Expert Systems With Applications 148 (2020) 113246

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems

Yiying Zhang, Zhigang Jin*

School of Electrical and Information Engineering, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, PR China

ARTICLE INFO

Article history:
Received 25 May 2019
Revised 27 October 2019
Accepted 24 January 2020
Available online 27 January 2020

Keywords:
Group teaching
Swarm intelligence
Global optimization
Engineering design

ABSTRACT

In last 30 years, many metaheuristic algorithms have been developed to solve optimization problems. However, most existing metaheuristic algorithms have extra control parameters except the essential population size and stopping criterion. Considering different characteristics of different optimization problems, how to adjust these extra control parameters is a great challenge for these algorithms in solving different optimization problems. In order to address this challenge, a new metaheuristic algorithm called group teaching optimization algorithm (GTOA) is presented in this paper. The proposed GTOA is inspired by group teaching mechanism. To adapt group teaching to be suitable for using as an optimization technique, without loss of generality, four simple rules are first defined. Then a group teaching model is built under the guide of the four rules, which consists of teacher allocation phase, ability grouping phase, teacher phase and student phase. Note that GTOA needs only the essential population size and stopping criterion without extra control parameters, which has great potential to be used widely. GTOA is first examined over 28 well-known unconstrained benchmark problems and the optimization results are compared with nine state-of-the-art algorithms. Experimental results show the superior performance of the proposed GTOA for these problems in terms of solution quality, convergence speed and stability. Furthermore, GTOA is used to solve four constrained engineering design optimization problems in the real world. Simulation results demonstrate the proposed GTOA can find better solutions with faster speed compared with the reported optimizers.

1. Introduction

Optimization problems can be found in almost all engineering fields. Thus the development of optimization techniques is very essential for engineering applications, which is an interesting research direction for researchers. Most of the conventional optimization techniques require the gradient information and hence they cannot be used to solve non-differentiable functions (Rao, Savsani & Vakharia, 2012). Moreover, such techniques usually suffer from getting trapped in a local optimum in solving complex optimization problems with many local optima (Eskandar, Sadollah, Bahreininejad & Hamdi, 2012; Mirjalili, 2016). However, many real-world engineering optimization problems are very complex, whose objective functions usually have more than one local optima (Cheng & Prayogo, 2014; Zhang, Jin & Chen, 2019b). The drawbacks of conventional optimization techniques have encouraged researchers to develop better optimization methods to solve real-world engineering optimization problems.

A growing interest has been observed in metaheuristic methods over the last two decades. Here, it is worth mentioning that research worldwide in the metaheuristic field has produced optimization approaches that have proven superior to the conventional optimization approaches (Cheng & Prayogo, 2014; Rao et al., 2012). At present, metaheuristic methods have been used successfully to solve a lot of engineering optimization problems, such as multi-robot path planning (Nazarahari, Khanmirza & Doostie, 2019), unmanned aerial vehicles navigation (Kuroki, Young & Haupt, 2010), the opinion leader detection in online social network (Jain, Katarya & Sachdeva, 2020), the identification of influential users in social network (Zareie, Sheikhahmadi & Jalili, 2020); the deployment of unmanned aerial vehicles (Wang, Ru, Wang & Huang, 2019), the data collection system of Internet of Things (Huang, Wang, Wang & Yang, 2019), the localization in wireless sensor network localization (Liao, Kao & Li, 2011). Metaheuristic methods commonly operate by combining some defined rules and randomness to simulate natural phenomena (Eskandar et al., 2012; Lee & Geem, 2005). Although every metaheuristic algorithm has its own characteristics, all metaheuristic algorithms have the following three common features: nature-inspired, randomness and adjustable parameters.

* Corresponding author.
E-mail addresses: zhangyiying@tju.edu.cn (Y. Zhang), zgjin@tju.edu.cn (Z. Jin).

https://doi.org/10.1016/j.eswa.2020.113246
0957-4174/

Nomenclature

u	mean value
δ	standard deviation
t	current iteration
F	teaching factor
N	population size
D	the dimension of the problem
$\mathbf{x}_i^t$	the knowledge of the ith student at time t
$\mathbf{T}_i^t$	the knowledge of the teacher at time t
$\mathbf{M}^t$	the mean knowledge of the group at time t
$\mathbf{x}_{\text{teacher},i}^t$	the knowledge of the ith student at time t after the teacher phase
$\mathbf{x}_{\text{student},i}^t$	the knowledge of the ith student at time t after the student phase
$\mathbf{G}^t$	the optimal solution at time t
$\mathbf{x}_{\text{first}}^t$	the knowledge of the best student at time t
$\mathbf{x}_{\text{second}}^t$	the knowledge of the second best student at time t
$\mathbf{x}_{\text{third}}^t$	the knowledge of the third best student at time t
$\mathbf{X}_{\text{good}}^t$	the outstanding group
$\mathbf{X}_{\text{bad}}^t$	the average group
a, b, c, d, e	random numbers between 0 and 1
$T_{\max}$	the maximum number of function evaluations
T_{current}	the current number of function evaluations
GWO	grey wolf optimizer
PSO	particle swarm optimization
NNA	neural network algorithm
WOA	whale optimization algorithm
SCA	sine cosine algorithm
SSA	salp swarm algorithm
TLBO	teaching-learning-based optimization
GTOA	group teaching optimization algorithm

To the best of our knowledge, most research on the classification of metaheuristic algorithms focus on inspiration sources (Eskandar et al., 2012; Kaveh & Bakhshpoori, 2016; Mirjalili, 2016; Zhang, Xiao, Gao & Pan, 2018a) and no attempts to divide metaheuristic algorithms based on the adjustable parameters types have been reported in the literature. Next we give the classification method of metaheuristc algorithms according to the adjustable parameters types.

The adjustable parameters of metaheuristic algorithms generally include common parameters and special parameters. Common parameters are essential for every metaheuristic algorithm, which usually include population size and stopping criterion (e.g. the maximum number of function evaluations or the maximum number of iterations). Such parameters can be called Type I parameters. As for special parameters, they reflect the features of algorithms themselves, which can be divided into two categories as follows. Some special parameters need to be set in the initialization phase of algorithms, which can be called Type II parameters, such as crossover rate and mutation rate in differential evolution (Storn & Price, 1997) and the inertia weight in particle swarm optimization (Shi & Eberhart, 1998). Moreover, another some special parameters are usually associated with the type I parameters like the control parameter "α" in grey wolf optimizer (Mirjalili, Mirjalili & Lewis, 2014), which can be called Type III parameters. Based on the different types of parameters, metaheuristic algorithms can be divided into as follows: (1) Type I parameters-based algorithms, such as teaching-learning-based optimization (TLBO) (Rao, Savsani & Vakharia, 2011) and neural network algorithm (NNA) (Sadollah, Sayyaadi & Yadav, 2018); (2) Type I and II parameters-based algorithms, such as particle swarm optimization (PSO) (Shi & Eberhart, 1998), differential evolution (DE) (Storn & Price, 1997), harmony search (Lee & Geem, 2005), biogeography-based optimization (Simon, 2008) and cuckoo search (CS) (Yang & Deb, 2009); (3) Type I and III parameters-based algorithms, such as gray wolf optimizer (GWO) (Mirjalili et al., 2014), whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016), sine cosine algorithm (SCA) (Mirjalili, 2016) and salp swarm algorithm (SSA) (Mirjalili et al., 2017); (4) Type I, II and III parameters-based algorithms, such as water cycle algorithm (WCA) (Eskandar et al., 2012).

Although many metaheuristic algorithms have been used successfully to solve different types of optimization problems, we think that it is very necessary for researchers to achieve more simple and efficient Type I parameters-based metaheuristic algorithms due to the following three reasons.

Firstly, most existing metaheuristic algorithms are related to Type II and III parameters. Although appropriate Type II and III parameters are good for the optimization performance of these algorithms, a perfect metaheuristic algorithm should avoid Type II and III parameters considering their drawbacks. The major disadvantages of metaheuristic algorithms with Type II parameters can be summarized as follows: (1) for an unknown optimization problem, how to determine the optimal values of these parameters is a very hard task; (2) different optimization problems have different characteristics, which usually need different optimal values. For instance, the discovery probability in CS is the Type II parameter, which is employed to balance exploration and exploitation of CS. The discovery rate is set to 0.25 according to the authors of CS (Yang & Deb, 2014), which means exploration and exploitation take about 0.75 and 0.25 of the total search time, respectively. However, some variants of CS with adaptive discovery probability (Mlakar, Fister & Fister, 2016; Rakhshani & Rahati, 2017; Valian, Tavakoli, Mohanna & Haghi, 2013; Wang & Zhou, 2016) have been proven to be more effective than original CS in solving some practical problems.

Secondly, few existing metaheuristic algorithms belong to Type I parameters-based algorithms. Although TLBO and NNA are Type I parameters-based algorithms, the two algorithms have their own drawbacks. TLBO is inspired by the traditional teaching. TLBO has the fast convergence speed while TLBO is easy to fall into a local optimum in solving complex optimization problems (Chen, Zou, Li, Wang & Li, 2015; Ouyang, Gao, Kong, Zou & Li, 2015). NNA is one of the newest metaheuristic algorithms, which has strong global search ability benefiting from the unique structure of artificial neural networks while it has been proven to have a slow convergence speed (Sadollah et al., 2018).

Thirdly, No Free Lunch (NFL) theorem (Wolpert & Macready, 1997) was proposed in 1997, which has great guidance to the development of optimization algorithms. According to NFL, there is no metaheuristic best suited for solving all optimization problems. More specifically, a metaheuristic algorithm may present very promising results on a set of optimization problems while it may show poor performance on another set of optimization problems. Thus NFL plays an important role in promoting the rapid development of metaheuristic algorithms.

Considering the above reasons, a novel metaheuristic optimization algorithm called group teaching optimization algorithm (GTOA) is proposed for solving global optimization problems. The proposed method is inspired by group teaching mechanism. Group teaching is a common teaching model, which can be briefly described as follows. The students first are divided into some groups according to the defined rules. Then combining with the characteristics of every group, the teacher uses the specific teaching method to improve the group's knowledge. Like TLBO and NNA, the proposed GTOA is also a Type I parameters-based metaheuristic algorithm. Moreover, GTOA has a simple framework and is easy to implement. The main contribution of this paper can be stated as follows:

(1) A new classification method for existing metaheuristic algorithms is presented based on parameter types.
(2) A novel optimization approach called GTOA is proposed for global optimization, which is inspired by group teaching.
(3) In order to adapt group teaching to be suitable for using as an optimization technique, a group teaching model is built.
(4) 28 well-known unconstrained benchmark test functions and four constrained engineering design problems are employed to check the performance of the proposed method.

The remaining of this paper is organized as follows: In Section 2, the basic idea and the frame of the proposed GTOA are introduced. The specific implement of the proposed method is described in Section 3. The proposed method is examined using a wide set of test beds and details are discussed in Section 4. Finally, conclusions are given in Section 5.

2. The proposed GTOA

In this section, the basic idea of GTOA is first introduced. Then the framework of the proposed method is presented in detail.

2.1. Basic idea

The proposed GTOA is inspired by group teaching mechanism. In order to understand this further, a brief explanation on the basics of group teaching is presented as follows.

Confucius is a great well-known educationist and ideologist, who first started private education in China history. He first time put forward the teaching idea of “teaching students according to their aptitude”. In other words, this teaching idea is that the teacher should formulate appropriate teaching methods for different students based on their different characteristics. To better illustrate this idea, we offer an interesting example about Confucius selected from Analects of Confucius (Watson, 2007). Three students asked Confucius the same question: what is perfect virtue? However, Confucius gave them different answers based on their personality:

(1) The first student is called Hui Yan, who is a bright and eager man. Confucius told him, “To subdue one’s self and return to propriety, is perfect virtue.”
(2) The second student is called Niu Sima, who is talkativeness and impetuousness. Confucius answered, “The man of perfect virtue is cautious and slow in his speech.”
(3) The third student is called Ci Duanmu, who has great ambition but puny ability. Confucius said, “As the man of perfect virtue, wishing to be established himself, he seeks also to establish others; wishing to be enlarged himself, he seeks also to enlarge others.”

At present, the prevailing group teaching can be regarded as a specific practice of “teaching students according to their aptitude”. More specifically, group teaching is aimed at highlighting students’ subjectivity, which is to adapt the school education to the differences of students by offering various courses and teaching methods. In fact, there are a lot of differences among students, such as intelligence, learning attitude, learning ability and economic conditions. Thus although group teaching is an effect way to improve the overall quality of students, there is no uniform mode of group teaching in practice.

2.2. The framework of the proposed GTOA

The idea of the proposed GTOA is aimed at improving the knowledge of the whole class by simulating the group teaching mechanism. Considering various differences among students, it is rather complicated for group teaching to be implemented in practice. In order to adapt group teaching to be suitable for using as an optimization technique, we first assume population, decision variables and fitness value are analogous to the students, the subjects offered to students and the knowledge of students, respectively. Then a simple group teaching model without loss of generality is built based on the following four defined rules:

(1) The ability of accepting knowledge is the only difference among students. The greater the differences in the ability of accepting knowledge among students, the bigger the challenge of the teacher in the formulation of teaching plan.
(2) A good teacher tends to pay more attention to students with poor ability of accepting knowledge than students with strong ability of accepting knowledge.
(3) During the spare time, one student can gain his or her knowledge by self-learning and interaction with other students.
(4) A good teacher allocation mechanism is very helpful for improving knowledge of students.

There are four phases in the proposed group teaching model, which includes teacher allocation phase, ability grouping phase, teacher phase and student phase as shown in Fig. 1. Next the four phases are described in detailed combining with the defined four rules.

2.2.1. Ability grouping phase

Without loss of generality, the knowledge of the whole class is assumed to be in normal distribution (Rao et al., 2012). The normal distribution can be defined as

$$f(x) = \frac{1}{\sqrt{2\pi}\,\delta} e^{\frac{-(x-u)^2}{2\delta^2}} \quad (1)$$

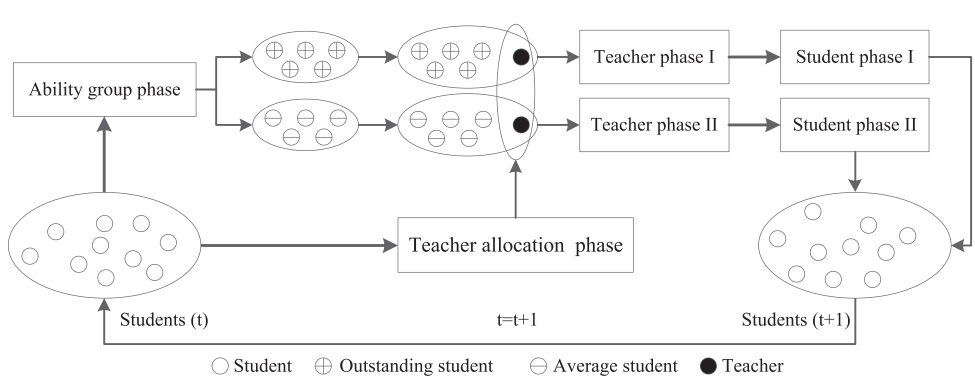

Fig. 1. The framework of the proposed GTOA (t is the current number of iterations).

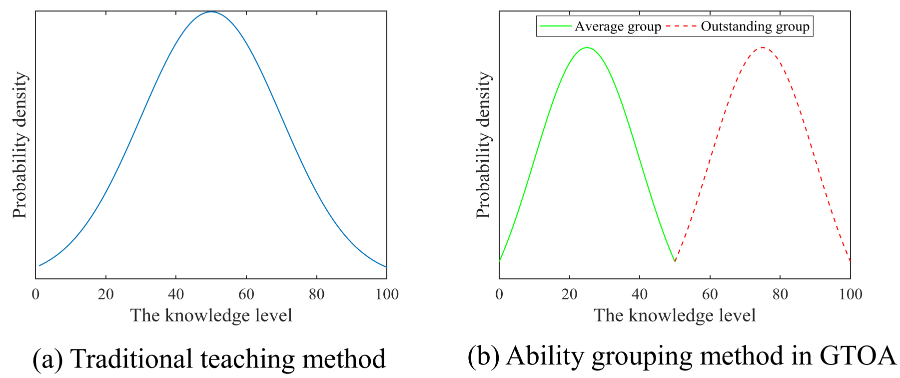

Fig. 2. The results distribution model of two different teaching methods.

Where x is the value for which the normal distribution function is required, u is the mean knowledge of the whole class and δ is the standard deviation. The standard deviation δ reflects the differences of knowledge among students. The larger the value of δ is, the greater the differences of knowledge among students are. A good teacher considers not only how to improve the mean knowledge u, but also how to reduce the standard deviation δ. In order to achieve this goal, the teacher should make a proper teaching plan for his or her students.

Fig. 2(a) presents the distribution of knowledge using traditional teaching method. Obviously, Fig. 2(a) shows a relative dispersed distribution. To better show the feature of group teaching, without loss of generality, all students are divided into two small groups according to their ability of accepting knowledge in GTOA. Note that the two groups are equally important in GTOA. Thus the two groups have the same number of students. One group with strong ability of accepting knowledge can be called outstanding group. Another group with poor ability of accepting knowledge can be called average group. Fig. 2(b) presents the distribution of knowledge based on ability grouping of the proposed GTOA. It is clear that outstanding group and average group in Fig. 2(b) are much smaller than Fig. 2(a) in terms of the standard deviation δ. Considering the first rule, it is easier for the teacher to adopt ability grouping method rather than traditional teaching method in terms of making the teaching plan. Note that the standard deviation of outstanding group and average group in Fig. 2(b) may be larger with the conduction of teaching activities. To address this issue, ability grouping is a dynamic process in GTOA, which is performed again after a learning cycle.

2.2.2. Teacher phase

Teacher phase means one student learns knowledge from his or her teacher, which corresponds to the defined second rule. The teacher makes different teaching plans for average group and outstanding group in the proposed GTOA.

$$\mathbf{x}_{\text{student},i}^{t+1} = \begin{cases} \mathbf{x}_{\text{teacher},i}^{t+1} + e \times \left(\mathbf{x}_{\text{teacher},i}^{t+1} - \mathbf{x}_{\text{teacher},j}^{t+1}\right) + g \times \left(\mathbf{x}_{\text{teacher},i}^{t+1} - \mathbf{x}_i^t\right), f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) < f\left(\mathbf{x}_{\text{teacher},j}^{t+1}\right) \\ \mathbf{x}_{\text{teacher},i}^{t+1} - e \times \left(\mathbf{x}_{\text{teacher},i}^{t+1} - \mathbf{x}_{\text{teacher},j}^{t+1}\right) + g \times \left(\mathbf{x}_{\text{teacher},i}^{t+1} - \mathbf{x}_i^t\right), f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) \geq f\left(\mathbf{x}_{\text{teacher},j}^{t+1}\right) \end{cases} \quad (7)$$

Teacher phase I: In view of the strong ability of accepting knowledge, a teacher focuses on improving the knowledge of the outstanding group as a whole in the proposed GTOA as done in TLBO. More specifically, the teacher can try his or her best to improve the mean knowledge of the whole class. In additional, the differences of accepting knowledge among students also need to be considered. Thus the student of the outstanding group can gain his or her knowledge by

$$\mathbf{x}_{\text{teacher},i}^{t+1} = \mathbf{x}_i^t + a \times \left(\mathbf{T}^t - F \times \left(b \times \mathbf{M}^t + c \times \mathbf{x}_i^t\right)\right) \quad (2)$$

$$\mathbf{M}^t = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^t \quad (3)$$

$$b + c = 1 \quad (4)$$

Where t is the current number of iterations, N is the number of students, $\mathbf{x}_i^t$ is the knowledge of student i at time t, $\mathbf{T}^t$ is the knowledge of teacher at time t, $\mathbf{M}^t$ is the mean knowledge of this group at time t, F is the teaching factor that decides the teaching results of the teacher, $\mathbf{x}_{\text{teacher},i}^{t+1}$ is the knowledge of student i at time t by learning from his or her teacher, a, b and c are there random numbers in the range [0, 1]. The value of F can be either 1 or 2 as done in Rao et al. (2012).

Teacher phase II: Considering the poor ability of accepting knowledge, a teacher pays more attention to the average group than outstanding group based on the second rule, who tends to improve the knowledge of the students from the perspective of individuals. Thus the student of the average group can gain his or her knowledge by

$$\mathbf{x}_{\text{teacher},i}^{t+1} = \mathbf{x}_i^t + 2 \times d \times \left(\mathbf{T}^t - \mathbf{x}_i^t\right) \quad (5)$$

Where d is a random number in the range [0, 1].

In addition, one student may not gain knowledge by the teacher phase, which can be addressed by (take the minimum problem as an example)

$$\mathbf{x}_{\text{teacher},i}^{t+1} = \begin{cases} \mathbf{x}_{\text{teacher},i}^{t+1}, f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) < f\left(\mathbf{x}_i^t\right) \\ \mathbf{x}_i^t, f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) \geq f\left(\mathbf{x}_i^t\right) \end{cases} \quad (6)$$

2.2.3. Student phase

The student phase including the student phase I and the student phase II corresponds to the mentioned third rule. During spare time, one student can gain his or her knowledge by two different ways: one through self-learning and the other through interaction with other students, which can be expressed as

Where e and g are two random numbers in the range [0, 1], $\mathbf{x}_{\text{student},i}^{t+1}$ is the knowledge of student i at time t by learning from the student phase and $\mathbf{x}_{\text{teacher},j}^{t+1}$ is the knowledge of student j at time t by learning from the teacher. As for the student j ($j \in$ {1, 2, ..., $i-1$, $i+1$, ..., N}), he or she is randomly selected. In Eq. (7), the second item and the third item on the right mean learning from the other student and self-learning, respectively.

In addition, one student may not gain knowledge by the student phase, which can be addressed by (take the minimum problem as an example)

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_{\text{teacher},i}^{t+1}, & f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) < f\left(\mathbf{x}_{\text{student},i}^{t+1}\right) \\ \mathbf{x}_{\text{student},i}^{t+1}, & f\left(\mathbf{x}_{\text{teacher},i}^{t+1}\right) \geq f\left(\mathbf{x}_{\text{student},i}^{t+1}\right) \end{cases} \tag{8}$$

Where $\mathbf{x}_i^{t+1}$ is the knowledge of student i at time $t+1$ after a learning cycle.

2.2.4. Teacher allocation phase

Based on the defined fourth rule, how to make a good teacher allocation mechanism is very important for improving the knowledge of students. In GWO, the first three best solutions obtained so far are saved, which are used to guide the hunting of wolves. Inspired by the hunting behavior in GWO, the teacher allocation in proposed method can be expressed as

$$\mathbf{T}^t = \begin{cases} \mathbf{x}_{\text{first}}^t, & f\left(\mathbf{x}_{\text{first}}^t\right) \leq f\left(\frac{\mathbf{x}_{\text{first}}^t + \mathbf{x}_{\text{second}}^t + \mathbf{x}_{\text{third}}^t}{3}\right) \\ \frac{\mathbf{x}_{\text{first}}^t + \mathbf{x}_{\text{second}}^t + \mathbf{x}_{\text{third}}^t}{3}, & f\left(\mathbf{x}_{\text{first}}^t\right) > f\left(\frac{\mathbf{x}_{\text{first}}^t + \mathbf{x}_{\text{second}}^t + \mathbf{x}_{\text{third}}^t}{3}\right) \end{cases} \tag{9}$$

Where $\mathbf{x}_{\text{first}}^t$, $\mathbf{x}_{\text{second}}^t$ and $\mathbf{x}_{\text{third}}^t$ are the first, second and third best students, respectively. In order to accelerate the convergence of the proposed GTOA, outstanding group and average group share the same teacher.

2.3. Comparison between TLBO and GTOA

Like GTOA, TLBO is also inspired from the teaching phenomenon in the classroom. A fundamental difference between TLBO and GTOA is that TLBO and GTOA imitate traditional teaching and group teaching, respectively. More specifically, their differences can be summarized as follows:

(1) In the teacher phase, GTOA considers the differences of accepting knowledge among students to make two different teaching methods as shown in Eqs. (2) and (5). However, TLBO uses the same teaching method for all students, which neglects the differences of accepting knowledge among students.
(2) In the student phase, GTOA uses self-learning and interaction with other students to gain knowledge while TLBO only concerns the interaction with other students.
(3) The ability grouping phase is introduced to GTOA, which is the distinct feature of the proposed GTOA. However, TLBO has not this phase.
(4) The best student is regarded as teacher in TLBO while GTOA defines a teacher allocation mechanism related to the first three best students.

3. Implementation of the proposed GTOA for optimization

In the following, the step-wise procedure for the implementation of GTOA is given and GTOA is explained with the aid of the flowchart in Fig. 3.

Step 1: Initialization information

(1.1) Initialization parameters.

These parameters include the maximum number of function evaluations $T_{\max}$, the current number of function evaluations T_{current} (T_{current}=0), population size N, the lower bounds of design variables $\mathbf{l}$, the upper bounds of design variables $\mathbf{u}$, dimension of problem D and fitness function $f(\bullet)$.

(1.2) Initialization population

A random population $\mathbf{X}^t$ is generated on the basis of the initialization parameters, which can be described as

$$\mathbf{X}^t = \left[\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_N^t\right]^{\mathrm{T}} = \begin{bmatrix} x_{1,1}^t & x_{1,2}^t & \cdots & x_{1,D}^t \\ x_{2,1}^t & x_{2,2}^t & \cdots & x_{2,D}^t \\ \vdots & \vdots & & \vdots \\ x_{N,1}^t & x_{N,2}^t & \cdots & x_{N,D}^t \end{bmatrix} \tag{10}$$

$$x_{i,j}^t = l_i + (u_i - l_i) \times \kappa \tag{11}$$

Where κ is a random number in the range [0, 1].

Step 2: Population evaluation.

The fitness values of individuals are calculated and the optimal solution $\mathbf{G}^t$ is selected. The current number of function evaluations T_{current} is updated by

$$T_{\text{current}} = T_{\text{current}} + N \tag{12}$$

Step 3: Termination criteria.

If the current number of function evaluations T_{current} is greater than the maximum number of function evaluations $T_{\max}$, the algorithm stops and the optimal solution $\mathbf{G}^t$ is outputted. Otherwise, go to Step 4.

Step 4: Teacher allocation phase.

The first three best individuals are selected. Then the teacher $\mathbf{T}^t$ is calculated by Eq. (9).

Step 5: Ability grouping phase.

The population is divided into two groups based on the fitness values. The best half individuals form the outstanding group and the rest individuals make up the average group. The two groups share the same teacher. The outstanding group and the average group are marked as$\mathbf{X}_{\text{good}}^t$ and $\mathbf{X}_{\text{bad}}^t$, respectively.

Step 6: Teacher phase and student phase.

(6.1) For the group $\mathbf{X}_{\text{good}}^t$, the teacher phase is implemented based on Eqs. (2), (3), (4) and (6). Then the student phase is conducted according to Eqs. (7) and (8). Finally, the new population $\mathbf{X}_{\text{good}}^{t+1}$ is obtained.

(6.2) For the group $\mathbf{X}_{\text{bad}}^t$, the teacher phase is implemented based on Eqs. (5) and (6). Then the student phase is conducted according to Eqs. (7) and (8). Finally, the new population $\mathbf{X}_{\text{bad}}^{t+1}$ is obtained.

Step 7: Construct population.

The population $\mathbf{X}_{\text{good}}^{t+1}$ and the population $\mathbf{X}_{\text{bad}}^{t+1}$ compose a new population $\mathbf{X}^{t+1}$.

Step 8: Population evaluation.

The fitness values of individuals are calculated and the optimal individual $\mathbf{G}^t$ is selected.

The current number of function evaluations T_{current} is updated by

$$T_{\text{current}} = T_{\text{current}} + 2N + 1 \tag{13}$$

Then Step 3 is executed.

4. Experimental studies

This section is to prove the validation of the proposed GTOA in solving global optimization problems, which is divided into two parts. 28 complex unconstrained benchmark problems are tested in Section 4.1, with the results compared against nine other meta-heuristic algorithms. Four constrained engineering design problems are examined in Section 4.2 and the obtained results are compared with other reported solutions.

4.1. GTOA for solving unconstrained benchmark functions

4.1.1. Benchmark functions

In this section, 28 benchmark functions are used to evaluate the proposed GTOA. These functions are classical functions, which are

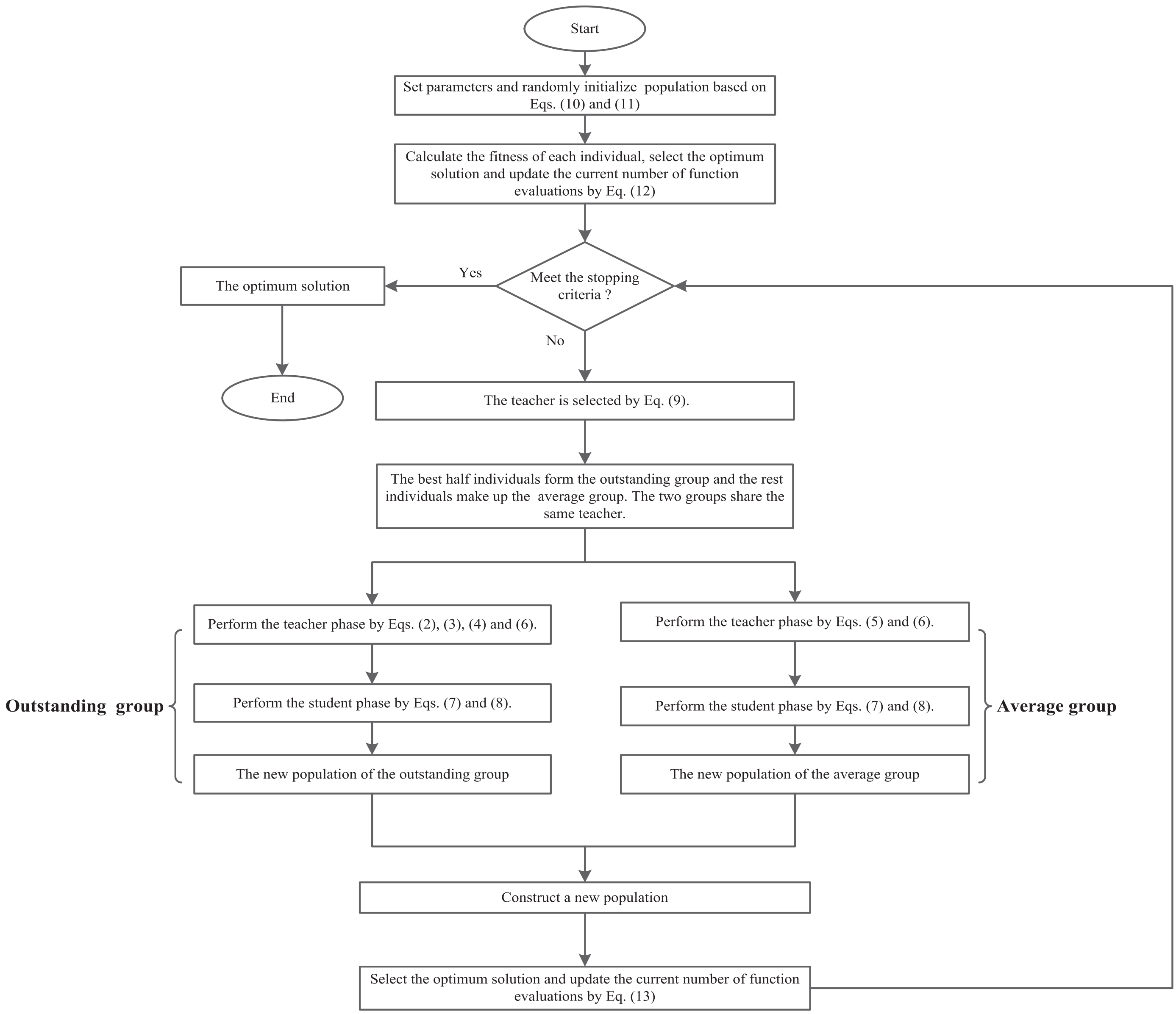

Fig. 3. Flowchart for the proposed GTOA.

often employed to check the optimization performance of different algorithms (Ewees, Abd Elaziz & Houssein, 2018; Long, Wu, Liang & Xu, 2019; Lu, Gao & Yi, 2018; Mirjalili, 2016; Pence, Cesmeli, Senel & Cetisli, 2016; Rakhshani & Rahati, 2017; Sun, Wang, Chen and Liu, 2018b; Yao, Liu & Lin, 1999). All test functions should be minimized and have been listed in Tables 1–3 where *D* indicates dimension of the function, *Range* is the boundary of the function's search space and *Optimal* is the global optimum.

More specifically, the benchmark functions used can be divided into four groups: unimodal functions (F_1- F_8), general multimodal functions (F_9- F_{20}), rotated multimodal functions (F_{21}- F_{24}) and fixed-dimension multimodal functions (F_{25}- F_{28}). Generally, multimodal functions have more than one local minimum, which are more complicated compared with unimodal functions. As shown in Tables 1–3, 20 of 28 benchmark functions are multimodal functions, which provide a better examination of global optimization ability of the proposed method.

4.1.2. Parameters settings

The applied algorithms: In order to better show the optimization performance of the proposed GTOA in solving the unconstrained

Table 1
Unimodal benchmark functions.

No.	Formulation	*D*	Range	Optimal
F_1	$f(x)=\sum_{i=1}^{D} {x_i}^2$	30	[−100,100]	0
F_2	$f(x)=\sum_{i=1}^{D} i\times x_i^2$	30	[−10,10]	0
F_3	$f(x)=\sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	30	[−100,100]	0
F_4	$f(x)=\sum_{i=1}^{D} i\times x_i^4$	30	[−1.28,1.28]	0
F_5	$f(x)=\max_i\{\lvert x_i\rvert, 1\le i\le D\}$	30	[−100,100]	0
F_6	$f(x)=\sum_{i=1}^{D} (\lfloor x_i+0.5\rfloor)^2$	30	[−100,100]	0
F_7	$f(x)=\sum_{i=1}^{D} i\times x_i^4+random[0,1)$	30	[−1.28,0.64]	0
F_8	$f(x)=\sum_{i=1}^{D} \lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	30	[−10,10]	0

Table 2
General multimodal benchmark functions.

No.	Formulation	D	Range	Optimal
F_9	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$	30	[−100,100]	0
F_{10}	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}{x_i}^2-\prod_{i=1}^{D}\cos\frac{x_i}{\sqrt{i}}+1$	30	[−600,600]	0
F_{11}	$f(x)=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	30	[−10,10]	0
F_{12}	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}{x_i}^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i)+20+e$	30	[−32,32]	0
F_{13}	$f(x)=\sum_{i=1}^{D}{x_i}^2+(\sum_{i=1}^{D}0.5\times i\times x_i)^2+(\sum_{i=1}^{D}0.5\times i\times x_i)^4$	30	[−10,10]	0
F_{14}	$f(x)=0.5+\frac{\sin^2(\sqrt{\sum_{i=1}^{D}x_i^2})-0.5}{(1+0.001(\sum_{i=1}^{D}x_i^2))^2}$	30	[−100,100]	0
F_{15}	$f(x)=x_1^2+10^6\sum_{2}^{D}x_i^6$	30	[−100,100]	0
F_{16}	$f(x)=10^6.x_1^2+\sum_{2}^{D}x_i^6$	30	[−1,1]	0
F_{17}	$f(x)=\sum_{i=1}^{D-1}[x_i^2+2x_{i+1}^2-0.3\cos(3\pi x_i)-0.4\cos(4\pi x_{i+1})+0.7]$	30	[−15,15]	0
F_{18}	$f(x)=\sum_{i=2}^{D}\lvert x_i\rvert^{i+1}$	30	[−1,1]	0
F_{19}	$f(x)=0.1D+0.1\sum_{i=2}^{D}\cos(5\pi x_i)+\sum_{i=1}^{D}x_i^2$	30	[−1,1]	0
F_{20}	$f(x)=1-\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1\sqrt{\sum_{i=1}^{D}x_i^2}$	30	[−100,100]	0

Table 3
Several special multimodal benchmark functions (RM: Rotated multimodal, FM: Fix-dimension multimodal).

No.	Formulation	D	Type	Range	Optimal
F_{21}	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}{y_i}^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi y_i)+20+e, y_i=M*x_i$	30	RM	[−32,32]	0
F_{22}	$f(x)=10D+\sum_{i=1}^{D}[y_i^2-10\cos(2\pi y_i)], y_i=M*x_i$	30	RM	[−5.12,5.12]	0
F_{23}	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}{y_i}^2-\prod_{i=1}^{D}\cos\frac{y_i}{\sqrt{i}}+1, y_i=M*x_i$	30	RM	[−600,600]	0
F_{24}	$f(x)=10D+\sum_{i=1}^{D}[y_i^2-10\cos(2\pi y_i)], y_i=\begin{cases}M*x_i, \lvert x_i\rvert<0.5\\ M*\frac{\mathrm{round}(2x_i)}{2}, \lvert x_i\rvert\geq 0.5\end{cases}$	30	RM	[−5.12,5.12]	0
F_{25}	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	FM	[−5,5]	−1.0316
F_{26}	$f(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})cos(x_1)+10$	2	FM	[−5,5]	0.398
F_{27}	$f(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	FM	[1,3]	−3.86
F_{28}	$f(x)=-\sum_{i=1}^{5}[(x-a_i)(x-a_i)^{\mathrm{T}}+c_i]^{-1}$	4	FM	[0,10]	−10.1532

benchmark problems, nine metaheuristic algorithms are selected to compare with GTOA, which include two Type I parameters-based metaheuristic algorithms (NNA and TLBO), three Type I and II parameters-based metaheuristic algorithms (DE, PSO and CS) and four Type I and III parameters-based metaheuristic algorithms (GWO, SCA, SSA and WOA). Note that how to make a fair comparison for different metaheuristic algorithms is a very difficult task due to the different characteristics of algorithms. For having fair comparison, like many other related studies (Ibrahim, Elaziz & Lu, 2018; Mirjalili, 2016; Mirjalili et al., 2014; Rakhshani & Rahati, 2017; Zhang, Kang, Cheng & Wang, 2018b), the parameters of the compared algorithms are taken from the original literature and have been shown in Table 4. In addition, the Type III parameters of GWO, SSA, SCA and WOA are related to the maximum number of function evaluations, which are not listed in Table 4 and can be found in the corresponding references.

The independent trials: Considering the results of a single run might be unreliable due to the stochastic nature of metaheuristics, all algorithms used are executed in 30 independent runs for the same test function. Then the average values of the obtained results are recorded.

Stopping criteria: The maximum number of function evaluations is often considered as the stopping criteria for metaheuristic algorithms. How to determine the maximum number of function evaluations all depends on the specific algorithms and the problems to be solved, which is not unity. In our experiments, it is 5000 multiples by dimension size for unimodal functions, multimodal functions and rotated multimodal functions (Sadollah et al., 2018; Zhang, Huang & Zhang, 2019a) and the maximum number of function evaluations is 50,000 for fixed-dimensional multimodal functions, which is sufficient for most test problems.

4.1.3. Evaluation indictors

Three quality indicators are used to compare the optimization performance among different algorithms, which can be described as follows.

Table 4
Parameter settings of the applied algorithms.

Algorithm	Parameter	Reference
GWO	Population size=30.	(Mirjalili et al., 2014)
NNA	Population size=50.	(Sadollah et al., 2018)
WOA	Population size=30.	(Mirjalili & Lewis, 2016)
SCA	Population size=30.	(Mirjalili, 2016)
SSA	Population size=30.	(Mirjalili et al., 2017)
TLBO	Population size=50.	(Rao et al., 2011)
GTOA	Population size=50.	–
CS	Population size=20, discover probability=0.25.	(Yang & Deb, 2009)
DE	Population size=100, differential amplification factor=0.5, crossover probability constant=0.9.	(Rahnamayan, Tizhoosh & Salama, 2008)
PSO	Population size=40, personal learning coefficient=1.4986, social learning coefficient=1.4986.	(Wang, Wu, Rahnamayan, Liu & Ventresca, 2011)

Table 5
The experimental results for 28 benchmark functions obtained by NNA, TLBO and GTOA.

No.	NNA			TLBO			GTOA		
	Mean	Std	TR	Mean	Std	TR	Mean	Std	TR
F_1	2.05E-21	5.22E-21	3	6.46E-268	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_2	3.35E-22	7.42E-22	3	7.93E-269	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_3	4.01E-15	1.06E-14	3	2.36E-264	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_4	7.04E-46	2.32E-45	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_5	4.03E-03	2.79E-03	3	6.39E-108	8.26E-108	2	**2.97E-287**	**0.00E+00**	1
F_6	**0.00E+00**	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	2
F_7	2.50E-03	1.30E-03	3	2.93E-04	1.25E-04	2	**8.16E-05**	**5.17E-05**	1
F_8	1.05E-11	1.44E-11	3	1.01E-133	1.38E-133	2	**0.00E+00**	**0.00E+00**	1
F_9	1.41E-04	8.82E-05	3	1.03E-57	2.75E-57	2	**0.00E+00**	**0.00E+00**	1
F_{10}	7.30E-03	1.54E-02	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_{11}	5.26E-06	9.34E-06	3	1.20E-268	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_{12}	3.19E-11	2.67E-11	3	5.98E-15	1.79E-15	2	**4.44E-15**	**0.00E+00**	1
F_{13}	1.06E-09	1.20E-09	3	3.57E-36	5.55E-36	2	**2.37E-226**	**0.00E+00**	1
F_{14}	3.95E-03	1.39E-03	3	3.13E-03	8.98E-10	2	**3.13E-03**	**1.05E-12**	1
F_{15}	2.50E-21	1.02E-20	3	5.05E-268	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_{16}	1.62E-24	2.98E-24	3	2.42E-270	**0.00E+00**	2	**0.00E+00**	**0.00E+00**	1
F_{17}	1.48E-17	5.63E-17	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_{18}	2.19E-39	8.13E-39	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_{19}	1.63E-16	2.47E-16	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_{20}	2.07E-01	5.21E-02	3	9.99E-02	1.16E-09	2	**9.98E-02**	**4.46E-13**	1
F_{21}	1.59E+00	1.14E+00	3	6.10E-15	1.80E-15	2	**4.44E-15**	**0.00E+00**	1
F_{22}	1.16E+02	3.74E+01	3	**1.17E+01**	**5.98E+00**	1	1.42E+01	1.80E+01	2
F_{23}	1.48E-02	1.76E-02	3	**0.00E+00**	**0.00E+00**	1.5	**0.00E+00**	**0.00E+00**	1.5
F_{24}	1.27E+02	3.67E+01	3	**2.34E+01**	**1.23E+01**	1	3.06E+01	2.00E+01	2
F_{25}	**−1.03E+00**	**4.86E-16**	2	**−1.03E+00**	6.78E-16	2	**−1.03E+00**	6.78E-16	2
F_{26}	**3.98E-01**	**0.00E+00**	2	**3.98E-01**	**0.00E+00**	2	**3.98E-01**	**0.00E+00**	2
F_{27}	**−3.86E+00**	**2.25E-15**	2	**−3.86E+00**	2.71E-15	2	**−3.86E+00**	2.71E-15	2
F_{28}	−6.03E+00	2.42E+00	3	**−1.02E+01**	**2.93E-14**	1.5	**−1.02E+01**	7.54E-09	1.5

The first one is the value-based method. Mean value and standard deviation are good indicators to measure the obtained solution quality. The smaller the mean value is, the stronger the global optimization ability of the algorithm is; the smaller the standard deviation is, the more stability the algorithm is. Tables 5 and 7 show the statistical results. In these tables, “Mean”, “Std” and “TR” indicate mean value, stand deviation and tied rank, respectively. Moreover, the best results are highlighted in bold.

The second one is the rank-based method. Tied rank (TR) (Rakhshani & Rahati, 2017; Sun et al., 2018b) is employed to intuitively compare the performance of the considered algorithms. More specifically, TR assigns rank 1 to the algorithm with the best mean value; rank 2 to the second best and rank M (the number of the considered algorithms) to the Mth best. If some algorithms have same results, they will share average of ranks. Tables 5 and 7 show the results of tied rank.

The third one is the statistical test-based method. Wilcoxon signed-rank test (Derrac, García, Molina & Herrera, 2011) is used to compare the performance between the proposed GTOA and other algorithms, which has been widely used to compare the optimization performance between two algorithms (Mafarja et al., 2018; Martínez-Peñaloza & Mezura-Montes (2018); Sun, Ma, Ren, Zhang & Jia, 2018a; Yi, Gao, Li, Shoemaker & Lu, 2019). In this research, the results of Wilcoxon signed-rank test with a significance level α=0.05 are produced by using the optimal solutions of 30 runs of obtained by GTOA and the compared algorithms. Tables 6 and 8 present the test results. In these tables, “H” is marked as “1”, which means there is significant difference between GTOA and the compared algorithm; “H” is marked as “0”, which indicates there is no significant difference between GTOA and the compared algorithm. “S” is marked as “+”, which imply the proposed method is superior to the compared algorithm; “S” is marked as “=”, which indicates the proposed method has the same performance with the compared algorithm. Moreover, the last column of each of these tables under the heading w/l/t represents the win, lose and tie counts of the proposed GTOA over the compared algorithms.

4.1.4. Comparison of GTOA with TLBO and NNA

Like TLBO and NNA, the proposed GTOA is also a Type I parameters-based metaheuristic algorithm. In this section, we compare the performance of the three algorithms in solving the unconstrained benchmark functions.

The statistical results obtained by NNA, TLBO and GTOA have been shown in Table 5. From Table 5, GTOA is superior to NNA on all test functions except four functions (i.e. F_6, F_{25}, F_{26} and F_{27}). Moreover, GTOA can offer the same solutions with NNA on F_6, F_{25},

Table 6
Wilcoxon signed-rank test results among GTOA, NNA and TLBO.

No.	GTOA vs. NNA			GTOA vs. TLBO		
	p-value	H	S	p-value	H	S
F_1	1.73E-06	1	+	1.73E-06	1	+
F_2	1.73E-06	1	+	1.73E-06	1	+
F_3	1.73E-06	1	+	1.73E-06	1	+
F_4	1.73E-06	1	+	1.73E-06	1	+
F_5	1.73E-06	1	+	1.73E-06	1	+
F_6	1.00E+00	0	=	1.00E+00	0	=
F_7	1.73E-06	1	+	1.73E-06	1	+
F_8	1.73E-06	1	+	1.73E-06	1	+
F_9	1.73E-06	1	+	1.73E-06	1	+
F_{10}	3.96E-05	1	+	1.95E-03	1	+
F_{11}	1.73E-06	1	+	1.73E-06	1	+
F_{12}	1.73E-06	1	+	1.68E-06	1	+
F_{13}	1.73E-06	1	+	1.73E-06	1	+
F_{14}	1.73E-06	1	+	1.73E-06	1	+
F_{15}	1.73E-06	1	+	1.73E-06	1	+
F_{16}	1.73E-06	1	+	1.73E-06	1	+
F_{17}	2.91E-04	1	+	1.71E-06	1	+
F_{18}	1.73E-06	1	+	1.73E-06	1	+
F_{19}	1.00E+00	0	=	6.10E-05	1	+
F_{20}	1.73E-06	1	+	1.73E-06	1	+
F_{21}	1.73E-06	1	+	1.73E-06	1	+
F_{22}	1.73E-06	1	+	1.73E-06	1	+
F_{23}	1.73E-06	1	+	1.73E-06	1	+
F_{24}	1.73E-06	1	+	2.13E-06	1	+
F_{25}	1.00E+00	0	=	1.00E+00	0	=
F_{26}	1.00E+00	0	=	1.00E+00	0	=
F_{27}	1.00E+00	0	=	1.00E+00	0	=
F_{28}	5.00E-01	0	=	5.00E-01	0	=
w/l/t			22/0/6		23/0/5	

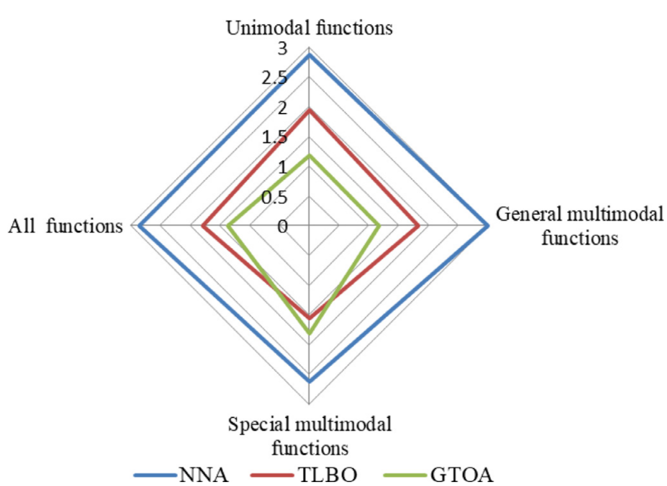

Fig. 4. Radar chart of the overall ranking obtained by NNA, TLBO and GTOA.

F_{26} and F_{27}. TLBO shows strong competitiveness, which can give the same solutions with GTOA on 10 functions (i.e. F_4, F_6, F_{10}, F_{17}, F_{18}, F_{19}, F_{23}, F_{25}, F_{26} and F_{27}) and surpass GTOA on three functions (i.e. F_{22}, F_{24} and F_{28}). However, GTOA still outperforms TLBO on more than half functions (i.e. F_1, F_2, F_3, F_5, F_7, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{20} and F_{21}). In addition, GTOA can get the global optimum solutions on 15 functions(i.e. F_1, F_2, F_3, F_4, F_6, F_8, F_9, F_{10}, F_{11}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{23}, F_{25}, F_{26} and F_{27}) and achieve quite approximate the global optimum solutions on two functions (i.e. F_5 and F_{13}). Besides, Fig. 4 gives radar chart of the overall ranking obtained by NNA, TLBO and GTOA on the basis of mean values of TR for different types of functions. It is clear that GTOA is superior to NNA on all types of functions. Although TLBO has a slight advantage over GTOA on special multimodal functions, TLBO is inferior to GTOA on unimodal functions, general multimodal functions and all test functions.

Table 6 presents the Wilcoxon signed-rank test results with a significance level α=0.05 among NNA, TLBO and GTOA for 28 benchmark functions. From Table 5, GTOA outperforms NNA on all test functions except F_6, F_{19}, F_{25}, F_{26}, F_{27} and F_{28}. GTOA is superior to TLBO on all functions except F_6, F_{25}, F_{26}, F_{27} and F_{28}. NNA and TLBO have no significance difference with GTOA on six (i.e. F_6, F_{19}, F_{25}, F_{26}, F_{27} and F_{28}) and five (i.e. F_6, F_{25}, F_{26}, F_{27} and F_{28}) functions, respectively. Obviously, GTOA can offer better solutions than NNA and TLBO on most test functions.

In order to compare the convergence performance of GTOA, NNA and TLBO, Fig. 5 shows several typical convergence curves. As can be seen from Fig. 5, NNA has a slow convergence speed on these selected functions compared with TLBO and GTOA. Moreover, NNA cannot offer better solutions than TLBO and GTOA on these selected functions. In terms of convergence speed and global search ability, TLBO outperforms NNA while TLBO is inferior to GTOA on these functions. In general, GTOA shows faster convergence speed and stronger global search ability than NNA and TLBO on these functions.

4.1.5. Comparison of GTOA with seven state-of-the-art algorithms

In this section, we compare the performance between GTOA and seven state-of-the-art algorithms (i.e. DE, CS, SCA, GWO, WOA, PSO and SSA).

The statistical results obtained by eight algorithms have been shown in Table 7. GTOA is superior to DE on all test functions except six functions (i.e. F_6, F_{19}, F_{25}, F_{26}, F_{27} and F_{28}). GTOA can surpass CS on all test functions except five functions (i.e. F_6, F_{25}, F_{26}, F_{27} and F_{28}). GWO shows strong global search ability, which can offer the same solutions with GTOA on 14 functions (i.e. F_1, F_2, F_4, F_6, F_{10}, F_{11}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{25}, F_{26} and F_{27}). Note that GWO is inferior to GTOA on 12 functions (i.e. F_3, F_5, F_7, F_8, F_9, F_{12}, F_{13}, F_{14}, F_{20}, F_{21}, F_{23} and F_{28}) while GWO only outperforms GTOA on two functions (i.e. F_{22} and F_{24}). In addition, GTOA can offer better solutions than SCA on all test functions except five functions (i.e. F_6, F_{17}, F_{19}, F_{25} and F_{26}). SSA and PSO are inferior to GTOA on all test functions except F_{25}, F_{26} and F_{27}. WOA has strong competitiveness, which can achieve the same solutions with GTOA on 15 functions (i.e. F_1, F_2, F_3, F_4, F_6, F_8, F_{11}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{25}, F_{26} and F_{27}) and can outperform GTOA on three functions (F_{12}, F_{14} and F_{21}).. However, GTOA still can get better results than WOA on 10 functions (i.e. F_5, F_7, F_9, F_{10}, F_{13}, F_{14}, F_{20}, F_{22}, F_{23}, F_{24} and F_{28}). Besides, Fig. 6 gives radar chart of the overall ranking obtained by eight different algorithms on the basis of mean values of TR for different types of functions. Obviously, GTOA can outperform DE, CS, GWO, SCA, WOA, SSA and PSO for all types of functions.

Wilcoxon signed-rank test results between GTOA and the compared algorithms are presented in Table 7. From Table 7, GTOA is superior to SSA on all test functions. GTOA can outperform DE on all test functions except six functions (i.e. F_6, F_{19}, F_{25}, F_{26}, F_{27} and F_{28}). CS is inferior to GTOA on all test functions except five functions (i.e. F_6, F_{25}, F_{26}, F_{27} and F_{28}). In addition, GTOA can offer better solutions than SCA on all test functions except four functions (i.e. F_6, F_7, F_{19} and F_{23}). GTOA can surpass PSO on all test functions except F_{25}, F_{26} and F_{27}. Besides, GWO and WOA have no significance difference on nine (i.e. F_4, F_6, F_{10}, F_{17}, F_{18}, F_{19}, F_{22}, F_{23} and F_{24}) and 15 functions (i.e. F_1, F_2, F_3, F_4, F_6, F_8, F_{10}, F_{11}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{20} and F_{23}), respectively. However, GTOA can achieve better solutions than GWO and WOA on 19 (i.e. F_1, F_2, F_3, F_5, F_7, F_8, F_9, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{20}, F_{21}, F_{25}, F_{26}, F_{27} and F_{28}) and 13 (i.e. F_5, F_7, F_9, F_{12}, F_{13}, F_{14}, F_{21}, F_{22}, F_{24}, F_{25}, F_{26}, F_{27} and F_{28}) functions.

In order to compare the convergence performance among the eight algorithms, Fig. 7 gives several typical convergence curves. As can be seen from Fig. 7, GTOA shows faster convergence speed than the compared algorithms on the selected functions. Moreover,

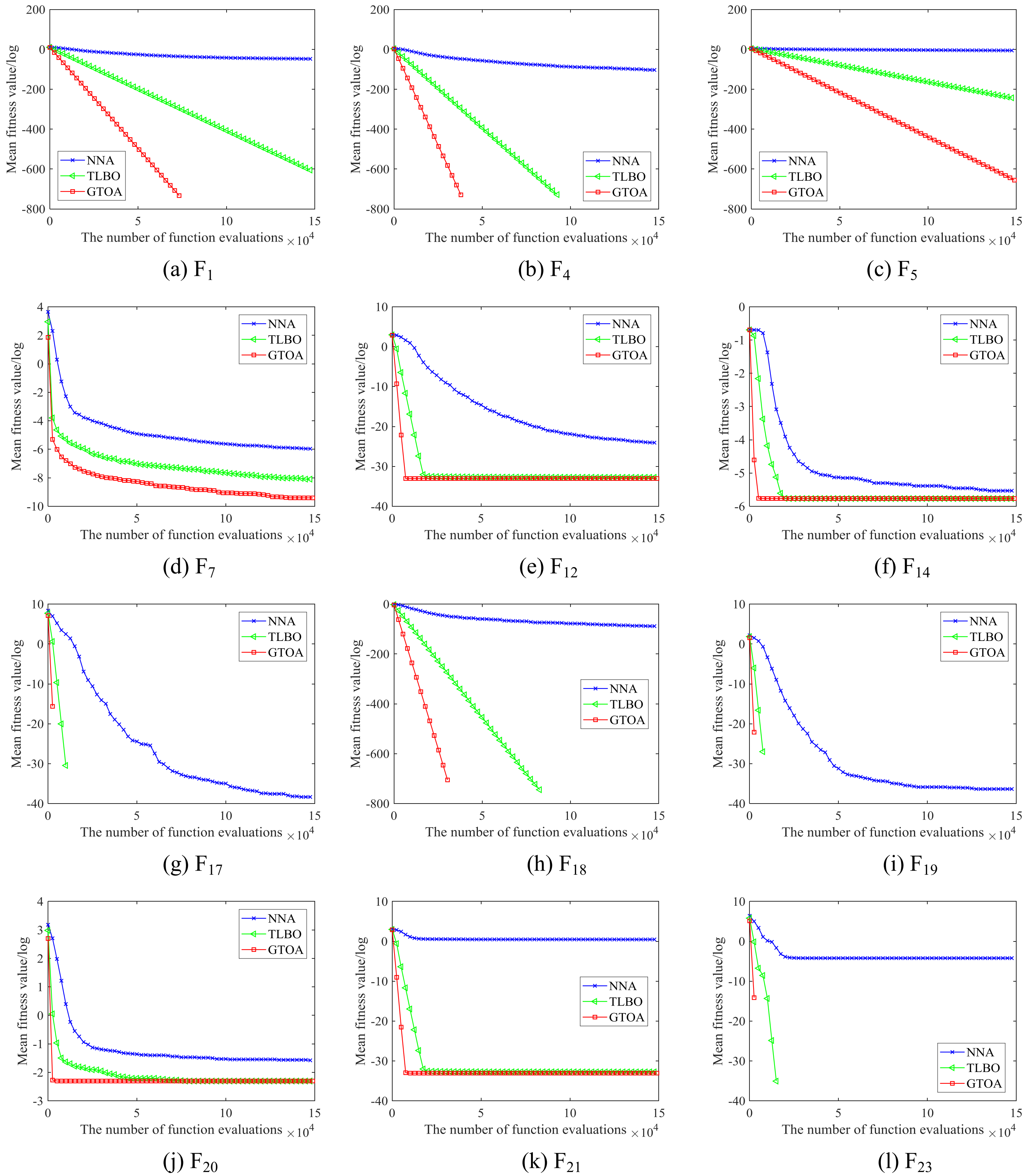

Fig. 5. Several typical convergence curves obtained by NNA, TLBO and GTOA.

GTOA can offer better solutions than CS, PSO and SSA on these functions. DE and SCA can get same solutions with GTOA on F_{17} and F_{19} while GTOA can achieve better solutions than DE and SCA on F_1, F_4, F_5, F_7, F_{10}, F_{11}, F_{13}, F_{15}, F_{20} and F_{23}. Besides, GWO and WOA can offer the same solutions with GTOA on seven (i.e. F_1, F_4, F_{10}, F_{11}, F_{15}, F_{17} and F_{19}) and six (i.e. F_1, F_4, F_{11}, F_{15}, F_{17} and F_{19}) functions, respectively. However, GWO and WOA cannot avoid the local optimum on three (i.e. F_5, F_{13} and F_{23}) and five functions (i.e. F_5, F_7, F_{10}, F_{13} and F_{23}).

4.2. GTOA for constrained engineering design optimization problems

In this section, the performance of GTOA is examined by solving four challenging engineering design optimization problems and the

Table 7
The experimental results obtained by eight metaheuristic algorithms.

No.	Metric	DE	CS	GWO	SCA	WOA	SSA	PSO	GTOA
F_1	Mean	3.42E-16	8.60E-25	**0.00E+00**	3.14E-26	**0.00E+00**	4.81E-09	8.66E-74	**0.00E+00**
	Std	2.53E-16	1.58E-24	**0.00E+00**	1.23E-25	**0.00E+00**	1.06E-09	2.21E-73	**0.00E+00**
	TR	7	6	2	5	2	8	4	2
F_2	Mean	4.18E-17	2.70E-25	**0.00E+00**	1.30E-23	**0.00E+00**	1.03E-08	5.94E-75	**0.00E+00**
	Std	4.37E-17	1.13E-24	**0.00E+00**	7.07E-23	**0.00E+00**	2.51E-09	2.01E-74	**0.00E+00**
	TR	7	5	2	6	2	8	4	2
F_3	Mean	2.37E-13	1.26E-21	2.95E-308	1.06E-23	**0.00E+00**	1.46E+06	1.78E-69	**0.00E+00**
	Std	2.36E-13	3.06E-21	**0.00E+00**	5.06E-23	**0.00E+00**	6.06E+05	6.80E-69	**0.00E+00**
	TR	7	6	3	5	1.5	8	4	1.5
F_4	Mean	2.16E-32	6.19E-27	**0.00E+00**	1.13E-21	**0.00E+00**	1.52E-24	3.11E-117	**0.00E+00**
	Std	4.04E-32	2.17E-26	**0.00E+00**	6.18E-21	**0.00E+00**	6.28E-25	1.27E-116	**0.00E+00**
	TR	5	6	2	8	2	7	4	2
F_5	Mean	4.93E-01	1.66E+01	4.05E-75	6.57E-01	1.28E+01	1.27E+00	3.24E-05	**2.97E-287**
	Std	1.21E+00	4.13E+00	1.17E-74	2.17E+00	1.94E+01	1.53E+00	3.59E-05	**0.00E+00**
	TR	4	8	2	5	7	6	3	1
F_6	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.50E+00	4.67E-01	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.35E+00	9.00E-01	**0.00E+00**
	TR	3.5	3.5	3.5	3.5	3.5	8	7	3.5
F_7	Mean	8.57E-03	9.23E-02	1.32E-04	4.33E-03	3.17E-04	1.87E-02	4.27E-03	**8.16E-05**
	Std	2.23E-03	3.60E-02	8.97E-05	3.93E-03	5.08E-04	6.01E-03	2.12E-03	**5.17E-05**
	TR	6	8	2	5	3	7	4	1
F_8	Mean	7.14E-08	1.76E-15	8.63E-179	4.83E-28	**0.00E+00**	5.24E-01	1.18E-31	**0.00E+00**
	Std	6.35E-08	5.60E-15	**0.00E+00**	1.76E-27	**0.00E+00**	7.33E-01	5.95E-31	**0.00E+00**
	TR	7	6	3	5	1.5	8	4	1.5
F_9	Mean	6.76E-01	2.72E-02	7.29E-84	2.20E+02	7.57E+01	2.92E-07	5.67E-08	**0.00E+00**
	Std	3.77E-01	2.92E-02	3.99E-83	6.16E+02	7.28E+01	9.01E-08	7.01E-08	**0.00E+00**
	TR	6	5	2	8	7	4	3	1
F_{10}	Mean	1.24E-15	7.45E-03	**0.00E+00**	1.84E-03	4.14E-04	6.73E-03	1.88E-02	**0.00E+00**
	Std	2.59E-15	1.68E-02	**0.00E+00**	1.01E-02	2.27E-03	7.17E-03	1.95E-02	**0.00E+00**
	TR	3	7	1.5	5	4	6	8	1.5
F_{11}	Mean	2.76E-09	3.89E-01	**0.00E+00**	1.02E-01	**0.00E+00**	2.65E+00	1.29E-15	**0.00E+00**
	Std	9.61E-09	2.25E-01	**0.00E+00**	5.58E-01	**0.00E+00**	1.63E+00	5.21E-16	**0.00E+00**
	TR	5	7	2	6	2	8	4	2
F_{12}	Mean	6.53E-09	2.00E+00	8.23E-15	1.61E+01	**3.02E-15**	2.18E+00	8.88E-01	4.44E-15
	Std	2.97E-09	1.13E+00	1.30E-15	7.09E+00	2.21E-15	5.88E-01	8.87E-01	**0.00E+00**
	TR	4	6	3	8	1	7	5	2
F_{13}	Mean	5.07E-01	2.58E-03	2.53E-109	1.56E-01	6.65E+02	2.65E-10	1.05E-13	**2.37E-226**
	Std	4.66E-01	7.78E-03	1.25E-108	5.54E-01	1.49E+01	9.34E-11	1.26E-13	**0.00E+00**
	TR	7	5	2	6	8	4	3	1
F_{14}	Mean	1.67E-02	4.00E-02	3.13E-03	3.13E-03	**2.29E-03**	1.16E-02	9.07E-03	3.13E-03
	Std	3.61E-03	2.49E-02	1.47E-11	3.69E-07	1.41E-03	2.20E-03	3.45E-03	**1.05E-12**
	TR	7	8	3	4	1	6	5	2
F_{15}	Mean	2.77E-16	1.12E-24	**0.00E+00**	6.69E-26	**0.00E+00**	4.89E-09	1.23E-73	**0.00E+00**
	Std	2.15E-16	3.54E-24	**0.00E+00**	3.63E-25	**0.00E+00**	8.50E-10	4.31E-73	**0.00E+00**
	TR	7	6	2	5	2	8	4	2
F_{16}	Mean	7.23E-20	1.62E-28	**0.00E+00**	2.04E-30	**0.00E+00**	3.21E-01	5.78E-76	**0.00E+00**
	Std	7.52E-20	3.53E-28	**0.00E+00**	8.12E-30	**0.00E+00**	1.96E-01	2.05E-75	**0.00E+00**
	TR	7	6	2	5	2	8	4	2
F_{17}	Mean	2.15E-15	5.73E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.06E+01	4.10E+00	**0.00E+00**
	Std	3.59E-15	2.27E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.41E+00	2.16E+00	**0.00E+00**
	TR	5	7	2.5	2.5	2.5	8	6	2.5
F_{18}	Mean	2.79E-34	1.38E-26	**0.00E+00**	4.67E-23	**0.00E+00**	4.71E-08	1.35E-164	**0.00E+00**
	Std	1.53E-33	5.53E-26	**0.00E+00**	2.49E-22	**0.00E+00**	2.28E-08	0.00E+00	**0.00E+00**
	TR	5	6	2	7	2	8	4	2
F_{19}	Mean	**0.00E+00**	7.39E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.26E+00	5.37E-01	**0.00E+00**
	Std	**0.00E+00**	7.52E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.42E-01	2.89E-01	**0.00E+00**
	TR	3	6	3	3	3	8	7	3
F_{20}	Mean	2.00E-01	7.80E-01	1.13E-01	1.19E-01	1.17E-01	5.23E-01	4.37E-01	**9.99E-02**
	Std	7.98E-04	1.88E-01	3.46E-02	4.00E-02	5.92E-02	9.35E-02	9.64E-02	**4.46E-13**
	TR	5	8	2	4	3	7	6	1
F_{21}	Mean	7.10E-09	2.78E+00	7.76E-15	1.38E-13	**3.26E-15**	1.56E+00	1.71E+00	4.44E-15
	Std	3.60E-09	1.61E+00	1.30E-15	6.66E-13	2.35E-15	9.16E-01	6.37E-01	**0.00E+00**
	TR	5	8	3	4	1	6	7	2
F_{22}	Mean	1.90E+02	9.01E+01	**1.25E+01**	8.44E+01	6.45E+01	6.58E+01	3.79E+01	1.42E+01
	Std	8.52E+00	2.00E+01	**1.33E+01**	7.77E+01	3.56E+01	1.72E+01	1.34E+01	1.80E+01
	TR	8	7	1	6	4	5	3	2
F_{23}	Mean	1.69E-14	5.82E-03	6.71E-03	2.36E-02	2.62E-03	9.44E-03	1.30E-02	**0.00E+00**
	Std	1.49E-14	1.25E-02	2.15E-02	1.29E-01	9.55E-03	8.98E-03	1.48E-02	**0.00E+00**
	TR	2	4	5	8	3	6	7	1
F_{24}	Mean	1.66E+02	8.82E+01	**2.22E+01**	1.29E+02	9.81E+01	7.54E+01	4.87E+01	3.06E+01
	Std	**1.06E+01**	2.19E+01	1.25E+01	5.12E+01	5.57E+01	1.80E+01	1.64E+01	2.00E+01
	TR	8	5	1	7	6	4	3	2
F_{25}	Mean	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	Std	**6.78E-16**	**6.78E-16**	1.42E-09	1.53E-05	5.78E-12	3.75E-15	**6.78E-16**	**6.78E-16**
	TR	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5

(*continued on next page*)

Table 7 (*continued*)

No.	Metric	DE	CS	GWO	SCA	WOA	SSA	PSO	GTOA
F_{26}	Mean	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**	**3.98E-01**
	Std	**0.00E+00**	**0.00E+00**	2.64E-08	4.67E-04	1.82E-07	1.73E-15	**0.00E+00**	**0.00E+00**
	TR	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
F_{27}	Mean	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	−3.85E+00	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**
	Std	**2.71E-15**	**2.71E-15**	2.17E-03	1.32E-03	2.13E-03	9.16E-15	**2.71E-15**	**2.71E-15**
	TR	4	4	4	8	4	4	4	4
F_{28}	Mean	**−1.02E+01**	**−1.02E+01**	−9.45E+00	−2.32E+00	−9.64E+00	−9.40E+00	−4.13E+00	**−1.02E+01**
	Std	**7.23E-15**	**7.23E-15**	1.84E+00	2.16E+00	1.55E+00	2.00E+00	1.20E+00	7.54E-09
	TR	2	2	5	8	4	6	7	2

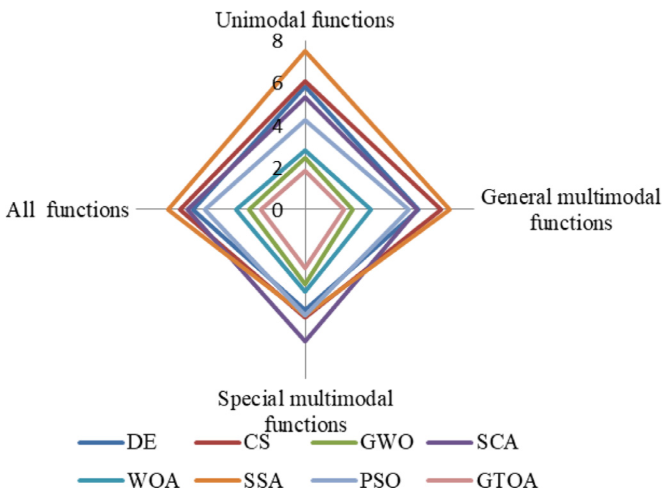

Fig. 6. Radar chart of the overall ranking obtained by eight different algorithms.

optimization results are compared with other reported optimizers. For all engineering optimization problems, the population size for GTOA was chosen as 50; the maximum number of function evaluations for GTOA was set based on the specific problems; GTOA was run 30 times for every problem.

4.2.1. Welded beam design problem

This is a classical engineering design problem that has been often used as a benchmark problem. This problem was proposed by Coello Coello (2000), whose goal is aim to design a welded beam for minimum cost. The constraints are as follows: (1) shear stress (τ); (2) bending stress in the beam (σ); (3) buckling load on the bar (p_b); (4) end deflection of the beam (δ); (5) side constraints. This problem has four variables including thickness of weld $h(x_1)$, length of attached part of bar $l(x_2)$, the height of the bar $t(x_3)$ and thickness of the bar $b(x_4)$. The mathematical formulation can be described as

$$\min f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$s.t.\begin{cases} g_1(x) = \tau(x) - \tau_{\max} \le 0 \\ g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \\ g_3(x) = x_1 - x_4 \le 0 \\ g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \\ g_5(x) = 0.125 - x_1 \le 0 \\ g_6(x) = \delta(x) - \delta_{\max} \le 0 \\ g_7(x) = P - P_c(x) \le 0 \end{cases} \tag{14}$$

Variable change $0.1 \le x_i \le 2 i = 1, 4; 0.1 \le x_i \le 10 i = 2, 3$

where $\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$, $\tau' = \frac{P}{\sqrt{2}x_1x_2}$, $\tau'' = \frac{MR}{J}$,

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

$J = 2\left(\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right)\right)$, $\sigma(x) = \frac{6PL}{x_4x_3^2}$, $\delta(x) = \frac{4PL^3}{Ex_3^3x_4}$,

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

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

$\tau_{\max} = 13,600\text{psi}, \sigma_{\max} = 30,000\text{psi}, \delta_{\max} = 0.25\text{in}$

This problem has been solved using CPSO (He & Wang, 2007b), WCA, SSA, GWO, WOA, TSA (Babalik, Cinar & Kiran, 2018), PVS (Savsani & Savsani, 2016) and QS (Zhang et al., 2018a). The comparisons for the best solutions obtained by different algorithms are presented in Table 9. From Table 9, it can be seen that GTOA, QS and PVS can achieve the best solution with an objective function value of $f(\boldsymbol{x})$ =1.724852. In terms of the number of function evaluations, GTOA reaches the best solution faster than QS and PVS using 10,000 function evaluations. Fig. 8 shows the convergence history of GTOA for finding the best solution of welded beam design problem.

4.2.2. Tubular column design problem

The objective of this problem is to design a uniform column of tubular section to carry a compressive load $P = 2500kgf$ at minimum cost (Gandomi, Yang & Alavi, 2013). Specifically, the column is made of material including a yield stress (σ_y) of 500kgf/cm^2, a modulus of elasticity (E) of $0.85 \times 10^6\text{kgf/cm}^2$, and a density ($\rho$) equal to 0.0025kgf/cm^3. There are two variables and six constraints in this problem. The mathematical formulation can be described as:

$$\min f(d,t) = 9.8dt + 2d$$

$$s.t.\begin{cases} g_1(d,t) = \frac{P}{\pi dt\sigma_y} - 1 \le 0 \\ g_2(d,t) = \frac{8PL^2}{\pi^3Edt(d^2+t^2)} - 1 \le 0 \\ g_3(d,t) = \frac{2.0}{d} - 1 \le 0 \\ g_4(d,t) = \frac{d}{14.0} - 1 \le 0 \\ g_5(d,t) = \frac{0.2}{t} - 1 \le 0 \\ g_6(d,t) = \frac{t}{0.8} - 1 \le 0 \end{cases} \tag{15}$$

Variable change $2 \le d \le 14, 0.2 \le t \le 0.8$

This engineering problem previously was optimized by EM (Rocha & Fernandes, 2009), HEM (Rocha & Fernandes, 2009), KH (Gandomi & Alavi, 2016) and CS (Gandomi et al., 2013). Table 10 presents the best solutions offered by GTOA and other algorithms. As can be seen from Table 10, GTOA is superior to EM, HEM, KH and CS in terms of solution quality and computational efficiency. Fig. 9 shows the convergence history of GTOA for finding the best solution of tubular column design problem.

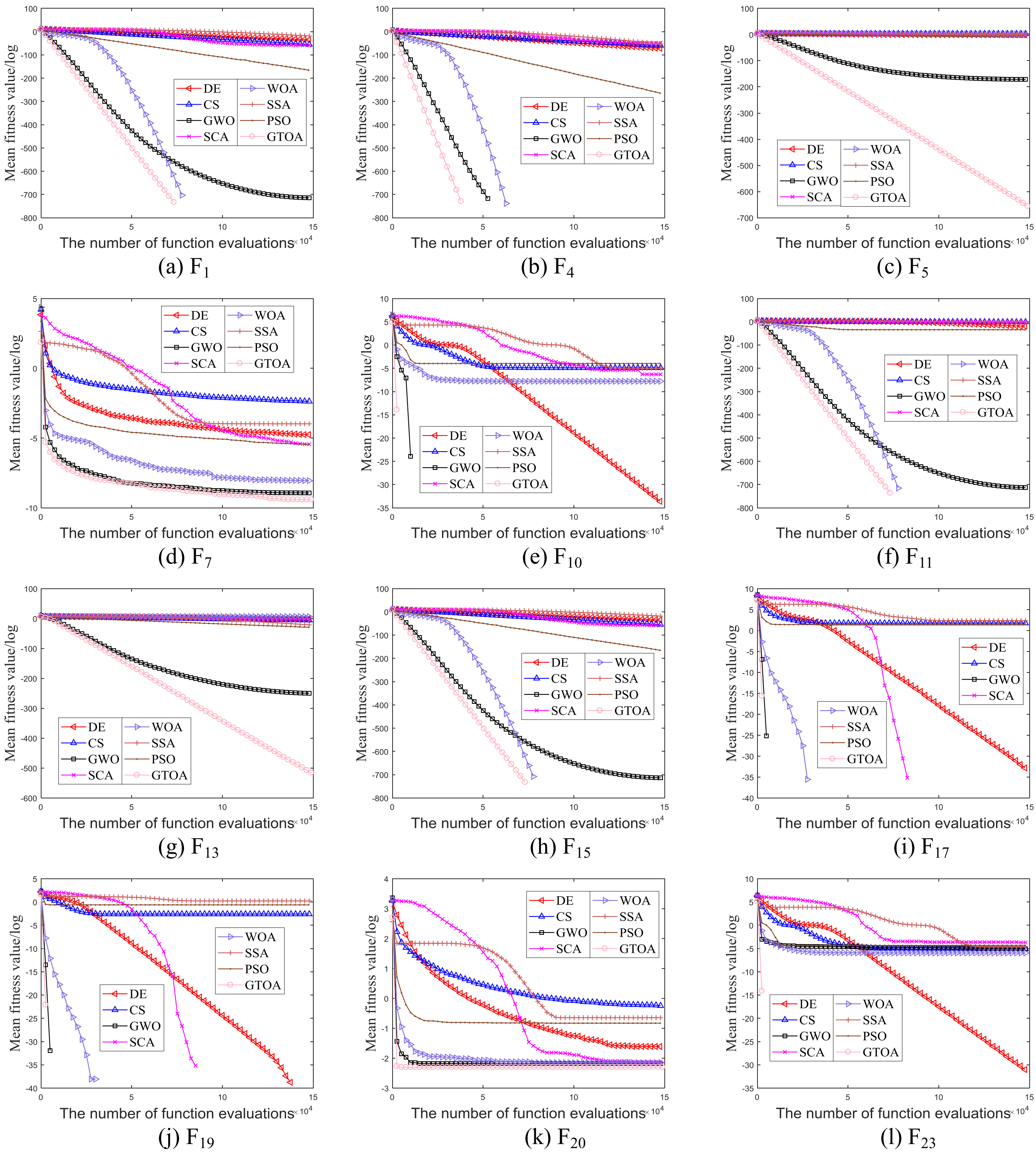

Fig. 7. Several typical convergence curves obtained by eight optimization algorithms.

4.2.3. Pressure vessel design problem

This problem was proposed by Kannan and Kramer (1994) and its target is to minimize the total cost of a pressure vessel considering the cost of material, forming and welding. In this problem, four design variables are considered, which include thickness of the shell $T_s(x_1)$, thickness of the head $T_h(x_2)$, inner radius $R(x_3)$, and length of the cylindrical section of the vessel $L(x_4)$. The mathematical formulation can be defined as

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$s.t. \begin{cases} g_1(x) = -x_1 + 0.0193x_3 \le 0 \\ g_2(x) = -x_2 + 0.00954x_3 \le 0 \\ g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296,000 \le 0 \\ g_4(x) = x_4 - 240 \le 0 \end{cases} \tag{16}$$

Variab lerange $0 \le x_i \le 100, i = 1, 2; 10 \le x_i \le 200, i = 3, 4$

Table 8
Wilcoxon signed-rank test results between GTOA and seven metaheuristic algorithms.

No.	GTOA vs. DE			GTOA vs. CS			GTOA vs. GWO			GTOA vs. SCA			GTOA vs. WOA			GTOA vs. SSA			GTOA vs. PSO		
	p-value	H	S	p-value	H	S	p-value	H	S	p-value	H	S	p-value	H	S	p-value	H	S	p-value	H	S
F1	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F2	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F3	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F4	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F5	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+
F6	1.00E+00	0	=	1.00E+00	0	=	1.00E+00	0	=	1.00E+00	0	=	1.00E+00	0	=	1.68E-06	1	+	3.91E-03	1	+
F7	1.73E-06	1	+	1.73E-06	1	+	1.57E-02	1	+	1.73E-06	1	+	5.67E-03	1	+	1.73E-06	1	+	1.73E-06	1	+
F8	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F9	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+
F10	3.96E-05	1	+	1.95E-03	1	+	1.00E+00	0	=	3.91E-03	1	+	1.00E+00	0	=	1.73E-06	1	+	2.68E-05	1	+
F11	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.67E-06	1	+
F12	1.73E-06	1	+	1.68E-06	1	+	6.80E-08	1	+	1.73E-06	1	+	2.70E-03	1	+	1.73E-06	1	+	1.66E-06	1	+
F13	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+
F14	1.73E-06	1	+	1.73E-06	1	+	5.29E-04	1	+	1.92E-06	1	+	4.23E-06	1	+	1.73E-06	1	+	1.73E-06	1	+
F15	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F16	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F17	2.91E-04	1	+	1.71E-06	1	+	1.00E+00	0	=	1.00E+00	0	=	1.00E+00	0	=	1.73E-06	1	+	1.68E-06	1	+
F18	1.73E-06	1	+	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+
F19	1.00E+00	0	=	6.10E-05	1	+	1.00E+00	0	=	1.00E+00	0	=	1.00E+00	0	=	1.73E-06	1	+	1.64E-06	1	+
F20	1.73E-06	1	+	1.73E-06	1	+	1.20E-03	1	+	1.73E-06	1	+	4.59E-01	0	=	1.73E-06	1	+	1.72E-06	1	+
F21	1.73E-06	1	+	1.73E-06	1	+	3.19E-07	1	+	1.56E-06	1	+	1.24E-02	1	+	1.73E-06	1	+	1.72E-06	1	+
F22	1.73E-06	1	+	1.73E-06	1	+	9.10E-01	0	=	2.47E-04	1	+	7.69E-06	1	+	1.92E-06	1	+	3.41E-05	1	+
F23	1.73E-06	1	+	1.73E-06	1	+	1.25E-01	0	=	1.00E+00	0	=	2.50E-01	0	=	1.73E-06	1	+	2.55E-06	1	+
F24	1.73E-06	1	+	2.13E-06	1	+	7.52E-02	0	=	4.73E-06	1	+	1.32E-05	1	+	1.73E-06	1	+	6.63E-04	1	+
F25	1.00E+00	0	=	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	2.76E-06	1	+	1.00E+00	=	+
F26	1.00E+00	0	=	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	3.91E-03	1	+	1.00E+00	=	+
F27	1.00E+00	0	=	1.00E+00	0	=	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	8.95E-07	1	+	1.00E+00	=	+
F28	5.00E-01	0	=	5.00E-01	0	=	1.73E-06	1	+	1.73E-06	1	+	1.73E-06	1	+	2.16E-05	1	+	8.71E-07	1	+
w/l/t			22/0/6			23/0/5			19/0/9			24/0/4			13/0/15			28/0/0			25/0/3

Table 9
The optimal solutions of several algorithms for welded beam design problem. "NFEs" stands for the number of function evaluations.

Algorithm	Optimal values for variables				Optimal cost	NFEs
	x_1	x_2	x_3	x_4		
CPSO(He & Wang, 2007b)	0.202369	3.544214	9.048210	0.205723	1.728024	240,000
WCA(Eskandar et al., 2012)	0.205728	3.470522	9.036620	0.205729	1.724856	46,450
SSA(Mirjalili et al., 2017)	0.2057	3.4714	9.0366	0.2057	1.72491	NA
GWO(Mirjalili et al., 2014)	0.205676	3.478377	9.03681	0.205778	1.72624	NA
WOA(Mirjalili & Lewis, 2016)	0.20536	3.48293	9.03746	0.206276	1.730499	9900
TSA(Babalik et al., 2018)	0.24415742	6.22306595	8.29555011	0.24440474	2.38241101	30,000
PVS(Savsani & Savsani, 2016)	NA	NA	NA	NA	1.724852	20,000
QS(Zhang et al., 2018a)	NA	NA	NA	NA	1.724852	20,000
GTOA	0.2057296	3.47048867	9.0366239	0.2057296	1.724852	10,000

Table 10
The optimal solutions of the considered algorithms for tubular column design problem.

Algorithms	Optimal values for variables		Optimal value	NFEs
	d	t		
EM(Rocha & Fernandes, 2009)	5.451083	0.291900	26.53380	17,308
HEM(Rocha & Fernandes, 2009)	5.451083	0.291990	26.53227	25,136
KH(Gandomi & Alavi, 2016)	5.451278	0.291957	26.53140	10,000
CS(Gandomi et al., 2013)	5.451390	0.291960	26.53217	15,000
GTOA	5.451156	0.291965	26.531328	9000

The problem of pressure vessel design was optimized previously by CPSO, HPSO (He & Wang, 2007a), NM-PSO (Zahara & Kao, 2009), G-QPSO (Coelho, 2010), WCA, GWO, CS, QS, PVS and TLBO (Rao et al., 2011). The statistical results obtained by 10 algorithms are presented in Table 11. As can be seen from Table 11, GTOA finds the best solution of $f(\boldsymbol{x})$ =5885.333. Fig. 10 shows the convergence history of GTOA for finding the best solution of pressure vessel problem.

4.2.4. Speed reducer design problem

The objective of this problem is to minimize the weight of speed reducer. This problem has seven variables including face width $b(x_1)$, module of teeth $m(x_2)$, number of teeth in the pinion $z(x_3)$, length of the first shaft between bearings $l_1(x_4)$, length of the second shaft between bearings $l_2(x_5)$, diameter of first shafts $d_1(x_6)$ and diameter of second shafts $d_2(x_7)$. The mathematical for-

Table 11
The optimal solutions obtained by 10 algorithms for pressure vessel design problem.

Algorithm	Optimal values for variables				Optimal value	NFEs
	x_1	x_2	x_3	x_4		
CPSO(He & Wang, 2007b)	0.8125	0.4375	42.0913	176.7465	6061.0777	240,000
HPSO(He & Wang, 2007a)	0.8125	0.4375	42.0984	176.6366	6059.7143	81,000
NM-PSO(Zahara & Kao, 2009)	0.8036	0.3972	41.6392	182.4120	5930.3137	80,000
G-QPSO(Coelho, 2010)	0.8125	0.4375	42.0984	176.6372	6059.7208	8000
GWO(Mirjalili et al., 2014)	0.8125	0.4345	42.089181	176.758731	6059.5639	NA
CS(Gandomi et al., 2013)	0.8125	0.4375	42.0984456	176.6365958	6059.7143348	15,000
QS(Zhang et al., 2018a)	NA	NA	NA	NA	6090.526	20,000
PVS(Savsani & Savsani, 2016)	NA	NA	NA	NA	6059.714	15,000
TLBO(Rao et al., 2011)	NA	NA	NA	NA	6059.714	20,000
GTOA	0.778169	0.38465	40.3196	200	5885.333	26,000

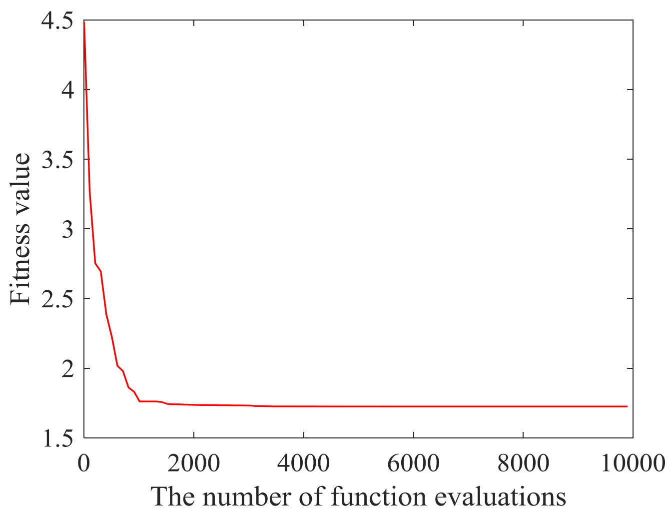

Fig. 8. Convergence history of GTOA for finding the best solution of welded beam design problem.

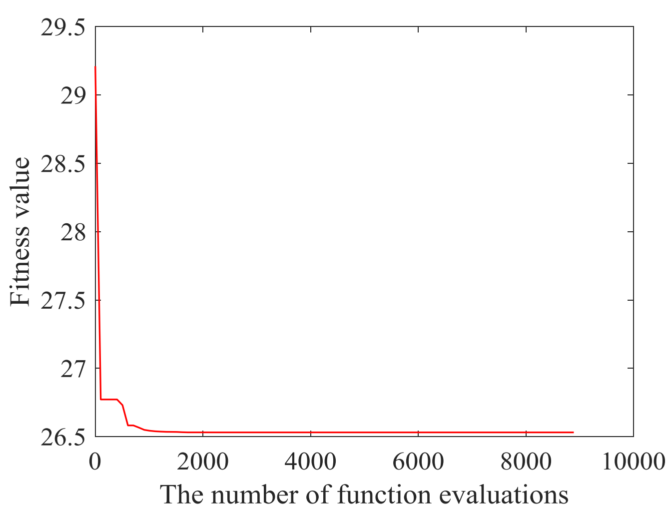

Fig. 9. Convergence history of GTOA for finding the best solution of tubular column problem.

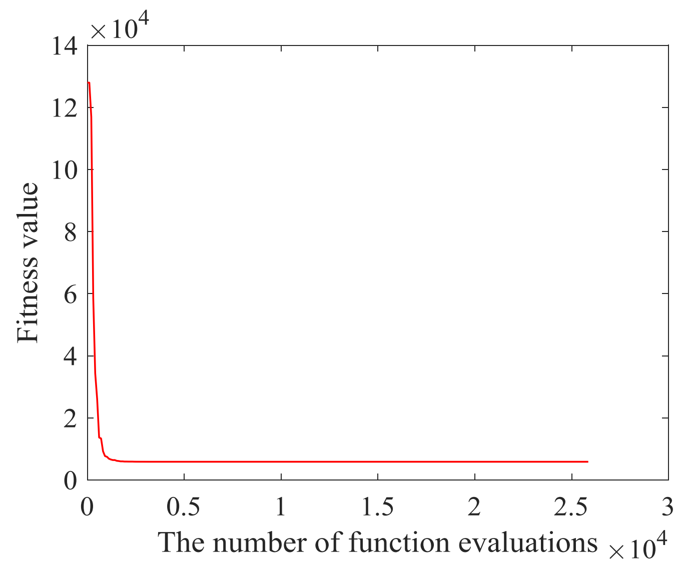

Fig. 10. Convergence history of GTOA for finding the best solution of pressure vessel problem.

mulation can be expressed as

$$\min f(x) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ -1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$$

$$s.t.\begin{cases} g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0 \\ g_2(x) = \frac{397.5}{x_1x_2^2x_3} - 1 \le 0 \\ g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0 \\ g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0 \\ g_5(x) = \frac{\left(\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5\times10^6\right)^{1/2}}{85x_7^3} - 1 \le 0 \\ g_6(x) = \frac{x_2x_3}{40} - 1 \le 0 \\ g_7(x) = \frac{5x_2}{x_1} - 1 \le 0 \\ g_8(x) = \frac{x_1}{12x_2} - 1 \le 0 \\ g_9(x) = \frac{1.5x_6+1.9}{x_4} - 1 \le 0 \\ g_{10}(x) = \frac{1.1x_7+1.9}{x_5} - 1 \le 0 \\ g_{11}(x) = \frac{\left(\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6\right)^{1/2}}{110x_6^3} - 1 \le 0 \end{cases} \quad (17)$$

Variable range $2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28,$
$7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5$

Table 12
The optimal solutions of eight optimization algorithms for speed reducer design problem.

Algorithm	Optimal values for variables							Optimal value	NFEs
	x_1	x_2	x_3	x_4	x_5	x_6	x_7		
PSO-DE(Liu et al., 2010)	3.5	0.7	17.0	7.3	7.8	3.350214	5.2866832	2996.348167	54,350
WCA(Eskandar et al., 2012)	3.5	0.7	17.0	7.3	7.715319	3.350214	5.286654	2994.471066	15,150
HEAA(Wang et al., 2009)	3.500022	0.7	17.0	7.300427	7.715377	3.350230	2.286663	2994.499107	24,000
QS(Zhang et al., 2018a)	NA	NA	NA	NA	NA	NA	NA	2996.348165	25,000
MBA(Sadollah et al., 2013)	3.5	0.7	17.0	7.300033	7.715772	3.350218	5.286654	2994.482453	6300
PVS(Savsani & Savsani, 2016)	NA	NA	NA	NA	NA	NA	NA	2996.471066	15,150
TLBO(Rao et al., 2011)	NA	NA	NA	NA	NA	NA	NA	2996.348170	20,000
GTOA	3.5	0.7	17.0	7.3	7.71532	3.35021467	5.28665446	**2994.471066**	**9000**

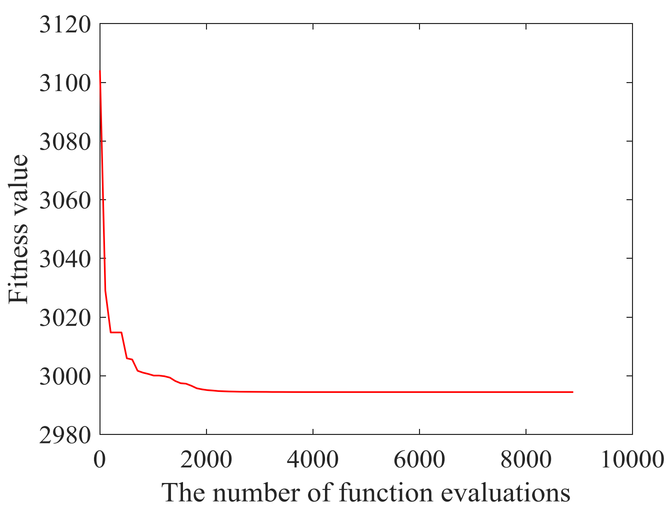

Fig. 11. Convergence history of GTOA for finding the best solution of speed reducer problem.

The optimization algorithms previously applied to this problem include PSO-DE (Liu, Cai & Wang, 2010), WCA, HEAA (Wang, Cai, Zhou & Fan, 2009), QS, MBA (Sadollah, Bahreininejad, Eskandar & Hamdi, 2013), PVS and TLBO. The comparisons of the statistical results for this problem are shown in Table 12. It is observed form Table 12 that GTOA, WCA and PVS outperform other algorithms. However, GTOA reaches the best solution faster than WCA and PVS using 9000 function evaluations. Fig. 11 shows the convergence history of GTOA for finding the best solution of speed reducer design problem.

5. Conclusions

This study presents a novel population-based optimization algorithm for solving global optimization problems, which is inspired by group teaching approach. The proposed algorithm called the group teaching optimization algorithm (GTOA) consists of teacher allocation phase, teacher phase, student phase and ability grouping phase. In addition, unlike most of existing meta-heuristic algorithms, GTOA does not need to preset extra control parameters except two essential parameters (population size and the stopping criterion). Note that the two essential parameters are necessary for all metaheuristic algorithms. In this research, the performance of GTOA is tested by 28 unconstrained benchmark functions and four constrained engineering design problems. According to the experimental results, GTOA can offer better solutions for most of test problems than the compared algorithms in terms of solutions quality and computational efficiency. This also fully indicates the superiority of the proposed GTOA on the ability of escaping from the local optimal solutions.

In further research, there are many works to do, which can be divided into the following two parts. Firstly, group teaching is a very complicated process. Our proposed GTOA is based on a simple group teaching model. There is a lot of room for improvement of the GTOA, such as the teacher allocation mechanism, the teaching methods for different groups and the criterion of ability grouping. Thus we will give some variants of the basic GTOA in further study. Secondly, simple frame, few parameters and strong global search ability are the remarkable characteristics of GTOA, which makes it has great potential to be used to solve different types of optimization problems. In order to expand the applications of the GTOA, we will try to use GTOA to solve complex practical optimization problems about the context of expert and intelligent systems, such as the opinion leader detection in online social network, multi-controller placement in software-defined networks, multi-robot path planning and the deployment optimization of multi-unmanned aerial vehicle.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Yiying Zhang: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Zhigang Jin:** Supervision, Writing - review & editing.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Babalik, A., Cinar, A. C., & Kiran, M. S. (2018). A modification of tree-seed algorithm using Deb's rules for constrained optimization. *Applied Soft Computing, 63*, 289–305. doi:10.1016/j.asoc.2017.10.013.

Chen, D., Zou, F., Li, Z., Wang, J., & Li, S. (2015). An improved teaching–learning-based optimization algorithm for solving global optimization problem. *Information Sciences, 297*, 171–190. doi:10.1016/j.ins.2014.11.001.

Cheng, M.-. Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures, 139*, 98–112. doi:10.1016/j.compstruc.2014.03.007.

Coelho, L. dos S. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*(2), 1676–1683. doi:10.1016/j.eswa.2009.06.044.

Coello Coello, C. A. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*(2), 113–127. doi:10.1016/S0166-3615(99)00046-9.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18. doi:10.1016/j.swevo.2011.02.002.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures, 110–111*, 151–166. doi:10.1016/j.compstruc.2012.07.010.

Ewees, A. A., Abd Elaziz, M., & Houssein, E. H. (2018). Improved grasshopper optimization algorithm using opposition-based learning. *Expert Systems with Applications, 112*, 156–172. doi:10.1016/j.eswa.2018.06.023.

Gandomi, Amir H., & Alavi, A. H. (2016). An introduction of Krill Herd algorithm for engineering optimization. *Journal of Civil Engineering and Management, 22*(3), 302–310. doi:10.3846/13923730.2014.897986.

Gandomi, Amir Hossein, Yang, X.-. S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(1), 17–35. doi:10.1007/s00366-011-0241-y.

He, Q., & Wang, L. (2007a). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186*(2), 1407–1422. doi:10.1016/j.amc.2006.07.134.

He, Q., & Wang, L. (2007b). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*(1), 89–99. doi:10.1016/j.engappai.2006.03.003.

Huang, P., Wang, Y., Wang, K., & Yang, K. (2019). Differential evolution with a variable population size for deployment optimization in a UAV-Assisted IoT data collection system. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1–12. doi:10.1109/TETCI.2019.2939373.

Ibrahim, R. A., Elaziz, M. A., & Lu, S. (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Systems with Applications, 108*, 1–27. doi:10.1016/j.eswa.2018.04.028.

Jain, L., Katarya, R., & Sachdeva, S. (2020). Opinion leader detection using whale optimization algorithm in online social network. *Expert Systems with Applications, 142*, 113016. doi:10.1016/j.eswa.2019.113016.

Kannan, B., & Kramer, S. N. (1994). An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*(2), 405–411.

Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: A novel physically inspired optimization algorithm. *Computers & Structures, 167*, 69–85. doi:10.1016/j.compstruc.2016.01.008.

Kuroki, Y., Young, G. S., & Haupt, S. E. (2010). UAV navigation by an expert system for contaminant mapping with a genetic algorithm. *Expert Systems with Applications, 37*(6), 4687–4697. doi:10.1016/j.eswa.2009.12.039.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*(36), 3902–3933. doi:10.1016/j.cma.2004.09.007.

Liao, W.-. H., Kao, Y., & Li, Y.-. S. (2011). A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks. *Expert Systems with Applications, 38*(10), 12180–12188. doi:10.1016/j.eswa.2011.03.053.

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*(2), 629–640. doi:10.1016/j.asoc.2009.08.031.

Long, W., Wu, T., Liang, X., & Xu, S. (2019). Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Systems with Applications, 123*, 108–126. doi:10.1016/j.eswa.2018.11.032.

Lu, C., Gao, L., & Yi, J. (2018). Grey wolf optimizer with cellular topological structure. *Expert Systems with Applications, 107*, 89–114. doi:10.1016/j.eswa.2018.04.012.

Mafarja, M., Aljarah, I., Heidari, A. A., Faris, H., Fournier-Viger, P., Li, X., et al. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Systems, 161*, 185–204. doi:10.1016/j.knosys.2018.08.003.

Martínez-Peñaloza, M.-. G., & Mezura-Montes, E. (2018). Immune generalized differential evolution for dynamic multi-objective environments: An empirical study. *Knowledge-Based Systems, 142*, 192–219. doi:10.1016/j.knosys.2017.11.037.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133. doi:10.1016/j.knosys.2015.12.022.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. doi:10.1016/j.advengsoft.2017.07.002.

Mirjalili, S., & Lewis, A. (2016). The Whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. doi:10.1016/j.advengsoft.2016.01.008.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf optimizer. *Advances in Engineering Software, 69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007.

Mlakar, U., Fister, I., & Fister, I. (2016). Hybrid self-adaptive cuckoo search for global optimization. *Swarm and Evolutionary Computation, 29*, 47–72. doi:10.1016/j.swevo.2016.03.001.

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications, 115*, 106–120. doi:10.1016/j.eswa.2018.08.008.

Ouyang, H., Gao, L., Kong, X., Zou, D., & Li, S. (2015). Teaching-learning based optimization with global crossover for global optimization problems. *Applied Mathematics and Computation, 265*, 533–556. doi:10.1016/j.amc.2015.05.012.

Pence, I., Cesmeli, M. S., Senel, F. A., & Cetisli, B. (2016). A new unconstrained global optimization method based on clustering and parabolic approximation. *Expert Systems with Applications, 55*, 493–507. doi:10.1016/j.eswa.2016.02.036.

Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation, 12*(1), 64–79. doi:10.1109/TEVC.2007.894200.

Rakhshani, H., & Rahati, A. (2017). Snap-drift cuckoo search: a novel cuckoo search optimization algorithm. *Applied Soft Computing, 52*, 771–794. doi:10.1016/j.asoc.2016.09.048.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303–315. doi:10.1016/j.cad.2010.12.015.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–Learning-Based optimization: An optimization method for continuous non-linear large scale problems. *Information Sciences, 183*(1), 1–15. doi:10.1016/j.ins.2011.08.006.

Rocha, A. M. A. C., & Fernandes, E. M. G. P. (2009). Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. *International Journal of Computer Mathematics, 86*(10–11), 1932–1946. doi:10.1080/00207160902971533.

Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*(5), 2592–2612. doi:10.1016/j.asoc.2012.11.026.

Sadollah, A., Sayyaadi, H., & Yadav, A. (2018). A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm. *Applied Soft Computing, 71*, 747–782. doi:10.1016/j.asoc.2018.07.039.

Savsani, P., & Savsani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling, 40*(5–6), 3951–3978. doi:10.1016/j.apm.2015.10.040.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69–73). doi:10.1109/ICEC.1998.699146.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation, 12*(6), 702–713. doi:10.1109/TEVC.2008.919004.

Storn, R., & Price, K. (1997). Differential evolution – A Simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359. doi:10.1023/A:1008202821328.

Sun, G., Ma, P., Ren, J., Zhang, A., & Jia, X. (2018a). A stability constrained adaptive alpha for gravitational search algorithm. *Knowledge-Based Systems, 139*, 200–213. doi:10.1016/j.knosys.2017.10.018.

Sun, Y., Wang, X., Chen, Y., & Liu, Z. (2018b). A modified whale optimization algorithm for large-scale global optimization problems. *Expert Systems with Applications, 114*, 563–577. doi:10.1016/j.eswa.2018.08.027.

Valian, E., Tavakoli, S., Mohanna, S., & Haghi, A. (2013). Improved cuckoo search for reliability optimization problems. *Computers & Industrial Engineering, 64*(1), 459–468. doi:10.1016/j.cie.2012.07.011.

Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences, 181*(20), 4699–4714. doi:10.1016/j.ins.2011.03.016.

Wang, J., & Zhou, B. (2016). A hybrid adaptive cuckoo search optimization algorithm for the problem of chaotic systems parameter estimation. *Neural Computing and Applications, 27*(6), 1511–1517. doi:10.1007/s00521-015-1949-1.

Wang, Y., Cai, Z., Zhou, Y., & Fan, Z. (2009). Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural and Multidisciplinary Optimization, 37*(4), 395–413. doi:10.1007/s00158-008-0238-3.

Wang, Y., Ru, Z., Wang, K., & Huang, P. (2019). Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. *IEEE Transactions on Cybernetics*, 1–14. doi:10.1109/TCYB.2019.2935466.

Watson, B. (2007). *The analects of confucius*. Columbia University Press.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. doi:10.1109/4235.585893.

Yang, X., & Deb, S. (2009). Cuckoo search via lévy flights. In *Proceedings of the World Congress on Nature Biologically Inspired Computing (NaBIC)* (pp. 210–214). doi:10.1109/NABIC.2009.5393690.

Yang, X.-. S., & Deb, S. (2014). Cuckoo search: Recent advances and applications. *Neural Computing and Applications, 24*(1), 169–174. doi:10.1007/s00521-013-1367-1.

Yao, X., Liu, Y, & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82–102. doi:10.1109/4235.771163.

Yi, J., Gao, L., Li, X., Shoemaker, C. A., & Lu, C. (2019). An on-line variable-fidelity surrogate-assisted harmony search algorithm with multi-level screening strategy for expensive engineering design optimization. *Knowledge-Based Systems, 170*, 1–19. doi:10.1016/j.knosys.2019.01.004.

Zahara, E., & Kao, Y.-. T. (2009). Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert Systems with Applications, 36*(2, Part 2), 3880–3886. doi:10.1016/j.eswa.2008.02.039.

Zareie, A., Sheikhahmadi, A., & Jalili, M. (2020). Identification of influential users in social network using gray wolf optimization algorithm. *Expert Systems with Applications, 142*, 112971. doi:10.1016/j.eswa.2019.112971.

Zhang, J., Xiao, M., Gao, L., & Pan, Q. (2018a). Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Applied Mathematical Modelling, 63*, 464–490. doi:10.1016/j.apm.2018.06.036.

Zhang, K., Huang, Q., & Zhang, Y. (2019a). Enhancing comprehensive learning particle swarm optimization with local optima topology. *Information Sciences, 471*, 1–18. doi:10.1016/j.ins.2018.08.049.

Zhang, X., Kang, Q., Cheng, J., & Wang, X. (2018b). A novel hybrid algorithm based on biogeography-based optimization and grey wolf optimizer. *Applied Soft Computing, 67*, 197–214. doi:10.1016/j.asoc.2018.02.049.

Zhang, Y., Jin, Z., & Chen, Y. (2019b). Hybrid teaching–learning-based optimization and neural network algorithm for engineering design optimization problems. *Knowledge-Based Systems*. doi:10.1016/j.knosys.2019.07.007.