Advances in Engineering Software 110 (2017) 69–84

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

A novel meta-heuristic optimization algorithm: Thermal exchange optimization

A. Kaveh*, A. Dadras

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

ARTICLE INFO

Article history:
Received 6 December 2016
Revised 23 March 2017
Accepted 29 March 2017
Available online 18 April 2017

Keywords:
Optimization
Metaheuristic algorithm
Thermal exchange optimization
Mathematical functions
Mechanical systems

ABSTRACT

This paper introduces a new optimization algorithm based on Newton's law of cooling, which will be called Thermal Exchange Optimization algorithm. Newton's law of cooling states that the rate of heat loss of a body is proportional to the difference in temperatures between the body and its surroundings. Here, each agent is considered as a cooling object and by associating another agent as environment, a heat transferring and thermal exchange happens between them. The new temperature of the object is considered as its next position in search space. The performance of the algorithm is examined by some mathematical functions and four mechanical benchmark problems.

1. Introduction

In engineering design, choosing design variables that fulfill all design requirements and have the lowest possible cost is concerned, i.e. the main objective is to comply with basic standards but also to achieve economic designs [1]. In practice, a designer is usually satisfied with good solutions, which are obtained by heuristic or metaheuristic algorithms. Meta-heuristics represent a family of approximate optimization techniques that have gained a lot of popularity in the past two decades. These are among the most promising and successful techniques. Metaheuristics provide acceptable solutions in a reasonable time for solving hard and complex problems in science and engineering [2].

Some of the metaheuristic approaches are inspired by nature. These can be classified into three categories in terms of the source of inspiration:

(a) Evolutionary Algorithms. The Evolutionary Algorithm (EA) proposed by Fogel et al. [3], De Jong [4] and Koza [5], and the Genetic Algorithm (GA) proposed by Holland [6] and Goldberg [7] are inspired by biological evolution, such as reproduction, mutation, recombination and selection.

(b) Swarm Algorithms. These methods mimic the processes of decentralized, self-organized systems, which can be either natural or artificial in nature. Studies on animal behavior led to Ant Colony Optimization (ACO) proposed by Dorigo et al. [8] which follows the processes of an ant colony searching for food, the navigation ability of dolphins was mimicked in Dolphin echolocation proposed by Kaveh and Farhoudi [9], Grey wolf optimizer formulated by Mirjalili et al. [10] that mimics the hunting process of grey wolves, Eberhart and Kennedy's Particle Swarm Optimizer (PSO) [11] imitates animal flocking behaviors.
This class of algorithms are often inspired by animal's behaviors of looking for food, locating, flocking and their other smart techniques.

(c) Physical algorithms. These methods are inspired by the physical laws. Of these, Water Evaporation Optimization (WEO) proposed by Kaveh and Bakhshpoori [12] imitate the evaporation of a tiny amount of water molecules, Charged System Search (CSS) introduced by Kaveh and Talatahari [13] which utilizes the governing Coulomb law from electrostatics and the Newtonian laws of mechanics. Simulated Annealing proposed by Kirkpatrick et al. [14] is inspired from annealing in metallurgy, the Big Bang–Big Crunch algorithm proposed by Erol and Eksin [15] that mimics the Big Bang and Big Crunch theory, Hsiao, et al. [16] developed the Space Gravitational Optimization and Gravitational Search Algorithm (GSA) presented by Rashedi et al. [17] that is based on the law of gravity and the Vibrating Particles System (VPS) developed by Kaveh and Ilchi Ghazaan [18] mimics the free vibration of single degree of freedom systems with viscous damping.

In recent years a big number of novel methods have been developed and applied to different multidisciplinary optimization

* Corresponding author.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

http://dx.doi.org/10.1016/j.advengsoft.2017.03.014

problems. Some of these methods are listed in the following: Glowworm Swarm Optimization (GSO) [19], Firefly Algorithm (FA) [20], Monkey Search (MS) [20], Bat Algorithm (BA) [21], Krill Herd (KH) Algorithm [22], Bird Mating Optimizer (BMO) [23], Tug of war optimization [24], Social Spider Optimization (SSO–C) [25] and Water Cycle Algorithm (WCA) [26].

A wide range of algorithms have been introduced, improved and applied by the first author and his students which some of them are mentioned above.

The main objective of this paper is to present a new optimization algorithm based on principles from physics, which is called Thermal Exchange Optimization (TEO). In this, each agent is considered as a cooling object and by associating another agent as surrounding fluid, a heat transferring and thermal exchanging happens between them. This algorithm utilizes Newton's law of cooling to update the temperatures. The process is repeated until the satisfaction of the termination conditions.

Simulated annealing (SA) is a well-known metaheuristic algorithm [14]. Partly, SA and TEO use common terminologies, but their operations are technically different. SA's operation can be summarized as follows:

SA models the physical process of heating a material and then slowly lowering the temperature to decrease defects and to produce crystals, thus minimizing the system energy. At each iteration of SA, a new position is randomly generated. The distance of the new position from the current position, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that lower the objective, but also, with a certain probability, positions that raise the objective. By accepting positions that raise the objective, the algorithm avoids being trapped in local minima in early iterations and is able to explore globally for better solutions.

However in TEO, temperature of each object is its position and by grouping the objects, they start to exchange it. New temperatures will be their new positions.

The rest of this paper is organized as follows. In Section 2, a brief overview of the TEO is presented. Section 3 uses benchmark functions to compare TEO with some other popular optimization methods. Section 4 verifies the parameters of the algorithm and its performance during iterations. Finally, conclusions are derived in Section 5.

2. Thermal exchange optimization

2.1. Back ground

2.1.1. Newton's law of cooling

Newton's law of cooling states that the rate of heat loss of a body is proportional to the difference in temperatures between the body and its surroundings. The following is the Newton's law of cooling in his own words:

"The iron was laid not in a calm air, but in a wind that blew uniformly upon it, that the air heated by the iron might be always carried off by the wind and the cold air succeed it alternately; for thus equal parts of air were heated in equal times, and received a degree of heat proportional to the heat of the iron", as shown in Fig. 1, Refs. [27,28].

2.1.2. Theory

The modern lumped parameter approach to transient cooling is given in many textbooks, e.g. [29,30]. We assume the overall heat transfer coefficient to be h, and the physical properties are constant. The shape of the solid region is irrelevant (except that it will affect the calculation of h). The object starts at time $t = 0$ at a high temperature 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 and its surface area is A. The rate of heat loss from the surface is:

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

where

A = area for heat flow m^2, T = temperature K

h = heat transfer coefficient $W\ m^{-2}K^{-1}$, t = time s

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

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

where

V = volume m^3, ρ = density $kg\ m^{-3}$ and c = spaecific heat $J\ kg^{-1}K^{-1}$

Integration gives

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

The integration is only valid when $\frac{hAt}{V\ \rho c}$ is constant, i.e. not a function of T, so we can write

$$\beta = \frac{hA}{V\ \rho c} \tag{4}$$

And from Eq. (3):

$$\frac{T - T_b}{T_0 - T_b} = \exp(-\beta \mathrm{t}) \tag{5}$$

This equation can be rearranged as:

$$T = T_b + (T_0 - T_b)\exp(-\beta t) \tag{6}$$

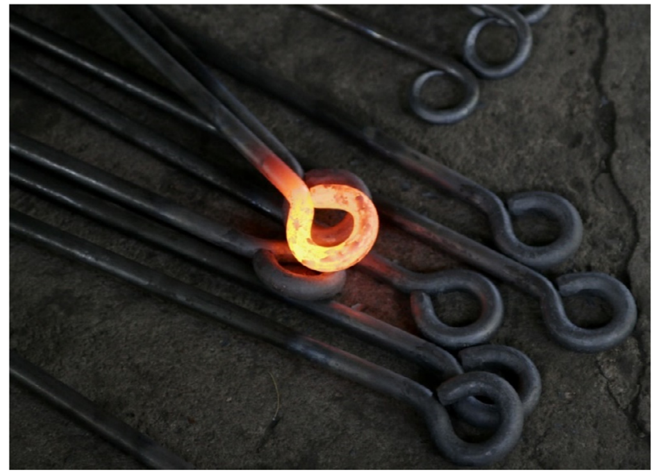

Fig. 1. Hot iron objects, transferring heat to the surrounding environment.

2.2. Presentation of thermal exchange optimization

The main objective of this section is to formulate the new effective physically-based meta-heuristic algorithm which is called Thermal Exchange Optimization (TEO).

2.2.1. Inspiration

In TEO algorithm, some agents are defined as the cooling objects and the remaining agents are supposed to represent the environment, then we do it contrariwise. Choosing the cooling objects and environment ones are similar to the grouping of bodies in CBO and ECBO [31–33].

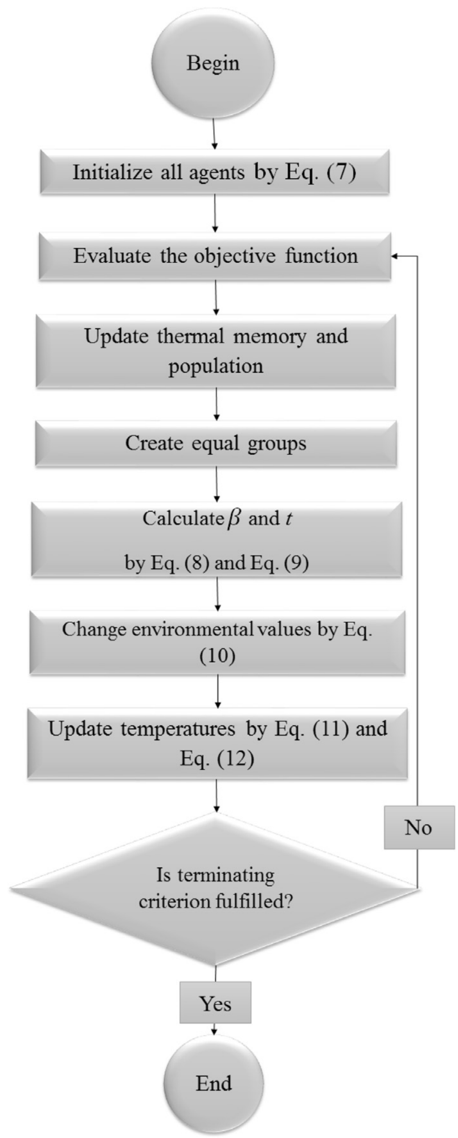

Fig. 2. Flowchart of the TEO algorithm.

2.2.2. Steps

The flowchart of the TEO is illustrated in Fig. 2 and the steps involved are given in the following:

Step 1: Initialization

The initial temperature of all the objects are determined in an m-dimensional search space.

$$T_i^0 = T_{min} + random.(T_{max} - T_{min}) \tag{7}$$

where T_i^0 is the initial solution vector of the ith object. Here T_{min},T_{max} are the bounds of design variables; random is a random vector which each component is in interval [0,1]; n is the number of objects.

Step 2: Evaluation

The objective function calculates the cost value of every object.

Step 3: Saving

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

Step 4: Creating groups

Agents are divided into two equal groups. The pairs of agents are defined according to Fig. 3. For instance T_1 is an environment object for $T_{\frac{n}{2}+1}$ cooling object, and vice versa.

Step 5: Defining β

In the nature when an object has lower β, it exchanges the temperature slightly. By inspiring this feature a similar formulation is suggested. The value of β for each object is evaluated according to Eq. (8). In this, lower cost object has lower β and it changes the position slightly.

$$\beta = \frac{Cost\,(object)}{Cost\,(worst\ object)} \tag{8}$$

Step 6: Defining t

Time is associated with the iteration number. In this formulation the value of t for each agent is evaluated according to Eq. (9) as

$$t = \frac{iteration}{Max\ iteration} \tag{9}$$

Step 7: Escaping from local optima (i)

Metaheuristic algorithms should have the ability to escape from traps, when agents get close to a local optimum. Steps 7 and 9 are utilized to escape from the trap. In these steps the environmental temperature is changed by Eq. (10), where c_1 and c_2 are the controlling variables.

$$T_i^{env.} = (1 - (c_1 + c_2 \times (1 - t)) \ \times random.) \times T_i'^{\,env.} \tag{10}$$

where c_1 and c_2 are the controlling variables. $T_i'^{\,env.}$ is the previous temperature of the object, which is modified to $T_i^{env.}$.

- $(1 - t)$ is considered to decrease the randomness by nearing to the last iterations. By nearing to the end of the process, t increases, leading to a linear decrease in randomness and increasing 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 a decreasing process is not employed ($c_2 = 0$; *as said above*) c_1 involves the randomness.

Here, $C = 0$ $(c_1 = c_2 = 0)$, none of the above mentioned mechanisms are employed and 1 is multiplied with the previous temperature. In this paper c_1 and c_2 are chosen from {0 or 1}.

Step 8: Updating the agents

According to the previous steps and Eq. (6), new temperature of each object is updated by

$$T_i^{new} = T_i^{env.} + \left(T_i^{Old} - T_i^{env.}\right) \exp\left(-\beta t\right) \tag{11}$$

Step 9: Escaping from local optima (ii)

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

$$T_{i,j} = T_{j,min} + random\ .\left(T_{j,max} - T_{j,min}\right) \tag{12}$$

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

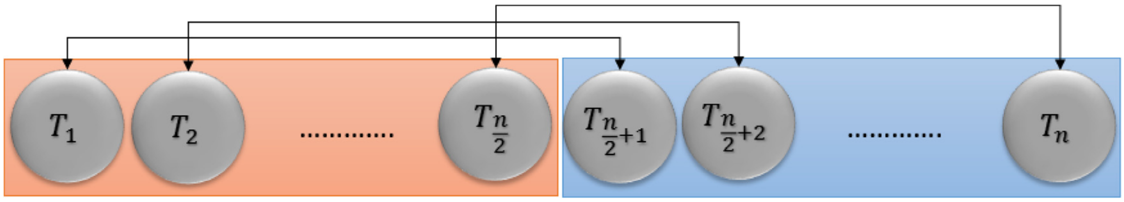

Fig. 3. Heat transferring groups and the pairs of environment and cooling objects.

Table 1
Rosenbrock function.

Function name	Function	Dim	Interval	Global minimum
Rosenbrock	$f(\mathbf{X}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	30	[−30,30]	0

order to protect the structures of the agents, only one dimension is changed. This mechanism provides opportunities for the agents to move all over the search space thus providing better diversity.

Step 10: Checking terminating conditions

The optimization process will be 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 will be stopped and the best found solution will be reported.

3. Verification of the algorithm

This section, first explains how the proposed method employs exploitation and exploration of the search space then different values for c_1,c_2, *pro* and *TM* are selected and the sensitivity of the average of 30 independent alternating runs is discussed and the behavior of the algorithm is verified.

3.1. Exploration and exploitation

These are two important features that should to be considered for any metaheuristic algorithm. Exploration means the ability of the algorithm to escape from local minima and to avoid trapping by generating new solutions at all over the search space. On the other hand, exploitation tries to improve good solutions by generating similar ones, which helps converging the algorithm. A good balance may lead to good efficiency. However, this itself is an unresolved optimization task.

Metaheuristics usually use random based parameters to perform exploration. TEO explores the search space by involving random parameters as explained in Steps 7 and 9.

Also the exploitation is considered by focusing on the best found solutions. For this purpose, TEO utilizes thermal memory (TM), which saves a number of best-so-far solutions and replace them with worst objects, as mentioned in Step 3.

Furthermore TEO is implicitly equipped to elitism mechanism. In this case, 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. (11), the cooling object will have tendency to the environmental temperature (better position) and vice versa.

3.2. Sensitivity analysis

The selected function for this analysis is Rosenbrock [34] which is a non-convex function (Table 1, Fig. 4).

As written in Table 2, c_1, c_2 are selected from {0 or 1} to reduce the complexity of the problem. Thus four scenarios are considered to verify the parameters, as shown in Fig. 5. The best parameter tuning for each scenario is marked with red rectangle in the figure, also they are given in Table 3. By considering $c_1 = 1$ in the second scenario, the cost is decreased, using $c_2 = 1$ in the third scenario, decreases the cost and the number of TMs. These show the importance of the defined parameters in improving the algorithm operation and reducing the required memory in a sample problem.

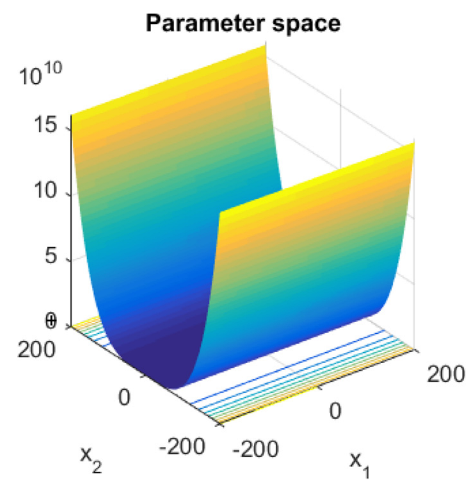

Fig. 4. Rosenbrock function.

Table 2
Values chosen for the sensitivity analysis.

Parameter	Values
c_1	{0 or 1}
c_2	{0 or 1}
pro	Factors of 0.05 from [0,0.5]
TM	Integers from [0,10]

Table 3
Parameter tuning for different scenarios.

Scenarios Parameters	Scenario I $c_1 = 0,\ c_2 = 0$	Scenario II $c_1 = 0,\ c_2 = 1$	Scenario III $c_1 = 1,\ c_2 = 0$	Scenario IV $c_1 = 1,\ c_2 = 1$
Pro	0.4	0.3	0.2	0.15
TM	10	8	5	4
Cost	818.8724	28.568813	28.724357	28.664437

3.3. Convergence curves

For this section, Griewank function is chosen, which is a challenging multi modal function with multiple maxima and minima. The function is presented in Table 4, also its two dimensional diagram is drawn in Fig. 6.

The convergence curves of some algorithms for this function are shown in Fig. 7. According to Table 5, the TEO algorithm obtained the global minima (=0) in 170 iterations, while the comparing algorithms trapped in local minima. This can be a good example for

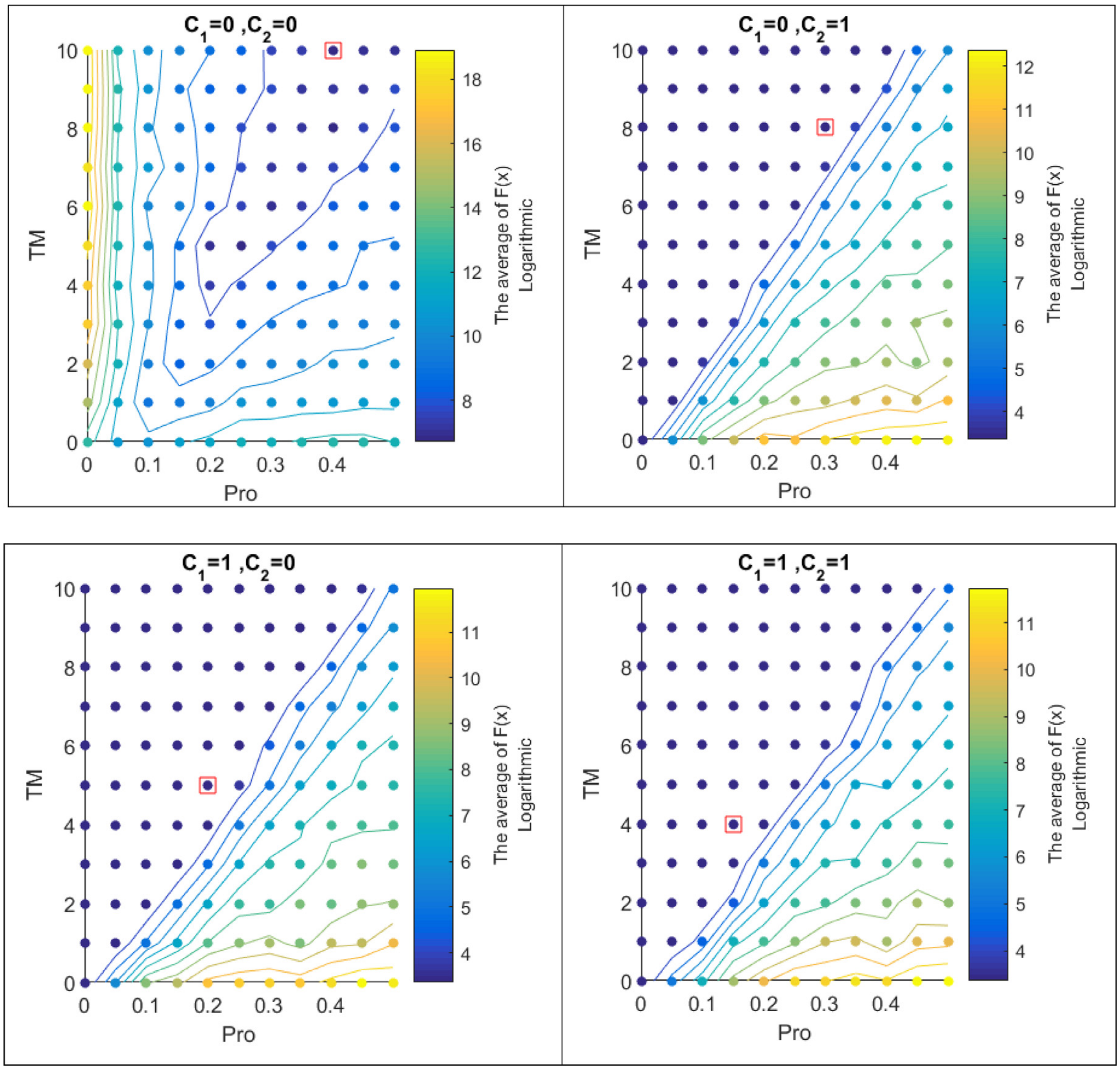

Fig. 5. Results of the sensitivity analysis carried out for the scenarios.

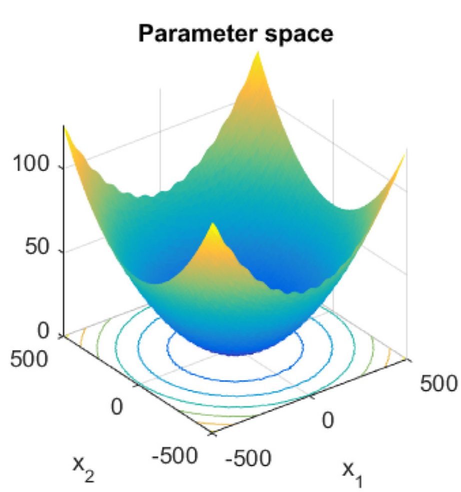

Fig. 6. Griewank function.

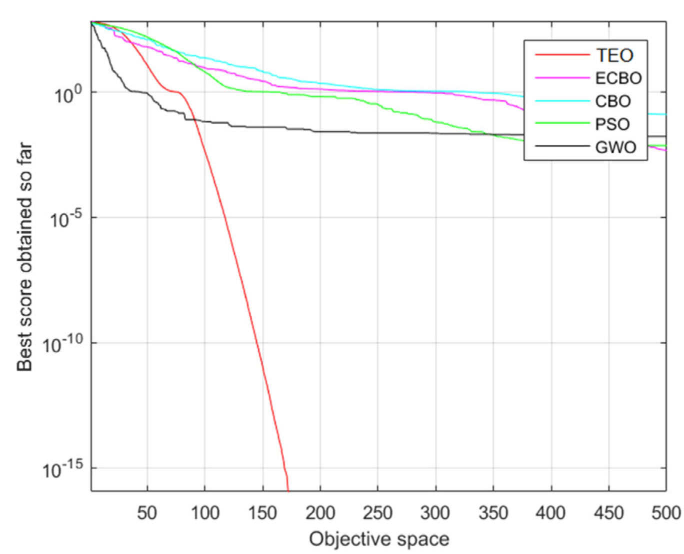

Fig. 7. Convergence curves for several algorithms.

Table 4
Griewank function.

Function name	Function	Dim	Interval	Global minimum
Griewank	$f(\mathbf{X}) = 1 + \frac{1}{200}\sum_{i=1}^{2} x_i^2 - \prod_{i=1}^{2}\cos(\frac{x_i}{\sqrt{i}})$	30	[−600,600]	0

Table 5
Comparing the results of the methods.

Method	Best score
CBO	0.258678
ECBO	0.004520
PSO	0.007408
GWO	0.029375
Present work	0

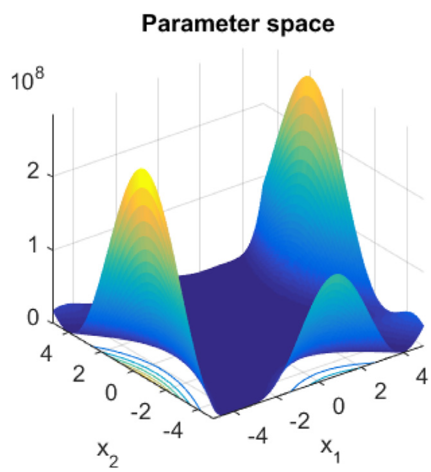

Fig. 8. Goldstein and Price function.

showing the fast convergence and the strong exploration capability of TEO.

3.4. Verification of the search history

In this part, the positions of the objects are studied in different iterations. For this purpose, we chose Goldstein and Price function which has many local optima and is 2-dimensional. Also the maximum number of iterations is limited to 200. The function is detailed in Table 6 and plotted in Fig. 8.

The positions of the objects in different iterations are illustrated in Fig. 9 by blue stars and the saved objects in TM are marked with red rectangles. The algorithm obtained the global minima. The stars out of squares in 150 and 200 iterations are indicating that the global search mechanism is controlled but it is working even in the final iteration.

4. Benchmark functions

In order to verify the features of the new algorithm, some numerical examples are considered from literature and compared to some other popular optimization methods. The examples contain 13 uni-modal and multi-modal functions which were tackled by many researchers [35,36]. Also four engineering problems are verified by the algorithm and the results are compared to those of other algorithms.

4.1. Mathematical optimization problems

These problems are described in Table 7. The first five functions (f1–f5) are unimodal functions and f6–f13 are multimodal. Dim indicates the dimension of each function, Range is the boundaries of the function's search space, and fmin is the global optimum value.

All problems are solved considering 30 agents, 500 iterations and the algorithm is run 30 times on each benchmark function frequently to obtain reliable results. Fig. 10 illustrates the 2D versions of the used benchmark functions.

4.1.1. Results and discussion

The statistical results (average and standard deviation) are reported in Table 8. For verification, the results of the TEO are compared to those of GWO (animal behavior-based) [10], GSA (physically-based) [16], PSO (Swarm Intelligence-based) [37] and DE (Fast Evolutionary programming) [38].

According to the results of Table 8, TEO outperforms all the other considered algorithms for all of the functions, but it is in the second place in F5. Outperforming in uni-modal problems show good exploitation capability of the algorithm. TEO has obtained very good results in multi-modal functions which have many local optima. It has found the global minima in F7, F8, F9, F11 and F13 which indicates the global search capability of the TEO. On the other hand achieving standard deviations which are equal or near to zero, shows good robustness of TEO.

4.2. Engineering optimization problems

In this section the TEO is examined by four engineering benchmark problems and the penalty approach is used for constraint handling. Parameter settings of TEO on engineering design problems are listed in Table 9 and the number of iterations are limited to 10,000.

4.2.1. Welded beam design

The first engineering problem considers the design optimization of the welded beam shown in Fig. 11. The goal of this problem is to find the minimum manufacturing cost of the welded beam subjected to constraints on shear stress (s), bending stress (r), buckling load (Pc), deflection (d) and side constraints. The design variables are the thickness of the weld $h(=x_1)$, length of attached part of the bar $l(=x_2)$, the height of the bar $t(=x_3)$ and thickness of the bar $b(=x_4)$.

The objective function can be mathematically be stated as:

$$f_{cost}(\mathbf{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \qquad (13)$$

Constraints are:

$$\begin{aligned}
g_1(\mathbf{X}) &= \tau(\{x\}) - \tau_{max} \le 0\\
g_2(\mathbf{X}) &= \sigma(\{x\}) - \sigma_{max} \le 0\\
g_3(\mathbf{X}) &= x_1 - x_4 \le 0\\
g_4(\mathbf{X}) &= 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0\\
g_5(\mathbf{X}) &= 0.125 - x_1 \le 0\\
g_6(\mathbf{X}) &= \delta(\{x\}) - \delta_{max} \le 0\\
g_7(\mathbf{X}) &= P - P_c(\{x\}) \le 0
\end{aligned} \qquad (14)$$

where

$$\tau(\mathbf{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

Table 6
Goldstein and Price function.

Function name	Function	Dim	Interval	Global minimum
Goldstein and Price	$f(\mathbf{X}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2]$ $\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 - 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	2	[−2,2]	3

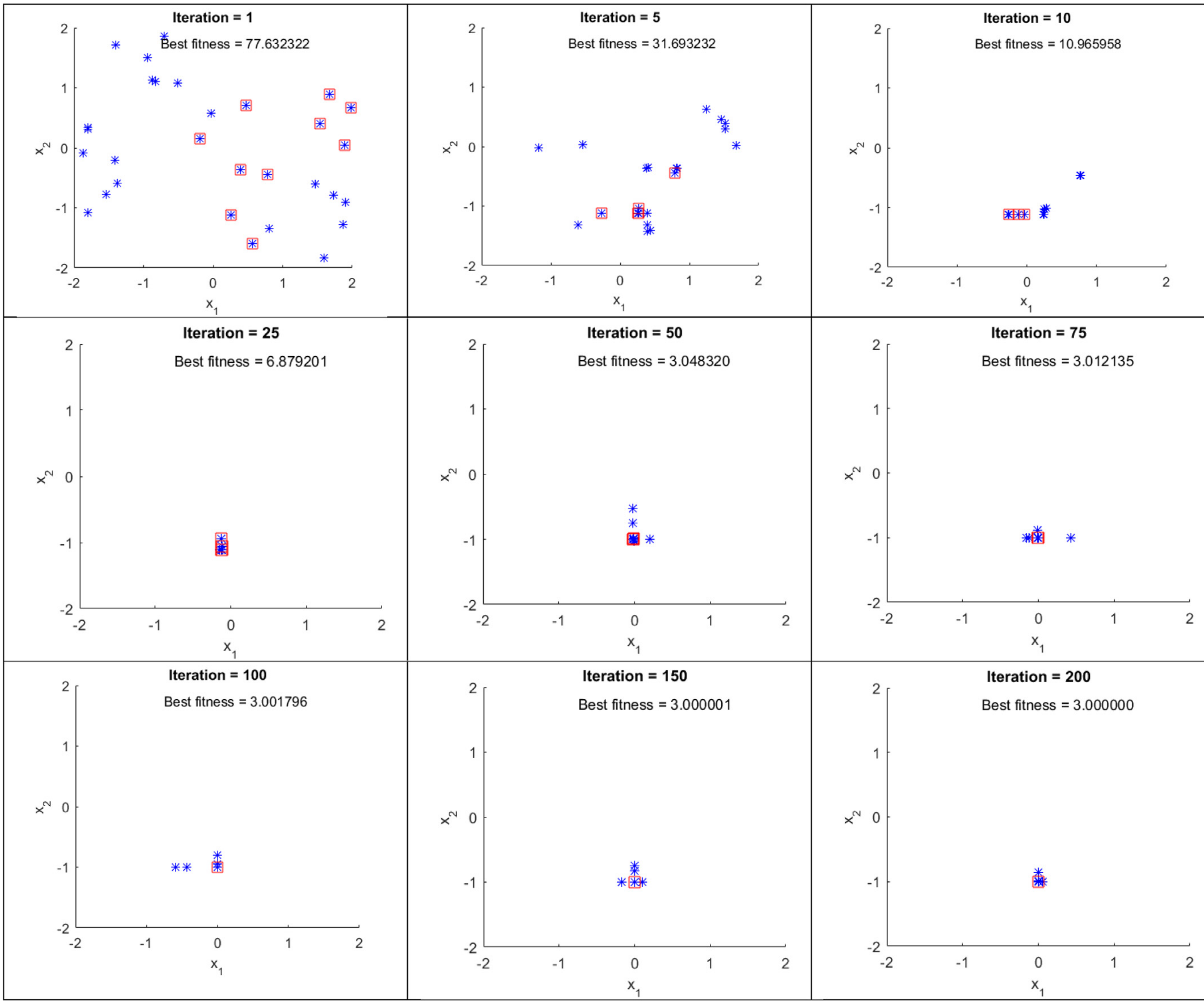

Fig. 9. Positions of the objects in different iterations.

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\mathbf{X}) = \frac{6PL}{x_4x_3^2}, \delta(\mathbf{X}) = \frac{4PL^3}{Ex_3^3x_4}$$

$$P_c(\mathbf{X}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000lb, L = 14in,$$
$$E = 30 \times 10^6 psi, G = 12 \times 10^6 psi \tag{15}$$

Variable bounds are:

$$0.1 \le x_1 \le 2$$
$$0.1 \le x_2 \le 10$$
$$0.1 \le x_3 \le 10$$
$$0.1 \le x_4 \le 2 \tag{16}$$

This problem has been solved by Deb [39], Coello [40] using GA-based methods also by Lee and Geem [41] using HS Algorithm. Radgsdell and Phillips [42] compared optimal results of different optimization methods which were mainly based on mathematical optimization algorithms. The optimal results of different methods are listed in Table 10. The statistical results are summarized in Table 11. According to this table, TEO is able to find a design with the minimum cost. Additionally, TEO could offer a competitive set of statistical results in less number of function evaluations as shown in Fig. 12.

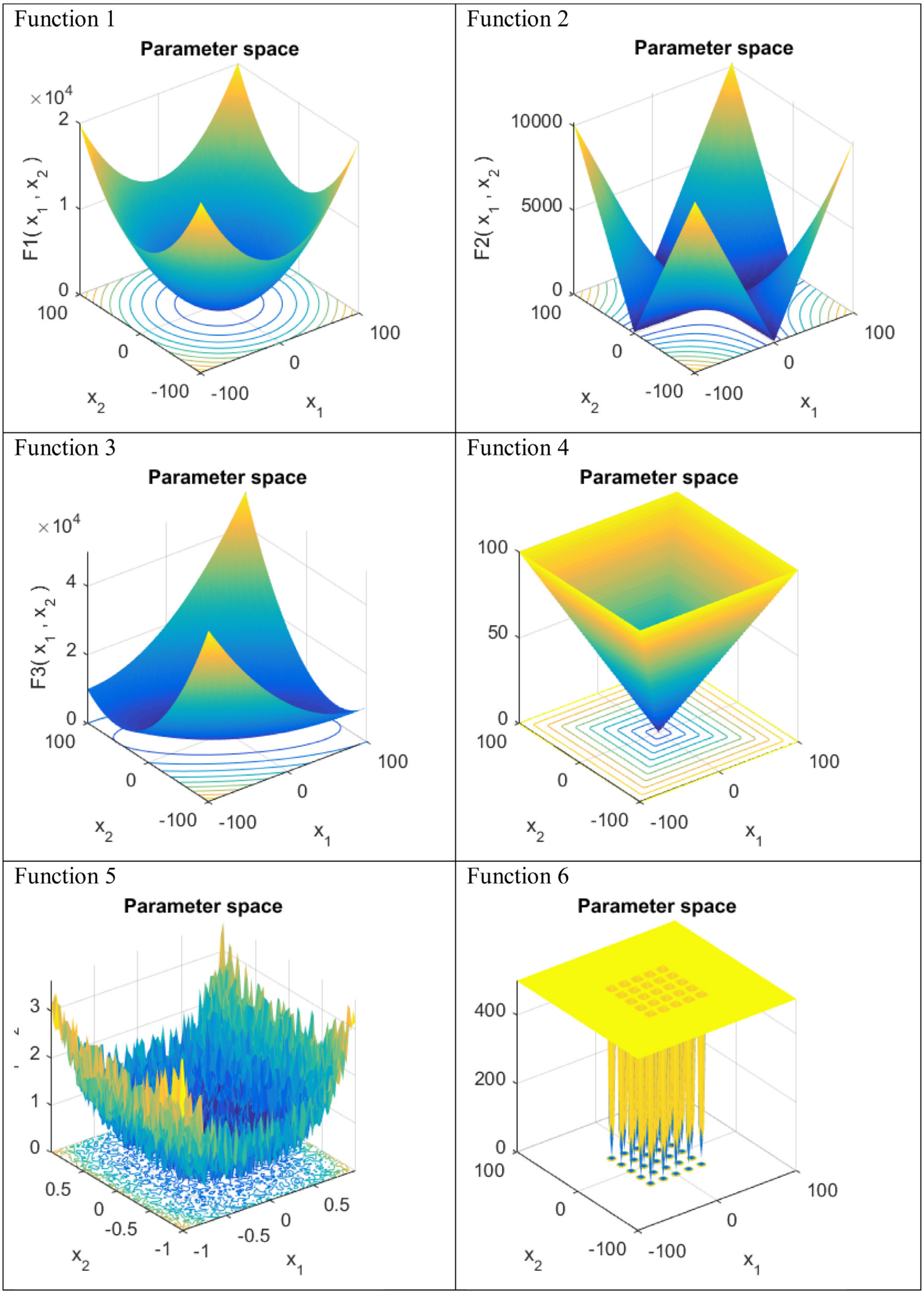

Fig. 10. A perspective view for some of functions with dimension 2.

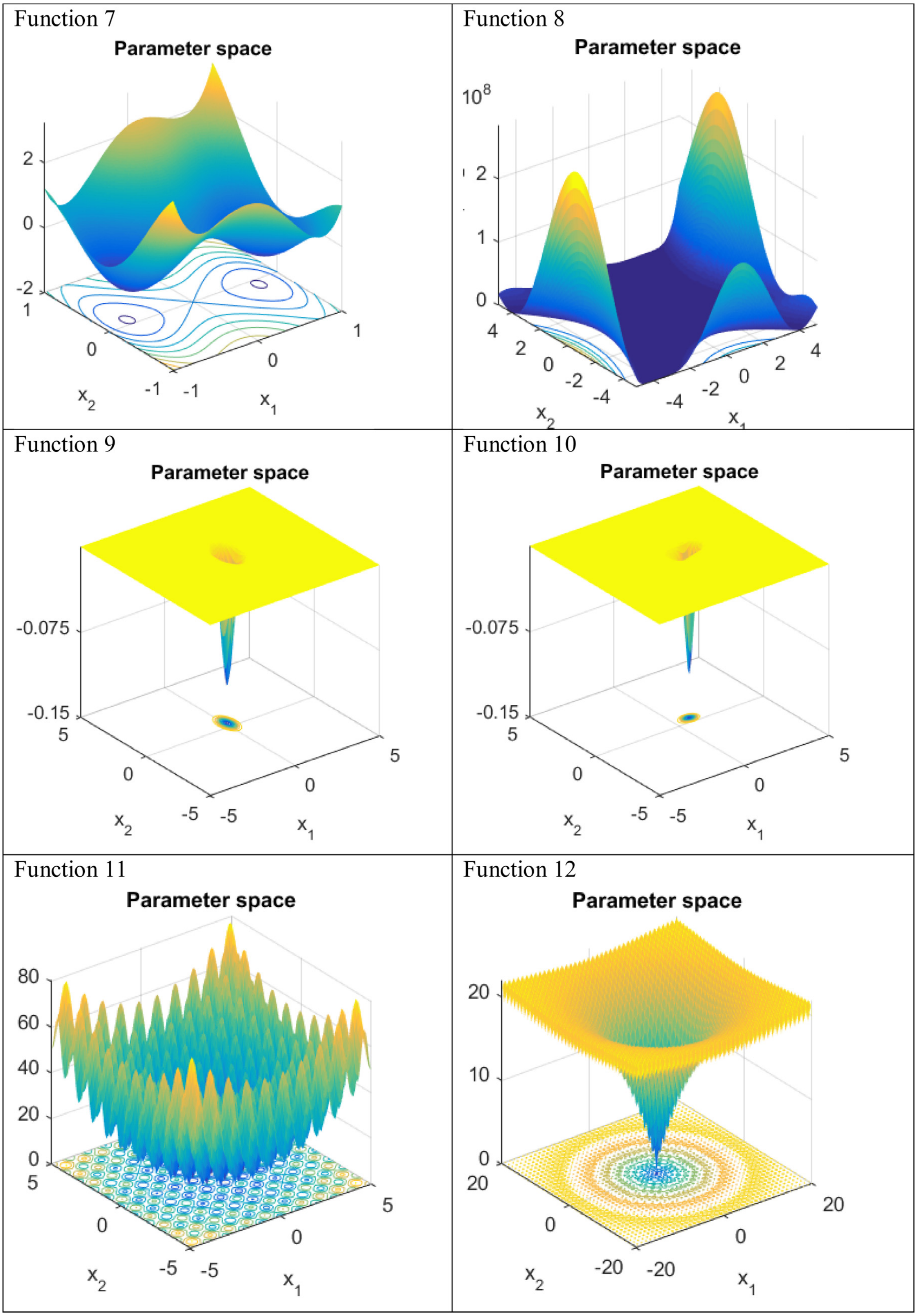

Fig. 10. Continued

Table 7
Uni-modal and multi-modal benchmark functions.

Function	Dim(n)	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} {x_i}^2$	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[−100,100]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} {x_i}^2 \right)^2$	30	[−100,100]	0
$f_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{n} i{x_i}^4 + random[0,1)$	30	[−100,100]	0
$f_6(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	[−65,65]	1
$f_7(x) = 4{x_1}^2 - 2.1{x_1}^4 + \frac{1}{3}{x_1}^6 + x_1x_2 - 4{x_2}^2 + 4{x_2}^4$	2	[−5,5]	−1.0316
$f_8(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2)]$	2	[−2,2]	3
$f_9(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2 \right)$ $a = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}$ and $p = \begin{bmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}$	3	[1,3]	−3.862782
$f_{10}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2 \right)$ $a = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 17 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{bmatrix}$ and $p = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$	6	[0,1]	− 3.322368
$f_{11}(x) = -\sum_{i=1}^{n} [{x_i}^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$f_{12}(x) = -20\exp\left(-0.2^2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} {x_i}^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	[−32,32]	0
$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[−600,600]	0

Table 8
Results of uni-modal and multi modal benchmark functions.

F	Present work		GSA		PSO		GWO		DE	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	3.0306E-102	1.0010E-101	2.53E-16	9.67E-17	0.000136	0.000202	6.59E-28	6.34E-05	8.2E-14	5.9E-14
F2	1.6357E-55	8.6580E-56	0.055655	0.194074	0.042144	0.045421	7.18E-17	0.029014	1.5E-09	9.9E-10
F3	3.0606E-49	1.6661E-48	896.5347	318.9559	70.12562	22.11924	3.29E-06	79.14958	6.8E-11	7.4E-11
F4	3.4477E-50	2.8251E-50	7.35487	1.741452	1.086481	0.317039	5.61E-07	1.315088	0	0
F5	4.9923E-05	4.8522E-05	0.089441	0.04339	0.122854	0.044957	0.002213	0.100286	0.00463	0.0012
F6	0.998004	1.123681E-07	5.859838	3.831299	3.627168	2.560828	4.042493	4.252799	0.998004	3.3E-16
F7	-1.03163	2.170053E-08	0.003673	0.001647	−1.03163	6.25E-16	−1.03163	−1.03163	−1.03163	3.1E-13
F8	3.000000	2.554632E-13	1.03163	4.88E16	3	1.33E-15	3.000028	3	3	2E-15
F9	-3.862782	3.724816E-09	0.397887	0	−3.86278	2.58E-15	−3.86263	3.86278	N/A	N/A
F10	-3.286326	0.062508	-3	4.17E15	−3.26634	0.060516	Wrong*	−3.25056	N/A	N/A
F11	0	0	25.96841	7.470068	46.70423	11.62938	0.310521	47.35612	69.2	38.8
F12	8.88182e-16	0	0.062087	0.23628	0.276015	0.50901	1.06E-13	0.077835	9.7E-08	4.2E-08
F13	0	0	27.70154	5.040343	0.009215	0.007724	0.004485	0.006659	0	0

Error: The reported value is less than global minima.
N/A= Not available.

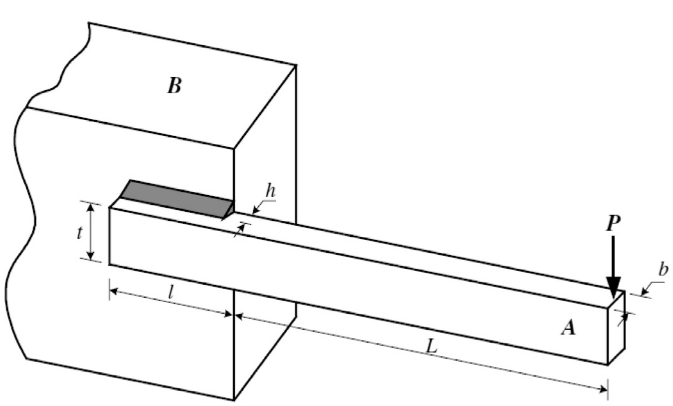

Fig. 11. Schematic of a welded beam design.

Table 9
The parameter settings of TEO on engineering design problems.

Problem	C_1	C_2	Pro	TM
Welded beam design	0	1	0.15	5
tension/compression spring design	1	1	0.3	5
stepped cantilever beam design	0	0	0.3	4
pressure vessel design	1	1	0.25	5

4.2.2. Design of tension/compression spring

This problem is described by Belegundu [55] and Arora [48]. It consists of minimizing the weight of a tension/compression spring subjected to constraints on shear stress, surge frequency and minimum deflection as shown in Fig. 13.

The objective function can be mathematically stated as:

$$f_{\text{cost}}(\mathbf{X}) = (x_3 + 2)x_2x_1^2 \tag{17}$$

Table 10
Optimum results for welded beam design.

Methods	$h(=x_1)$	$l(=x_2)$	$t(=x_3)$	$b(=x_4)$	f_{cost}
RANDOM	0.4575	4.7313	5.0853	0.66	4.1185
DAVID	0.2434	6.2552	8.2915	0.2444	2.3841
SIMPLEX	0.2792	5.6256	7.7512	0.2796	2.5307
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815
GA1 [39]	0.248900	6.173000	8.178900	0.253300	2.433116
GA2 [40]	0.208800	3.420500	8.997500	0.210000	1.748310
HS [41]	0.2442	6.2231	8.2915	0.2443	2.3807
GSA [16]	0.182129	3.856979	10	0.202376	1.879952
CPSO [46]	0.202369	3.544214	9.04821	0.205723	1.728024
ESs [45]	0.199742	3.61206	9.0375	0.206082	1.7373
FGA [51]	0.205986	3.471328	9.020224	0.206480	1.728226
CDE [62]	0.203137	3.542998	9.033498	0.206179	1.733462
GA3 [43]	0.205986	3.471328	9.020224	0.206480	1.728226
Present work	0.205681	3.472305	9.035133	0.205796	1.725284

Table 11
Statistical results of different algorithms for welded beam design.

Methods	Best	Mean	Worst	Std Dev
GP [42]	2.3815	N/A*	N/A	N/A
GA1 [39]	2.433116	N/A	N/A	N/A
GA2 [40]	1.748309	1.771973	1.785835	0.011220
GA3 [43]	1.728226	1.792654	1.993408	0.074713
CPSO [44]	1.728024	1.748831	1.782143	0.012926
ESs [45]	1.737300	1.813290	1.994651	0.070500
MGA [47]	1.824500	1.919000	1.995000	5.37E-02
DE [49]	1.733461	1.768158	1.824105	2.22E-02
CDE [62]	1.733461	1.768158	1.824105	0.022194
Rank-iMDDE [72]	1.724852309	1.724852309	1.724852309	9.06E-16
Present work	1.725284	1.768040	1.931161	0.0581661

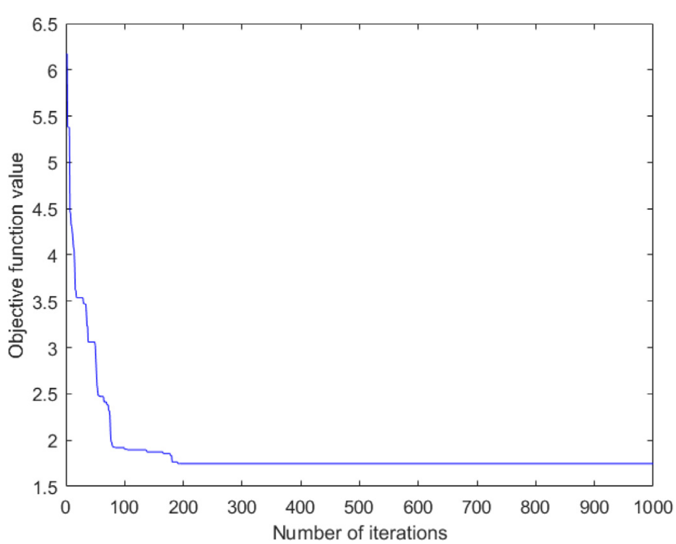

Fig. 12. Function values versus number of iterations for the welded beam problem.

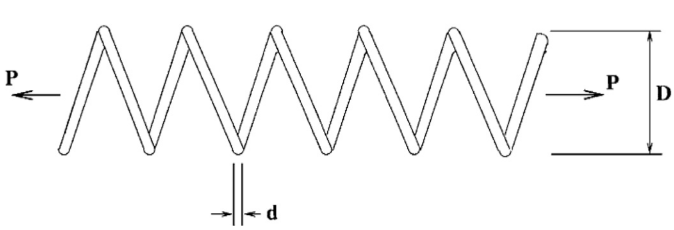

Fig. 13. Schematic of the tension/compression spring.

Table 12
Optimum results obtained by various algorithms for the tension/compression spring problem.

Methods	$d(=x_1)$	$D(=x_2)$	$N(=x_3)$	f_{cost}
Belegundu [55]	0.050000	0.315900	14.250000	0.012833
Arora [48]	0.053396	0.399180	9.185400	0.012730
GA2 [39]	0.051480	0.351661	11.632201	0.012704
GA3 [43]	0.051989	0.363965	10.890522	0.012681
CPSO [44]	0.051728	0.357644	11.244543	0.012674
ESs [50]	0.051643	0.355360	11.397926	0.012698
GSA [16]	0.050276	0.323680	13.525410	0.012702
DEDS [73]	0.051689	0.356717	11.288965	0.012665
HEAA [74]	0.051689	0.356729	11.288293	0.012665
DELC [67]	0.051689	0.356717	11.288965	0.012665
BA [61]	0.05169	0.35673	11.2885	0.012665
WEO [12]	0.051685	0.356630	11.294103	0.012665
Rank-iMDDE [72]	0.051689	0.35671718	11.288999	0.012665
WCA [26]	0.05168	0.356522	11.30041	0.012665
Present work	0.051775	0.3587919	11.168390	0.012665

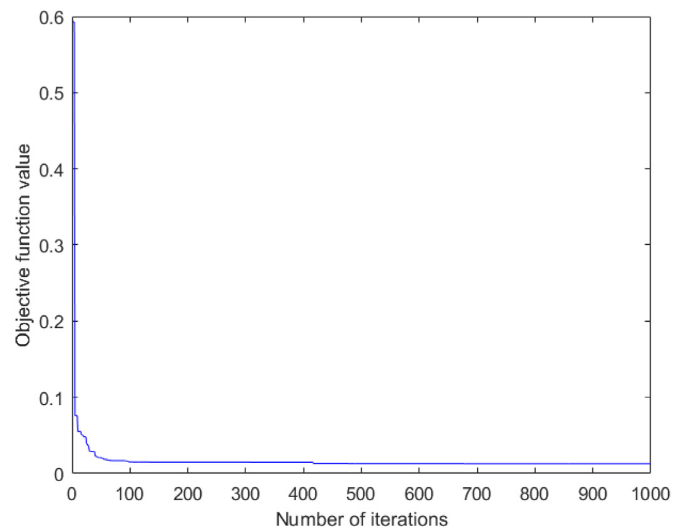

Fig. 14. Function values versus number of iterations for the tension/compression spring.

to be minimized and constraints

$$
\begin{aligned}
g_1(\mathbf{X}) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0 \\
g_2(\mathbf{X}) &= \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \le 0 \\
g_3(\mathbf{X}) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0 \\
g_4(\mathbf{X}) &= \frac{x_1 + x_2}{1.5} - 1 \le 0.
\end{aligned} \tag{18}
$$

The design space is bounded by

$$
\begin{aligned}
0.05 &\le x_1 \le 2 \\
0.25 &\le x_2 \le 1.3 \\
2 &\le x_3 \le 15.
\end{aligned} \tag{19}
$$

It can be seen from Table 12 that optimum design found by the TEO has the minimum cost. The mean of 30 independent designs by TEO is only 0.03% more than BIANCA, which is the best mean quoted so far in literature, Table 13. However, the standard deviation of TEO is the minimum value among other methods. Fig. 14 is the convergence curve of this problem, which indicates the good convergence rate of the TEO.

Table 13
Statistical results of different methods for the tension/compression spring problem.

Methods	Best	Mean	Worst	Std Dev
Belegundu [47]	0.0128334	N/A*	N/A	N/A
Arora [48]	0.0127303	N/A	N/A	N/A
GA2 [40]	0.0127048	0.012769	0.012822	3.9390e-5
GA3 [43]	0.0126810	0.0127420	0.012973	5.9000e-5
CPSO [44]	0.0126747	0.012730	0.012924	5.1985e-5
ESs [50]	0.012698	0.013461	0.16485	9.6600e-4
SCM [75]	0.012669	0.012923	0.016717	5.90E-04
CDE [62]	0.01267	0.012703	0.01279	2.70E-05
AATM [76]	0.012668	0.012708	0.012861	4.50E-05
CVI-PSO [69]	0.012666	0.012731	0.012843	5.58E-05
BIANCA [70]	0.012671	0.012681	0.012913	5.12E-05
BA [61]	0.012665	0.013501	0.016895	0.00142
WCA [26]	0.012665	0.012746	0.012952	8.06e-05
Rank-iMDDE [72]	0.012665233	0.012665264	0.01266765	2.45E-07
Present work	0.012665	0.012685	0.012715	4.4079e-06

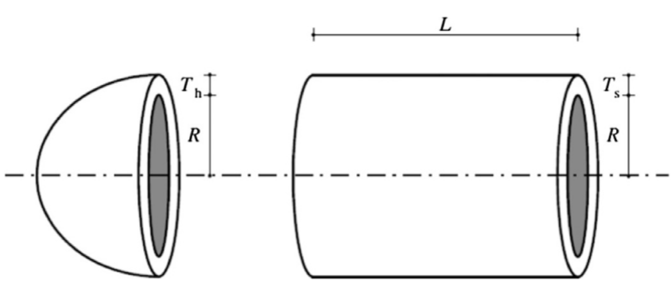

Fig. 15. Schematic of a pressure vessel.

4.2.3. Pressure vessel design

Another well-known engineering optimization problem is the design of a pressure vessel. This problem has been tackled by both mathematical and heuristic approaches. A cylindrical pressure vessel capped at both ends by hemispherical heads is presented in Fig. 15. The objective is to minimize the total cost, including the cost of material, forming and welding:

$$f_{cost}(x) = 0.6224x_1x_3x_4 + 1.7881x_2{x_3}^2 + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3 \tag{20}$$

Table 15
Statistical results of different methods for the pressure vessel design problem.

Methods	Best	Mean	Worst	Std Dev
GA [39]	6059.95	6177.25	6469.32	130.9297
CPSO [44]	6061.08	6147.13	6363.8	86.4545
ESs [50]	6059.75	6850	7332.88	426
IACO [34]	6059.73	6081.78	6150.13	67.2418
CSS [13]	6059.09	6067.91	6085.48	10.2564
BA [61]	6059.71	6179.13	6318.95	137.223
WEO [12]	6059.71	6138.61	6410.19	129.9033
CDE [62]	6061.08	6085.23	6371.05	43
GA4 [56]	6059.946	6177.253	6469.322	130.9297
HPSO [63]	6059.714	6099.932	6288.677	86.2
G-QPSO [65]	6059.721	6440.379	7544.493	448.4711
MDDE [66]	6059.702	6059.702	6059.702	1.00E-12
DELC [67]	6059.7143	6059.7143	6059.7143	2.10E-11
COMDE [68]	6059.714335	6059.714335	6059.714335	3.62E-10
CVI-PSO [70]	6059.7143	6820.4101	6292.1231	2.88E+02
MVDE [71]	6059.9384	6447.3251	6182.0022	1.22E+02
BIANCA [70]	6059.9384	6447.3251	6182.0022	1.22E+02
Rank-iMDDE [72]	6059.714335	6059.714335	6059.714335	7.57E-07
CBO [31]	5889.911	5934.201	6213.006	63.5417
NM-PSO [64]	5930.314	5946.79	5960.056	9.161
Present work	5887.511073	5942.565917	6134.187981	62.2212

where $x_1$is the thickness of the shell (T_s), x_2 is the thickness of the head (T_h), x_3 is the inner radius (R) and x_4 is the length of cylindrical section of the vessel (L), not including the head.

The constraints are as follows:

$$\begin{aligned} g_1(x) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(x) &= -x_2 + 0.0193x_3 \le 0 \\ g_3(x) &= -\pi\ {x_3}^2x_4 - \frac{4}{3}\pi {x_3}^3 + 1296000 \le 0 \\ g_4(x) &= x_4 - 240 \le 0 \end{aligned} \tag{21}$$

The design space is bounded by:

$$0 \le x_1, x_2 \le 99,\ 10 \le x_3, x_4 \le 200 \tag{22}$$

Table 14 provides the optimum designs obtained by TEO and other metaheuristic algorithms. The optimal result is obtained by TEO and it seriously challenges them. Table 15 presents the statistical results yielded by TEO and other algorithms. Table 15 shows the second place of TEO in mean value among other strong methods, however CBO is dominated by ETO in standard deviation. Constraint values are checked in Table 16. As it can be seen all con-

Table 14
Optimum designs obtained by various algorithms for the pressure vessel design problem.

Methods	$T_s(=x_1)$	$T_h(=x_2)$	$R(=x_3)$	$L(=x_4)$	f_{cost}
Sandgren [52]	N/A	N/A	N/A	N/A	8129.103
Kannan and Kramer [53]	N/A	N/A	N/A	N/A	7198.042
Deb and Gene [54]	N/A	N/A	N/A	N/A	6410.381
GA [39]	0.812500	0.437500	42.097398	176.654050	6059.9463
CPSO [44]	0.812500	0.437500	42.091266	176.746500	6061.0777
ESs [50]	0.812500	0.437500	42.098087	176.640518	6059.7456
IACO [34]	0.812500	0.437500	42.098353	176.637751	6059.7258
CSS [13]	0.812500	0.437500	42.103624	176.572656	6059.0888
BA [61]	0.812500	0.437500	42.098445	176.636595	6059.7143
WEO [12]	0.812500	0.437500	42.098444	176.636622	6059.71
CDE [62]	0.812500	0.437500	42.0984	176.6376	6059.7340
GA3 [43]	0.812500	0.437500	42.0974	176.6540	6059.9463
HPSO [63]	0.812500	0.437500	42.0984	176.6366	6059.7143
G-QPSO [65]	0.812500	0.437500	42.0984	176.6372	6059.7208
MDDE [66]	0.812500	0.437500	42.098446	176.636047	6059.701660
DELC [67]	0.812500	0.437500	42.0984456	176.6365958	6059.7143
BIANCA [70]	0.812500	0.437500	42.096800	176.658000	6059.9384
Rank-iMDDE [72]	0.812500	0.437500	42.09844560	176.636595	6059.714335
NM-PSO [64]	0.803600	0.397200	41.6392	182.4120	5930.3137
CBO [31]	0.779946	0.385560	40.409065	198.76232	5889.911
Present work	0.779151	0.385296	40.369858	199.301899	5887.511073

Table 16
Constraint checking of pressure vessel design problem.

Constraints	Values
g_1	-1.2712×10^{-5}
g_2	-1.6838×10^{-4}
g_3	-0.5143
g_4	-40.6981

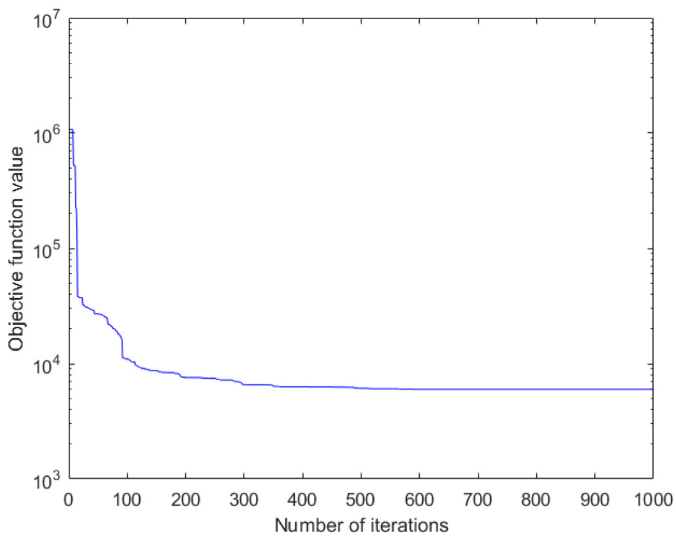

Fig. 16. Function values versus number of iterations for the pressure vessel.

straints are satisfied. The convergence curve is figured in Fig. 16, where the good converging behavior of TEO can be observed.

4.2.4. Stepped cantilever beam

Unlike the past examples, the last problem is not utilized by researchers extensively. But this is a good benchmark to verify the capability of the optimization methods for solving continuous, discrete, and mixed variable structural design problems. This was originally adopted by Thanedar and Vanderplaats [57]. The beam is designed for minimum volume. The height and width in all steps of the cantilever beam are chosen to be the design variables. These 10 variables are illustrated in Fig. 17. Except for bending stress constraints, a specified aspect ratio is imposed such that the ratio of height to width in the steps of the beam is limited to be less than 20. Some design variables are set to be integer or continuous, while the others are to be selected from discrete sets. Thus, the problem shows a characteristic of the integer, continuous and discrete behavior. The objective function is formulated as follows:

$$\text{Minimize} \quad V = 100D(b_1h_1 + b_2h_2 + b_3h_3 + b_4h_4 + b_5h_5)$$

$$\begin{aligned} g_1 &= \frac{6Pl_5}{b_5{h_5}^2} - \sigma_d \le 0 \\ g_2 &= \frac{6P(l_5+l_4)}{b_4{h_4}^2} - \sigma_d \le 0 \\ g_3 &= \frac{6P(l_5+l_4+l_3)}{b_3{h_3}^2} - \sigma_d \le 0 \\ g_4 &= \frac{6P(l_5+l_4+l_3+l_2)}{b_2{h_2}^2} - \sigma_d \le 0 \\ g_5 &= \frac{6P(l_5+l_4+l_3+l_2+l_1)}{b_1{h_1}^2} - \sigma_d \le 0 \end{aligned} \tag{23}$$

One displacement constraint on the tip deflection is to be less than the allowable deflection (Δ_{max}):

$$g_6 = \frac{pl^3}{3E}\left(\frac{1}{I_5} + \frac{1}{I_4} + \frac{1}{I_3} + \frac{1}{I_2} + \frac{1}{I_1}\right) - \Delta_{max}) \le 0 \tag{24}$$

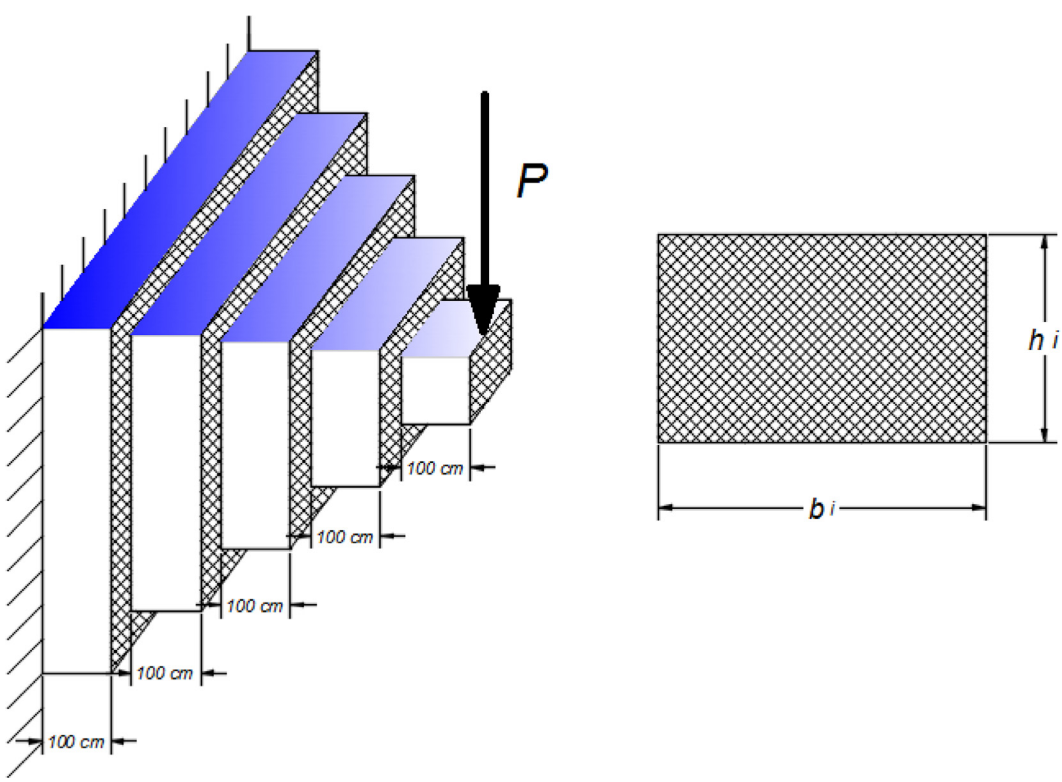

Fig. 17. Schematic of a stepped cantilever beam.

Table 17
Design variables for the stepped cantilever beam.

design variables	Type	sets & Bounds (cm)
b_1	Integer	[1,5]
h_1	Integer	[30,65]
b_2,h_2	Discrete	{2.4,2.6,2.8,3.1}
b_3,h_3	Discrete	{45.0,50.0,55.0,60.0}
b_4,b_5	Continuous	[1,5]
h_4,h_5	Continuous	[30,65]

A specific aspect ratio of 20 has to be maintained between the height and width of each of the five cross sections of the beam:

$$g_7 = \frac{h_5}{b_5} - 20 \le 0$$
$$g_8 = \frac{h_4}{b_4} - 20 \le 0$$
$$g_9 = \frac{h_3}{b_3} - 20 \le 0$$
$$g_{10} = \frac{h_2}{b_2} - 20 \le 0$$
$$g_{11} = \frac{h_1}{b_1} - 20 \le 0 \tag{25}$$

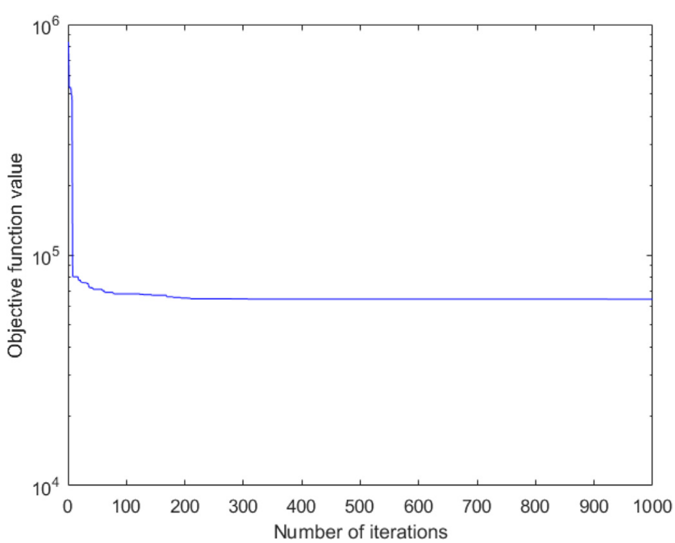

Fig. 18. Function values versus number of iterations for the stepped cantilever beam.

Table 19
Constraint checking of stepped cantilever beam design.

Constraints	Value
g_1	−33.157060
g_2	−25.554426
g_3	−153.846154
g_4	−1203.412423
g_5	−111.111111
g_6	−2.447883
g_7	−0.303792
g_8	−0.140500
g_9	−0.769231
g_{10}	−2.258065
g_{11}	0

The assigned force $p = 50{,}000$ N and the modulus of elasticity is $E = 200$ GPa. The sets of design variables are detailed in Table 17.

This problem has been solved with continuous, discrete and mixed variables in different cases in the literature [58–60]. The optimal result of the TEO is compared with several methods in Table 9. The Maximum iterations of the TEO and PSO were set to 10,000, and 30 individuals were employed.

As it is shown in Table 18, the optimal design found by Rank-iMDDE has the lower volume than those of the other compared methods and TEO is in the second place. In Table 19, the state of constraint satisfaction by optimal solution of the TEO is checked. However, TEO could offer a competitive results in less number of function evaluations as shown in Fig. 18. It can be seen from Table 20 that all g values are less than or equal to zero and the constraints are all satisfied.

5. Conclusions

This paper presents a new optimization algorithm which has used the Newton's law of cooling. Unlike most of the other meta-heuristics, that joint the location in space as agent's positions, TEO utilizes temperature dimension. The initial parameters are investigated carefully and sometimes have led to lower computational costs. The presented algorithm is conceptually simple and is relatively easy to implement. The results achieved by solving various mathematical and engineering problems show the good performance of the method in terms of global search, robustness and fast convergence. TEO can be employed as a search engine in most of the optimization problems. Also it might be a source of inspiration for the future algorithms or improved and hybridized with other methods.

Table 18
Optimal designs obtained by various algorithms for the stepped cantilever beam.

Design variables(cm)	Methods							
	Present work	PSO	BA [77]	GOAS [51]	Linear approximate discrete	Conservative approximate discrete	Continuous/ round up	Precise discrete
B_1	3	5	2.99204	3	3	3	3	3
B_2	3.1	3.1	2.77756	3.1	3.1	3.1	3.1	3.1
B_3	2.6	2.6	2.52359	2.6	2.6	2.6	2.6	2.6
B_4	2.21629046531169	2.204556	2.20455	2.27	2.279	2.3	2.276	2.262
B_5	1.76910085340763	1.749757	1.74977	1.75	1.75	1.8	1.75	1.75
H_1	60	47	59.84087	60	60	60	60	60
H_2	55	55	55.55126	55	55	55	55	55
H_3	50	50	50.4718	50	50	50	50	50
H_4	44.014420969625	44.09111	44.09106	45.25	45.553	45.5	45.528	45.233
H_5	34.8445777966116	34.995140	34.99537	35	35.004	35	34.995	34.995
Volume (cm^3)	63,994.018919	69,393.430796	61914.9	64,815	64,403	64,558	73,555	64,537

Acknowledgment

The first author is grateful to the Iran National Science Foundation for the support.

References

[1] Kaveh A. Advances in metaheuristic algorithms for optimal design of structures. second ed. Switzerland: Springer Verlag; 2017.
[2] Talbi E. Metaheuristics from design to implementation. Hoboken: John Wiley & Sons; 2009.
[3] Fogel LJ, Owens AJ, MJWalsh. Artificial intelligence through simulated evolution. Chichester: Wiley; 1966.
[4] de Jong K. Analysis of the behavior of a class of genetic adaptive systems Ph.D. Thesis. Ann Arbor, MI: University of Michigan; 1975.
[5] J.R. Koza, Genetic programming a paradigm for genetically breeding populations of computer programs to solve problems. Report No. STAN-CS-90-1314, Stanford University, Stanford, CA (1990).
[6] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press; 1975.
[7] Goldberg DE. Genetic algorithms in search optimization and machine learning. Boston: Addison-Wesley; 1989.
[8] Dorigo M, Maniezzo V, Colorni A. The ant system optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet B 1996;26:29–41.
[9] Kaveh A, Farhoudi N. A new optimization method dolphin echolocation. Adv Eng Softw 2013;59:53–70.
[10] Mirjalili SA, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61.
[11] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan; 1995.
[12] Kaveh A, Bakhshpoori T. Water evaporation optimization: a novel physically inspired optimization algorithm. Comput Struct 2016;167:69–85.
[13] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213:267–86.
[14] Kirkpatrick S, Gelatt C, Vecchi M. Optimization by simulated annealing. Science 1983;220(4598):671–80.
[15] Erol OK, Eksin I. New optimization method: big Bang–Big Crunch. Adv Eng Softw 2006;37:106–11.
[16] Hsiao Ying-Tung, Chuang Cheng-Long, Jiang Joe-Air, Chien Cheng-Chih. A novel optimization algorithm: space gravitational optimization. In: IEEE international conference on systems, man and cybernetics (SMC); October 2005. p. 2323–8.
[17] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179:2232–48.
[18] Kaveh A, Ilchi Ghazaan M. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech. 2016 First Online: 20 September. doi:10.1007/s00707-016-1725-z.
[19] Krishnanand KN, Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell 2009;3(2):87–124.
[20] Mucherino A, Seref O. Monkey search: a novel metaheuristic search for global optimization. AIP Conf Proc 2007;953(1):162–73.
[21] Yang X-S. Nature-inspired metaheuristic algorithms. Luniver Press; 2008. p. 128.
[22] Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 2012;17(12):4831–45.
[23] Askarzadeh A. Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci Numer Simul 2014;19(4):1213–28.
[24] Kaveh A, Zolghadr A. A novel meta-heuristic algorithm: tug of war optimization. Asian J Civil Eng 2016;6(4):469–92.
[25] Cuevas E. A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 2013;40(16):6374–84.
[26] Eskandar H. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 2012;110–111:151–66.
[27] Newton I. Scala graduum caloris (in Latin). Philosoph Trans R Soc London 1701;22:824–9.
[28] Newton I. A scale of the degrees of heat. Philosoph Trans R Soc London Abridged 1809;4:572–5.
[29] Winterton RHS. Heat transfer. Oxford: Oxford University Press; 1997.
[30] Winterton RHS. Newton's law of cooling. Contemp Phys 1999;40(3):205–12.
[31] Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 2014;139:18–27.
[32] Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables. Adv Eng Softw. 2014;77:66–75.
[33] Kaveh A, Ilchi Ghazaan M. Computer codes for colliding bodies optimization and its enhanced version. Int J Optim Civ Eng 2014;4:321–32.
[34] Rosenbrock HH. An automatic method for finding the greatest or least value of a function. Comput J 1960;3:175–84.
[35] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999;3:82–102.
[36] Digalakis J, Margaritis K. On benchmarking functions for genetic algorithms. Int J Comput Math 2001;77:481–506.
[37] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks; 1995. p. 1942–8.
[38] Storn R, Price K. Differential evolutionary simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11: 341–359.
[39] Deb K. Optimal design of a welded beam via genetic algorithms. AIAA J 1991;29:2013–15.
[40] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Indust 2000;41:113–27.
[41] Lee KS, Geem Z. A new structural optimization method based on the harmony search algorithm. Comput Struct 2004;82:781–98.
[42] Ragsdell KM, Phillips DT. Optimal design of a class of welded structures using geometric programming. ASME J Eng Ind Ser B. 1976;98:1021–5.
[43] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament. IEEE Trans Reliab 1992;41:576–82.
[44] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problem. Eng Appl Artif Intel 2007;20:89–99.
[45] Montes EM, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37:443–73.
[46] Renato AK, Leandro Dos Santos C. Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 2006;36:1407–16.
[47] Coello CAC. Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Environ Syst 2000;17:319–46.
[48] Arora JS. Introduction to optimum design. New York: McGraw-Hill; 1989.
[49] Lampinen J. A constraint handling approach for the differential evolution algorithm. IEEE Trans Evol Comput 2002:1468–73.
[50] Montes EM, Coello CAC. A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 2005;9(1):1–17.
[51] Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 2000;186(2–4):311–38.
[52] Sandgren E. Nonlinear integer and discrete programming in mechanical design. In: Proceedings of the ASME design technology conference; 1988. p. 95–105.
[53] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. Trans ASME J Mech Des 1994;116:318–20.
[54] Deb K, Gene AS. A robust optimal design technique for mechanical component design. In: Dasgupta D, Michalewicz Z, editors. Evolutionary algorithms in engineering applications. Berlin: Springer; 1997. p. 497–514.
[55] Belegundu AD. A study of mathematical programming methods for structural optimization PhD thesis. Iowa, USA: Department of Civil and Environmental Engineering, University of Iowa; 1982.
[56] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 2002;16:193–203.
[57] Thanedar PB, Vanderplaats GN. Survey of discrete variable optimization for structural design. J Struct Eng 1995;121:301–5.
[58] Huang MW, Arora JS. Optimal design with discrete variables some numerical experiments. Int J Numer Methods Eng 1997;40:165–88.
[59] Lamberti L, Pappalettere C. Move limits definition in structural optimization with sequential linear programming. Part II. Numer Examp Comput Struct 2003;81:215–38.
[60] Erbatur F, Hasancebi O, Tutuncil I, Kihc H. Optimal design of planar and structures with genetic algorithms. Comput Struct. 2000;75:209–24.
[61] Gandomi A. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2013;22(6):1239–55.
[62] Huang FZ, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 2007;186(1):340–56.
[63] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 2007;186:1407–22.
[64] Zahara E, Kao YT. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 2009;36:3880–6.
[65] Coelho LDS. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 2010;37:1676–83.
[66] Mezura-Montes E, Coello Coello CA, Velázquez-Reyes J, Muñoz-Dávila L. Multiple trial vectors in differential evolution for engineering design. Eng Optim 2007;39(5):567–89.
[67] Wang L, Li L-P. An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscipl Optim 2010;41:947–63.
[68] Mohamed AW, Sabry HZ. Constrained optimization based on modified differential evolution algorithm. Inf Sci 2012;194:171–208.
[69] Mazhoud KHadj-Hamou, Bigeon J, Joyeux P. Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. Eng Appl Artif Intel 2013;26(4):1263–73.
[70] Montemurro M, Vincenti A, Vannucci P. The automatic dynamic penalisation method (ADP) for handling constraints with genetic algorithms. Comput Methods Appl Mech Eng 2013;256 70–87.
[71] de Melo VV, Carosio GL. Investigating multi-view differential evolution for solving constrained engineering design problems. Expert Syst Appl 2013;40(9):3370–7.
[72] Gong Wenyin, Cai Zhihua, Liang Dingwen. Engineering optimization by means of an improved constrained differential evolution. Comput Methods Appl Mech Eng 2014;268:884–904 1 January.

[73] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 2008;178:3043–74.
[74] Wang Y, Cai Z, Zhou Y, Fan Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. Struct Multidisc Optim 2009;37:395–413.
[75] Ray T, Liew K. Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 2003;7(4):386–96.
[76] Wang Y, Cai Z, Zhou Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. Int J Numer Methods Eng 2009;77(11):1501–34.
[77] Yang X-S, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. Eng Comput 2012;29(5):464–83.