Smart Structures and Systems, *Vol. 21, No. 3 (2018) 263-278*
DOI: https://doi.org/10.12989/sss.2018.21.3.263

Improved thermal exchange optimization algorithm for optimal design of skeletal structures

A. Kaveh*[1], A. Dadras[2] and T. Bakhshpoori[3]

[1]Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846-13114, Iran
[2]Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846-13114, Iran
[3]Faculty of Technology and Engineering, Department of Civil Engineering, East of Guilan, University of Guilan, Rudsar-Vajargah, Iran

(Received January 16, 2018, Revised February 20, 2018, Accepted February 23, 2018)

Abstract. Thermal Exchange Optimization (TEO) is a newly developed algorithm which mimics the thermal exchange between a solid object and its surrounding fluid. In this paper, an improved version of the TEO is developed to fix the shortcomings of the standard version. To demonstrate the viability of the new algorithm, the CEC 2016's single objective problems are considered along with the discrete size optimization of benchmark skeletal structures. Problem specific constraints are handled using a fly-back mechanism. The results show the validity of the improved TEO method compared to its standard version and a number of well-known algorithms.

Keywords: improved thermal exchange optimization; metaheuristic; discrete structural optimization; optimum design; steel structures

1. Introduction

Nowadays, resource depletion has urged human beings to achieve the maximum output from the limited amount of available resources. Therefore, in engineering design, using the lowest cost design variables which fulfill the requirements of the design codes, has attracted enormous interest among the engineers. The low-cost design as a mathematical minimization problem is usually a constrained, discrete, highly complex, and multi-modal problem. The dependency of the classical gradient based methods to the initial good starting point and the acquisition of gradient information make solving such problems either costly or even impossible. The mentioned weaknesses of the traditional methods and the growth of the computational resources have led to a new generation of methods called metaheuristics, which are not restricted in the aforementioned manner (Kaveh 2017, Talbi 2009).

In recent years, a significant number of metaheuristics are introduced and applied to engineering problems. Some of the most well-known metaheuristic algorithms are Genetic Algorithm (GA), inspired by biological evolution (Goldberg and Samtani 1986, Koumousis and Georgiou 1994, Yang and Soh 1997), Harmony Search (HS) that mimics the improvisation process of jazz musicians (Geem *et al*. 2001, Maheri and Narimani 2014, Saka *et al*. 2011), Pigeon Colony Algorithm (PCA) based on the features of a pigeon colony flying and homing process (Yi *et al*. 2016), Grasshopper optimization algorithm (GOA) according to the characteristics of the swarming of grasshoppers (Saremi *et al.* 2017) and Lion Pride Optimizer (LPO) which mimics the lion pride behavior (Kaveh and Mahjoubi 2017, Wang *et al*. 2012). In addition to these algorithms there are some other algorithms which are based on the physics laws, for instance the Ray Optimization (RO) is based on Snell's light refraction law (Kaveh and Khayatazad 2012) or the Colliding Bodies Optimization (CBO) upon the one-dimensional collisions between bodies (Kaveh and Mahdavi 2014).

Thermal exchange optimization (TEO) is one of the new physics based algorithms, based on the Newton's cooling law. This law states that the rate of heat loss of a body is proportional to the difference between the body and its surroundings temperatures. In TEO, each search agent is considered as a cooling object and by associating another agent as the surrounding fluid, thermal exchange happens between them. The new temperature of the object is considered as its next position in the search space. The results, achieved by solving various mathematical and engineering problems, have shown the good performance of the method in terms of global and local search, robustness and fast convergence (Kaveh and Dadras 2017b). Subsequently, an off-line tuned version of TEO algorithm was introduced by the authors and applied to optimization-based damage identification problem, where GWO algorithm (Mirjalili *et al*. 2014) was utilized to tune the TEO at the meta-level (Kaveh and Dadras 2017a). However, a good balance between global and local search abilities of metaheuristics has always been of great interest for obtaining accurate results using less computational cost (Gholizadeh 2013, Wang *et al*. 2014). The main purpose of

∗Corresponding author, Professor
E-mail: alikaveh@iust.ac.ir

http://www.techno-press.com/journals/sss&subpage=7 ISSN: 1738-1584 (Print), 1738-1991 (Online)

this study is to propose some modifications to the TEO mechanisms for improving the convergence of the algorithm and obtaining more accurate results. The improved version is named as ITEO. According to the thermal exchange between an object and its surrounding fluid, the temperature of the object is closer to that of the fluid. In search agents, monotonic convergence is not favorable, especially when the fluid is in a worse position.

Having many design variables, the large size of the search space and the presence of many constraints make the discrete optimization of skeletal steel structures a suitable means to investigate the efficiency of the newly developed algorithms (Hasançebi and Azad 2015). On the other hand, in practice the design variables are usually chosen from a predefined available list of sections, making the discrete size optimization more workable. In this regard, the proposed ITEO and its original version were applied to the solution of four benchmark test cases: a spatial 25-bar truss, a spatial 72-bar truss, a three-bay fifteen-story frame, and a three-bay twenty four-story frame with 8, 16, 11, and 20 design variables, respectively. Comparing the results of the ITEO with those of the TEO and the available results of some well-known metaheuristics show the high efficiency of the presented ITEO.

The rest of this paper is organized as follows. In Section 2, a brief overview of the TEO is presented and the new improved version of this algorithm (ITEO) is provided. Section 3 uses four benchmark skeletal structures with discrete variables to compare the performance of the ITEO to those of the standard TEO and some other popular optimization methods. Finally, conclusions are derived in Section 4.

2. Optimization algorithms

In this section the standard TEO is outlined, and the improvements made to TEO are presented.

2.1 Standard thermal exchange optimization algorithm

TEO is a physically-inspired algorithm which uses the Newton’s law of cooling. In this algorithm, the temperature solution for the body with lumped thermal capacity is employed to update the position of the agents in the search space.

2.1.1 Theory

Assume that the overall heat transfer coefficient of a body is equal to *h* and the object has high temperature T_0 at time $t = 0$ and is suddenly placed in a different environment where it is cooled by surrounding fluid at a constant temperature T_b. The volume of the solid is *V* (in m^3) and its surface area, which the heat flow (Q) takes place on it is *A* (in m^2). The rate of heat loss from the surface is

$$\frac{dQ}{dt} = h(T_0 - T_b)A \tag{1}$$

in which *T*, *h*, and *t* stand for temperature in $°K$, heat transfer coefficient in $Wm^{-2}k^{-1}$, and time in *Sec*, respectively.

The heat loss in time dt is $h(T_0 - T_b)A\,dt$, and this is equal the change in the stored heat as the temperature falls *dT*, i.e.

$$V\rho c dT = -hA(T - T_b)dt \tag{2}$$

where ρ and *c* are the density ($kg\ m^3$) and specific heat ($J\ kg^{-1}\ K^{-1}$), respectively.

Integration results in

$$\frac{T - T_b}{T_0 - T_b} = \exp(-\frac{hA}{V\rho c}\mathrm{t}) \tag{3}$$

Assuming $\frac{hA}{V\rho c}$ as β, the result of the integration can be rearranged as follows

$$T = T_b + (T_0 - T_b)\exp(-\beta t) \tag{4}$$

2.1.2 TEO Algorithm

The TEO is made up of ten steps as follows:

Step 1: Initialization

The initial temperature of all the objects is determined randomly in an *m*-dimensional search space by

$$T_i^0 = T_{min} + rand \times (T_{max} - T_{min}) \tag{5}$$

where T_i^0 is the initial solution vector of the *i*th object, T_{min} and T_{max} are the minimum and maximum allowable values of design variables, *rand* is a random vector with components in the interval [0, 1]. The length of the vectors is equal to the number of design variables.

Step 2: Evaluation

The cost function specifies the cost value of each object. The weight of the structure is considered as the cost function in this paper.

Step 3: Saving

A Thermal Memory (TM) is utilized to save a number of the best-so-far solutions. Therefore in this step, the saved solution vectors are added to the population, and the same numbers of current worst agents are removed. Finally, agents are sorted according to their related objective function values in an ascending order. Considering a memory which saves some historically best *T* vectors and their related objective function values, can improve the performance of the algorithm in terms of local search and convergence without increasing the computational cost.

Step 4: Creating groups

Since an object and its surrounding fluid are necessary to perform thermal exchanging, two equal groups containing object and environment are created. The pairs of objects are defined according to Fig. 1, where for instance T_1 is environment for $T_{\frac{n}{2}+1}$ and vice versa. This type of pairing is originally proposed in CBO algorithm (Kaveh and Mahdavi 2014).

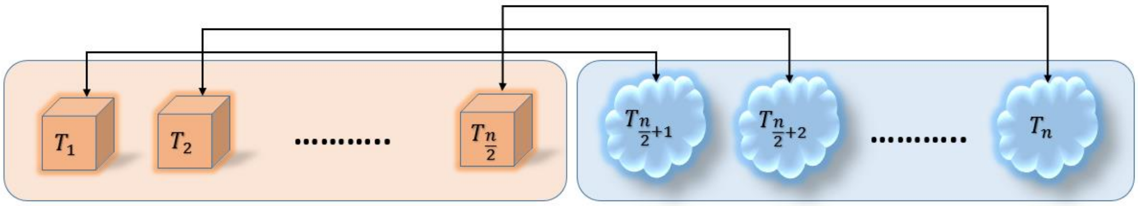

Fig. 1 Pairs of environment and cooling objects

Step 5: Definition of the β Parameter

The value of β for each agent is evaluated according to

$$\beta = \frac{Cost(object)}{Cost(worst\ object)} \tag{6}$$

where $Cost(object)$ is the weight of the current design and $Cost(worst\ object)$ is the highest weight among all of the updated design scenarios. It can be seen that the value of β and the cost of the object are directly proportional.

Step 6: Definition of t

The value of t for each agent is calculated from

$$t = \frac{iteration}{Max\ iteration} \tag{7}$$

Step 7: Escaping from local optima (i)

Non-convex search spaces are usually made of several local optimums. Meta-heuristic algorithms should have the ability to escape from the traps when agents get close to the local optimum. Thus steps 7 and 9 are devised to escape from these traps. In this step the environmental temperature is changed by Eq. (8), where c_1 and c_2 are controlling variables.

$$T_i^{env.} = \left(1 - \left(c_1 + c_2 \times (1-t)\right) \times rand\right) \times T_i'^{\,env.} \tag{8}$$

where $T_i'^{\,env.}$ is the non-changed environmental temperature. (1-t) is considered to decrease the randomness by nearing to the final iterations. This helps the convergence and exploitation. c_2 controls (1-t). For instance, this can be considered equal to zero, when the decreasing is not required. c_1 controls the size of the random steps. Furthermore when the decreasing is not employed (c_2=0), c_1 involves the randomness.

Step 8: Updating the search agents

The new temperature of each object is calculated by Eq. (9) which is inspired by Eq. (4).

$$T_i^{new} = T_i^{env.} + \left(T_i^{Old} - T_i^{env.}\right).\exp(-\beta t) \tag{9}$$

If the cooling object is in worse position compared to the environmental object, then the ratio β will have higher value, and $\exp(-\beta t)$ will tend to have lower value. Therefore, according to Eq. (9), the cooling object will have higher tendency to the environmental temperature and vice versa.

Step 9: Escaping from local optima (ii)

Pro is defined to specify whether a component of each cooling object must be changed or not. For each agent, *Pro* is compared with rand (i) (i=1, 2, ..., n) which is a random number uniformly distributed within (0, 1). If rand (i) < *Pro*, one dimension of the ith agent is selected randomly and its value is regenerated as follows

$$T_{i,j} = T_{j,min} + random\,.\left(T_{j,max} - T_{j,min}\right) \tag{10}$$

where $T_{i,j}$ is the jth variable of the ith agent. $T_{j,min}$ and $T_{j,max}$ respectively, are the lower and upper bounds of the jth variable. In order to protect the structures of the agents, only one dimension of design vector is changed. This mechanism provides the second global search opportunity for the agents to move all over the search space thus providing better diversity.

Step 10: Checking the termination conditions

The optimization process is terminated after a fixed number of iterations. If the criterion is not satisfied it goes to step 2 for a new round of iteration, otherwise, the process stops and the best-found solution is reported.

2.2 Improved thermal exchange optimization

In order to avoid the repeated descriptions, only the performed improvements are stated here. The improved version is named as Improved TEO and abbreviated as ITEO. At the end, the pseudocode of the ITEO and constraint handling approach are provided.

2.2.1 Improvement on t parameter

The value of t is changed to

$$t = \left(\frac{iteration}{Max\ iteration}\right)^Z, \qquad 0 < Z < 1 \tag{11}$$

where the parameter Z is considered as the power of the previous version of t. According to this definition and as it is shown in Fig. 2, t slows the speed of tending to 1 and the converging speed to search wider space. The numbers on the curves are the values of Z.

2.2.2 Improvement on β parameter

The value of β is changed to

$$\beta = \left(\frac{Rank(object)}{Rank(worst\ object)}\right)^Z, 0 < Z < 1 \tag{12}$$

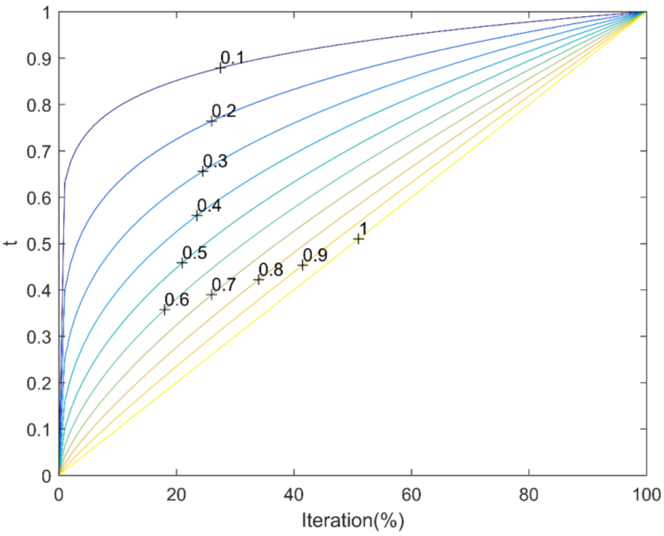

Fig. 2 The modified *t* by considering various values for *Z* parameter and passed iteration percent

The previous definition of β results in a division by zero, when $Cost(worst\ object) = 0$, (the worst object is not defined before) hence, the new definition is replaced. $Rank$ is the rank of the objects, when they are sorted in an ascending order of the objective function value. Z exponent has the same role as stated in the previous section.

2.2.3 Improvement on thermal updating equation

Taylor series expansion in the exponential function (e^x) is according to Eq. (13)

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (13)$$

By utilizing this expansion in Eq. (9) the following equation can be obtained

$$T_i^{new} = T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 - \beta.t + \frac{(-\beta.t)^2}{2} + \frac{(-\beta.t)^3}{6} + \cdots) \quad (14)$$

Since both β and t are small fractions and also divided by numbers greater than one, for reducing the computational burden, nonlinear small terms are simplified to $rand.\beta.t$

$$T_i^{new} = T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 - \beta.t + rand.\beta.t) \quad (15)$$

Herein $rand$ is a random number between zero and 1, so the this equation can be rearranged as Eq. (15)

$$T_i^{new} = T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 - (1 - rand).\beta.t) \quad (16)$$

and

$$T_i^{new} = T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 - rand.\beta.t) \quad (17)$$

In Teaching-Learning-Based Optimization (TLBO) algorithm the state of each search agent (student) within the search space, moves toward better agent or away from worse agent (Rao *et al.* 2011). By using this mechanism, the following additional modifications are imposed on the updating equation

$$T_i^{new} = \begin{cases} T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 - rand.\beta.t) & Cost(old) < Cost(env.) \quad (18a) \\ T_i^{env.} + (T_i^{Old} - T_i^{env.}).(1 + rand.\beta.t) & Cost(old) > Cost(env.) \quad (18b) \end{cases}$$

These equations can be simplified in the following form

$$T_i^{new} = T_i^{env.} + (T_i^{Old} - T_i^{env.}) (1 - \mathrm{sign}(Cost(env.) - Cost(old)).rand.\beta.t) \quad (19)$$

where *sign* finds the sign function of the value. Schematics of these equations are illustrated in Fig. 3. In the standard TEO, temperatures get closer together and the algorithm focuses on the space between the objects. As it can be seen from Fig. 3, in the new version new temperature is getting away from the high cost object, exploring the outer space.

2.2.4 Pseudo-code of the ITEO

The pseudo-code of the ITEO is now presented as follows:

Procedure of Thermal Exchange Optimization
begin
  Initialize all agents
  while *(termination condition not met) do*
      for *each search agent*
          Calculate the fitness
      end for
  Update thermal memory and sort population
  Create equal groups
      for *each agent*
      Calculate t and β by Eq. (11) and Eq. (12)
      Change environmental temperature by Eq. (8)
      Update temperature by Eqs. (18a, 18b)
      end for
  Regenerate violating variables
  end while
end

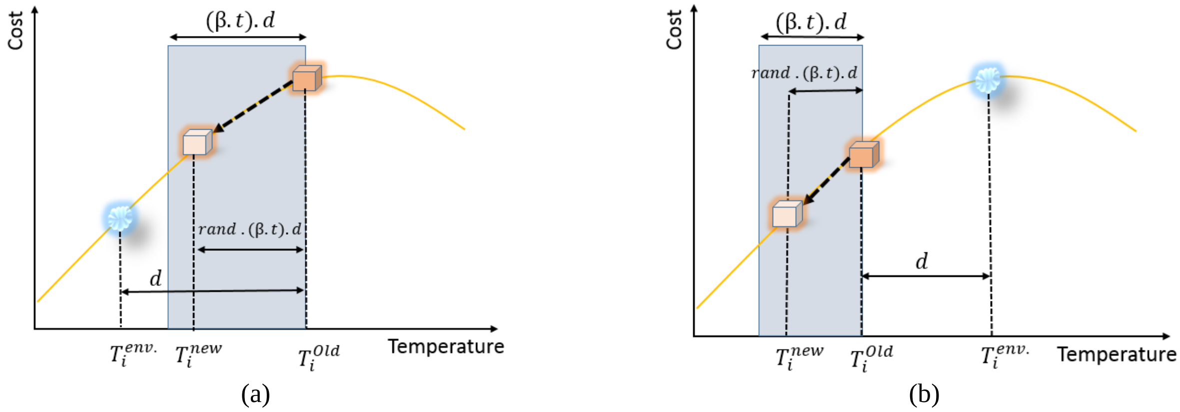

Fig. 3 Schematics of search agent updating, according to: (a) Eq. (18(a)) and (b) Eq. (18(b))

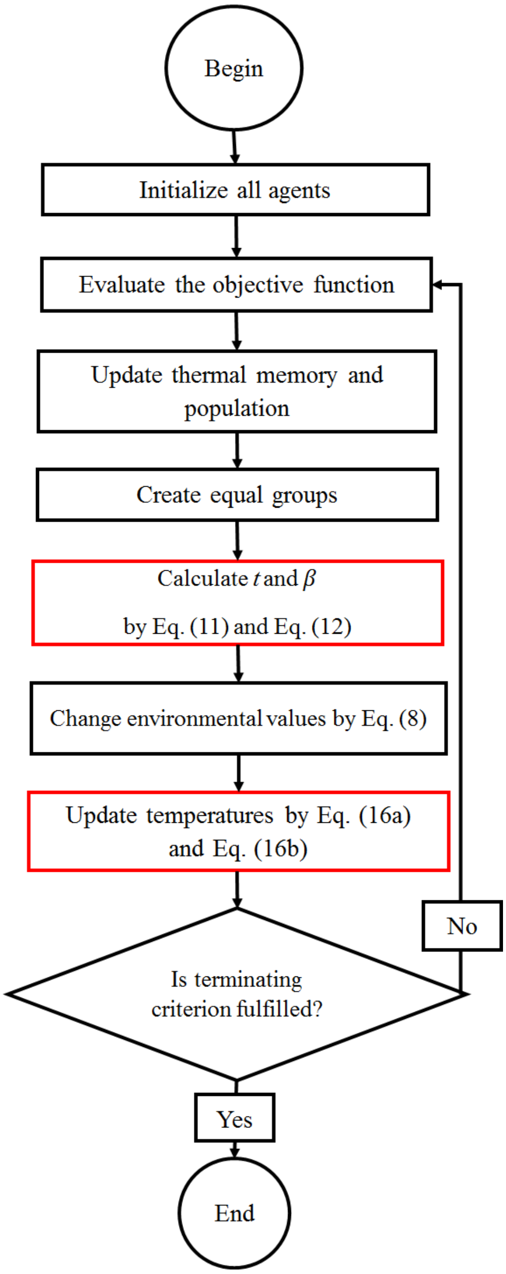

Fig. 4 Flowchart of the ITEO algorithm

In the above pseudocode, the second ***for*** loop is related to the temperature updating which can be considered as the most important part of the ITEO. Flowchart of the proposed ITEO is depicted in Fig. 4. It is clear that the structure of the algorithm is simple and the implementation is quite easy.

2.3 Constraint handling

One of the important issues in constrained optimization problems is the approach of constraint handling. Many techniques are proposed to handle the constraints. A prevalent and successful technique is penalty approach (Yeniay 2005). This approach is utilized by the following equation:

$$W_p = W \times (1 + \varepsilon_1 . \upsilon)^{\varepsilon_2} \tag{20}$$

where W is the weight of the structure without penalty, W_p is the penalized weight that is the cost function and υ is the total constraints violation. Constants ε_1 and ε_2 must be selected considering the exploration and the exploitation rate of the search space. In this study ε_2 has a linear increase from 1.5 to 3 and ε_1 is set to one.

Sometimes the search agents leave the bounds, in this paper flyback mechanism is employed to bring back them on the bounds which are left.

3. Numerical examples

For investigating the performance of the proposed algorithm two following subsection are provided. In the first subsection, TEO and its improved version (ITEO) are evaluated on the IEEE-CEC 2016 benchmark problems (Rueda Torres and Erlich 2016). In the second subsection, four well-known structural examples with discrete variables are utilized to test the new algorithm by optimization of the weight of the benchmark structures. The results are compared by those of some other well-established optimizers and some discussions are provided to show the efficiency of the improved version.

3.1 Computationally expensive problems

As mentioned above, the performance of the proposed algorithm is evaluated on 15 benchmark CEC 2016 single-objective bound constrained problems. The list of the test problems are provided in Table 1, the details about the functions can be found in reference (Chen *et al.* 2014).
In all problems, the following points are considered:

- Dimensions are $D = 10, 30$.
- Search range is $[-100,100]^D$.
- The number of function evaluations are limited to 50*D as the termination criteria.
- The optimization is independently repeated 20 times for each case.
- The error value $OF = TF_i\ (x) - F_i^*$ is defined as the objective function, where F_i^* is the theoretical global optimum of the *i*th benchmark function reported in Table 1.

Here, c_1, c_2, *Pro*, *TM* and *Z* are equal to 1, 1, 0.1, 2 and 0.5, respectively. According to the method of evaluation suggested in (Rueda Torres and Erlich 2016), the statistical results are given in Tables 2 and 3. The better results between TEO and ITEO are written in bold style. As seen in most cases the ITEO achieved better results in comparison with TEO, especially in 30 dimension problems.

The complexity of the algorithms is measured by $\overline{T_1}/T_0$, where $\overline{T_1}$ is the average optimization time for the above functions and T_0 is the computing time of a predefined process given in (Chen *et al.* 2014). The ratios are provided in Fig. 5. As seen in most cases, the complexity of ITEO is a little more than TEO, also, the complexity of functions with 30 dimensions are higher than 10 dimensions.

The total score of each algorithm with 10D and 30D problems are measured by the following equation

$$Total\ Score = \sum_{1}^{15} mean(f_{ai})\,|_D + \sum_{1}^{15} median(f_{ai})\,|_D \tag{21}$$

where f_a for each function is calculated by

$$f_a = 0.5(f_{MaxFEs} + f_{0.5MaxFEs}) \tag{22}$$

As provided in Table 4, the improved version obtained better scores in comparison with the basic version.

3.2 Discrete optimization of skeletal structures

The proposed algorithm is tested using various values of parameters, such as number of objects, c_1, and c_2 by varying number of objects = 5, 10, and 15; c_1 = - 0.5, -0.25, 0.25, and 0; and c_2= -0.5,0, 0.5, 1. Based on our simulations, the proposed ITEO needs at least 10 number of objects and results in the most efficient performance of the algorithm. Considering c_1, c_2, *Pro*, *TM* and *Z* equal to -0.25, 1, 0.3, 4 and 0.5, respectively, results in a good performance for almost all the discrete test problems. The maximum number of algorithm iterations is considered as 2000 for the stopping criteria in all the structural test cases. The consistency of the algorithm is verified by running all the problems for 30 independent runs with different random initial solutions, and the results for the representative sample run and statistical results of all independent runs are reported. The results of the proposed algorithm are compared with those of some other published results.

3.2.1 Spatial 25-bar truss

The first example is a 25-bar transmission tower as illustrated in Fig. 6. This structure is studied widely in structural optimization to examine numerous meta-heuristic algorithms. The material density is 0.1 lb/in^3 (2768 kg/m^3), and the modulus of elasticity is 10^7 psi (68950 MPa). Twenty five members are divided into eight groups, as follows: (1) A_1, (2) A_2–A_5, (3) A_6–A_9, (4) A_{10}–A_{11}, (5) A_{12}–A_{13}, (6) A_{14}–A_{17}, (7) A_{18}–A_{21}, and (8) A_{22}–A_{25}. The allowable displacements and stresses are limited respectively to ±0.35 in (±8.89 mm) for each node and ±40 ksi (275.80 MPa).

Table 1 Summary of the CEC' 16 expensive optimization test problems

Categories	No.	Functions	Related basic functions	Fi*
Unimodal function	1	Rotated Bent Cigar Function	Bent Cigar Function	100
	2	Rotated Discus Function	Discus Function	200
Simple Multimodal functions	3	Shifted and Rotated Weierstrass Function	Weierstrass Function	300
	4	Shifted and Rotated Schwefel's Function	Schwefel's Function	400
	5	Shifted and Rotated Katsuura Function	Katsuura Function	500
	6	Shifted and Rotated HappyCat Function	HappyCat Function	600
	7	Shifted and Rotated HGBat Function	HGBat Function	700
	8	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Griewank's Function Rosenbrock's Function	800
	9	Shifted and Rotated Expanded Scaffer's F6 Function	Expanded Scaffer's F6 Function	900
Hybrid functions	10	Hybrid Function 1 (N=3)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	1000
	11	Hybrid Function 2 (N=4)	Griewank's Function Weierstrass Function Rosenbrock's Function Scaffer's F6 Function	1100
	12	Hybrid Function 3 (N=5)	Katsuura Function HappyCat Function Griewank's Function Rosenbrock's Function Schwefel's Function Ackley's Function	1200
Composition functions	13	Composition Function 1 (N=5)	Rosenbrock's Function High Conditioned Elliptic Function Bent Cigar Function Discus Function High Conditioned Elliptic Function	1300
	14	Composition Function 2 (N=3)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	1400
	15	Composition Function 3 (N=5)	HGBat Function Rastrigin's Function Schwefel's Function Weierstrass Function High Conditioned Elliptic Function	1500

Table 2 The statistical results for 10D problems

	Best		Worst		Median		Mean		Std	
Function	ITEO	TEO	ITEO	TEO	ITEO	TEO	ITEO	TEO	ITEO	TEO
f1	3.00E+10	**2.90E+10**	**6.78E+10**	8.24E+10	**5.16E+10**	5.61E+10	**5.12E+10**	5.33E+10	**1.26E+10**	1.67E+10
f2	5.64E+06	**4.70E+07**	**6.72E+08**	2.85E+09	**2.07E+08**	2.84E+08	**2.33E+08**	4.68E+08	**1.75E+08**	6.23E+08
f3	**9.6137**	10.156	**13.766**	13.781	**12.115**	12.134	**12.094**	12.099	1.1314	**0.85284**
f4	**2110.7**	2376.4	**3236**	3335.5	**2772.4**	2833.6	**2765.2**	2829.9	**283.37**	311.34
f5	2.1165	**1.8489**	4.5814	**3.9289**	**2.7313**	2.9048	**2.8484**	2.89	0.65242	**0.57429**
f6	11.184	**8.6745**	**21.16**	27.311	**17.69**	18.275	**17.039**	19.154	**3.3023**	5.4477
f7	291.15	**207.53**	**596.93**	671.23	502.01	**377.68**	464.72	**386.79**	**93.806**	110.47
f8	5.32E+07	**9.37E+06**	8.20E+09	**2.87E+09**	3.95E+08	**1.54E+08**	1.47E+09	**5.63E+08**	2.36E+09	**7.84E+08**
f9	**4.1501**	4.7719	**4.999**	5	**4.7042**	4.9969	**4.6882**	4.9742	0.20991	**0.057268**
f10	8.36E+06	**4.64E+06**	**9.06E+08**	1.30E+10	**9.65E+07**	1.80E+09	**1.84E+08**	3.31E+09	**2.37E+08**	3.80E+09
f11	**29.214**	92.393	**1365**	1.92E+05	**148.65**	3762.8	**259.39**	22555	**328.95**	46123
f12	**18053**	48917	**1.72E+07**	3.46E+07	**3.82E+05**	1.99E+06	**1.58E+06**	3.90E+06	**3.77E+06**	7.59E+06
f13	947.93	**561.38**	**4413.8**	4809.6	**3456.5**	1899.2	3111.7	**2315.1**	**1087.3**	1352.5
f14	239.78	**236.3**	**580.4**	1083.9	371.2	**352.8**	**379.58**	447.65	**97.791**	229.48
f15	**542.02**	575.2	713.6	**678.81**	**637.55**	639.21	637.16	**636.92**	37.676	**27.7**

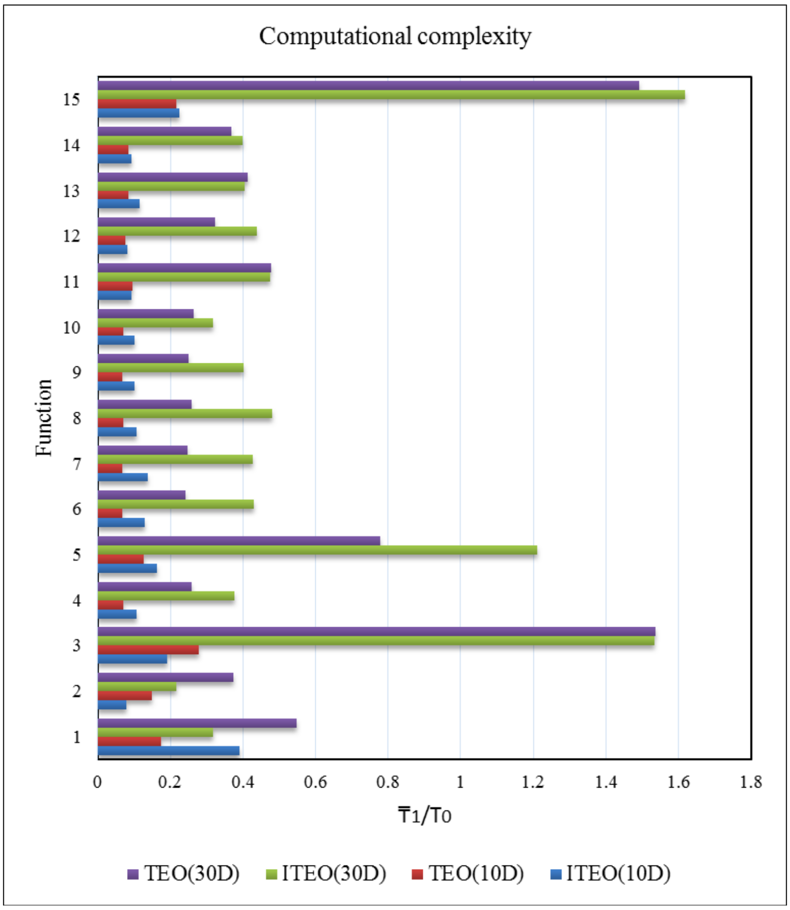

Fig. 5 The computational complexity of TEO and ITEO for different functions

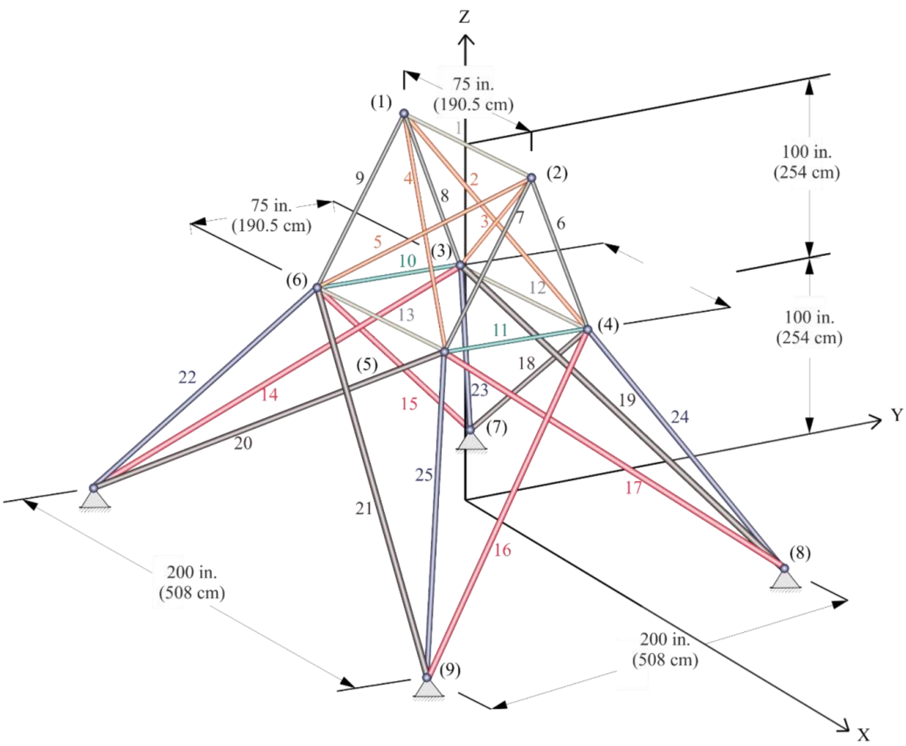

Fig. 6 Schematic of the spatial 25-bar truss

Table 3 The statistical results for 30D problems

	Best		Worst		Median		Mean		Std	
Function	ITEO	TEO	ITEO	TEO	ITEO	TEO	ITEO	TEO	ITEO	TEO
f1	**5.89E+10**	1.10E+11	**1.07E+11**	1.38E+11	**8.02E+10**	1.26E+11	**7.93E+10**	1.25E+11	1.31E+10	**9.09E+09**
f2	**3.66E+06**	3.79E+06	2.66E+09	**2.22E+07**	1.72E+07	**7.08E+06**	1.68E+08	**9.46E+06**	5.89E+08	**5.52E+06**
f3	**38.657**	38.864	47.216	**46.63**	**45.01**	45.387	44.801	**44.532**	**1.735**	1.9964
f4	**8815.4**	9004.2	**10445**	10714	**9844.2**	10030	**9845.2**	10072	411.08	**405.96**
f5	**2.3526**	3.7162	**4.9555**	5.263	**4.153**	4.2683	**4.125**	4.3837	0.60001	**0.45309**
f6	**11.598**	15.308	**17.013**	20.677	**14.954**	19.073	**14.636**	18.965	1.6525	**1.2807**
f7	**433.25**	556.15	**652.09**	825.63	**547.1**	767.31	**545.26**	753.09	65.935	**64.409**
f8	**3.92E+07**	5.07E+08	**1.67E+09**	5.11E+09	**3.71E+08**	2.14E+09	**4.75E+08**	2.58E+09	**4.09E+08**	1.46E+09
f9	14.684	**13.995**	14.997	**14.728**	14.945	**14.385**	**14.923**	14.399	**0.087448**	0.15623
f10	3.15E+07	**2.71E+07**	1.97E+11	**1.30E+09**	**1.26E+08**	4.06E+08	**1.31E+10**	4.80E+08	4.54E+10	3.73E+08
f11	**442.75**	1724	6.19E+05	**27107**	**940.87**	5708.3	**32046**	7937.2	1.38E+05	**6529.5**
f12	**3507**	11950	5.47E+13	**6.67E+09**	**2.75E+05**	1.31E+07	**2.75E+12**	7.27E+08	1.22E+13	**1.92E+09**
f13	**3194.7**	5456.6	**31716**	35473	**11132**	25453	**12682**	25223	8284.3	**7329.2**
f14	**309.85**	486.61	**974.16**	1679.6	**510.94**	1068.9	**539.84**	1113.6	159.84	**326.77**
f15	**1364.8**	1425	**1531**	1650.1	**1474.8**	1505.5	**1460.8**	1507.8	**45.072**	55.181

Table 4 The total score obtained by the algorithms

Dimension	ITEO	TEO
10	2.5970e+11	4.5512e+12
30	3.2432e+14	1.0619e+16

Table 5 Optimal design comparison for the 25-bar spatial truss

Element group	Optimal cross-sectional areas (in^2)					
	WEO	CBO	CS	MBA	TEO	ITEO
	(Kaveh and Bakhshpoori 2016)	(Kaveh and Ilchi Ghazaan 2015)	(Kaveh and Bakhshpoori 2013)	(Sadollah *et al.* 2012)	Present study	
1	0.1	0.1	0.1	0.1	0.1	0.2
2	0.3	0.3	0.3	0.3	0.3	0.7
3	3.4	3.4	3.4	3.4	3.4	3.1
4	0.1	0.1	0.1	0.1	0.1	0.3
5	2.1	2.1	2.1	2.1	1.3	0.7
6	1.0	1.0	1.0	1.0	0.9	0.9
7	0.5	0.5	0.5	0.5	0.8	1.1
8	3.4	3.4	3.4	3.4	3.4	3.3
Best weight (lb)	484.85	484.85	484.85	484.85	486.324	**484.605**
Mean Weight (lb)	485.598	486.87	485.01	484.89	491.622	**484.821**

In engineering, usually the design sections are selected from a predefined list, similarly in this structure the cross-sectional areas are considered from 0.1 to 3.4 in^2 (0.6452 to 21.94 cm^2) with 0.1 in^2 (0.6452 cm^2) increment. Detailed descriptions can be found in (Kaveh 2017).

In this example, c_1 and c_2 are equal to 0.25 and -0.5, respectively. As it can be seen from Table 5, the optimal design and average results found by ITEO has the minimum weight compared to other algorithms. The optimal sections found by ITEO differs from those of other algorithms while the other algorithms have found the same sections. The applied modifications have effectively improved the algorithm and the mean weight obtained by ITEO is lower than the best design of standard TEO. The convergence histories are represented in Fig. 7 showing that the curves related to ITEO are generally under the curves of TEO, this confirms again the statement about the efficiency of ITEO.

3.2.2 Spatial 72-bar truss

The spatial 72-bar truss is studied widely in structural optimization. Fig. 8 represents topology and geometry of the structure. The modulus of elasticity and material density are 10^4 ksi (68,950 MPa) and 0.1 lb/in^3 (2,767.990 kg/m^3), respectively. Seventy-two members are divided into eight groups, as follows: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}–A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66} (15), A_{67}–A_{70}, and (16) A_{71}– A_{72}.

Member groups can be designed from a list of 64 available ready sections. The final design must satisfy the stress limits of ±25 ksi (±172.375 MPa) and node displacement limits of ±0.25 in. (±0.635 cm). The truss subjected to multiple load cases. Detailed information can be found in (Kaveh 2017).

Table 6 Optimal design comparison for the 72-bar spatial truss

Element group	WEO (Kaveh and Bakhshpoori 2016)	ICA (Kaveh and Talatahari 2010)	CBO (Kaveh and Ilchi Ghazaan 2015)	CS (Kaveh and Bakhshpoori 2013)	MBA (Sadollah *et al.* 2012)	TEO	ITEO
						Present work	
1	1.99	1.99	1.62	1.800	0.196	1.8	1.800
2	0.563	0.442	0.563	0.563	0.563	0.563	0.563
3	0.111	0.111	0.111	0.111	0.442	0.111	0.111
4	0.111	0.141	0.111	0.111	0.602	0.25	0.250
5	1.228	1.228	1.457	1.266	0.442	1.228	1.266
6	0.442	0.602	0.442	0.563	0.442	0.442	0.442
7	0.111	0.111	0.111	0.111	0.111	0.111	0.196
8	0.111	0.141	0.111	0.111	0.111	0.141	0.141
9	0.563	0.563	0.602	0.563	1.266	0.602	0.563
10	0.563	0.563	0.563	0.442	0.563	0.563	0.442
11	0.111	0.111	0.111	0.111	0.111	0.111	0.111
12	0.111	0.111	0.111	0.111	0.111	0.141	0.250
13	0.196	0196	0.196	0.196	1.800	0.196	0.141
14	0.563	0.563	0.602	0.602	0.602	0.602	0.766
15	0.391	0.307	0.391	0.391	0.111	0.391	0.391
16	0.563	0.602	0.563	0.563	0.111	0.563	0.442
Best weight (lb)	389.33	392.84	391.07	389.87	390.72	396.65	**389.15**
Mean weight (lb)	391.20	N/A	403.71	N/A	395.432	416.74	**390.82**

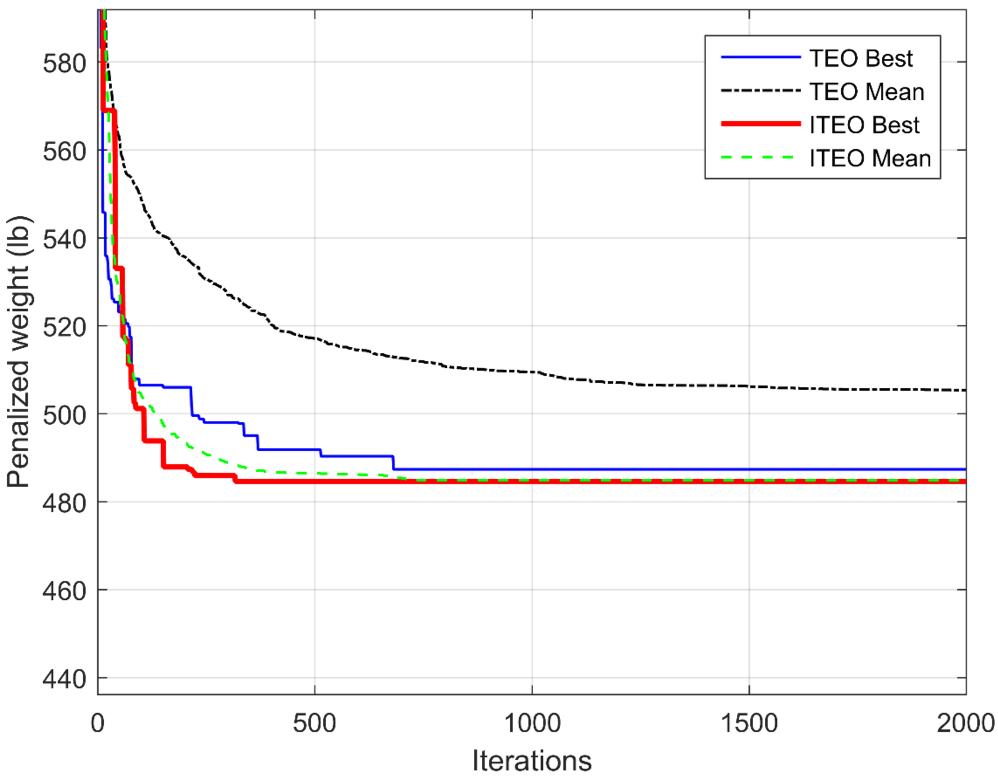

Fig. 7 Convergence curves recorded for the 25-bar truss problem

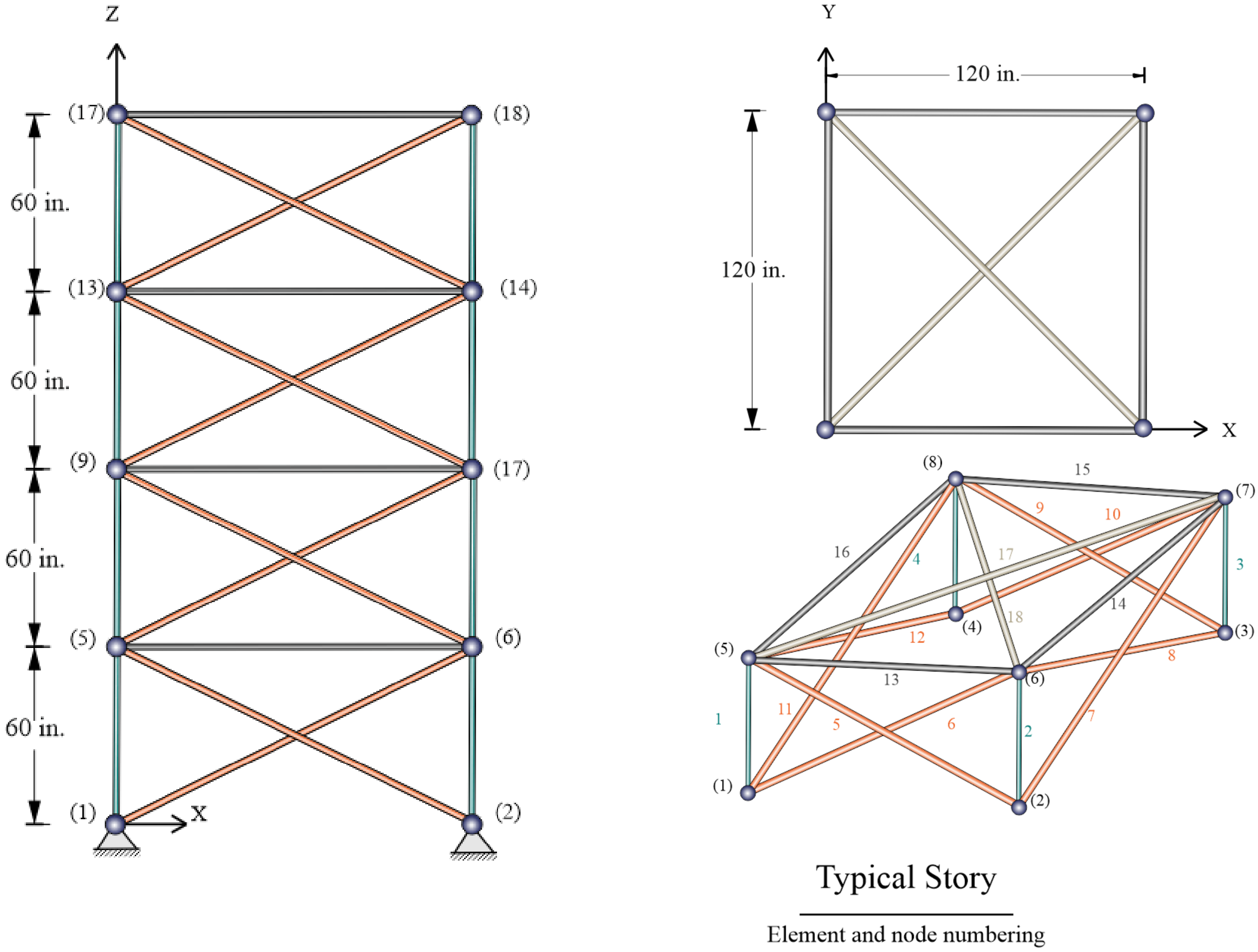

Fig. 8 Schematic of the spatial 72-bar truss

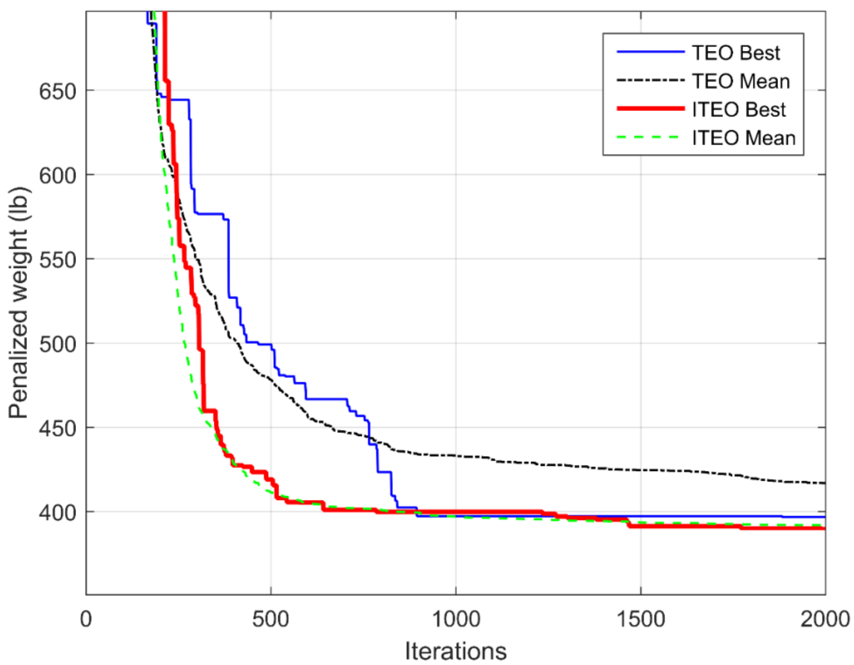

Fig. 9 Convergence curves recorded for the spatial 72-bar truss

Table 7 Optimal design comparison for the three-bay fifteen-story frame

Element group	Optimal cross-sectional areas				
	WEO	CBO	ICA	CSS	TEO	ITEO
	(Kaveh and Bakhshpoori 2016)	(Kaveh and Ilchi Ghazaan 2015)	(Kaveh and Talatahari 2010)	(Kaveh and Talatahari 2012)	Present work	
1	W14×90	W24×104	W24×117	W21×147	W14×90	W24×131
2	W36×170	W40×167	W21×147	W18×143	W30×173	W21×147
3	W30×90	W27×84	W27×84	W12×87	W12×79	W30×90
4	W24×104	W27×114	W27×114	W30×108	W30×116	W24×104
5	W24×68	W21×68	W14×74	W18×76	W24×68	W21×93
6	W12×87	W30×90	W18×86	W24×103	W12×106	W30×90
7	W8×48	W8×48	W12×96	W21×68	W16×50	W8×48
8	W14×68	W21×68	W24×68	W14×61	W18×71	W21×68
9	W10×33	W14×34	W10×39	W18×35	W8×31	W14×38
10	W16×45	W8×35	W12×40	W10×33	W16×45	W8×35
11	W21×44	W21×50	W21×44	W21×44	W21×44	W21×44
Best weight (lb)	88710.97	93795	93486	92723	90514.58	**87754.57**
Mean weight (lb)	90649.49	98738	N/A	N/A	99737.430	**90477.73**

Table 8 Comparison of optimization results obtained by TEO and some other metaheuristics for the 3-bay 24-story frame problem

Element group	Optimal cross-sectional areas					
	WEO	CBO	ICA	CSS	TEO	ITEO
	(Kaveh and Bakhshpoori 2016)	(Kaveh and Ilchi Ghazaan 2015)	(Kaveh and Talatahari 2010)	(Kaveh and Talatahari 2012)	Present work	
1	W14×176	W14×109	W14×132	W14×132	W14×176	W14X159
2	W14×145	W14×159	W14×120	W14×109	W14×145	W14X99
3	W14×145	W14×120	W14×145	W14×109	W14×109	W14X132
4	W14×132	W14×90	W14×82	W14×90	W14×90	W14X132
5	W14×109	W14×74	W14×61	W14×61	W12×14	W8X13
6	W14×109	W14×68	W14×43	W14×38	W12×14	W6X15
7	W14×90	W14×30	W14×38	W14×38	W8×13	W4X13
8	W14×82	W14×38	W14×22	W14×22	W6×8.5	W8X10
9	W14×74	W14×159	W14×99	W14×109	W5×16	W6X16
10	W14×68	W14×132	W14×109	W14×109	W5×16	W8X18
11	W14×61	W14×99	W14×82	W14×99	W5×16	W10X17
12	W14×43	W14×82	W14×90	W14×90	W6×15	W10X15
13	W14×34	W14×68	W14×74	W14×82	W6×15	W5X16
14	W14×34	W14×48	W14×61	W14×68	W8×13	W6X12
15	W14×34	W14×34	W14×30	W14×34	W6×9	W10X12
16	W14×22	W14×22	W14×22	W14×22	W6×8.5	W6X8.5
17	W30×90	W30×90	W27×102	W30×90	W30×90	W30X90
18	W21×50	W21×50	W8×18	W6×15	W8×18	W10X19
19	W21×48	W24×55	W24×55	W24×55	W24×68	W24X68
20	W12×19	W8×28	W6×8.5	W6×8.5	W8×18	W6X20
Best weight (lb)	**202626**	215874	212640	212364	206328.06	202890.04
Mean weight (lb)	204954.03	225071	N/A	215226	225862.37	**204741.97**

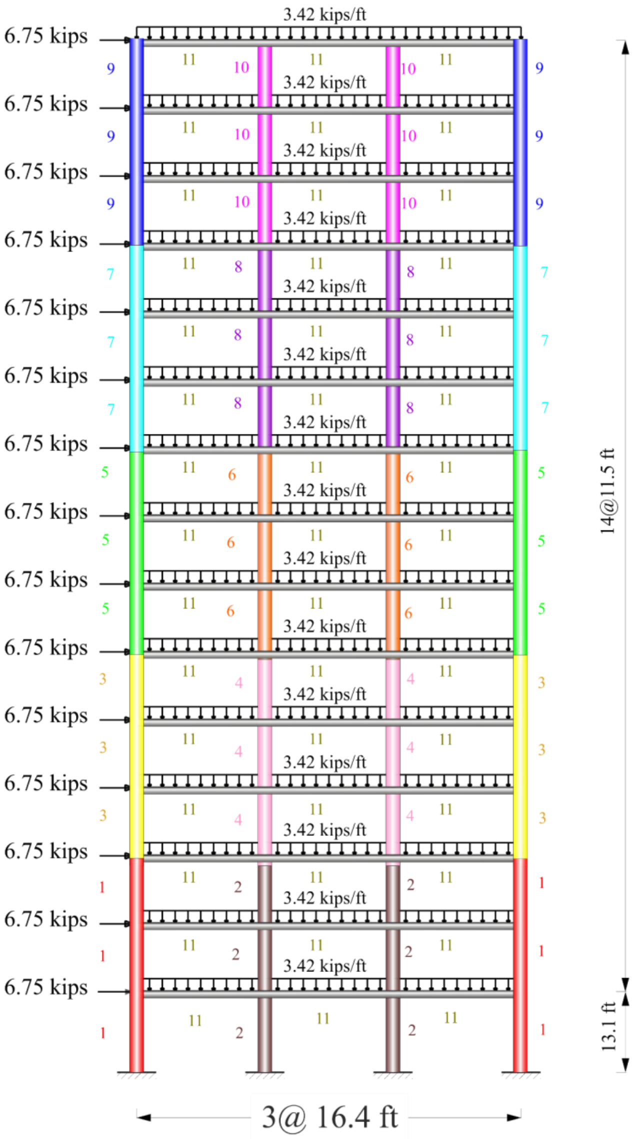

Fig. 10 Schematic of the three-bay fifteen-story frame

The optimal solutions found by various algorithms are provided in Table 6. The ITEO has obtained the minimum weight among the compared algorithms. The mean weight obtained by ITEO is lighter than the lightest obtained by TEO and it is almost 7% better than that of TEO, which shows, the adjustments of this paper have improved TEO significantly. This also can be observed in the convergence curves illustrated in Fig. 9.

3.2.3 Three-bay fifteen-story frame

This benchmark frame consists of 105 members which are divided into 10 groups for columns and a beam element group chosen from 267 W-sections. As it can be seen from Fig. 10, the frame is subjected to uniform and lateral service loading. The design must satisfy the displacement and AISC-LRFD combined strength constraints, also, the lateral sway of the top story is limited to 9.25 in (23.5 cm).

The optimum and mean of the obtained by different algorithms are collected in Table 7. The optimal design found by ITEO is the best design reported in the table. The average weight of the ITEO calculated from 30 independent runs is also lighter than that of other algorithms. ITEO also performed better than the standard TEO with lower iterations. The convergence curves of the best solution and average of 30 independent runs, presented in Fig. 11, show the good converging behavior of ITEO compared to that of the TEO.

3.2.4 Three-bay twenty four-story frame

The last problem is a 168-member, three-bay twenty four-story frame originally designed by Davison and Adams (Davison and Adams 1974). Fig. 12 displays the topology, configuration, service loading conditions and numbering of member groups for the frame. The column members are categorized into 16 groups chosen from 267 W-shapes and 4 beam groups limited to W14 sections.

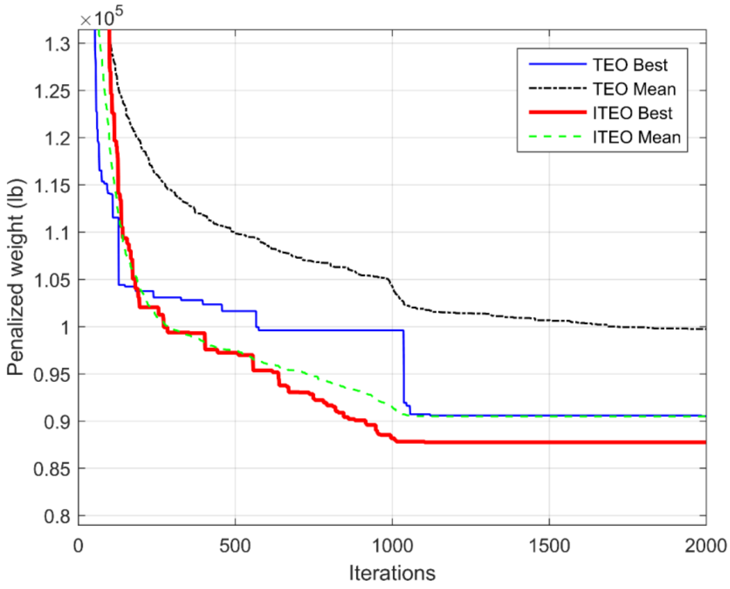

Fig. 11 Convergence curves for three-bay fifteen-story frame problem

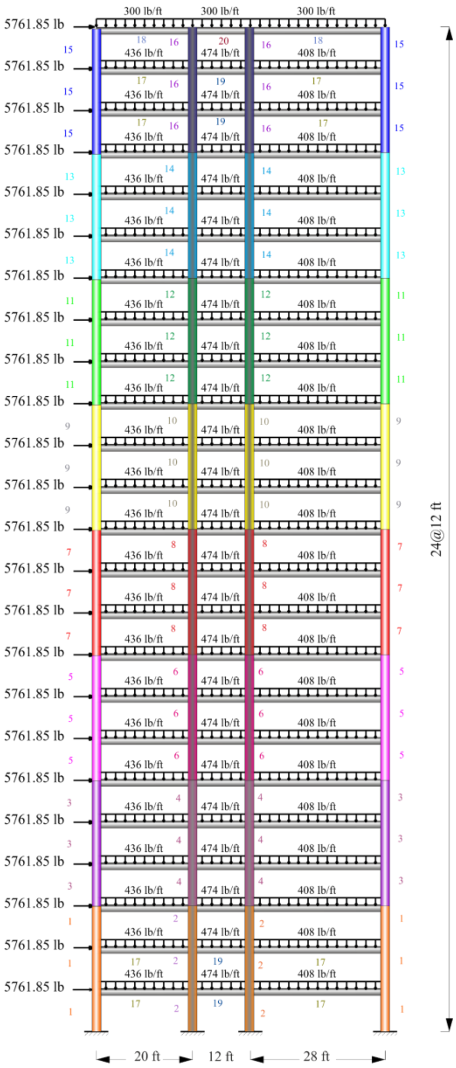

Fig. 12 Schematic of the three-bay twenty four-story frame

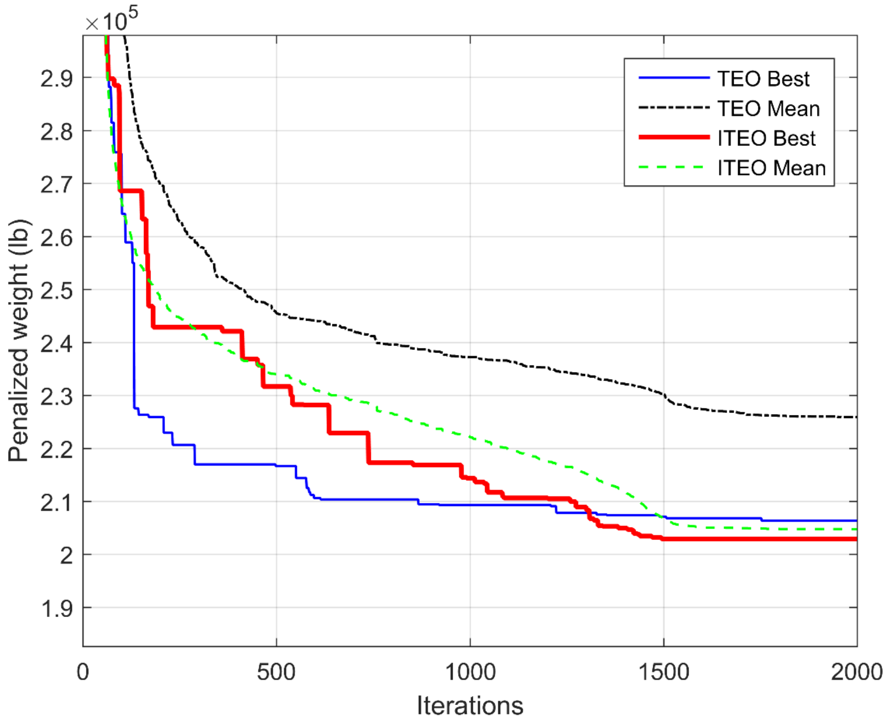

Fig. 13 Comparison of the convergence curves for the TEO and ITEO algorithms for the three-bay twenty four-story frame

According to the optimum results reported in Table 8, ITEO has found the second minimum weight among other powerful methods, but this should be noted that the weight obtained by WEO is about 0.1% lower than the improved version of ITEO and the mean weight of ITEO is better than other algorithms. As illustrated in Fig. 13, the average convergence rate of ITEO is much faster than TEO.

4. Conclusions

TEO is a recently developed optimization algorithm which is inspired by cooling law of Newton. In this paper, an improved version of the TEO (abbreviated as ITEO) is presented. ITEO uses the linear statement instead of exponential statement utilized in the standard version and this leads to additional saving in the computational cost. The applicability of the ITEO has been studied considering CEC 16's computationally expensive problems and the design optimization of benchmark structures with discrete cross-sectional areas. According to the results achieved by 30 independent runs, the ITEO has displayed a competitive performance compared to the other well-known algorithms. As it can be seen from the first example, ITEO has found a new better design while other algorithms have converged to identical designs. This shows the different search approach of the ITEO. According to the results, the modifications have improved the performance of the algorithm and the new version has outperformed the standard TEO in most of the studied examples.

References

Chen, Q., Liu, B., Zhang, Q., Liang, J., Suganthan, P. and Qu, B. (2014), Problem definition and evaluation criteria for CEC 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore, Tech. Rep.

Davison, J.H. and Adams, P.F. (1974), "Stability of braced and unbraced frames", *J. Struct. Div*., **100**(2), 319-334.

Geem, Z.W., Kim, J.H. and Loganathan, G. (2001), "A new heuristic optimization algorithm: harmony search", *Simulation*, **76**(2), 60-68.

Gholizadeh, S. (2013), "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization", *Comput. Struct.*, **125**, 86-99. doi:10.1016/j.compstruc.2013.04.024

Goldberg, D.E. and Samtani, M.P. (1986), Engineering optimization via genetic algorithm, Paper presented at the Electronic Computation:

Hasançebi, O. and Azad, S.K. (2015), "Adaptive dimensional search: A new metaheuristic algorithm for discrete truss sizing optimization", *Comput. Struct*., **154**, 1-16. doi:https://doi.org/10.1016/j.compstruc.2015.03.014

Kaveh, A. (2017), Applications of metaheuristic optimization algorithms in civil engineering, Springer, Switzerland.

Kaveh, A. and Bakhshpoori, T. (2013), "Optimum design of space trusses using cuckoo search algorithm with lévy flights. Iranian", *J. Sci. Technol. T. Civil Eng*., **37**(1), 1-15.

Kaveh, A. and Bakhshpoori, T. (2016), "An accelerated water evaporation optimization formulation for discrete optimization of skeletal structures", *Comput. Struct*., **177**, 218-228. doi:https://doi.org/10.1016/j.compstruc.2016.08.006

Kaveh, A. and Dadras, A. (2017a), "A novel meta-heuristic

optimization algorithm: Thermal exchange optimization", *Adv. Eng. Softw*., **110**. 69-84
Kaveh, A. and Dadras, A. (2017b), "Structural damage identification using an enhanced thermal exchange optimization algorithm", *Eng. Optimiz*., 1-22. doi:10.1080/0305215X.2017.1318872
Kaveh, A. and Ilchi Ghazaan, M. (2015), "A comparative study of CBO and ECBO for optimal design of skeletal structures", *Comput. Struct*., **153**, 137-147. doi:https://doi.org/10.1016/j.compstruc.2015.02.028
Kaveh, A. and Khayatazad, M. (2012), "A new meta-heuristic method: Ray optimization", *Comput. Struct*., **112-113**, 283-294. doi:https://doi.org/10.1016/j.compstruc.2012.09.003
Kaveh, A. and Mahdavi, V.R. (2014), "Colliding bodies optimization: A novel meta-heuristic method", *Comput. Struct*., **139**, 18-27. doi:https://doi.org/10.1016/j.compstruc.2014.04.005
Kaveh, A. and Mahjoubi, S. (2017), "Design of multi-span steel box girder using lion pride optimization algorithm", *Smart Struct. Syst*., **20**(15), 607-618.
Kaveh, A. and Talatahari, S. (2010), "Optimum design of skeletal structures using imperialist competitive algorithm", *Comput. Struct*., **88**(21-22), 1220-1229. doi:https://doi.org/10.1016/j.compstruc.2010.06.011
Kaveh, A. and Talatahari, S. (2012), "Charged system search for optimal design of frame structures", *Appl. Soft Comput*., **12**(1), 382-393. doi:https://doi.org/10.1016/j.asoc.2011.08.034
Koumousis, V.K. and Georgiou, P.G. (1994), "Genetic algorithms in discrete optimization of steel truss roofs", *J. Comput. Civil Eng*., **8**(3), 309-325.
Maheri, M.R. and Narimani, M. (2014), "An enhanced harmony search algorithm for optimum design of side sway steel frames", *Comput. Struct*., **136**, 78-89.
Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014), "Grey wolf optimizer", *Adv. Eng. Softw*., **69**, 46-61.
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**(3), 303-315. doi:https://doi.org/10.1016/j.cad.2010.12.015
Rueda Torres, J. and Erlich, I. (2016), Solving the CEC2016 Real-Parameter Single Objective Optimization Problems through MVMO-PHM: Technical report.
Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2012), "Mine blast algorithm for optimization of truss structures with discrete variables", *Comput. Struct.*, **102**, 49-63.
Saka, M.P., Aydogdu, I., Hasancebi, O. and Geem, Z.W. (2011), Harmony Search Algorithms in Structural Engineering. (Eds., X.S. Yang and S. Koziel), Computational Optimization and Applications in Engineering and Industry (145-182), Berlin, Heidelberg: Springer Berlin Heidelberg.
Saremi, S., Mirjalili, S. and Lewis, A. (2017), "Grasshopper optimisation algorithm: Theory and application", *Adv. Eng, Softw.*, **105**, 30-47.
Talbi, E.G. (2009), Metaheuristics: from design to implementation, John Wiley & Sons.
Wang, B., Jin, X. and Cheng, B. (2012), "Lion pride optimizer: An optimization algorithm inspired by lion pride behaviour", *Science China Inform. Sciences*, **55**(10), 2369-2389.
Wang, D., Wu, Z., Fei, Y. and Zhang, W. (2014), "Structural design employing a sequential approximation optimization approach", *Comput. Struct*., **134**, 75-87. doi:https://doi.org/10.1016/j.compstruc.2013.12.004
Yang, J. and Soh, C.K. (1997), "Structural optimization by genetic algorithms with tournament selection", *J. Comput. Civil Eng.*, **11**(3), 195-200.
Yeniay, Ö. (2005), "Penalty function methods for constrained optimization with genetic algorithms", *Math. Comput. Appl*., **10**(1), 45-56.
Yi, T.H., Wen, K.F. and Li, H.N. (2016), "A new swarm intelligent optimization algorithm: Pigeon Colony Algorithm (PCA)", *Smart Struct. Syst*., **18**(3), 425-448. doi:10.12989/sss.2016.18.3.425

CC