space to proceed with the global solution. It is clear that the population might have feasible or infeasible solutions as well. In optimization algorithms, the solution composes of design variables and is qualified according to its fitness. Similarly the optimization algorithms, in the TLBO a class and a student in that class represent the population (*pop*) randomly generated with pre-defined size (*np*) and the candidate solution (x^i , i=1 to *np*), respectively. Each subject taught to the students represents the design variable which is an integer value representing the sequence numbers of predefined discrete set and the combination of it denotes the design variables of objective function. The candidate solution composes of design variables ($x^i = [x_j, x_2, \ldots, x_{nd}]$, i=1 to *np*; j= 1 to *nd)* and is qualified according to its fitness ($f(x^i)$). The solution having best fitness (i.e.,

$f_{min}(.)$) in the population is determined as the teacher ($x_{teacher}$). Then a formula is employed to update the solution *i* according to the best solution and the mean of the solutions as follows

$$x^{new,i} = x^i + r(x_{teacher} - T_F\, x_{mean}) \tag{1}$$

where $x^{new,i}$ and x^i are the modified and existing solution of *i*, *r* is a random number varying between 0 and 1, T_F is a teaching factor being either 1 or 2, which is again a heuristic step and decided randomly with equal probability as T_F = round[1 + rand (0,1) {2-1}] (Rao *et al.* 2011). The random number *r* is generated for each design variables while T_F is produced just one times for x_{mean}. $x_{teacher}$ is the best solution and x_{mean} is the mean of the solutions calculated vector based, Eq.(2).

$$x_{mean} = \begin{bmatrix} \bar{x}_1^t & \bar{x}_2^t & \cdots & \bar{x}_{nd-1}^t & \bar{x}_{nd}^t \end{bmatrix} \Rightarrow x_{mean} = \bar{x}_z^t = \frac{\sum_{t=1}^{np} x_z^t}{np}$$

$$\text{where} \quad pop = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{nd-1}^1 & x_{nd}^1 \\ x_1^2 & x_2^2 & \cdots & x_{nd-1}^2 & x_{nd}^2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1^{np-1} & x_2^{np-1} & \cdots & x_{nd-1}^{np-1} & x_{nd}^{np-1} \\ x_1^{np} & x_2^{np} & \cdots & x_{nd-1}^{np} & x_{nd}^{np} \end{bmatrix} \quad \text{and} \quad z = 1,\ldots,nd, \quad t = 1,..,np \tag{2}$$

where *np* and *nd* are the number of solutions(population size) and the design variables, *pop* is the population composed of *np* rows and *nd* columns. If the modified solution $x^{new,i}$ produces better objective function value than x^i change x^i to $x^{new,i}$, otherwise preserve x^i.

After the Teaching Phase, an exchange information operation is applied during the Learning Phase. In this case, it is aimed for the solution to learn something new interacting randomly with other solutions through group discussions, presentations, formal communications, etc. A learner with a certain solution will learn new information if the other learners with different solutions have more knowledge than him or her (Rao *et al.* 2011). So, for the minimization problem the modification formula requiring the exchange information between the solution *i* and the solution *j* can be expressed as

$$\begin{aligned} x^{new,i} &= x^i + r(x^i - x^j) && \text{if } f(x^i) < f(x^j) \\ x^{new,i} &= x^i + r(x^j - x^i) && \text{if } f(x^i) > f(x^j) \\ \\ \text{if } f(x^{new,i}) &> f(x^i) && x^i = x^i \\ \text{if } f(x^{new,i}) &< f(x^i) && x^i = x^{new,i} \end{aligned} \tag{3}$$

in which $x^{new,i}$ is the modified solution of the existing solution x^i, x^j is the any solution to be different from x^i, and $f(x^{new,i})$ and $f(x^i)$ show the objective function values for the solutions $x^{new,i}$ and x^i, respectively. This operation is repeated until the number of solutions (i, j = 1 to np, $i \neq j$). At the end of this operation, a cycle is completed for the TLBO.

The entire process explained above is continued until reaching the termination criterion. The maximum number of cycle is taken into account as the termination criterion.

3. Optimum design of truss by TLBO

The cost of a truss structure associated with the volume of the material used is generally taken as the objective function while satisfying the design constraints, such as the allowable stresses in members and/or nodal deflection limits. If the topology of the truss keeps on fixed the main factor affecting the cost of the truss would be the cross sectional area of each member in truss. Therefore, the value of cross-sectional area for each group of structural elements can be considered as discrete or continuous design variable. In case of sizing optimization problems with discrete design variables, the major task is to select an optimal cross-section of the elements from a permissible list of standard sections that minimize the weight of the structure while satisfying the design constraints. Typically truss designs are limited by allowable material stresses and structural displacements.

In the current work, penalty function concept is preferred to handle the constraints in contrast to (Rao *et al.* 2011), where a heuristic constraint method developed by Deb (2000) was used. Therefore, for each candidate truss design, a penalty function is applied to the structural weight reflecting the degree of constraint violation. Thus, the structural weight taken as the objective function is modified to search the designs with the smallest structural weight that satisfies the design constraints. In this study, an optimal truss design is sought, which has the minimum weight of the structure while it does not exceed the allowable values for compressive and/or tensile stress in each member and deflection of any connection. A truss optimization problem can thus be expressed as follows

$$\begin{aligned} &\text{Minimize} \quad f(x)=\sum_{i=1}^{ng}\rho A_i \sum_{j=1}^{ne} L_j \\ &\text{subject to:} \quad g_{str,k}=\sigma_k/\sigma_{a,k}-1 \qquad k=1,..,nm \\ &\qquad\qquad\quad g_{d,r}=\delta_r/\delta_{a,r}-1 \qquad r=1,..,n \end{aligned} \tag{4}$$

In Eq. (4), x is the vector for the design variables taken as the values of cross-sectional area for each group of structural elements ($x=[A_1\ A_2\ \ldots\ A_{ng}]^T$), $f(x)$ is the objective function for minimum weight of truss, ρ is the material density, *ng* is the total numbers of groups in the truss, *ne* is the total numbers of elements in group i, A_i is the cross-section area of ith group and L_j is the length of jth element for ith group, $g_{str,k}$ and $g_{d,r}$ are the violation value of normalized stress and displacement of the kth element and of the rth node of truss structure, respectively. σ_k and $\sigma_{a,k}$ are the stress in each member k of the truss and the maximum allowable stresses indicating tension or compression stress depending on the axial force , δ_r and $\delta_{a,r}$ are the displacement of node r of the truss and the allowable displacement imposed on node r. Finally, *nm* and n represent the number of stress and displacement constraints. The penalized weight reflecting the feasibility of the candidate truss design is then written as

$$\varphi(x)=f(x)(1+C)^{\varepsilon} \tag{5}$$

where C is the value of total constraints violation and ε is the positive penalty exponent. The total constraints violation C is a function of the summation of the stress and deflection constraints defined as

$$\begin{aligned} &C=\sum_{k=1}^{nm} c_{str}+\sum_{r=1}^{n} c_d \\ &\text{where } c_{str}=\max(g_{str,k},0) \ \text{ and } \ c_d=\max(g_{d,r},0) \end{aligned} \tag{6}$$

The steps of the TLBO algorithm used in the current work for truss optimization are demonstrated with stepwise manner as follows:

• Define *np*, *nd* and the maximum number of cycle adopted as the termination criterion

• Initialize a population; As stated above, the population, *pop*, represents a class, which composes of students. The population, *pop* (see Eq. (2)), is filled with randomly generated students (solutions) according to the population size, *np*, and number of design variables, *nd*.

• Evaluate Eq. (5) for each x^i (i=1,…,*np*) in the population

• Find the best learner (i.e., best solution is to produce the minimum penalized weight, $\varphi(.)$) in the population and assign him/her to the teacher ($x_{teacher} = x^i$ where x^i produces minimum $\varphi(.)$)

• Calculate mean of each group of learners (x_{mean}) in the current population and perform the Teaching Phase by the help of Eq. (2) and Eq. (1)

• Accept all $x^{new,i}$ instead of x^i if $f(x^{new,i})$ is better than $f(x^i)$ and update the population

• Improve the learners' knowledge by utilizing the knowledge of some other learner through the Eq. (3). It means that perform the Learning Phase.

• Accept all $x^{new,i}$ instead of x^i if $f(x^{new,i})$ is better than $f(x^i)$ and update the population

• Control the termination criterion (i.e., maximum number of cycle). In the case that it is met, show the results. Otherwise continue the procedure from steps 3 to 8

The presented optimum design algorithm employing a Teaching–Learning Based Optimization (TLBO) technique for discrete optimization of trusses is coded in MATLAB environment and all computations are performed in a PC with the Pentium® 4 2.66 GHz processor and 1.0 GB RAM .

3.1 TLBO truss design parameters

Recalling the definition of TLBO algorithm given above, it is worthy to say that the algorithm requires easy software programming with relatively few parameters to control the algorithm performance. T_F defined in the TLBO algorithm is taken as 2. The value of the penalty function exponent, given in Eq. (5), is considered as 2 (Camp *et al*. 2005). Numerical results presented in the study show the best solutions obtained among the twenty independent runs performed. Since these are compared with other nature-inspired methods, a predetermined number of truss analyses being minimum than that determined from the given references are taken into account as the termination criterion to show the computational efficiency and overall algorithm performance of TLBO.

4. Design examples

4.1 Ten-bar truss design examples

A ten bar plane truss shown in Fig. 3 is studied by many researchers for comparison purposes. It is simple enough to take area of cross section of each member as a discrete design variable. This structure has been previously studied for discrete design variables by Rajeev and Krishnamoorthy (1992) using GA, Kripka (2004) using SA (Simulated Annealing), Li *et al.* (2009) using HPSO, Camp and Bichon (2004) using ACO, Camp (2007) using BB-BC (Bing Bang-Big Crunch), and Sönmez (2011) using ABC. The geometry, support, and loading conditions for the ten-bar cantilevered truss are also illustrated in Fig. 3. All members were assumed to be constructed from a material with the Young modulus, E, of 10^4 ksi, the density, ρ, of 0.10 lb/in^3, and the allowable

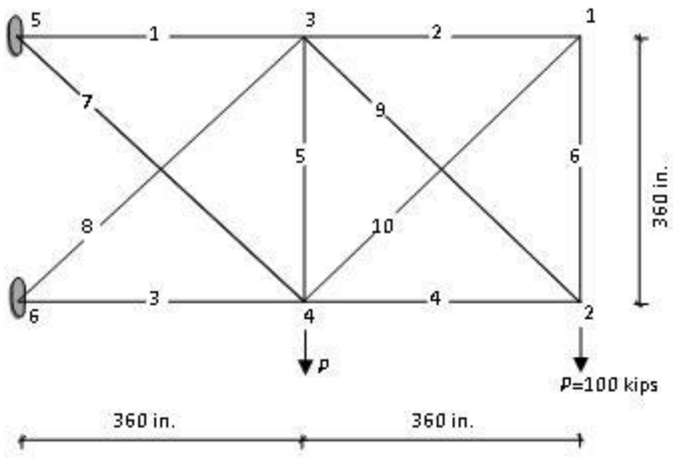


Fig. 3 Configuration of ten bar truss

Table 1 Designs for ten bar truss

Element group	Member	Cross-sectional areas (in.^{2}) GA	SA	HPSO	ACO	BB-BC	ABC	TLBO
1	1	33.50	33.50	30.00	33.50	33.50	33.50	33.50
2	2	1.62	1.62	1.62	1.62	1.62	1.62	1.62
3	3	22.90	22.90	22.90	22.90	22.90	22.90	22.90
4	4	14.20	14.20	13.50	14.20	14.20	14.20	14.20
5	5	1.62	1.62	1.62	1.62	1.62	1.62	1.62
6	6	1.62	1.62	1.62	1.62	1.62	1.62	1.62
7	7	7.97	7.97	7.97	7.97	7.97	7.97	7.97
8	8	22.90	22.90	26.50	22.90	22.90	22.90	22.90
9	9	22.00	22.00	22.00	22.00	22.00	22.00	22.00
10	10	1.62	1.62	1.80	1.62	1.62	1.62	1.62
f_{best} (lb)		5,490.74	5,490.74	5,531.98	5,490.74	5,490.74	5,490.74	5,490.74
$f_{avg.}$ (lb)		-	-	-	5,510.52	5,494.17	5,510.35	5,510.54
$f_{std.}$ (lb)		-	-	-	23.19	12.42	21.513	22.22
$N_{analyses}$		8,000	-	50,000	10,000	8,694	25,800	8,040
Violation								0.00

Note: GA=Mahfouz (1999); SA=Kripka (2004); HPSO=Li et al. (2009); ACO=Camp and Bichon (2004) BB-BC=Camp (2007); ABC=Sönmez (2011); TLBO=This study.

stress of ±25 ksi. The objective of the problem is to minimize the weight of the structure. Constraints are imposed on member stresses (excluding buckling) and displacements. The displacements of the free nodes in both directions had to be less than ±2 in. The prescribed sections used for the possible cross-sectional areas for each member are 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5 in^2 (Camp and Bichon 2004) .

Table 1 summarizes the best designs developed by Mahfouz (1999), Kripka (2004), Li *et al.*

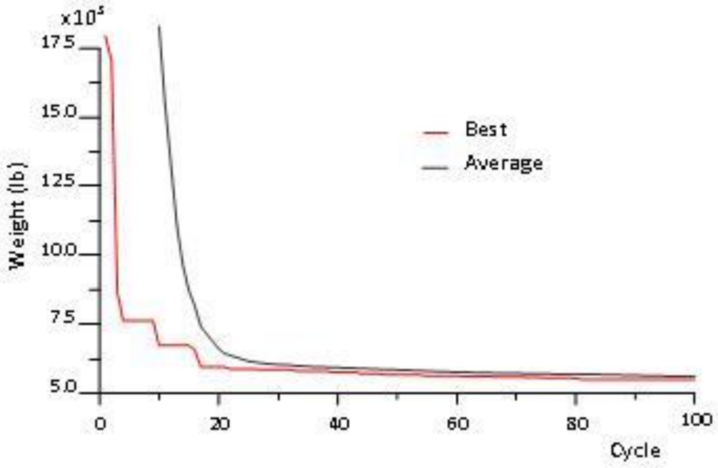


Fig. 4 Convergence history of ten bar truss

(2009), Camp and Bichon (2004), Camp (2007), Sönmez (2011), and by the TLBO algorithm explained in this study. The reported minimum number of truss analyses in the given references is 8,000 (Mahfouz 1999) and 8,694 (Camp 2007) to converge to a solution. Therefore, 100 generations with a population size 40 resulting in 8,040 truss analyses are taken into account in TLBO algorithm as the termination criteria and the best design developed by the TLBO algorithm, a truss weighting 5,490.74 lb, is presented in the last column of Table 1. The average and the standard deviation of the designs obtained for a series of 20 design runs are also included in Table 1.

It is worthy to state that the result obtained in this study show a remarkable agreement with the previous studies including Mahfouz (1999), Camp and Bichon (2004), Camp (2007), and Sönmez (2011). However, the TLBO algorithm exhibits more computational efficiency over HPSO, ACO, and ABC from the truss analyses point of view. Moreover, the number of truss analyses required for the TLBO algorithm is lighter than the BB-BC algorithm while it is the same with the GA. The number of truss analyses is also decreased and increased to investigate the effects on the best solution obtained using TLBO. It is observed that when the number of truss analyses is varied from 8,000 to 10,000, 12,000, and 25,000, respectively, the best solution to be reached is the same with the one obtained using GA, ACO, BB-BC, ABC. However, it is not encountered any solution being lighter than or the same with the best solution presented in Table 1 when the number of truss analyses is changed from 8,000 to 7,000 and 6,500. Typical design history for the best optimum design and average truss weight of 20 designs for the 10-bar truss is illustrated in Fig. 4.

4.2 Twenty-five bar space truss design

Another benchmark problem used to test the improvements in the algorithm proposed by the researcher is the 25-bar space truss shown in Fig. 5. Members of the truss are organized into 8 groups, and the displacements at joints are restricted at 0.35 in. in the directions of x, y and z, respectively. A set of available sections used for this problem is 0.1 to 3.4 in.^{2} with a 0.1 in.^{2} increment (Camp *et al.* 1998, Camp *et al.* 2005, Camp 2007). The young modulus, E, is 10^7 psi, the density, ρ, is 0.1 lb/in.^{3}, and the allowable stress is ±40 ksi. A single load case applied to the truss is presented in Table 2.

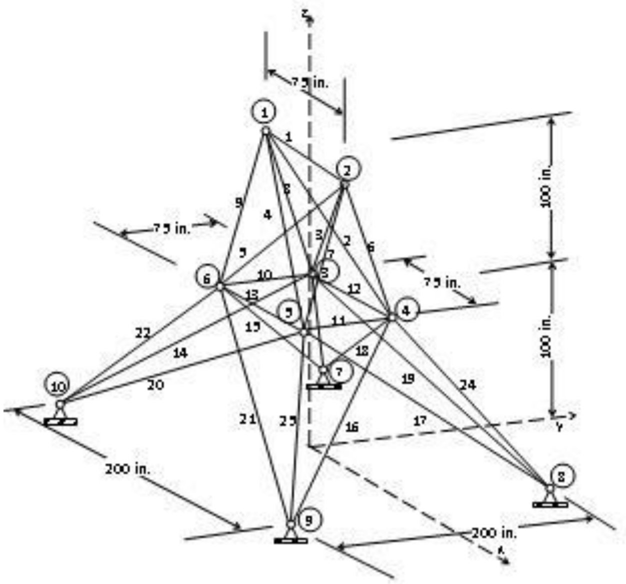


Fig. 5 Configuration of the 25-bar truss

Table 2 Load case for 25-bar truss

Node	F_x (kip)	F_y (kip)	F_z (kip)
1	1.0	-10.0	-10.0
2	0.0	-10.0	-10.0
3	0.5	0.0	0.0
6	0.6	0.0	0.0

Table 3 Designs for the 25 bar truss

		Cross-sectional areas (in.^{2})						
Element group	Member	GA	SA	HPSO	ACO	BB-BC	ABC	TLBO
1	1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	2-5	0.5	0.4	0.3	0.3	0.3	0.3	0.3
3	6-9	3.4	3.4	3.4	3.4	3.4	3.4	3.4
4	10, 11	0.1	0.1	0.1	0.1	0.1	0.1	0.1
5	12, 13	1.9	2.2	2.1	2.1	2.1	2.1	2.1
6	14-17	0.9	1.0	1.0	1.0	1.0	1.0	1.0
7	18-21	0.5	0.4	0.5	0.5	0.5	0.5	0.5
8	22-25	3.4	3.4	3.4	3.4	3.4	3.4	3.4
f_{best} (lb)		485.05	484.33	484.85	484.85	484.85	484.85	484.85
$f_{avg.}$ (lb)		-	-	-	486.46	485.10	484.94	486.54
$f_{std.}$ (lb)		-	-	-	4.71	0.44	-	2.74
$N_{analyses}$		-	40,000	25,000	7,700	2,420	24,250	2,420
Violation								0.00

Note: GA=Camp et al. (1998); SA=Kripka (2004); HPSO=Li et al. (2009); ACO=Camp and Bichon (2004); BB-BC=Camp (2007); ABC=Sönmez (2011); TLBO=This study.

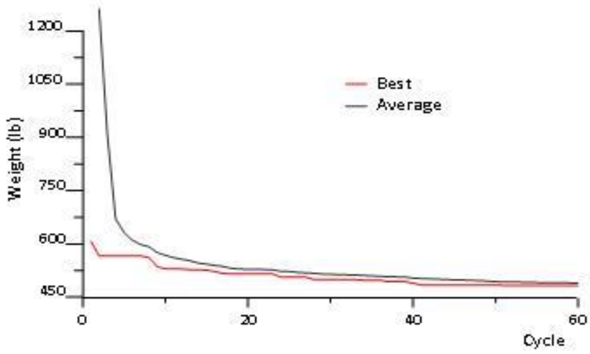


Fig. 6 Convergence history of the 25 bar truss

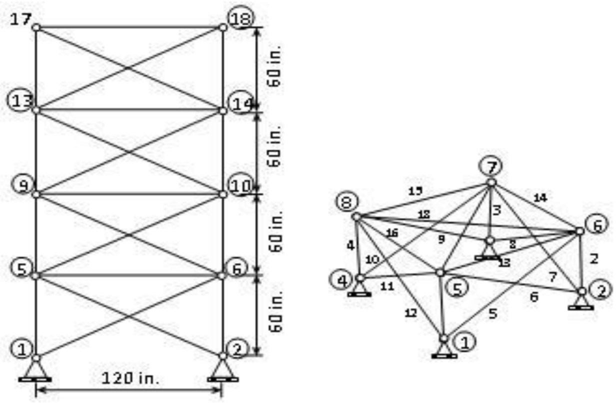


Fig. 7 Geometry and element of the 72-bar truss with node and element numbering schemes

The best truss design that weighs 484.85 lb developed by the TLBO and the other algorithms are presented in Table 3. The TLBO algorithm produces identical design to that found by Li *et al.* (2009), Camp and Bichon (2004), Camp (2007), and Sönmez (2011). This design is lighter than the design presented by Camp *et al*. (2005) while it is roughly the same with that published by Kripka (2004). However, the TLBO algorithm required a 2,420 truss analyses to converge to a solution, which is very small than the 40,000 analyses required by Kripka (2004). The number of truss analyses required for the TLBO algorithm is lighter than the algorithms listed in Table 3 excepting the BB-BC. Although both the TLBO and the BB-BC need nearly the same number of truss analyses, the BB-BC algorithm found the best design in Phase 2 scheme following a procedure called Phase 1 requiring the 6,670 truss analyses. 60 generations with a population size of 20 resulting in 2,420 truss analyses are taken into account in TLBO algorithm as the termination criterion since the reported minimum number of truss analyses in the given references is 2,420 (Camp 2007). Fig. 6 shows the design history for the best optimum design and average truss weight of 20 designs for the 25-bar truss.

4.3 Seventy-two bar space truss design

Fig. 7 shows the configuration of the 72-bar space truss and its node and element numbering

Table 4 Load cases for the 72-bar truss

Case	Node	F_x (kip)	F_y (kip)	F_z (kip)
1	17	0.0	0.0	-5.0
	18	0.0	0.0	-5.0
	19	0.0	0.0	-5.0
	20	0.0	0.0	-5.0
2	17	5.0	5.0	-5.0

Table 5 Designs for the 72 bar truss

		Cross-sectional areas (in.^{2})					
Element group	Member	GA	HPSO	ACO	BB-BC	ABC	TLBO
1	1–4	0.161	2.100	1.948	1.858	0.156	1.877
2	5–12	0.544	0.600	0.508	0.506	0.553	0.516
3	13–16	0.379	0.100	0.101	0.100	0.391	0.100
4	17,18	0.521	0.100	0.102	0.100	0.597	0.100
5	19–22	0.535	1.400	1.303	1.248	0.520	1.270
6	23–30	0.535	0.500	0.511	0.527	0.515	0.513
7	31–34	0.103	0.100	0.101	0.100	0.101	0.100
8	35,36	0.111	0.100	0.100	0.101	0.103	0.100
9	37–40	1.310	0.500	0.561	0.521	1.271	0.517
10	41–48	0.498	0.500	0.492	0.517	0.512	0.517
11	49–52	0.110	0.100	0.100	0.100	0.100	0.100
12	53,54	0.103	0.100	0.107	0.101	0.100	0.100
13	55–58	1.910	0.200	0.156	0.157	1.843	0.157
14	59–66	0.525	0.500	0.550	0.551	0.517	0.546
15	67–70	0.122	0.300	0.390	0.392	0.102	0.409
16	71,72	0.103	0.700	0.592	0.592	0.100	0.567
f_{best} (lb)		383.12	388.94	380.24	379.85	379.89	379.70
$f_{avg.}$ (lb)		-	-	383.16	382.08	380.05	382.56
$f_{std.}$ (lb)		-	-	3.66	1.912	-	5.48
$N_{analyses}$		-	50,000	18,500	19,621	50,000	6,440
Violation							0.00

Note: GA=Erbatur et al. (2000); HPSO=Li et al. (2009); ACO=Camp and Bichon (2004); BB-BC=Camp (2007); ABC=Sönmez (2011); TLBO=This study.

patterns. The structural members of the space truss are divided into 16 groups after being connected in order to impose symmetry. The material has a modulus of elasticity of 10^7 psi and a mass density of 0.1 lb/in.^{3}. The maximum displacement at the upper most joints 1, 2, 3, and 4 in either *x*, *y*, or *z* directions are not allowed to exceed 0.25 in. and the allowable stress for all members is ±25 ksi. The set of available sections used for this problem is 0.1 to 3.0 in.^{2} with a 0.001 in.^{2} increment (Erbatur *et al.* 2000). Table 4 lists the values and directions of the two independent loads cases applied to the 72-bar space truss.

The lightest 72-bar truss designed by the TLBO algorithm weighs 379.70 lb, which is 0.04% lighter than the best discrete variable design presented by (Camp 2007). In a series of 20 design runs, the average weight of TLBO algorithm designs is 382.56 lb with a standard deviation of 5.48 lb and the percent difference between the best solution and the average solution is 0.75%.

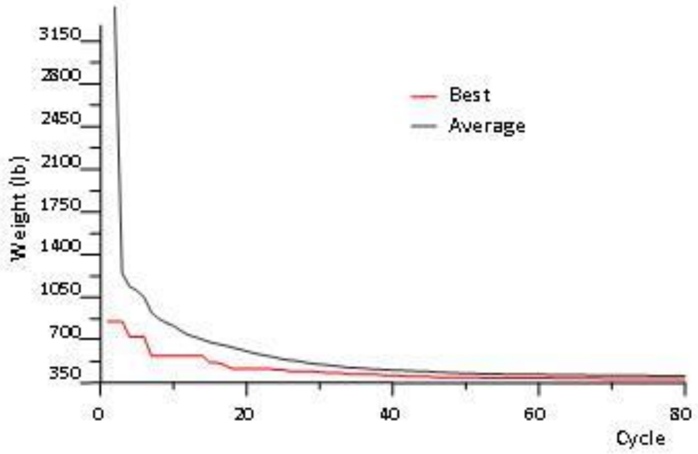


Fig. 8 Convergence history of the 72 bar truss

The reported minimum number of truss analyses among the given references is 18,500 (Camp and Bichon 2004) and 80 generations with a population size 40 resulting in 6,440 truss analyses are taken into account in TLBO algorithm as the termination criterion. Table 5 compares the TLBO design with other optimization techniques. The TLBO algorithm requires a 6,440 truss analyses to converge to a solution, which is a significant reduction in computational cost when compared to the number of analyses required by the HPSO, ACO, and ABC designs. The design history for the best optimum design and average truss weight of 20 designs for the 72-bar truss are shown in Fig. 8.

4.4 582 Bar space tower design

The geometry and group numbering of a 582 bar space tower, previously studied by Sönmez (2011) are given in Fig. 9. The structural members of the space tower are linked together into 32 groups. The modulus of elasticity and the material density of all members are 29000 ksi and 0.283 lb/in.^{3}, respectively. The members are subjected to stress limitations of ±21.6 ksi. The maximum displacement of all the nodes is not allowed to exceed 3.15 in. for all directions. A single loading condition is considered to be applied such that the lateral loads of 1.12 kips are applied to all nodes in both x and y-directions, and vertical loads of −6.74 kips and −3.37 kips are applied, respectively, to all nodes in the upper and lower parts of the tower in z direction. The maximum slenderness ratio of i-th member is limited to 300 and 200 for tension and compression, respectively ($\lambda_i = K_i L_i / r_i \leq \lambda_{allowed}$, in here K_i is the effective length factor which was taken to be 1, L_i is the length and r_i is minimum radii of gyration). A W-shape list of AISC profiles used in this problem is presented in Table 6.

Table 7 lists the designs developed by the TLBO algorithm and the others. The lightest TLBO design results a tower weight of 363,568.37 lb, which is 0.6% lighter than the one obtained using ABC (Sönmez 2011). In a series of 20 design runs, the average weight of TLBO algorithm design is 363,666.22 lb with a standard deviation of 50.64 lb and the percent difference between the best solution and the average solution is 0.0027%. The TLBO algorithm requires approximately 35,050

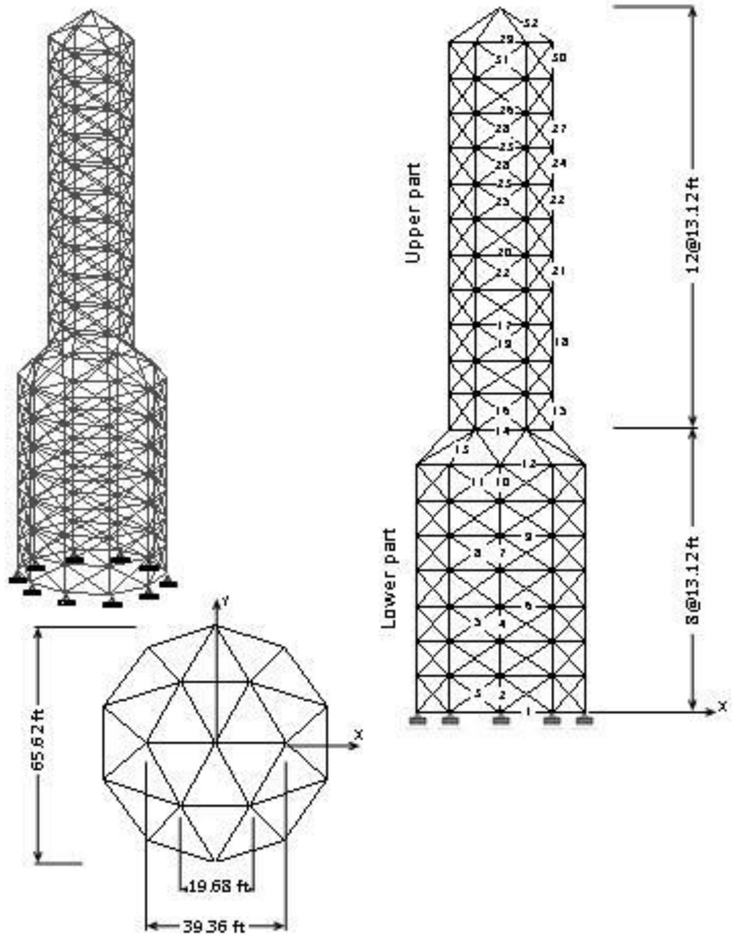


Fig. 9 The 582-bar space tower

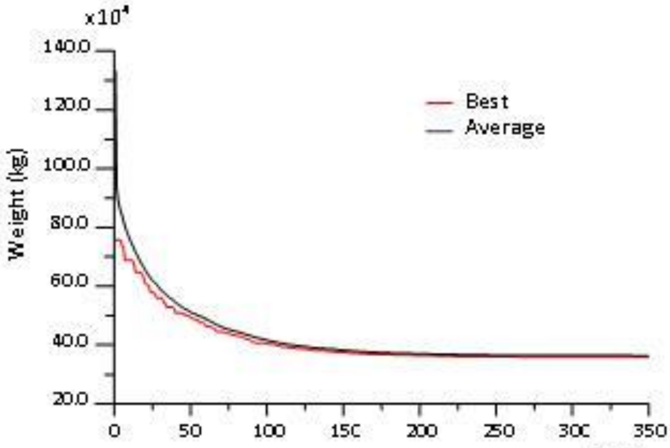


Fig. 10 Convergence history of the 582 bar space tower

frame analyses in order to yield to the best optimum design (350 generations with a 50 population size).

TLBO enables to reach the best optimum design with less truss analyses when compared to 98,650 analyses required by ABC. Therefore, it can be concluded that the TLBO algorithm exhibits well computational efficiency than ABC from the point of view of the best optimum design. The design history for the best optimum design and the average truss weight of 20 designs for the 582-bar space tower are shown in Fig. 10.

Table 6 W shape profile list designs from AISC for the 582 bar space tower

W-shape profile list					
W8 × 21	W10 × 54	W14 × 99	W12 × 152	W24 × 229	W30 × 292
W10 × 22	W12 × 58	W16 × 100	W18 × 158	W36 × 230	W40 × 297
W8 × 24	W10 × 60	W10 × 100	W14 × 159	W44 × 230	W36 × 300
W6 × 25	W14 × 61	W21 × 101	W27 × 161	W12 × 230	W14 × 311
W12 × 26	W21 × 62	W24 × 104	W24 × 162	W14 × 233	W33 × 318
W8 × 28	W12 × 65	W12 × 106	W12 × 170	W30 × 235	W30 × 326
W12 × 30	W16 × 67	W14 × 109	W30 × 173	W27 × 235	W36 × 328
W14 × 30	W10 × 68	W21 × 111	W40 × 174	W33 × 241	W44 × 335
W8 × 31	W12 × 72	W10 × 112	W24 × 176	W36 × 245	W14 × 342
W10 × 33	W14 × 74	W27 × 114	W14 × 176	W40 × 249	W33 × 354
W14 × 34	W18 × 76	W30 × 116	W27 × 178	W24 × 250	W36 × 359
W8 × 35	W10 × 77	W24 × 117	W21 × 182	W12 × 252	W14 × 370
W16 × 36	W12 × 79	W33 × 118	W12 × 190	W14 × 257	W14 × 398
W14 × 38	W14 × 82	W18 × 119	W30 × 191	W27 × 258	W14 × 426
W10 × 39	W27 × 84	W14 × 120	W24 × 192	W36 × 260	W14 × 455
W8 × 40	W18 × 86	W21 × 122	W14 × 193	W30 × 261	W14 × 500
W12 × 40	W12 × 87	W24 × 131	W27 × 194	W44 × 262	W14 × 550
W14 × 43	W10 × 88	W14 × 132	W40 × 199	W33 × 263	W14 × 605
W12 × 45	W16 × 89	W12 × 136	W33 × 201	W40 × 277	W14 × 665
W10 × 45	W14 × 90	W14 × 145	W30 × 211	W12 × 279	W14 × 730
W14 × 48	W21 × 93	W27 × 146	W14 × 211	W24 × 279	
W10 × 49	W27 × 94	W24 × 146	W40 × 215	W36 × 280	
W12 × 50	W12 × 96	W21 × 147	W27 × 217	W14 × 283	
W12 × 53	W18 × 97	W36 × 150	W33 × 221	W33 × 291	

Table 7 Designs for the 582 bar space tower

	Cross-sectional areas (in.^{2})	
Element group	ABC	TLBO
1	W8 × 21	W8 × 21
2	W10 × 77	W14 × 74
3	W8 × 24	W8 × 24
4	W14 × 61	W14 × 61
5	W8 × 24	W8 × 24
6	W8 × 21	W8 × 21
7	W12 × 50	W10 × 49
8	W8 × 24	W8 × 24
9	W8 × 21	W8 × 21
10	W10 × 49	W14 × 48

Table 7 Continued

11	W8 × 24	W8 × 24
12	W10 × 68	W16 × 67
13	W18 × 76	W10 × 77
14	W14 × 48	W12 × 50
15	W10 × 77	W14 × 82
16	W8 × 31	W8 × 24
17	W8 × 21	W8 × 21
18	W21 × 62	W12 × 65
19	W8 × 24	W8 × 24
20	W8 × 21	W8 × 21
21	W14 × 43	W12 × 45
22	W8 × 24	W8 × 24
23	W8 × 21	W8 × 21
24	W8 × 24	W12 × 26
25	W8 × 24	W8 × 24
26	W8 × 21	W8 × 21
27	W8 × 21	W8 × 21
28	W8 × 24	W8 × 24
29	W8 × 21	W8 × 21
30	W8 × 21	W8 × 21
31	W8 × 24	W8 × 24
32	W8 × 24	W8 × 24
f_{best} (lb)	365906.3	363568.37
$f_{avg.}$ (lb)	366088.4	363666.22
$f_{worst.}$ (lb)	369162.2	368759.60
$N_{analyses}$	98650	35050
Violation		0.00

Note: ABC=Sönmez [22]; TLBO=This study.

5. Conclusions

A optimization method, TLBO, based on concepts proposed by Rao *et al.* (2011), is applied to discrete forms of structural optimization to design low-weight trusses. Through a series of benchmark-type, except for 582 bar space tower, truss optimization problems, the TLBO algorithm demonstrates that it can routinely minimize the overall weight of truss structures while satisfying material and performance constraints.

The main characteristics of the TLBO algorithm are its simplified numerical structure and its independence on a number of parameters to define the algorithm's performance. TLBO does not need any algorithm parameters to be tuned. In contrast, GA requires the crossover probability, mutation rate, and selection method; PSO requires learning factors, the variation of weight, and the maximum value of velocity; ABC requires the limit value; and HS requires the harmony memory consideration rate, pitch adjusting rate, and number of improvisations (Rao *et al.* 2011), simple ACO requires four parameters with an additional search space reduction parameter for multiphase applications, and BB-BC requires the upper limit value on the search space, the center of mass weighting factor with an additional search space reduction parameter for multiphase applications.

From results presented in this study, the TLBO algorithm seems to demonstrate good performance as well as GA, SA, HPSO, ACO, BB-BC and ABC techniques. The consistency of the results is proved by the small deviation of the average solution from the best solution (averaged less than 0.8% for all presented example designs).

Consequently, the TLBO algorithm works effectively, shows well performance and can be applied efficiently for the optimization of engineering design applications.

References

Camp, C., Pezeshk, S. and Cao, G. (1998), "Optimized design of two dimensional structures using genetic algorithm", *J. Struct. Eng.*, **124**(5), 551-559.

Camp, C.V. (2007), "Design of space trusses using big bang–big crunch optimization", *J. Struct. Eng.*, **133**(7), 999-1008.

Camp, C.V. and Bichon, B.J. (2004), "Design of space trusses using ant colony optimization", *J. Struct. Eng.*, **130**(5), 741-751.

Camp, C.V., Bichon, B.J. and Stovall, S.P. (2005), "Design of steel frames using ant colony optimization", *J. Struct. Eng.*, **131**(3), 369-379.

Capriles, P.V.S.Z., Fonseca, L.G., Barbosa, H.J.C. and Lemonge, A.C.C. (2007), "Rank-based ant colony algorithms for truss weight minimization with discrete variables", *Commun. Numer. Meth. Eng.*, **23**, 553-575.

Chen, S.Y. and Rajan, S.D. (2000), "A robust genetic algorithm for structural optimization", *Struct. Eng. Mech.*, **10**(4), 313-336.

Deb, K. (2000), "An efficient constraint handling method for genetic algorithm", *Comput. Methods. Appl. Mech. Eng.*, **186**, 311-338.

Değertekin, S.O. (2007), "Optimum design of steel frames using harmony search algorithm", *Struct. Multidisc. Optim.*, **36**, 393-401.

Erbatur, F., Hasançebi, Ö., Tütüncü, I. and Kılıç, H. (2000), "Optimal design of planar and space structures with genetic algorithms", *Comput. Struct.*, **75**, 209-224.

Karaboğa, D. (2005), "An idea based on honey bee swarm for numerical optimization", Technical Report, TR06, Erciyes University, Turkey.

Karaboğa, D. and Baştürk, B. (2007), "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", *Proceeding of LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, Cancun, Mexico, June.

Kaveh, A. and Talatahari, S. (2010), "Optimal design of truss structures via the charged system search algorithm", *Struct. Multidisc. Optim.*, **37**, 893-911.

Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", *Proceeding of IEEE International Conference on Neural Networks*, Perth, Western Australia, November.

Kripka, M. (2004), "Discrete optimization of trusses by simulated annealing", *J. Braz. Soc. Mech. Sci. Eng.*, **XXVI**, 170-173.

Lee, K.S. and Geem, Z.W. (2004), "A new structural optimization method based on the harmony search algorithm", *Comput. Struct.*, **82**, 781-798.

Li, L.J., Huang, Z.B. and Liu, F. (2009), "A heuristic particle swarm optimization method for truss structures with discrete variables", *Comput. Struct.*, **87**, 435-444.

Mahfouz, S.Y. (1999), "Design optimization of structural steelwork", Ph.D. thesis, Univ. Of Bradford, Yorkshire.

Nanakorn, P. and Meesomklin, K. (2001), "An adaptive penalty function in genetic algorithms for structural design optimization", *Comput. Struct.*, **79**, 2527-2539.

Rajeev, S. and Krishnamoorthy, C.S. (1992), "Discrete optimization of structures using genetic algorithms", *J. Struct. Eng.*, **118**(5), 1233-1250.

Rao, R.V., Savsani, V.J. and Vakharia, D.P. (2011), "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems", *Comput-Aided Des.*, **43**, 303-315.

Saka, M.P. (2007), "Optimum geometry design of geodesic domes using harmony search algorithm", *Adv. Struct. Eng.*, **10**(6), 595-606.

Sönmez, M. (2011), "Discrete optimum design of truss structures using artificial bee colony algorithm", *Struct. Multidisc. Optim.*, **43**, 85-97.

Toğan, V. (2011), "Design of planar steel frames using Teaching-Learning Based Optimization", *Eng. Struct.*, **34**, 225-232.

Toğan, V. and Daloğlu, A. (2006), "Optimization of 3d trusses with adaptive approach in genetic algorithms", *Eng. Struct.*, **28**, 1019-1027.

Toğan, V. and Daloğlu, A. (2006), "Shape and size optimization of 3d trusses with genetic algorithm", *Technical J. Turkish Chamber Civil Eng.*, **17**, 3809-3826.

Toğan, V. and Daloğlu, A. (2008), "An improved genetic algorithm with initial population strategy and self-adaptive member grouping", *Comput. Struct.*, **86**, 1204-1218.

Toğan, V. and Daloğlu, A. (2009), "Bridge truss optimization under moving load using continuous and discrete design variables in optimization methods", *Indian J. Eng. Mater. Sci.*, **16**, 245-258.