Journal of Optimization in Industrial Engineering
Vol.13, Issue 1, Winter & Spring 2020, 177-194
DOI:10.22094/JOIE.2019.1866904.1649

Research Article

Hybrid Teaching-Learning-Based Optimization and Harmony Search for Optimum Design of Space Trusses

Siamak Talatahari [a,b,*], Vahid Goodarzimehr [a], Nasser Taghizadieh[a]

[a] *Department of Civil Engineering, University of Tabriz, Tabriz, Iran*

[b] *Engineering Faculty, Near East University, Turkey*

Received 24 April 2019; Revised 27 August 2019; Accepted 01 September 2019

Abstract

The Teaching-Learning-Based Optimization (TLBO) algorithm is a new meta-heuristic algorithm which recently received more attention in various fields of science. The TLBO algorithm divided into two phases: Teacher phase and student phase; In the first phase a teacher tries to teach the student to improve the class level, then in the second phase, students increase their level by interacting among themselves. But, due to the lack of additional parameter to calculate the distance between the teacher and the mean of students, it is easily trapped at the local optimum and make it unable to reach the best global for some difficult problems. Since the Harmony Search (HS) algorithm has a strong exploration and it can explore all unknown places in the search space, it is an appropriate complement to improve the optimization process. Thus, based on these algorithms, they are merged to improve TLBO disadvantages for solving the structural problems. The objective function of the problems is the total weight of whole members which depends on the strength and displacement limits. Indeed, to avoid violating the limits, the penalty function applied in the form of stress and displacement limits. To show the superiority of the new hybrid algorithm to previous well-known methods, several benchmark truss structures are presented. The results of the hybrid algorithm indicate that the new algorithm has shown good performance.

Keywords: Teaching-learning-based optimization; Harmony search; Size optimization; Structural optimization; Continuous variables

1. Introduction

During recent years, shape and size optimization of structures has received very interestingly and meta-heuristic methods seem to be one of the best alternatives for solving this problem. Due to the high potential of the meta-heuristic methods to achieve the optimum design, many methods have been developed for structural optimization, such as Genetic Algorithms (GAs) models the process of natural evolution (Adeli and Cheng, 1994, Kaveh and Rahami 2016, Kaveh and Bijari, 2018) , Particle swarm optimization (PSO) is inspired from social behavior and instruction between a flock of birds (Kennedy and Eberhart, 1955), Harmony Search (HS) (Lee and Geem, 2004) is based on the behavior of musicians to find a better harmony, Ant Colony Optimization (ACO) (Dorigo, 1992) is developed for discrete optimization, Teaching-Learning-Based Optimization (TLBO) (Rao et al., 2011) simulates class-level learning, Charged System Search (CSS) (Kaveh and Talatahari, 2010) uses the governing Coulomb law from electrostatics and the Newtonian laws of mechanics, Hybrid Teaching-Learning-Based Optimization algorithm (HTLBO) (Talatahari and Goodarzimehr, 2019) is based on the standard TLBO and HS algorithm, Vibrating Particles System (VPS) (Kaveh and Ilchi, 2017) uses a free vibration of single degree of freedom systems with viscous damping, Big Bang-Big Crunch (Kaveh and Talatahari, 2009) uses the population averaging in a model of the evolution of the universe, Artificial Bee Colony algorithm (ABC) (Sonmez, 2011) is motivated by the intelligent behavior of honey bees, A Hybrid Particle Swarm Optimization and Genetic Algorithm (HPSOGA) (Omidinasab and Goodarzimehr, 2019) is based on the standard PSO and GA algorithm, and so on.

All of these meta-heuristic methods have a high power to solve the optimization problems and due to this advantage, they have been used in various fields of science. The algorithms have many similarities and most of them are population-based methods. It means that each of them has a population of solutions which obtain an optimal design in an iteration and selection process. But it's not possible to use an evolutionary algorithm to solve all the optimization problems. For this reason, we need to make some changes to the algorithm so that it can be used for various problems. Therefore, there are two challenges for these methods: The first one is in difficult problems that due to the large search space applies additional computational costs; The second is due to a large number of design variables and large feasible space, the algorithm would easily trap at a local search.

To overcome these problems, the TLBO and HS algorithms are merged because both of them have great complementary to each other. The TLBO advantage is on no required controlling parameters and the HS advantage is its exploration ability to discover all the unknown places in the search space. Also, the TLBO and HS algorithms have some disadvantages for solving difficult optimization problems. The TLBO falls easily at local points because there is no controlling parameter to measure the distance between the best student as a teacher and the mean of class. Vice versa, the HS disadvantage is

*Corresponding author Email address: siamak.talat@gmail.com

on its convergence speed. Therefore, it seems to hybridizing these methods results in an improved algorithm for solving the optimization problems.
Recently, several modified TLBO and HS algorithms have been presented; some of them are as: HS and PSO (HPSO) (Li et al., 2007), HS, PSO, and ACO (HPSACO) (Kaveh and Talatahari, 2009), Hybrid HS (HHS) (Cheng et al., 2016), Modified Teaching–Learning Based Optimization (TLBO) (Camp and Farshchin, 2014), Improved Teaching–Learning-Based Optimization (ITLBO) (Chen et al. 2015), Diversified Particle Swarm Optimization (DPSO) (Behnamian 2019), A Hybrid Grey based Two Steps Clustering and Firefly Algorithm (Faezy razi, Shadloo, 2017), A Multi-Objective Particle Swarm Optimization (MOPSO) (Fattahi, Samouei, 2016), time domain responses and Teaching-Learning-Based Optimization (TLBO) algorithm (Fallahian et al , 2018). However, developing a new method which improves the results or optimization process can be useful. In this paper, several truss structures with continuous design variables are tested using the hybrid TLBO and HS and the results are compared to standard TLBO and HS algorithm and some other meta-heuristic algorithms.

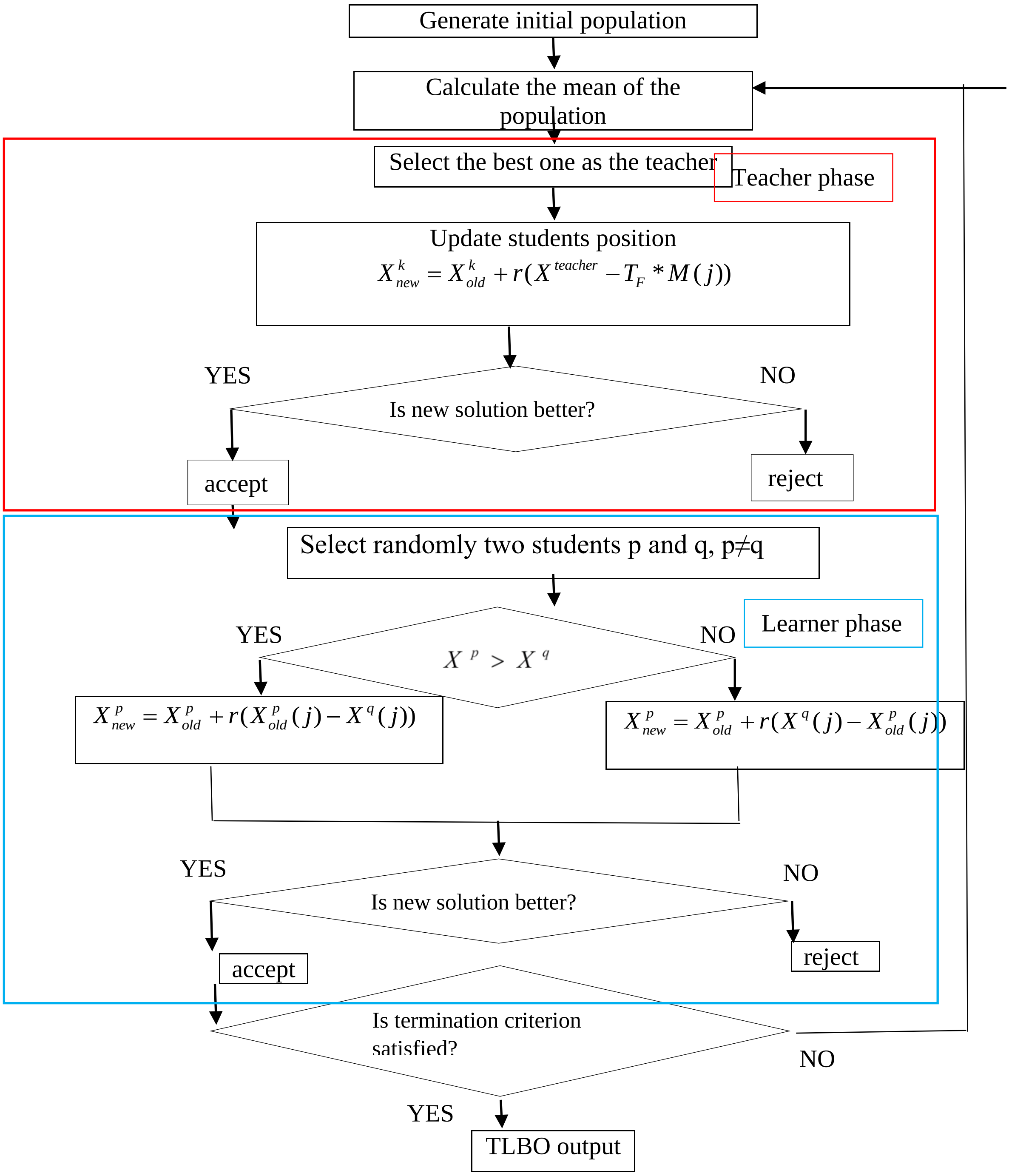

Fig. 1. The TLBO flow chart

2. Heuristic Teaching-learning-based Optimization and Harmony search for Truss Structures

2.1. Review of teaching-learning-based optimization algorithm

The TLBO algorithm is inspired by a classroom, which makes it able to provide a model for truss structures optimization. The TLBO algorithm at first proposed by Rao et al. (2011). In this method, a classroom consists of one teacher and some students, actually, one of the students who are better than other ones are chosen as the teacher. Then, the teacher increases the knowledge of the class-level by teaching the students. The number of students represents a population that is considered as design variables in various sciences. Figure 1 shows the optimization process.

2.1.1. Teacher phase

In this phase, at first the best student of the class is chosen as the teacher that tries to increase the average knowledge of the class-level. This phase can be formulated as:

$$X_{new}^{k} = X_{old}^{k} + r(X^{teacher} - T_F * M(j)) \quad (1)$$

$$M(j) = \frac{\sum_{K=1}^{N} \frac{X^{K}(j)}{F^{k}}}{\sum_{K=1}^{N} \frac{1}{F^{k}}} \quad (2)$$

Where the $X^{k}(j)$ denotes the *j*th design variable, T_F used as the teaching factor, *r* is a random number within the range of [0,1], *M(j)* denotes the mean of class. F^k is the penalized fitness function.

2.1.2. Learner Phase

In the learner phase, each learner increases their knowledge with the interaction between students and this procedure will be lead to an increase in overall knowledge of the class. The learner phase explained as below:

Randomly select *p* and *q* students from the class such a way that p and q are unequal, and we have:

If $X_p < X_q$

$$X_{new}^{p} = X_{old}^{p} + r(X_{old}^{p}(j) - X^{q}(j)) \quad (3)$$

Otherwise

$$X_{new}^{p} = X_{old}^{p} + r(X^{q}(j) - X_{old}^{p}(j)) \quad (4)$$

where *r* is random number within the range [0,1]. $X^P(j)$ denotes the *j*th design for the *p*th design vector.

2.2. Review of Harmony Search Algorithm

The HS algorithm is one of the easiest meta-heuristic methods that applied in the optimization problems, and it is inspired by the process of harmony such as during jazz improvisation. In other words, there is a similarity between finding an optimal solution to the optimization problem and the process of Jazz improvisation. This algorithm has lower mathematical requirements than other meta-heuristic methods. In the following, we intend to briefly explain the steps of the HS algorithm, which consist of step1 through 4. Figure 2 shows the optimization procedure of the HS algorithm.

Step 1: initialization: In the first step, the HS algorithm has several parameters that inquired to be adjusted to solve the optimization problem. Harmony Memory (HM), Harmony Memory Size (HMS), Harmony Memory Consideration Rate (HMCR), and Pitch Adjusting Rate (PAR). In this section, we must generate a population and store it in the HM.

$$HM = \begin{bmatrix} X^{1} \\ . \\ . \\ X^{HMS} \end{bmatrix}_{HMS \times mg} \quad (5)$$

Step 2: Initialize a new harmony from the HM: The HMCR is within [0,1] and used for considering the HM and choosing a new vector from the previous value. And (1-HMCR) sets the rate of randomly choosing one value from a possible range of values.

$$X_i^K = \begin{cases} select\ from\ \{X_i^1, X_i^2, \ldots\ldots\ldots, X_i^{HMS}\} & w.p.HMCR \\ select\ from\ the\ possible\ range & w.p.\ (1-HMCR) \end{cases} \quad (6)$$

Step 3: Updating the harmony memory :If a new harmony vector is better than the worst harmony in the HM, judged in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 4: Terminating criterion controlling: Repeat Steps 2 and 3 until the terminating criterion is satisfied.

Initialize the harmony memory(HM) with random vectors as many as the value

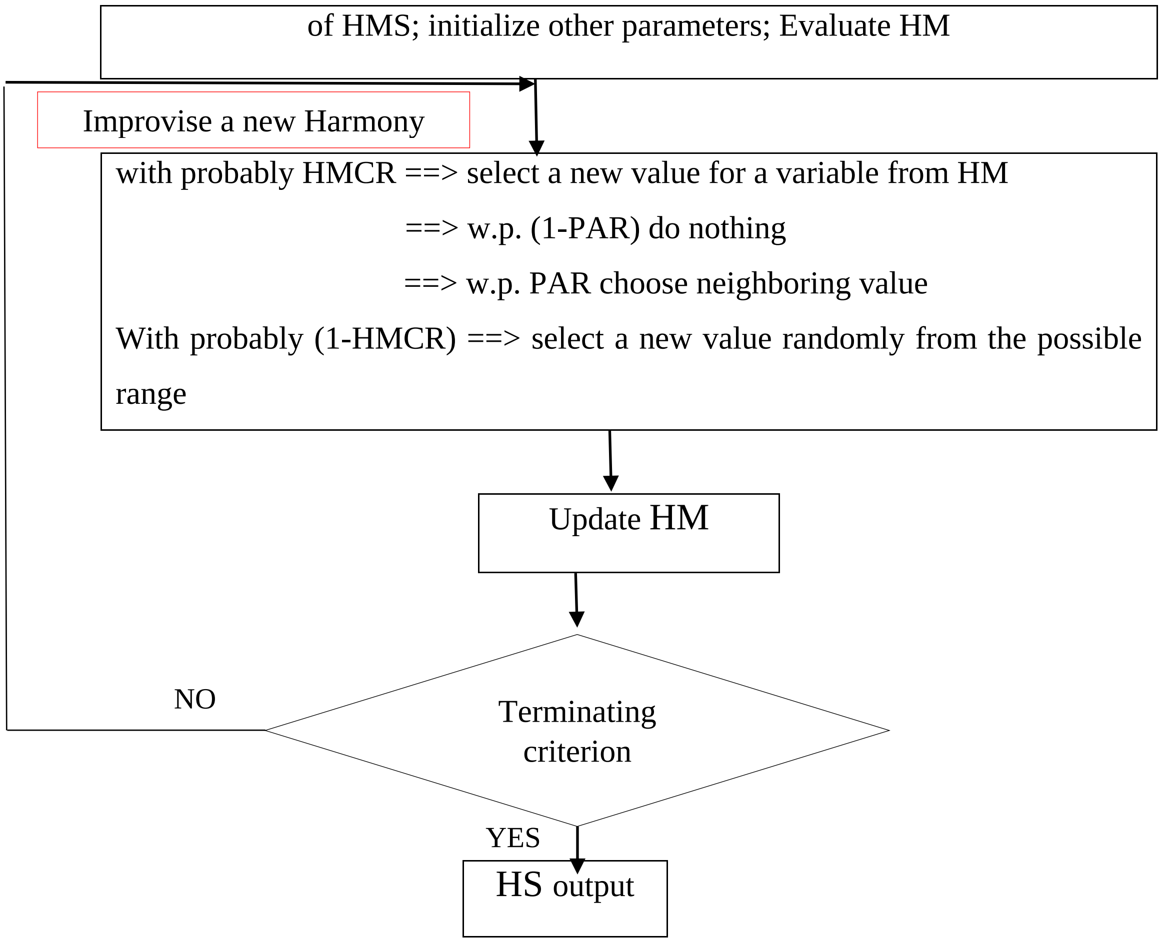

Fig. 2. The HS flow chart

3. Hybrid TLBO and HS Algorithm

After recognizing the best student in the class, TLBO algorithm tries to teach the rest of the class by using their knowledge to improve the level of the whole class. Then in the next stage, the students also try to improve their level by sharing information. In each iteration, the teacher will update their position based on its distance to the calculated mean of the class. An individual student also updates its position based on its distance to a randomly selected student from the class.

Eq. (1) shows that the teacher only improves the class level by using the mean of the students and the distance between the teacher and the mean of the class is not considered in the teacher phase. This makes the TLBO algorithm trapping at local points. In the next stage (learner phase), the optimization procedure may continue with a local point. But, the vector of learners is used as a vector of harmony entrance and then due to the strong exploration of the HS algorithm the optimization process will continue until the global optimum earned. So, we used HS-based mechanism to solve this problem. The new hybrid algorithm is presented as follows:

Step 1: As conventional TLBO algorithm, initialize a population of students; these students are like harmony in the HS algorithm.
Step 2: Calculate the mean of the population using the Eq. (2), because our perception of the class progress is under the improvement of the mean of the class level.
Step 3: Consider individual student fitness in order to choose the best one as the teacher. The teacher phase will be applying using the Eq. (1).
Step 4: Apply learner phase using Eq. (3 and 4).
Step 5: Generate a new student as:

$$X_{i,j} = \begin{cases} w.p.HMCR ==> select\ a\ new\ value\ from\ X_i^k \\ \qquad ==> w.p.(1-PAR)\ do\ nothing \\ \qquad ==> w.p.PAR\ choose\ neighboring\ value \\ w.p.(1-HMCR) ==> select\ a\ new\ value\ randomly \end{cases} \tag{7}$$

where $X_{i,j}$ is the jth variable of student i, the $HMCR$ is varying within [0,1] which sets a rate of choosing a value from the historic values stored in the X_i^k, (1-$HMCR$) sets the rate of choosing one value from the possible list of values. The pitch adjusting process is performed only after a value is chosen from X_i^k . the value (1-PAR) sets the rate of doing nothing, A PAR (pitch adjusting rate) of

0.5 indicates that the algorithm will choose a neighboring value with 50%×HMCR probability (Kaveh and Talatahari 2009). The scheme of the hybrid algorithm is shown in the figure 3.

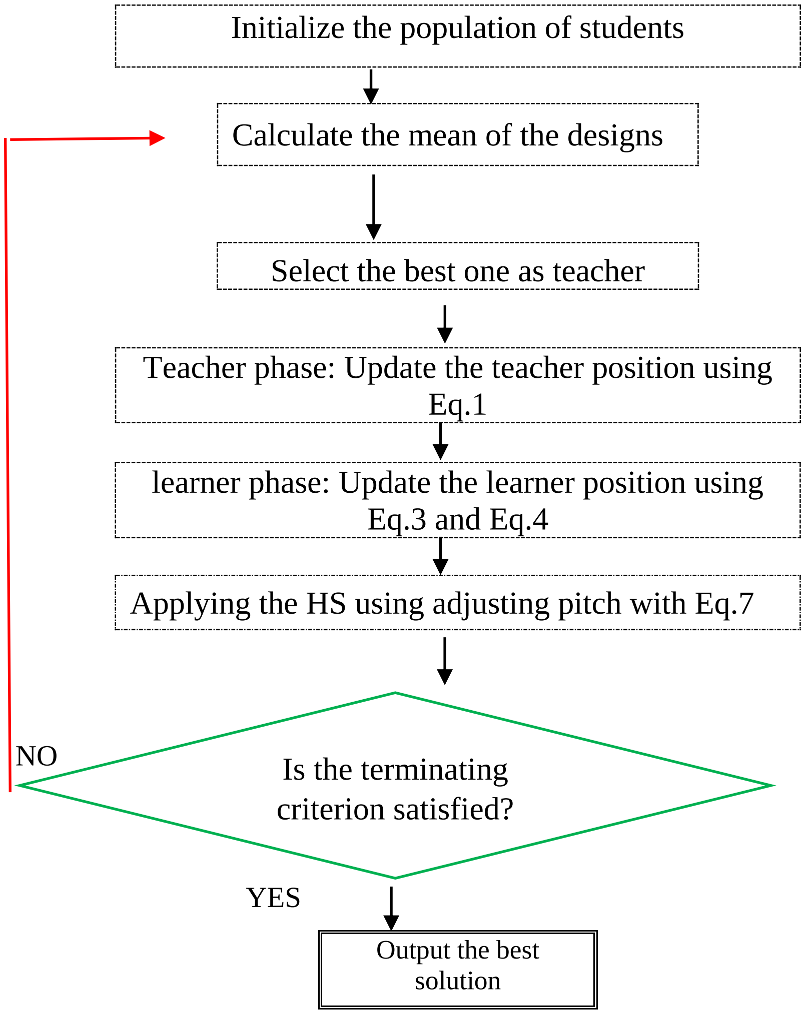

Fig. 3. the Hybrid TLBO and HS flow chart

4. Design of Truss Structure

One of the most common methods to optimizing truss structures is to minimize the cross-sectional area of elements which lead to an accepted construction cost. In the optimization of truss structures, some limitation such as strength for each member and displacement for each connection should be checked. As a result, the truss optimization procedure is as follow:

$$\begin{aligned} &\min imize \quad W = \sum_{e=1}^{N_m} \gamma_e . l_e . A_e \\ &subject\ to \quad \sigma^l < \sigma_e < \sigma^u \\ &\qquad\qquad A^l < A_e < A^U \\ &\qquad\qquad \delta^l < \delta_e < \delta^u \end{aligned} \tag{8}$$

Where W is the weight of truss; γ_e is the unit weight; l_e is the length of each member; A_e is the cross-sectional area of element e. This minimum design also has to satisfy the constraints on each member stress σ_e and deflection δ_c at each connection c. To control these constraints, a penalty method can be used as:

$$if \quad \sigma^{l} < \sigma_{e} < \sigma^{u} \quad then \quad \varphi_{\sigma}^{e} = 0 \tag{9}$$

$$if \quad \sigma_{e} < \sigma^{l} \quad or \quad \sigma_{e} > \sigma^{u} \quad then \quad \varphi_{\sigma}^{e} = \left| \frac{\sigma_{e} - \sigma^{l,u}}{\sigma^{l,u}} \right| \tag{10}$$

$$\varphi_{\sigma}^{k} = \sum_{e=1}^{N_m} \varphi_{\sigma}^{e} \tag{11}$$

$$if \quad \delta^{l} \le \delta_{c(x,y.z)} \le \delta^{u} \quad then \quad \varphi_{\delta(x,y,z)}^{c} = 0 \tag{12}$$

$$if \quad \delta_{c(x,y.z)} < \delta^{l} \quad or \quad \delta_{c(x,y.z)} > \delta^{u} \quad \varphi_{\delta(x,y,z)}^{c} = \left| \frac{\delta_{c(x,y.z)} - \delta^{l,u}}{\delta^{l,u}} \right| \tag{13}$$

$$\varphi_{\delta}^{k} = \sum_{c=1}^{N_m} [\varphi_{\delta x}^{c} + \varphi_{\delta y}^{c} + \varphi_{\delta z}^{c}] \tag{14}$$

The final penalty function ψ^{k} for a truss structure is as:

$$\psi^{k} = (1 + \varphi_{\sigma}^{k} + \varphi_{\delta}^{k})^{\varepsilon} \tag{15}$$

where ε is a positive penalty coefficient. The value of penalized weight can be defined as:

$$F^{k} = \psi^{k} . w^{k} \tag{16}$$

5. Numerical Examples

The truss structures with continuous variables are optimized using the presented method; containing: 25-bar spatial truss with 8 design variables; 72-bar spatial truss with 16 design variables; 200-bar planar truss with 29 design variables; 26-Story, 942-bar spatial truss with 59 design variables. In all examples, the results of hybrid TLBO and HS are compared to other heuristic-based methods.

5.1. Twenty-five bar spatial truss

The geometry of a 25-bar spatial truss is shown in Figure 4. the structure is subjected to load cases as presented in Table 1. The limits of stress for each member of the structure is listed in Table 2. All notes in X, Y, and Z directions are subjected to the allowable displacements ±0.35in. There are 8 groups of design variables with the minimum cross-section area of 0.01 in^2 and a maximum one of 3.4 in^2. Elements information is presented in Table 3. The unit weight is 0.1lb/in^3 and the modulus of elasticity is 10^7 psi.

Table 1
Multiple loading for the 25-bar truss

case	node	P_x (kips)	P_y (kips)	P_z (kips)
1	1	1.0	10.0	-5.0
	2	0.0	10.0	-5.0
	3	0.5	0.0	0.0
	6	0.5	0.0	0.0
2	1	0.0	20.0	-5.0
	2	0.0	-20.0	-5.0

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

Table 2
The 25-bar truss allowable stress

Elements group	members	Compression(ksi)	Tension(ksi)
1	1	35.092	35
2	2,3,4,5	11.59	35
3	6,7,8,9	17.305	35
4	10,11	35.092	35
5	12,13	35.092	35
6	14,15,16,17	6.759	35
7	18,19,20,21	6.959	35
8	22,23,24,25	11.082	35

Table 3
Elements information

Group of elements							
1	2	3	4	5	6	7	8
1(1,2)	2:(1,4)	6:(2,4)	10(6.3)	12:(3,4)	14:(3,10)	18:(4,7)	22:(6,10)
	3:(2.3)	7:(2,5)	11:(5,4)	13:(6,5)	15:(6,7)	19:(3,8)	23:(3,7)
	4:(1,5)	8:(1,3)			16:(4,9)	20:(5,10)	24:(4,8)
	5:(2,6)	9:(1,6)			17:(5,8)	21:(6,9)	25:(5,9)

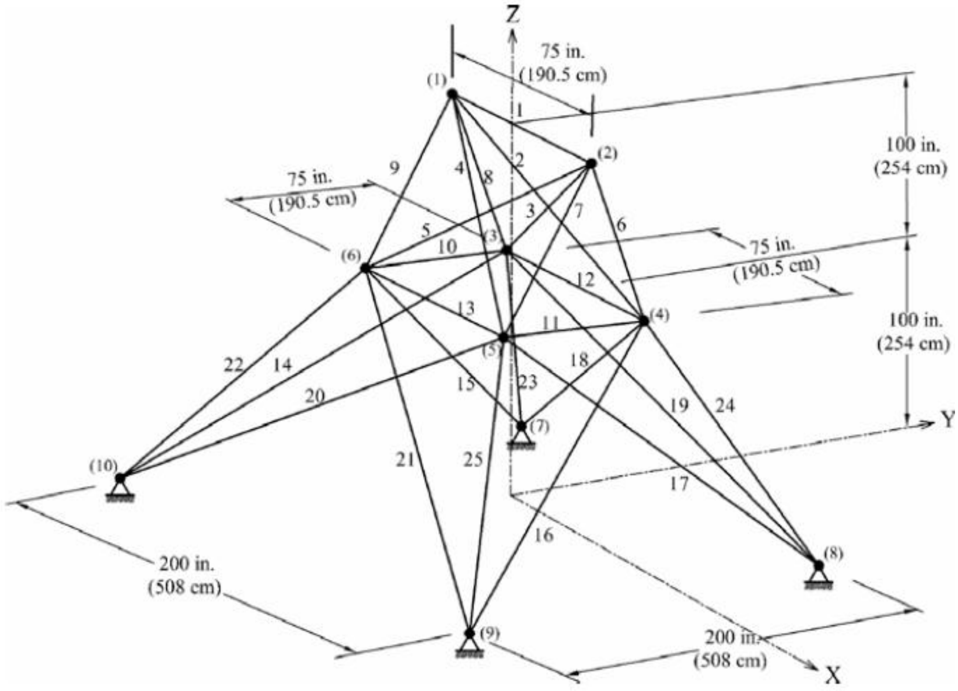

Fig. 4. Topology of the 25-bar spatial truss

The results of the new algorithm and some other selected meta-heuristic ones are listed in Table 4. The best weight of 25-bar truss is obtained by the hybrid algorithm that is equal to 545.15 lb. The new method needs 11,750 searches to find the best result. Although, the design obtained by HS (Lee and Geem, 2004) algorithm is 544.38 lb. and is a little lighter than the new hybrid performance and accuracy than other meta-heuristic algorithms when the standard division and mean results of method but it achieved the best weight after 15000 analyses which 21% more than the present method. The HBB-BC (Kaveh and Talatahari, 2009) algorithm achieved the best solution of 545.16 lb. after 12500 search which is more than the result of the new algorithm. Not only the hybrid method shows an improvement in the best solution, but it has the best different runs are compared. The best weight of the standard TLBO (Camp and Farshchin, 2014) is 545.175

lb. which obtained after 12,199 analyses, the best weight of the ABC algorithm (Sonmez, 2011) is 545.19 lb. which requires 500,000 searches. the best weight of the HPSO (Li et al., 2007) is 545.19 lb. requires 125,000 searches, the best weight of the GAs (Cao, 1996) is 545.80 lb., the best weight of the CMLPSA (Lamberti, 2008) is 545.86 lb., and the best weight of the ACO (Camp and Bichon, 2004) is 545.530 lb. which obtained after 16,500 analyses. Figure 5 shown the convergence history of the Hybrid TLBO and HS algorithm for the 25-bar spatial truss.

Table 4
Performance comparison for 25-bar spatial truss with continuous variables

variables		Cross-sectional area(in^2)							
Element group	members	Cao 1996	Camp and Bichon 2004	Li et al. 2007	Lamberti 2008	Kaveh Talatahari 2009	Sonmez 2011	Camp and Farshchin 2014	This work
1	1	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
2	2,3,4,5	2.0119	2.0000	1.9700	1.9870	1.9930	1.9790	1.9878	1.9826
3	6,7,8,9	2.9493	2.9660	3.0160	2.9935	3.056	3.0030	2.9914	3.0005
4	10,11	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0102	0.0100
5	12,13	0.0295	0.0120	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
6	14,15,16, 17	0.6838	0.6890	0.6940	0.6840	0.6650	0.6900	0.6828	0.6520
7	18,19,20, 21	1.6798	1.6790	1.6810	1.6769	1.6420	1.6790	1.6775	1.6730
8	22,23,24, 25	2.6798	2.6680	2.6430	2.6621	2.6790	2.6520	2.6640	2.6614
Weight(lb)		545.80	545.530	545.19	545.86	545.16	545.19	545.175	545.15
W_{avg}(lb)		-	546.34	-	-	545.66	-	545.483	545.48
W_{stdv}(lb)		-	0.94	-	-	0.367	-	0.306	0.314
$N_{analysis}$		-	16500	125,000	400	12,500	5e5	12,199	11750

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

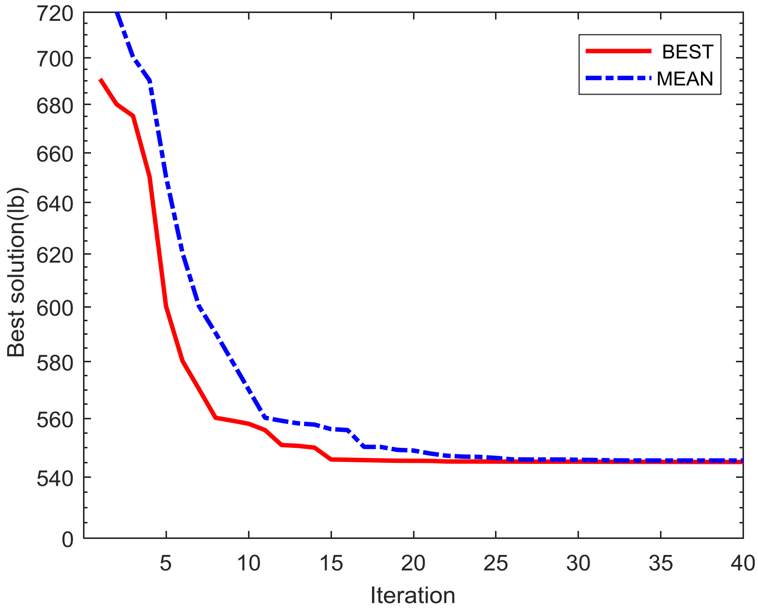

Fig. 5. The convergence history of 25-bar spatial truss

5.2. Seventy-two bar spatial truss

In the second example, the 72-bar spatial truss is considered as shown in Figure 6. The structure is subjected to multiple loading conditions listed in Table 5. The modulus of elasticity is 1e7 psi. the unit weight of the material is 0.1 lb/in^3. The members are subjected to the allowable stress limits of ±25ksi, and the maximum displacement of each node is ±0.25 in through X, Y, and Z direction. There are 16 group of design variables with a minimum 0.1in^2 and maximum 3.0 in^2 : (A1) 1-4, (A2) 5-12, (A3) 13-16, (A4) 17-18, (A5) 19-22, (A6) 23-30, (A7) 31-34, (A8) 35-36, (A9) 37-40, (A10) 41-48, (A11) 49-52, (A12) 53-54, (A13) 55-58, (A14) 59-66, (A15) 67-70, (A16) 71-72.

Table 5
Multiple loading for the 72-bar truss

case	node	P_x (kips)	P_y (kips)	P_z (kips)
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

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

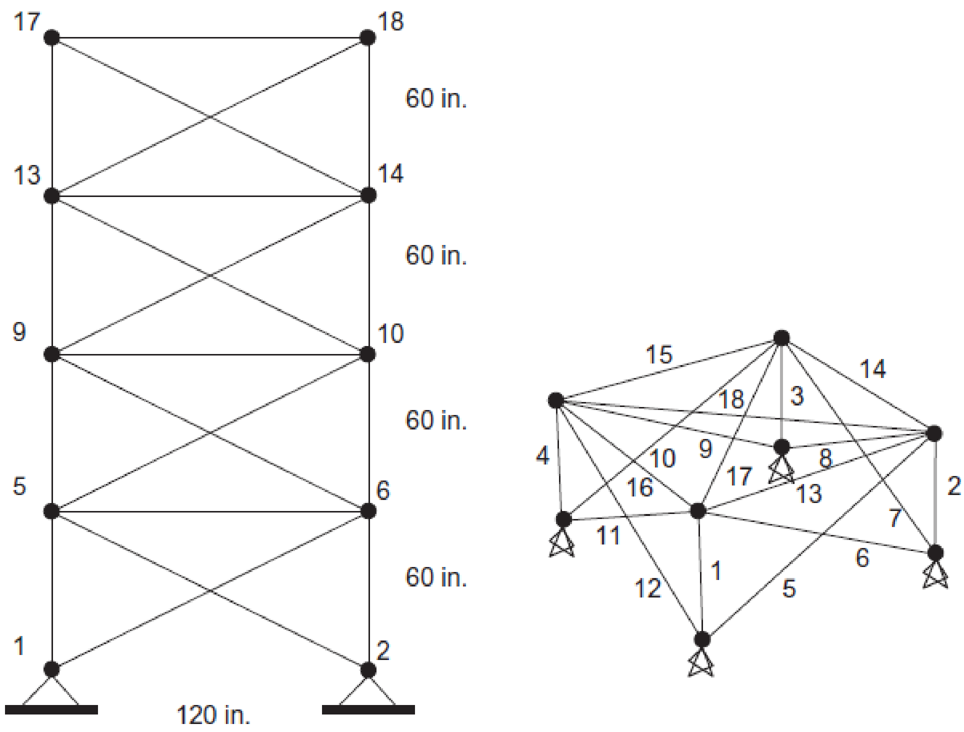

Fig. 6 Topology and elements definition of 72-bar truss;
(a) dimension and node numbering; (b) the pattern of element numbering.

Th

e results of the meta-heuristic algorithms are listed in Table 6. The best weight of the hybrid method is 379.6158 lb. Although the result of HS (Lee and Geem, 2004) algorithm is slightly small than that obtained by the new algorithm, it needs 20,000 analyses (37% more than it for the new method). The best weight for other meta-heuristic algorithms are 379.632 lb., 379.66 lb., 380.32 lb., 380.24 lb., 379.85 lb. and 381.91 lb. for TLBO (Camp and Farshchin, 2014), HBB-BC (Kaveh and Talatahari, 2009), GA (Cao, 1996), ACO (Camp and Bichon, 2004), BB-BC (Camp, 2007) and PSO (Perez and Behdinan, 2007), respectively. The standard deviation for the hybrid method is 0.033 lb. while it is 0.149 lb. for the standard TLBO (Camp and Farshchin, 2014). Also, the required search for reaching the best weight is 12,600 for the new method which lower than the required ones for the standard TLBO and HBB-BC algorithms. The average weight of the hybrid method is 379.6187 lb. which is the lightest one. Figure 7 shows the convergence history of the new algorithm for the 72-bar spatial truss.

Table 6
Performance comparison for 72-bar spatial truss with continuous variables

Variables		Cross-sectional area(in^2)							
Element group	members	Cao 1996	Camp and Bichon 2004	Lee and Geem 2004	Camp 2007	Perez and Behdinan 2007	Kaveh Talatahari 2009	Camp and Farshchin 2014	This work
1	1-4	1.8562	1.9480	1.7900	1.8577	1.7427	1.9042	1.8807	1.8904
2	5-12	0.4933	1.5080	0.5210	0.5059	0.5185	0.5162	0.5142	0.5115
3	13-16	0.1000	0.1010	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	17-18	0.1000	0.1020	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
5	19-22	1.2830	1.3030	1.2290	1.2476	1.3079	1.2582	1.2711	1.2631
6	23-30	0.5028	0.5110	0.5220	0.5269	0.5193	0.5035	0.5151	0.5134
7	31-34	0.1000	0.1010	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
8	35-36	0.1000	0.1000	0.1000	0.1012	0.1000	0.1000	0.1000	0.1000
9	37-40	0.5177	0.5610	0.5170	0.5209	0.5142	0.5178	0.5317	0.5215
10	41-48	0.5227	0.4920	0.5040	0.5172	0.5464	0.5214	0.5134	0.5154
11	49-52	0.1000	0.1000	0.1000	0.1004	0.1000	0.1000	0.1000	0.1000
12	53-54	0.1049	0.1070	0.1010	0.1005	0.1095	0.1007	0.1000	0.1000
13	55-58	0.1557	0.1560	0.1560	0.1565	0.1615	0.1566	0.1565	0.1564
14	59-66	0.5501	0.5500	0.5470	0.5507	0.5092	0.5421	0.5429	0.5466
15	67-70	0.3981	0.3900	0.4420	0.3922	0.4967	0.4132	0.4081	0.4105
16	71-72	0.6749	0.5920	0.5900	0.5922	0.5619	0.5756	0.5733	0.5703
Weight(lb)		380.32	380.24	379.27	379.85	381.91	379.66	379.632	379.6158
W_{avg}(lb)		-	383.16	-	382.08	-	381.82	379.7596	379.6187
W_{stdv}(lb)		-	3.66	-	1.912	-	1.202	.149	0.033
$N_{analysis}$		15,000	18,500	20,000	19,679	8,000	13,200	15,000	12,600

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

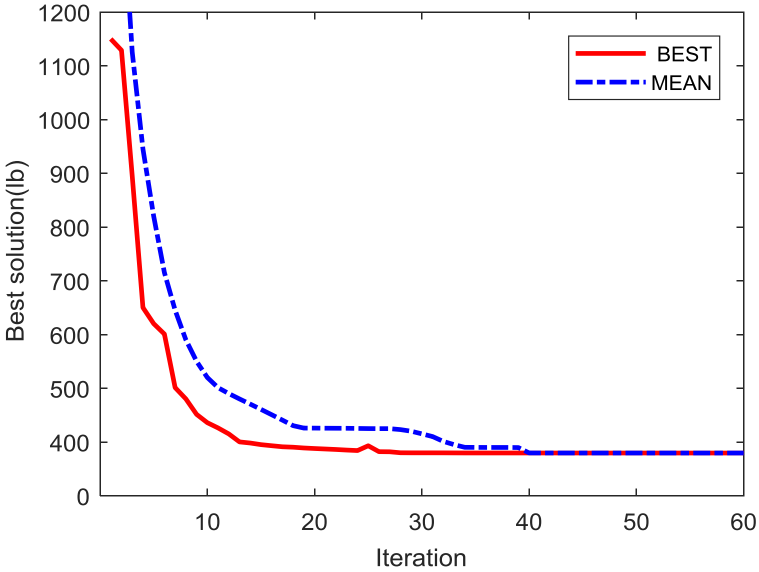

Fig. 7. The convergence history of 72-bar spatial truss

5.3. Two-hundred bar planar truss

In this example, size optimizing of the 200-bar planar truss is considered. Figure 8 shows the topology of the 200-bar truss. This truss structure has been optimized using other methods by Lamberti and Pappalettere (2003), Lamberti, (2008), Lee and Geem, (2004), and Farshi and Alinia, (2010). The modulus of elasticity is 3e7 psi. the material unit weight is 0.283 lb/in^3. The structure is subjected to only stress limitation of ±10 ksi. Because of the symmetry of the truss, variables are linked into 29 groups as shown in Table 7. The structure is subjected to three load cases as shown in Table 8.

Table 7
Element information

group	elements	group	elements
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, 197, 198, 200
14	95, 96, 97, 98, 99, 100	29	196, 199
15	102, 105, 108, 111, 114		

Table 8
Multiple loading for the 200-bar truss

case	Node	P_x (kips)	P_y (kips)	P_z (kips)
1	1, 6, 15, 20, 29, 34, 43, 48, 57, 62, 71	1.0	0.0	0.0
2	1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74, 75	0.0	10.0	0.0
3	Combination of case 1 and case 2			

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

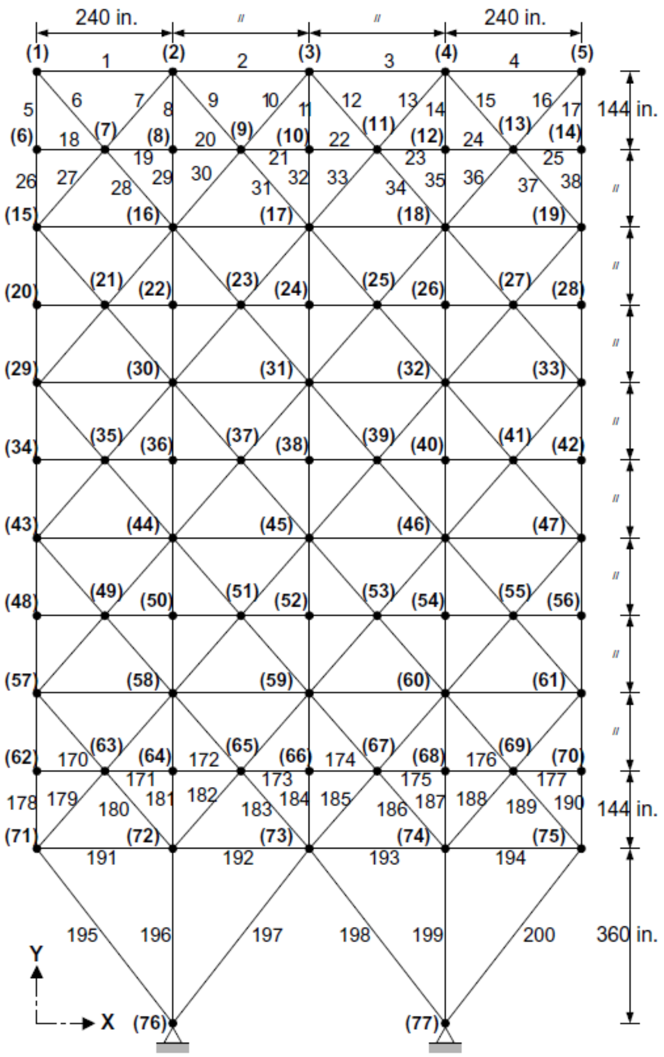

Fig. 8. Topology and element condition of 200-bar planar truss

Table 9 shows the comparison of results developed by the hybrid algorithm with other meta-heuristic methods. The best weight of 200-bar planar truss designed by the new algorithm is 25,447.81 lb. which is less than the designs developed by Lamberti (2008) and Lee and Geem (2004). Also, Lee and Geem (2004) achieved the best weight after 48,000 analyses while the hybrid TLBO finds it after 30,000 analyses. The design developed by Farshi and Alinia (2010) is 25,456.57 lb. which is more than that obtained by the hybrid method. Figure 9 shows a typical convergence history for 200-bar planar truss design using the new algorithm.

Table 9
Performance comparison for 200-bar planar truss with continuous variables

variables		Cross-sectional area(in^2)			
Element group	members	Lee and Geem 2004	Lamberti 2008	Farshi and Alinia 2010	This work
1	1,2,3,4	0.1253	0.1467	0.1470	0.1456
2	5,8,11, 14,17	1.0157	0.9400	0.9450	0.9378
3	19,20, 21,22, 23,24	0.1069	0.1000	0.1000	0.1000
4	18,25, 56, 63,94,101,132,139,170,177	0.1096	0.1000	0.1000	0.1000
5	26,29, 32, 35,38	1.9369	1.9400	1.9451	1.9425
6	6,7,9, 10,12, 13,15, 16, 27,28, 30,31, 33,34, 36,37	0.2686	0.2962	0.2969	0.2967
7	39,40, 41,42	0.1042	0.1000	0.1000	0.1000
8	43,46, 49, 52,55	2.9731	3.1040	3.1062	3.1060
9	57,58, 59, 60,61,62	0.1309	0.1000	0.1000	0.1002
10	64,67, 70,73,76	4.1831	4.1040	4.1052	4.1625
11	44,45, 47, 48,50, 51,53, 54, 65,66, 68,69, 71, 72,74,75	0.3967	0.4034	0.4039	0.4036
12	77,78, 79,80	0.4416	0.1922	0.1934	0.1924
13	81,84, 87, 90,93	5.1873	5.4282	5.4289	5.0918
14	95,96, 97, 98,99,100	0.1912	0.1000	0.1000	0.1196
15	102,105,108,111,114	6.241	6.4282	6.4289	6.4131
16	82,83, 85, 86,88, 89,91, 92,103,104, 106,107,109,110,112,113	0.6994	0.5738	0.5745	0.5639
17	115,116,117,118	0.1158	0.1325	0.1339	0.1337
18	119,122,125,128,131	7.7643	7.9726	7.9737	7.8233
19	1, 2, 3,4	0.1000	0.1000	0.1000	0.1000
20	5, 8,11, 14,17	8.8279	8.9726	8.9737	8.8269
21	19,20, 21,22, 23,24	0.6986	0.7048	0.7053	0.7055
22	18,25, 56, 63,94,101,132,139,170,177	1.5563	0.4202	0.4215	0.4213
23	26,29, 32, 35,38	10.9806	10.8666	10.8675	10.9513
24	6, 7, 9, 10,12, 13,15, 16, 27,28, 30,31, 33,34, 36,37	0.1317	0.1000	0.1000	0.1000
25	39,40, 41,42	12.1492	11.8666	11.8674	11.9560
26	43,46, 49, 52,55	1.6373	1.0344	1.0349	1.0346
27	57,58, 59, 60,61,62	5.0032	6.6838	6.6849	6.5043
28	64,67, 70,73,76	9.3545	10.8083	10.8101	10.8092
29	44,45, 47, 48,50, 51,53, 54, 65,66, 68,69, 71, 72,74,75	15.091	13.8339	13.8379	13.8344
	Weight(lb)	25,447.1	25,446.76	25,456.57	25,447.81
	W_{avg}(lb)	-	-	-	25,449.27

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

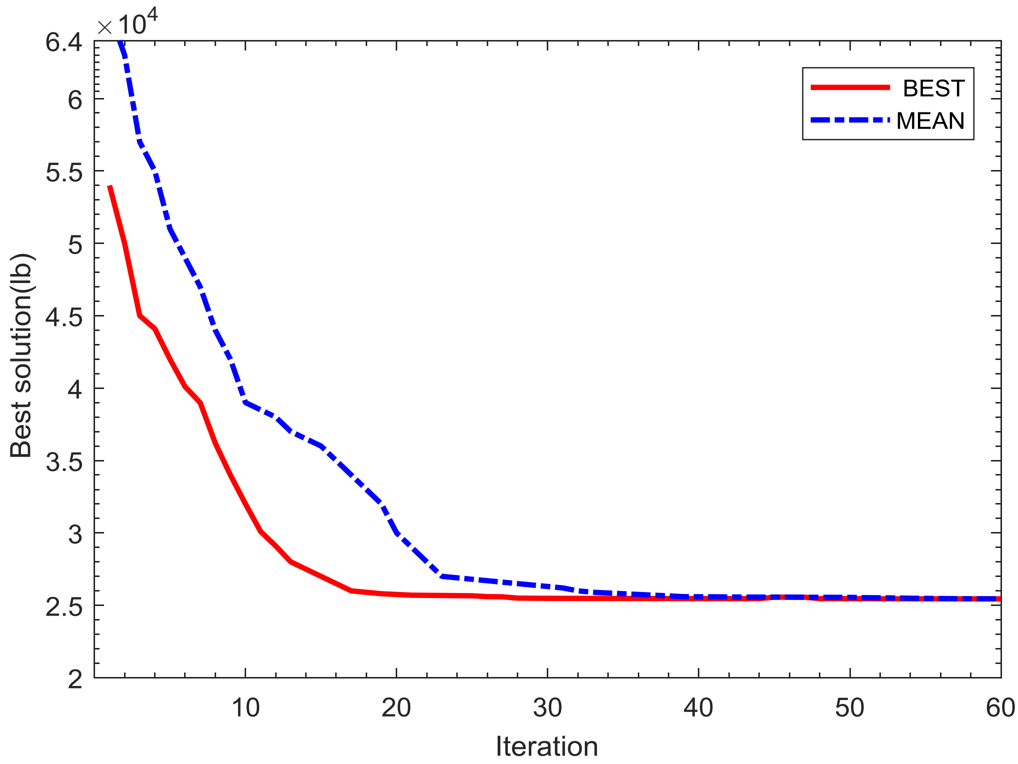

Fig. 9. The convergence history of 200-bar planar truss

5.4. Nine-hundred and Forty-two bar spatial truss

In the last example, the 26-story space truss containing 942 elements is considered in this section. Figure 10 shows the topology of the 942-bar spatial truss. Because of the symmetry of the spatial truss is divided into 59 groups. The material density is 0.1lb/in^3 and the modulus of elasticity is 1e7 Psi. the allowable cross-sectional areas in this example are selected from 1.0 in^2 to 200in^2. The members are subjected to the stress limitation of ±25ksi and the displacement limitation through X, Y, and Z direction is ±15 in. The structure is subjected to several load cases as: Case (1) The vertical load at each node in the first section is equal to 3kips, Case (2) The vertical load at each node in the second section is equal to 6kips, Case (3) The vertical load at each node in the third section is equal to 9kips, Case (4) The horizontal load at each node on the right side in the x-direction is equal to 1kips, Case (5) The horizontal load at each node on the left side in the x-direction is equal to 1.5kips, Case (6) The horizontal load at each node on the front side in the y-direction is equal to 1kips, Case (7)The horizontal load at each node on the backside in the y-direction is equal to 1kips.

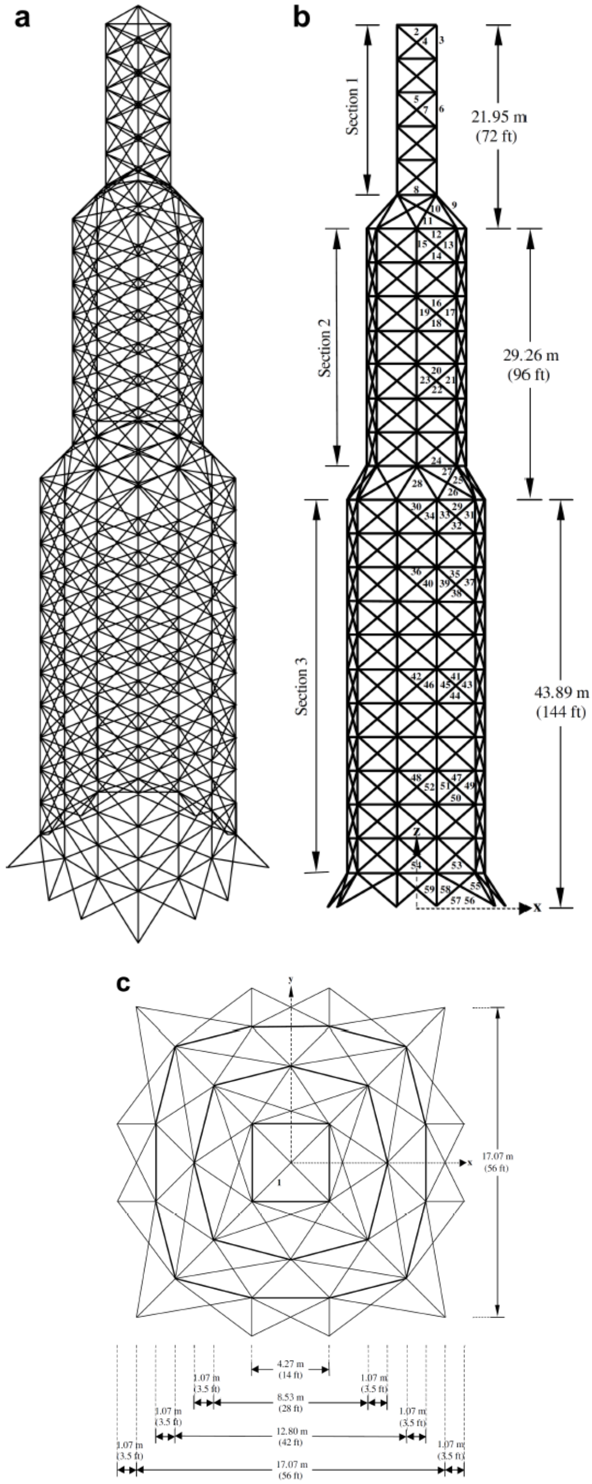

Fig. 10. Topology and elements definition of 942-bar truss;
(a) 3D dimension; (b) the pattern of element numbering; (c) top view.

The comparison results of different algorithms are listed in Table 10. As shown in the table, the lightest weight gained by the new algorithm is 140,871 lb. and it is the best solution between the existing results. Hasancebi (2008) achieved the best weight of 141,241 lb. using adaptive evolution strategies (ESs). The best weight of Adeli and Cheng (1994) is 170,000 lb. The best weight of Erbatur and Hasancebi (2000) is 143,436 lb. They used a simulated annealing (SA) method. As it is clear the best weight of the hybrid method is better than the other ones. Figure 11 shows the convergence history for 26-story, 942-bar spatial truss.

Table 10
Performance comparison for 26-story, 942-bar truss with continuous variables

variables	Cross-sectional area(in^2)			
Element group	Adeli and Cheng (1994)	Erbatur and Hasancebi (2000)	Hasancebi (2008)	This work
1	N/A	1.000	1.020	1.000
2	N/A	1.000	1.037	1.000
3	N/A	3.000	2.943	2.754
4	N/A	1.000	1.920	1.000
5	N/A	1.000	1.025	1.000
6	N/A	17.000	14.961	14.254
7	N/A	3.000	3.074	3.120
8	N/A	7.000	6.780	6.452
9	N/A	20.000	18.580	18.021
10	N/A	1.000	2.415	2.780
11	N/A	8.000	6.584	6.007
12	N/A	7.000	6.291	6.457
13	N/A	19.000	15.383	14.985
14	N/A	2.000	2.100	2.210
15	N/A	5.000	6.021	5.652
16	N/A	1.000	1.022	1.000
17	N/A	22.000	23.099	22.100
18	N/A	3.000	2.889	2.254
19	N/A	9.000	7.960	7.560
20	N/A	1.000	1.008	1.000
21	N/A	34.000	28.548	27.870
22	N/A	3.000	3.349	3.000
23	N/A	19.000	16.144	15.289
24	N/A	27.000	24.822	24.012
25	N/A	42.000	38.401	37.650
26	N/A	1.000	3.787	3.547
27	N/A	12.000	12.320	12.000
28	N/A	16.000	17.036	16.320
29	N/A	19.000	14.733	14.315
30	N/A	14.000	15.031	14.001
31	N/A	42.000	38.597	37.750
32	N/A	4.000	3.511	3.892
33	N/A	4.000	2.997	2.577
34	N/A	4.000	3.060	2.650
35	N/A	1.000	1.086	1.000
36	N/A	1.000	1.462	1.042
37	N/A	62.000	59.433	59.001
38	N/A	3.000	3.632	3.000
39	N/A	2.000	1.887	2.000
40	N/A	4.000	4.072	3.870
41	N/A	1.000	1.595	1.000
42	N/A	2.000	3.671	3.258
43	N/A	77.000	79.511	78.840
44	N/A	3.000	3.394	3.120
45	N/A	2.000	1.581	2.000
46	N/A	3.000	4.204	3.000
47	N/A	2.000	1.329	2.000
48	N/A	100.000	96.886	96.025
49	N/A	4.000	2.242	2.610
50	N/A	4.000	3.710	4.000
51	N/A	1.000	1.055	1.000
52	N/A	4.000	4.566	4.000
53	N/A	6.000	9.606	7.354
54	N/A	3.000	2.984	3.210
55	N/A	49.000	45.917	46.323
56	N/A	1.000	1.000	1.000
57	N/A	62.000	62.426	62.001
58	N/A	1.000	2.977	1.000
59	N/A	3.000	1.000	1.601
Weight(lb)	≅ 170,000	143,436	141,241	140,871
W_{avg}(lb)	-	-	-	141,265

Note: 1 in^2 = 6.452 cm^2 ; 1lb = 4.45 N

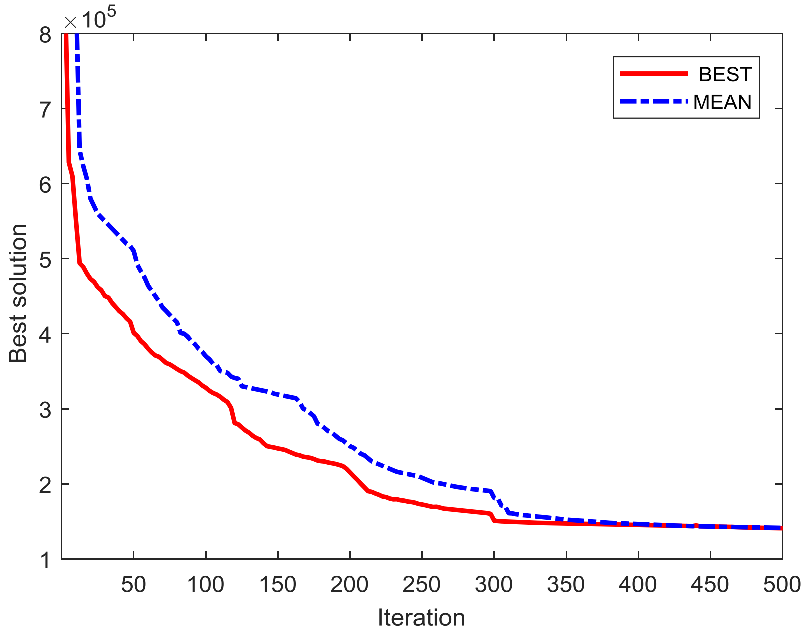

Fig. 11. The convergence history of 26- story, 942-bar spatial truss

6. Conclusion

Due to the difficulty of optimization methods, most of the meta-heuristic algorithms are unable to easily solve these problems. Recently, to find a solution to this problem, many researchers decided to hybridize the evolutionary algorithms. Indeed, they try to take advantage of these algorithms and merge them to eliminates their weaknesses. Investigation shows that the new hybrid algorithms are more efficient than the standard algorithms. Another advantage of the hybrid algorithms is that it could solve more difficult problems. In this paper, the hybrid TLBO and HS algorithm is developed based on the standard TLBO and HS algorithms to improve the performance of the new algorithm by identifying the merits and demerits of the standard TLBO and HS. The TLBO algorithm consists of two phases, the teacher phase, and the student phase and the main disadvantage of TLBO algorithm is in the teacher phase when the best student selected as a teacher and then it tries to improve the average of the class-level. There is no control parameter for measuring the distance between the teacher and the average of the class. This problem makes the algorithm trapping at local points and then the optimization will be continued by the local point.

To improve the mentioned problem, we used the HS algorithm which can explore all unknown places in search space to find the global optimum and it has the most complementary to the TLBO algorithm. This algorithm has lower mathematical requirements than other meta-heuristic methods and by applying some changes in parameters and operators, it can be adapted to different engineering problems. Therefore, to demonstrate the efficiency in both performance and convergence rate several truss structures have been optimized and the result proves the ability of the new hybrid algorithm. According to the high potential of this new algorithm, it can be used to solving the difficult optimization problem in structural engineering. Also, by making a slight change, it can be used to optimize the steel frames and concrete structures.

References

Adeli, H., Cheng, NT. (1994). Concurrent genetic algorithms for optimization of large structures. *J Aerospace Engineering*, 7:276–296

Behnamian, J. (2019). Diversified Particle Swarm Optimization for Hybrid Flowshop Scheduling. *Journal of Optimization in Industrial Engineering*, 12(2), 107-119. doi: 10.22094/joie.2018.671.1433

Camp, CV. (2007). Design of space trusses using big bang–big crunch optimization. *J Structural Engineering*, 133(7):999–1008

Camp, CV., Bichon, BJ. (2004). Design of space trusses using ant colony optimization. *J Structural Engineering*, 130(5):741–751

Camp, CV., Farshchin, M. (2014). Design of space trusses using modified teaching-learning based optimization. Computers & Structures. 62(63):87–97

Cao, G. (1996). Optimized design of framed structures using a genetic algorithm. PhD thesis. The University of Memphis TN.

Chen, D., Zou, F., Li, Z., Wang, J., Li, S. (2015). An improved teaching–learning-based optimization algorithm for solving global optimization problem. *Information Sciences*, 297(C):171–190

Dorigo, M. (1992). Optimization, learning and natural algorithms. PhD thesis, Dip. Elettronicae Informazione, Politecnico di Milano, Italy

Erbatur, F., Hasancebi, O. (2000). Optimal design of planar and space structures with genetic algorithms. *Computer and Structures*, 75:209-224

Faezy razi, F., Shadloo, N. (2017). A Hybrid Grey based Two Steps Clustering and Firefly Algorithm for Portfolio Selection. *Journal of Optimization in Industrial Engineering,* 10(22), 49-59. doi: 10.22094/joie.2017.276

Fattahi, P., Samouei, P. (2016). A Multi-Objective Particle Swarm Optimization for Mixed-Model Assembly Line Balancing with Different Skilled Workers. *Journal of Optimization in Industrial Engineering,* 9(20), 9-18. doi: 10.22094/joie.2016.246

Fallahian, S., Joghataie, A., Kazemi, M.T. (2010). Structural damage detection using time domain responses and Teaching-Learning-Based Optimization algorithm. *Scientia Iranica*, 25(6), 3088-3100

Farshi, B., Alinia, A. (2010). Sizing optimization of truss structures by method of centers and force formulation. *International Journal of Solids and Structures*, 47: 2508–2524

Hasancebi, O. (2008). Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures. Computers and Structures, 86:119-132

Kaveh, A. and Bijari, Sh. (2018). Simultaneous analysis, design and optimization of trusses via force method. Struct Eng Mech, Int J, 65(3), pp. 233-241

Kaveh, A., Ilchi, Ghazaan. M. (2017). A new meta-heuristic algorithm: Vibrating particles system. Scientia Iranica. *International Journal of Science and Technology*, 24: 551-566

Kaveh, A., Rahami, S. (2016). Analysis, design and optimization of structures using force method and genetic algorithm. Int. *J. Numer. Methods Eng*, 65, pp. 1570-1584

Kaveh, A., Talatahari, S. (2009). Size optimization of space trusses using big bang–big crunch algorithm. *Computers & Structures*, 87(17–18):1129–1140

Kaveh, A., Talatahari, S. (2009). Hybrid Algorithm of Harmony Search, Particle Swarm and Ant Colony for Structural Design Optimization. *Computers & Structures,* 3:642-03450

Kaveh, A., Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213: 267–289

Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks p, 1942–1948

Lamberti, L. (2008). An efficient simulated annealing algorithm for design optimization of truss structures. *Computers & Structures*, 86(19–20):1936–1953

Lee, KS., Geem, ZW. (2004). A new structural optimization method based on the harmony search algorithm.*Computers & Structures*, 82(9–10):781–798

Li, LJ., Huang, ZB., Liu, F., Wu, QH. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, 85(7–8):340–9

Omidinasab, F., Goodarzimehr, V. (2019). A Hybrid Particle Swarm Optimization and Genetic Algorithm for truss structures with discrete variables. *Journal of Applied and Computational Mechanics*, (), -. doi: 10.22055/jacm.2019.28992.1531

Perez, Ruben. E, Behdinan, Kamran. (2007). Particle Swarm Optimization in Structural Design. Itech Education and Publishing, Vienna, Austria.

Rao, RV., Savsani, VJ., Vakharia, DP. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer Aided Design*, 43(3):303–315

Sonmez, M. (2011). Artificial bee colony algorithm for optimization of truss structures. *Applied Soft Computing*, 11(2):2406–2418

Talatahari, S., Goodarzimehr, V. (2019). A Discrete Hybrid Teaching-Learning-Based Optimization algorithm for optimization of space trusses. *Journal of Structural Engineering and Geo-Techniques*, 9(1), 0-0.

Tuo, S., Yong, L., Deng, F., Li, Y., Lin, Y., Lu, Q. (2017). A hybrid algorithm based on Harmony Search and Teaching-Learning-Based Optimization for complex high-dimensional optimization problems, Pone10-1371

This article can be cited: Talatahari, S., Goodarzimehr, V., Taghizadieh, N. (2020) Hybrid Teaching-Learning-Based Optimization and Harmony Search for Optimum Design of Space Trusses.
Journal of Optimization in Industrial Engineering. 13 (1), 177-194.

http://www.qjie.ir/article_667916.html
DOI: 10.22094/JOIE.2019.1866904.1649