Information Sciences 456 (2018) 131–144

Contents lists available at ScienceDirect

Information Sciences

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

An improved teaching-learning-based optimization for constrained evolutionary optimization

Bing-Chuan Wang[a,*], Han-Xiong Li[a,b], Yun Feng[a]

[a] Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong
[b] State Key Laboratory of High Performance Complex Manufacturing, School of Mechanical and Electrical Engineering, Central South University, Changsha, China

ARTICLE INFO

Article history:
Received 14 November 2017
Revised 18 April 2018
Accepted 30 April 2018
Available online 4 May 2018

Keywords:
Constrained optimization
TLBO
Tradeoff
Diversity
Convergence
Constraints
Objective function

ABSTRACT

When extending a global optimization technique for constrained optimization, we must balance not only diversity and convergence but also constraints and objective function. Based on these two criteria, the famous teaching-learning-based optimization (TLBO) is improved for constrained optimization. To balance diversity and convergence, an efficient subpopulation based teacher phase is designed to enhance diversity, while a ranking-differential-vector-based learner phase is proposed to promote convergence. In addition, how to select the teacher in the teacher phase and how to rank two solutions in the learner phase have a significant impact on the tradeoff between constraints and objective function. To address this issue, a dynamic weighted sum is formulated. Furthermore, a simple yet effective restart strategy is proposed to settle complicated constraints. By adopting the ε constraint-handling technique as the constraint-handling technique, a constrained optimization evolutionary algorithm, i.e., improved TLBO (ITLBO), is proposed. Experiments on a broad range of benchmark test functions reveal that ITLBO shows better or at least competitive performance against other constrained TLBOs and some other constrained optimization evolutionary algorithms.

1. Introduction

Most scientific and engineering optimization problems can be formulated as constrained optimization problems (COPs). Hence, designing an effective constrained optimization algorithm is of great significance. Without loss of generality, a COP is described as follows:

$$\begin{aligned}&\text{minimize } f(\vec{x}),\ \vec{x}=(x_1,\ldots,x_D)\in S,\ L_i\le x_i\le U_i\\&\text{subject to }: g_j(\vec{x})\le 0,\ j=1,\ldots,l\\&h_j(\vec{x})=0,\ j=l+1,\ldots,m\end{aligned}$$

where $f(\vec{x})$ is the objective function; $\vec{x}=(x_1,\ldots,x_D)$ is a D-dimensional decision vector (solution); L_i and U_i are the lower and upper boundaries of x_i, respectively; $S=\prod_{i=1}^{D}[L_i,U_i]$ is the decision space; $g_j(\vec{x})$ is the jth inequality constraint; $h_j(\vec{x})$ is the $(j-l)$th equality constraint; l is the number of inequality constraints; and $(m-l)$ is the number of equality constraints.

* Corresponding author.
E-mail address: bingcwang3-c@my.cityu.edu.hk (B.-C. Wang).

https://doi.org/10.1016/j.ins.2018.04.083
0020-0255/

Conventionally, an equality constraint is always relaxed to two inequality constraints by a positive tolerance value δ:

$$|h_j(\vec{x})| - \delta \le 0. \tag{1}$$

Conventionally, when a nature-inspired algorithm is utilized to solve COPs, the degree of constraint violation on the jth constraint is calculated as follows:

$$G_j(\vec{x}) = \begin{cases} \max(0, g_j(\vec{x})), & 1 \le j \le l \\ \max(0, |h_j(\vec{x})| - \delta), & l+1 \le j \le m. \end{cases} \tag{2}$$

Consequently, the total degree of constraint violation on all constraints is calculated as follows:

$$G(\vec{x}) = \sum_{j=1}^{m} G_j(\vec{x}). \tag{3}$$

A solution $\vec{x}$ is said to be a feasible solution, if and only if $G(\vec{x})$ is equal to zero. Otherwise, $\vec{x}$ is called an infeasible solution. The aim of a constrained optimization algorithm is to seek the optimal feasible solution.

Various nature-inspired algorithms have been tailored to solve COPs due to their outstanding search performance [24]. Recently, a novel nature-inspired algorithm, that is, teaching-learning-based optimization (TLBO) has been proposed [33]. This method equates the whole optimization process with the teaching and learning process in a classroom. To be specific, each solution represents a student, each dimension of a solution represents a subject, and the objective function value is regarded as the output of a class, respectively. In the teacher phase, the best solution is considered as the teacher. Sequentially, a differential vector, which is directing to the teacher, is generated for each student to learn the outstanding performance. In the learner phase, each student randomly learns from other students. TLBO exhibits numerous promising features such as the ease of implementation, few algorithm-specific parameters, and powerful search ability. Zou et al. [50] proposed a dynamic group strategy to promote diversity, where each learner is able to learn from the mean vector of its corresponding group, rather than the mean vector of the whole class. In addition, in the learner phase, a quantum-behaved learning is executed with a certain probability. Yu et al. [45] employed several techniques, which involve a feedback phase, mutation crossover operation of differential evolution (DE) algorithms, and chaotic perturbation mechanism, to prevent TLBO's premature convergence. Waghmare [38] gave the correct understanding of TLBO algorithm, where the algorithm-specific parameter-less concept of TLBO is explained. Shao et al. [35] proposed a hybrid meta-heuristic based on the probabilistic teaching-learning mechanism to solve the no-wait flow shop scheduling problem with the makespan criterion. The method includes four phases, i.e., previewing before class, teaching phase, learning phase, and reviewing after class. Yu et al. [43] proposed a self-adaptive TLBO to identify the parameters of the photovoltaic model where each learner selects different learning phases self-adaptively. Similarly, Zamli et al. [47] utilized Mamdani fuzzy inference system to select global and local search operations adaptively. Due to the page limitation, other improved TLBOs for global optimization can be referred to [2,11].

Though various studies have been done to improve the performance of TLBO for global optimization, few studies have focused on constrained optimization. Degertekin and Hayalioglu [16] combined TLBO and a modified feasible-based mechanism to optimize truss structures. Bhattacharjee et al. [3] utilized TLBO to solve economic load dispatch problems involving different linear and nonlinear constraints. Banerjee et al. [1] presented a novel TLBO to solve economic load dispatch of the thermal unit without considering transmission losses. In these studies, TLBO has been directly integrated with existing constraint-handling techniques without modifications, which may limit its performance on constrained optimization to a great extent. Rao and Patel [32] proposed an elitist TLBO to tackle COPs. In this method, some elitist solutions are kept and then used to replace some solutions in the main population. To compare two solutions, the feasibility rule [41] is adopted. Though the elitist strategy can increase the convergence, it runs a high probability of being trapped in a local optimum. Moreover, the feasibility rule, which is also a greedy selection strategy, would make the situation more serious. Yu et al. [46] proposed a diversity learning TLBO for cyclic scheduling of an ethylene cracking furnace system. In this method, a new updating formula is presented for the learner phase. Mandal and Roy [23] incorporated quasi-opposition based learning into TLBO to accelerate convergence and improve solution quality. Niknam et al. [29] proposed a modified TLBO to solve the reserve constrained dynamic economic dispatch where a new phase named “modified phase” based on a self-adaptive learning mechanism is added. The modified TLBO includes a fitness weighted mean based teacher phase and a refined learner phase. Baykasoğlu et al. [2] tested TLBO on some combinatorial optimization problems. Recently, an improved constrained TLBO (ICTLBO) was proposed [44]. ICTLBO designs several strategies to prevent the premature convergence of the basic TLBO. It reveals overwhelming advantages over elitist TLBO and some other state-of-the-art constrained optimization evolutionary algorithms (COEAs). It can be observed that the main emphasis of most of the above methods is put on balancing diversity and convergence while the tradeoff between constraints and objective function is neglected to a certain degree. As we know, the core issue of constrained optimization is to balance constraints and objective function. Hence, such a neglect would limit their performance to a certain degree.

When extending a global optimizer to a constrained one, we should be concerned with two kinds of tradeoffs, i.e., the tradeoff between diversity and convergence, and the tradeoff between constraints and objective function. Researchers have explored the first tradeoff by developing various evolutionary algorithms (EAs) such as DE [39,49], particle swarm optimization (PSO) [30,48], artificial bee colony algorithm (ABC) [14,15], and TLBO [6]. A comprehensive survey on this topic can be referred to [13]. Among these algorithms, the subpopulation strategy is usually utilized to increase diversity [41], while ranking-differential-vector strategy is employed to promote convergence [17]. Wang et al. [41] proposed a subpopulation based replacement strategy for constrained optimization. In this method, the population is sorted according to objective function values. Afterward, the sorted population is divided into several subpopulations equally. By utilizing a ranking-differential-vector strategy, Gong et al. modified DE for constrained optimization [17]. The tradeoff between constraints and objective function has been considered in various constraint-handling techniques [24]. All of the referred experience can be borrowed to extend a global optimizer to solve COPs.

Based on the above observations, an improved TLBO (ITLBO) is proposed in this paper for constrained optimization. To balance diversity and convergence, an efficient subpopulation based teacher phase and a ranking-differential-vector-based learner phase are proposed. In terms of the tradeoff between constraints and objective function, a dynamic weighted sum is formulated to select a teacher and rank solutions. A simple yet effective restart strategy is also designed to tackle COPs with complicated constraints. The ε constraint-handling technique [36], which is adopted by the winner of IEEE CEC2010 competition [22], is taken as the constraint-handling technique. As a result, an alternative COEA (i.e., ITLBO), is presented. The main contributions of this paper are highlighted as follows:

- An efficient subpopulation strategy is designed to increase the diversity of the teacher phase.
- A novel ranking differential vector is presented to promote the convergence of the learner phase.
- A dynamic weighted sum is formulated to utilize the valuable knowledge summarized in the community of constrained evolutionary optimization.
- A simple yet effective restart strategy is presented to settle complicated constraints.
- Extensive experiments on a broad range of benchmark test functions have demonstrated that ITLBO provides state-of-the-art performance against other constrained TLBOs and COEAs.

The remaining structure of this paper is arranged as follows: Section 2 introduces the preliminary knowledge of this paper. The proposed ITLBO is elaborated in Section 3. Section 4 summarizes the experimental study and corresponding discussions. Finally, some concluding remarks are presented in Section 5.

2. Preliminary knowledge

TLBO, ICTLBO, and the ε constraint-handling technique, which are closely related to ITLBO, are briefly reviewed.

2.1. TLBO

A basic TLBO is composed of two stages: the teacher phase and the learner phase. In the teacher phase, each solution $\vec{x}_i (i \in \{1, \ldots, NP\})$ learns from the teacher, which can be formulated mathematically as follows:

$$\vec{x}_{i,new} = \vec{x}_{i,old} + rand \cdot (\vec{x}_{teacher} - T_F \cdot \vec{x}_{mean}). \tag{4}$$

Where NP is the population size; $\vec{x}_{i,old}$ is the solution before learning from the teacher; $\vec{x}_{i,new}$ is the new position of the solution; $rand$ is a random number which is uniformly distributed in [0,1]; $\vec{x}_{teacher}$ is the teacher, which is the solution with the best performance in the population; $\vec{x}_{mean}$ is the mean vector of all solutions, and $T_F = round(1 + rand)$ is randomly assigned a value of either 1 or 2. Due to the fact that each solution learns from the best solution, this operator has the property of convergence.

In the learner phase, the solutions learn from each other. To be specific, $\vec{x}_i$ learns from a randomly selected solution $\vec{x}_l$ as follows:

$$\vec{x}_{i,new} = \begin{cases} \vec{x}_{i,old} + rand \cdot (\vec{x}_l - \vec{x}_{i,old}), & \text{if } f(\vec{x}_l) < f(\vec{x}_{i,old}) \\ \vec{x}_{i,old} + rand \cdot (\vec{x}_{i,old} - \vec{x}_l), & \text{if } f(\vec{x}_{i,old}) \le f(\vec{x}_l). \end{cases} \tag{5}$$

As can be seen from Eq. (5), in this manner, diversity can be preserved to a certain degree. It is interesting to find that TLBO seems to be closely related to the mutation operators of DE [12,27,31]. In fact, TLBO is not a new metaphor per se but a rebranding of DE/PSO or a combination of DE and PSO [18,26].

2.2. ICTLBO

ICTLBO is a recent proposal which applies several techniques to enhance the TLBO's ability to solve COPs. Firstly, the population is divided into K subpopulations according to Euclidean distance. Then each learner learns from the teacher through a differential vector that points to the teacher from the sub-mean position. In addition, each subpopulation would exchange information with other subpopulations directly or indirectly. In summary, in the teacher phase, the solutions in the first subpopulation are updated mathematically as follows:

$$\vec{x}_{i,new,1} = \vec{x}_{i,old,1} + rand \cdot (\vec{x}_{teacher} - T_F \cdot SubMean_1) + rand \cdot (\vec{x}_{r1,1} - \vec{x}_{r2,1}). \tag{6}$$

And the solutions in the subsequent subpopulations learn from the teacher in the following manner:

$$\vec{x}_{i,new,k} = \vec{x}_{i,old,k} + rand \cdot (\vec{x}_{teacher} - T_F \cdot SubMean_k) + rand \cdot (\vec{x}_{SubBest,k-1} - \vec{x}_{r2,k}), \quad k \in \{2, \ldots, K\}, \tag{7}$$

where $\vec{x}_{i,old,k}$ and $\vec{x}_{i,new,k}$ are the solutions before and after learning in the kth ($k \in \{1, \ldots, K\}$) subpopulation, respectively; $SubMean_k$ is the mean position of the kth subpopulation; $\vec{x}_{SubBest,k-1}$ is the solution with the best performance in the $(k-1)$th subpopulation, and $\vec{x}_{r1,k}$ and $\vec{x}_{r2,k}$ are two mutually different solutions selected from the kth subpopulation. Note that neither $\vec{x}_{r1,k}$ nor $\vec{x}_{r2,k}$ is equal to $\vec{x}_{i,old,k}$.

In order to further improve the search ability, a diversity strategy is embedded into the learner phase. The new formula is described as follows:

$$\vec{x}_{i,new} = \vec{x}_{i,old} + rand \cdot (\vec{x}_l - \vec{x}_{i,old}) + rand \cdot (\vec{x}_{r1} - \vec{x}_{r2}), \quad \text{if } f(\vec{x}_l) < f(\vec{x}_{i,old}) \tag{8}$$

$$x^d_{i,new} = \begin{cases} x^d_{i,old}, & \text{if } rand_1 < rand_2 \\ x^d_{r1} + r^d \cdot (x^d_{r2} - x^d_{r3}), & \text{otherwise} \end{cases}, \text{ if } f(\vec{x}_{i,old}) \le f(\vec{x}_l), \tag{9}$$

where $d \in \{1, \ldots, D\}$ represents the dth dimension; $rand_1$ and $rand_2$ are two random values uniformly distributed in [0,1]; r^d is a random value uniformly distributed in [−1,1], and $\vec{x}_{r1}$, $\vec{x}_{r_2}$, and $\vec{x}_{r_3}$ are mutually different solutions randomly selected from the population. Note that these three solutions are different from $\vec{x}_i$. Additionally, the comparisons in the learner phase are based on the feasibility rule.

2.3. ε Constraint-handling technique

In the ε constraint-handling technique, when comparing two vectors, say $\vec{x}_i$ and $\vec{x}_j$, $\vec{x}_i$ is better than $\vec{x}_j$ if and only if the following conditions are satisfied:

$$\begin{cases} f(\vec{x}_i) < f(\vec{x}_j), & \text{if } G(\vec{x}_i) \le \varepsilon \wedge G(\vec{x}_j) \le \varepsilon \\ f(\vec{x}_i) < f(\vec{x}_j), & \text{if } G(\vec{x}_i) = G(\vec{x}_j) \\ G(\vec{x}_i) < G(\vec{x}_j), & \text{otherwise} \end{cases} \tag{10}$$

In Eq. (10), ε declines as the generation increases:

$$\varepsilon = \begin{cases} \varepsilon_0 \left(1 - \frac{t}{T}\right)^{cp}, & \text{if } t \le Tc \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$cp = -\frac{\log\varepsilon_0 + \lambda}{\log(1 - \frac{Tc}{T})}, \tag{12}$$

where ε_0 is the initial threshold which is set to the maximum degree of constraint violation of the initial population; t is the current generation number; T is the maximum generation number; λ is set to 10 in this paper, and Tc is a parameter to truncate the value of ε.

3. ITLBO

Two tradeoffs are critical to extending a global optimizer for constrained optimization. Bearing these in mind, the outstanding global optimizer TLBO, is improved to solve COPs in this paper. A simple yet effective restart strategy is also proposed to cope with complicated constraints. These two tradeoffs and the restart strategy will be introduced next.

3.1. Tradeoff between diversity and convergence

As we know, the basic TLBO is easy to be trapped in a local optimum. Accordingly, a variety of improved TLBOs have been proposed to prevent such premature convergence. In ICTLBO, a subpopulation strategy and a diversity strategy are designed to enhance the diversity of the teacher phase and the learner phase, respectively. However, the clustering process in ICTLBO is time-consuming not only due to the Euclidean distance between every two solutions but also because the Euclidean distance between each solution and the reference point must be calculated. In addition, each solution still learns from the exclusive teacher, which might also be easily trapped. As we know, the operator in the learner phase of the basic TLBO is more diverse than that in the teacher phase. Hence, an extra diversity strategy will make the basic TLBO more disturbed, which might hinder its ability to locate the feasible optimum. In view of these, a novel efficient subpopulation strategy is proposed to enhance the diversity of the teacher phase and a ranking differential vector is presented to reduce the disturbance of the learner phase. Consequently, the tradeoff between diversity and convergence can be achieved in both the teacher phase and the learner phase.

In the teacher phase of ITLBO, after sorting the population in ascending order based on objective function values, we divide it into K subpopulations with the same size. This can be considered as a simple and cheap niching technique [41]. The clustering process is very efficient because only a sorting procedure is needed. Subsequently, the solutions in the kth ($k \in \{1, \ldots, K\}$) subpopulation are updated as follows:

$$\vec{x}_{i,new,k} = \vec{x}_{i,old,k} + rand \cdot (SubTeacher_k - T_F \cdot (SubMean_k + \vec{x}_{i,old,k})/2) + rand \cdot (\vec{x}_{r1,k} - \vec{x}_{r2,k}), \tag{13}$$

where $SubTeacher_k$ is the solution with the smallest weighted sum in the ith subpopulation. Note that the weighted sum will be described in the next sub-section. As shown in the equation, each subpopulation has its own teacher which would prevent the premature convergence to a certain degree. In addition, each solution learns from the teacher through two differential vectors:

$$SubTeacher_k - T_F \cdot (SubMean_k + \vec{x}_{i,old,k})/2 = \frac{(SubTeacher_k - T_F \cdot SubMean_k)}{2} + \frac{(SubTeacher_k - T_F \cdot \vec{x}_{i,old,k})}{2}. \tag{14}$$

As described above, in addition to the differential vector between $SubTeacher_k$ and $SubMean_k$, another differential vector between $SubTeacher_k$ and $\vec{x}_{i,old,k}$ is introduced. In this manner, the diversity of the teacher phase can be further enhanced.

In the learner phase of ITLBO, when the weighted sum of $\vec{x}_{i,old}$, i.e., $FIT(\vec{x}_{i,old})$, is greater than that of $\vec{x}_l$, i.e, $FIT(\vec{x}_l)$, the solution is updated as the same to Eq. (8). When $FIT(\vec{x}_{i,old}) \le FIT(\vec{x}_l)$, we modify the Eq. (9) of ICTLBO in the following way to promote convergence:

$$x^d_{i,new} = \begin{cases} x^d_{i,old}, & \text{if} \quad rand_1 < rand_2 \\ x^d_{r1} + r^d \cdot \vec{V}^d, & \text{otherwise} \end{cases} \tag{15}$$

where $\vec{V}$ is a ranking differential vector which can be described as follows:

$$\vec{V} = \begin{cases} \vec{x}_{r2} - \vec{x}_{r3}, & \text{if } FIT(\vec{x}_{r2}) < FIT(\vec{x}_{r3}) \\ \vec{x}_{r3} - \vec{x}_{r2}, & \text{otherwise.} \end{cases} \tag{16}$$

As shown in Eqs. (15) and (16), by utilizing the ranking differential vector $\vec{V}$, which points to the region with the smaller weighted sum, convergence can be promoted.

In summary, by the above process, the tradeoff between diversity and convergence can be maintained in both the teacher phase and the learner phase of ITLBO.

3.2. Tradeoff between constraints and objective function

The tradeoff between constraints and objective function is another core issue to be addressed. In the proposed ITLBO, how to select a teacher and how to rank two solutions are closely related to this issue. To address this issue, a weighted sum FIT is formulated as follows:

$$FIT(\vec{x}_i) = pf \cdot f^{norm}(\vec{x}_i) + (1 - pf) \cdot G^{norm}(\vec{x}_i) \tag{17}$$

$$f^{norm}(\vec{x}_i) = \frac{f(\vec{x}_i) - f_{\min}}{f_{\max} - f_{\min}} \tag{18}$$

$$G^{norm}(\vec{x}_i) = \frac{G(\vec{x}_i) - G_{\min}}{G_{\max} - G_{\min}}, \tag{19}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and minimum objective function values in the population, respectively, and $G_{\max}$ and $G_{\min}$ are the maximum and minimum degree of constraint violation in the population, respectively. Note that when $f_{\max}$ is equal to $f_{\min}$, $f^{norm}(\vec{x}_i)$ is set to 0. Similarly, when $G_{\max}$ is equal to $G_{\min}$, $G^{norm}(\vec{x}_i)$ is set to 0. As described in Eqs. (17)–(19), a bigger pf would prefer more information of objective function while a smaller pf would prefer more information of constraints. Thus, pf is significant to the tradeoff between constraints and objective function.

However, it is not easy to select a proper pf due to the fact that it is optimization-stage-dependent and problem-dependent. Fortunately, in the community of constrained evolutionary optimization, many researchers have found that using more information of objective function at the early stage while less at the later stage is beneficial to constrained optimization [36,39]. The reasons are intuitive. At the early stage of optimization, the population is expected to explore the infeasible regions, which is beneficial to locating discrete feasible regions and seeking the optimum on the boundary between the infeasible and feasible regions. Hence, more information of objective function is preferred. On the contrary, at the later stage, the population needs to enter the feasible regions promptly. Too much information of objective function would hinder this process. Motivated by these observations, the parameter pf decreases dynamically as generation increases according to a trend function [17]:

$$pf = 1 - 0.5 \cdot \left(1 - \cos\left(\frac{t}{T} \cdot \pi\right)\right), \tag{20}$$

where t is the current generation number, and T is the maximum generation number. As shown in Fig. 1, pf decreases with the increase of generation. Consequently, more information of objective function would be used at the early stage while less would be utilized at the later stage. After attaining the weighted sum according to Eqs. (17)–(20), this value would be utilized to select a teacher and rank two solutions. In summary, by the above analyses and implementation, the tradeoff between constraints and objective function can be achieved. It should be noted that though similar techniques [9,10,28] have been utilized to balance diversity and convergence, they have been scarcely utilized to achieve the tradeoff between constraints and objective function.

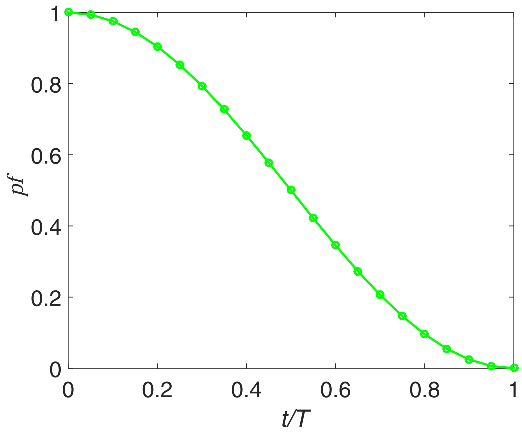

Fig. 1. The decreasing trend of *pf*.

Remark 1. The dynamic weighted sum method can be considered as a constraint-handling technique. According to the no free lunch theorem [42], it is impossible for a single constraint-handling technique to outperform all other techniques on every problem. Hence, we utilize the ε constraint-handling technique to select promising solutions for the next generation. In Section S-V of the supplementary file, experiments have been conducted to validate the effectiveness of combining these two constraint-handling techniques in this manner.

3.3. Restart strategy

In practice, some COPs have constraints with complicated properties such as multi-modal and nonlinearities. Due to the complex infeasible region formulated by these constraints, a COEA can easily be trapped in a local optimum. To remedy this weakness, a simple yet effective restart strategy is proposed.

Before executing this restart strategy, we need to judge whether the population has been trapped already. Conventionally, if the population converges in the infeasible region, the differences among all solutions are small. Consequently, the standard deviation of either objective function values or degree of constraint violation would be tiny. In addition, both the dynamic weighted sum method and the ε constraint-handling technique prefer more information of objective function at the early stage while less at the later stage. Hence, under normal circumstances, the differences among the degree of constraint violation of solutions would be relatively significant when all solutions are infeasible. In view of these, we can infer that the population has been trapped in the infeasible region if the following two conditions are satisfied:

- The whole population is infeasible.
- The standard deviation of degree of constraint violation is smaller than a predefined threshold μ.

Once these two conditions are satisfied, the restart strategy is triggered, that is, the population is regenerated randomly.

Based on these descriptions, an alternative constrained TLBO, that is, ITLBO, is presented. The whole framework of the algorithm is summarized in Algorithm 1 . Note that, in order to match with the ε constraint-handling technique, both the teacher phase and learner phase are executed with a probability of 0.5 for each solution.

4. Experimental study

4.1. Benchmark test functions and parameter settings

In order to validate the performance of ITLBO, two sets of benchmark test suites, which include test functions with various tough properties, were adopted. To be specific, these two test suites are benchmarks from the IEEE CEC2006 competition [20] and the IEEE CEC2010 competition [22], respectively.

The population size *NP* and the maximum number of function evaluations *MaxFEs* are described in Table 1. To reduce randomness, 25 independent runs were performed for each test function. In addition, the tolerance value δ was set to 0.0001. Note that the settings of *MaxFEs*, number of runs, and δ are in accordance with the suggestions in [20,22,44]. Additionally, they were kept the same in all compared algorithms. In addition, *K* in the subpopulation strategy, μ in the restart strategy, and T_c in the ε constraint-handling technique [36] were set to 10, 1E−08 and 0.5*T*, respectively.

Algorithm 1: The Framework of ITLBO.

Step (1) Initialization:
Step (1.1) Set the generation number $t = 1$; Set the maximum generation number $T = \left\lfloor \frac{MaxFEs}{NP} \right\rfloor$, where *MaxFEs* is the maximum function evaluations.
Step (1.2) Generate a random population with *NP* solutions, $P = \{\vec{x}_1, \ldots, \vec{x}_{NP}\}$ and evaluate the population $FV = \{(f(\vec{x}_1), G(\vec{x}_1)), \ldots, (f(\vec{x}_{NP}), G(\vec{x}_{NP}))\}$.
Step (1.3) Set the initial threshold value ε_0 of the ε constraint-handling technique as the maximum degree of constraint violation in the population. Afterward, calculate *cp* according to Eq. (12).
Step (1.4) Initialize the number of subpopulations *K*, and the threshold of the restart strategy, μ;
Step (2) Updating:
Step (2.1) Calculate the parameter pf according to Eq. (20).
Step (2.2) Calculate the weighted sum of all solutions according to Eqs. (17)–(19).
Step (2.3) Sort the population in ascending order according to objective function values and then divide it into *K* subpopulations equally.
Step (2.4) Execute the improved TLBO to generate offspring:
For $i = 1, \ldots, NP$ do
  IF $rand < 0.5$
    Execute the teacher phase according to Eq. (13).
  Else
    Execute the learner phase according to Eqs. (8), (15), and (16).
  End If
End For
Step (2.5) Evaluate the offspring.
Step (2.6) Execute the ε constraint-handling technique for selection.
Step (2.7) Execute the **restart strategy**.
Step (3) Stopping criteria: If the stopping criterion is satisfied, then stop the procedure and output the feasible solution with the smallest objective function value, otherwise, go to **Step (2)**.

Table 1
Maximum number of function evaluations *MaxFEs* and population size *NP*.

Test functions	*MaxFEs*	*NP*
24 test functions from IEEE CEC2006	2.4E+05	50
18 test functions with 10D from IEEE CEC2010	2.0E+05	80
18 test functions with 30D from IEEE CEC2010	6.0E+05	120

4.2. Experiments on 24 benchmark test functions from IEEE CEC2006

Firstly, ITLBO was tested on 24 test functions from IEEE CEC2006. The performance of ITLBO was compared with those of several COEAs with various EAs [44]: ICTLBO, ETLBO, DE with the feasibility rule, PSO with the feasibility rule, ABC with the feasibility rule, SAMODE, and ATMES.

The experimental results of these eight methods were summarized in Table 2 where "Mean OFV" and "Std Dev" represent the average and standard deviation of objective function values over 25 independent runs, respectively. Note that the experimental results of ICTLBO, ETLBO, DE, PSO, ABC, SAMODE, and ATMES were obtained from [44]. As we know, the optima of test functions from IEEE CEC2006 are provided in [20]. Hence, a run is successful if and only if $f(\vec{x}_{best}) - f(\vec{x}^*) < 10^{-4}$, where $\vec{x}^*$ is the true optimum calculated mathematically and $\vec{x}_{best}$ is the best solution sought by a COEA. Furthermore, the Mean OFV would be marked in bold, if a COEA can be successful over 25 runs consistently on the considered test function. If the COEA cannot achieve a feasible solution consistently, a symbol "*NA*" would be presented. Meanwhile, "–" represents that the results cannot be obtained from the original literature. There are no feasible solutions for g20 and it is extremely difficult to obtain a feasible solution for g22. Besides, most of the literature does not take these two test functions into consideration. Hence, we also exclude these two test functions and mainly focus on the remaining 22 test functions.

As shown in Table 2, ITLBO could achieve the true optima of 16 test functions successfully. ICTLBO could also satisfy the successful condition on 16 test functions. However, ETLBO, DE, PSO, ABC, SAMODE, and ATMES could not achieve the optima of more than 14 test functions. To further compare the eight methods statistically, the multi-problem Wilcoxon's test was implemented. As shown in Table 3, ITLBO performed better than DE, PSO, ABC, and ATMES at a 0.05 significance level, while no significant difference existed between ITLBO and the three algorithms: ICTLBO, ETLBO, and SAMODE. In summary, ITLBO outperformed or at least had competitive performance on test functions from IEEE CEC 2006.

Table 2
Experimental results of ITLBO and other seven selected methods over 25 independent runs on 22 test functions from IEEE CEC2006.

Test function	Criteria	ITLBO	ICTLBO	ETLBO	DE	PSO	ABC	SAMODE	ATMES
g01	Mean OFV	**−15**	**−15**	**−15**	−14.555	−14.71	**−15**	**−15**	**−15**
	Std Dev	0.00E+00	0.00E+00	0.00E+00	–	–	–	0.00E+00	0.00E+00
g02	Mean OFV	−0.80226	−0.799622	**−0.803169**	−0.665	−0.41996	−0.792412	−0.798735	−0.787637
	Std Dev	3.26E−03	5.17E−03	0.00E+00	–	–	–	8.80E−03	1.18E−02
g03	Mean OFV	**−1.0005**	**−1.0005**	−1.0003	−1	0.764813	−1	**−1.0005**	−0.9999
	Std Dev	2.58E−09	1.97E−13	1.40E−04	–	–	–	0.00E+00	1.02E−04
g04	Mean OFV	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**
	Std Dev	3.71E−12	7.43E−12	0.00E+00	–	–	–	0.00E+00	7.43E−12
g05	Mean OFV	**5126.4967**	**5126.4967**	5168.7194	5264.27	5135.973	5185.714	**5126.4967**	5127.7321
	Std Dev	2.78E−12	1.86E−12	5.41E+01	–	–	–	0.00E+00	2.15E+00
g06	Mean OFV	**−6961.8139**	**−6961.8139**	**−6961.8139**	−6954.434	**−6961.8139**	−6961.813	**−6961.8139**	**−6961.8139**
	Std Dev	0.00E+00	3.71E−12	0.00E+00	–	–	–	0.00E+00	3.71E−12
g07	Mean OFV	**24.3062**	**24.3062**	24.31	24.31	32.407	24.473	24.3069	24.31456
	Std Dev	1.51E−05	5.40E−14	7.11E−03	–	–	–	1.59E−03	1.42E−02
g08	Mean OFV	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
	Std Dev	1.42E−17	0.00E+00	0.00E+00	–	–	–	0.00E+00	6.12E−17
g09	Mean OFV	**680.63**	**680.63**	**680.63**	**680.63**	**680.63**	680.64	**680.63**	680.64
	Std Dev	3.36E−13	4.64E−13	0.00E+00	–	–	–	1.16E−05	1.26E−02
g10	Mean OFV	**7049.249**	7049.3128	7143.45	7147.334	7205.5	7224.407	7059.8134	7277.470
	Std Dev	4.29E−05	8.39E−02	1.13E+02	–	–	–	7.86E+00	1.97E+02
g11	Mean OFV	**0.7499**	**0.7499**	0.74998	0.901	**0.7499**	**0.75**	**0.7499**	**0.7499**
	Std Dev	1.13E−16	1.13E−16	7.06E−05	–	–	–	0.00E+00	2.82E−04
g12	Mean OFV	**−1**	**−1**	**−1**	**−1**	−0.998875	**−1**	**−1**	**−1**
	Std Dev	0.00E+00	0.00E+00	0.00E+00	–	–	–	0.00E+00	0.00E+00
g13	Mean OFV	0.054008	0.207885	0.83851	0.872	0.569358	0.968	**0.05392**	0.053959
	Std Dev	3.30E−04	1.92E−01	2.26E−01	–	–	–	1.75E−08	1.06E−05
g14	Mean OFV	**−47.7649**	**−47.7649**	−43.805	−29.2187	−40.871	−40.1071	−47.68115	−47.7279
	Std Dev	3.80E−05	2.10E−08	2.32E+00	1.36E+01	2.29E+00	7.14E+00	4.04E−02	5.05E−02
g15	Mean OFV	**961.72**	**961.72**	962.044	961.7537	965.5154	966.2868	**961.72**	961.7153
	Std Dev	5.80E−13	4.64E−13	4.39E−01	1.22E−01	3.72E+00	3.12E+00	0.00E+00	2.69E−04
g16	Mean OFV	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	−1.902816
	Std Dev	4.53E−16	2.79E−15	0.00E+00	2.34E−16	2.34E−16	2.34E−16	0.00E+00	8.41E−04
g17	Mean OFV	8959.8	8880.595253	8895.7544	8932.0444	8899.4721	8941.9245	**8853.5397**	8896.4008
	Std Dev	3.77E+01	3.69E+01	5.14E+01	4.68E+01	3.79E+01	4.26E+01	1.15E−05	3.27E+01
g18	Mean OFV	**−0.866025**	**−0.866025**	−0.865755	−0.86165	−0.8276	−0.86587	−0.866024	−0.843026
	Std Dev	1.68E−05	1.48E−13	5.09E−04	3.67E−03	1.11E−01	3.37E−04	7.04E−07	6.35E−02
g19	Mean OFV	32.662	32.6570	33.3699	32.768	36.6172	36.0078	32.75734	33.37968
	Std Dev	1.06E−02	1.56E−03	7.87E−02	6.28E−02	2.04E+00	1.83E+00	6.15E−02	3.52E−01
g21	Mean OFV	222.22	**193.72451**	206.118	366.9193	345.6569	275.5436	193.7713	*NA*
	Std Dev	4.84E+01	6.00E−11	2.99E+01	9.13E+01	6.36E+01	6.05E+01	1.96E−02	*NA*
g23	Mean OFV	−256.4	−400.03716	−352.263	−7.2642	−25.9179	−4.3254	−360.8176	*NA*
	Std Dev	1.42E+02	7.44E−02	2.33E+01	2.30E+01	4.30E+01	1.37E+01	1.96E+01	*NA*
g24	Mean OFV	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**
	Std Dev	9.06E−16	5.66E−15	0.00E+00	9.36E−16	9.36E−16	9.36E−16	0.00E+00	0.00E+00

Table 3
Results of the multiple-problem Wilcoxon's test for ITLBO and other seven selected methods on test functions from IEEE CEC2006.

Algorithm	R^+	R^-	p-value	α=0.1	α=0.05
ICTLBO	102.5	128.5	$\geq$0.2	No	No
ETLBO	163.0	90.0	$\geq$0.2	Yes	Yes
DE	224.5	28.5	7.515E-04	Yes	Yes
PSO	202.5	28.5	1.4872E-03	Yes	Yes
ABC	203.5	27.5	1.275E-03	Yes	Yes
SAMODE	137.0	116.0	$\geq$0.2	No	No
ATMES	207.5	46.0	7.444E-03	Yes	Yes

4.3. Experiments on 36 benchmark test functions from IEEE CEC2010

In order to further validate the performance of ITLBO, 36 test functions with 10D (10 dimensions)/30D (30 dimensions) from IEEE CEC2010 were adopted. These test functions are more complicated than those from IEEE CEC2006. Hence, they can evaluate the performance of a COEA adequately. Similarly, six COEAs with various EAs were chosen as competitors, these were, ICTLBO [44], MS-($\mu+\lambda$)-CDE [44], CMODE [40], Co-CLPSO [21], RGA [34], and EABC [25]. The true optima of these 36 test functions are not provided in [22]. Hence, the average and standard deviation of objective function values achieved

Table 4
Experimental results of ITLBO and other six selected methods over 25 independent runs on 18 test functions with 10D from IEEE CEC2010.

Test function	Criteria	ITLBO	ICTLBO	MS-$(\mu+\lambda)$–CDE	CMODE	Co-CLPSO	RGA	EABC
C01	Mean OFV	−7.47E−01	−7.44E−01	−7.42E−01	−7.47E−01	−7.34E−01	−7.21E−01	−7.16E−01
	Std Dev	1.87E−03	5.45E−03	1.17E−02	2.35E−13	1.78E−02	2.62E−02	2.69E−02
C02	Mean OFV	−2.03E+00	−1.72E+00	−2.25E+00	−1.48E+00 Δ	−2.27E+00	1.48E+00	−1.25E−01
	Std Dev	8.14E−02	8.14E−01	1.28E−02	4.88E−01	1.46E−02	4.94E−01	1.58E+00
C03	Mean OFV	0.00E+00	1.41E+09	5.91E+13	2.84E+00	3.55E−01	2.55E+12	2.45E+12
	Std Dev	0.00E+00	6.00E+09	1.22E+14	4.23E+00	1.78E+00	2.36E+07	1.01E+12
C04	Mean OFV	−1.00E−05	−1.00E−05	−1.00E−05	−9.99E−04	−9.34E−06	1.20E−03	8.56E−01
	Std Dev	3.39E−15	1.73E−21	6.77E−13	2.90E−08	1.07E−06	4.20E−03	3.01E+00
C05	Mean OFV	−4.84E+02	−6.82E+01	−4.69E+02	−4.50E+02 Δ	−4.84E+02	5.16E+01	3.65E+02
	Std Dev	1.11E−11	3.56E+01	4.87E+01	1.61E+02	1.96E−02	2.93E+01	1.17E+02
C06	Mean OFV	−5.79E+02	−5.46E+02	−5.09E+02	−5.78E+02	−5.79E+02	−1.18E+02	4.38E+02
	Std Dev	2.39E−04	3.30E+01	1.05E+02	1.60E−02	5.73E−04	−1.18E+02	8.60E+01
C07	Mean OFV	0.00E+00	3.80E−24	0.00E+00	6.69E−15	7.97E−01	6.21E+00	7.16E+01
	Std Dev	0.00E+00	1.50E−23	0.00E+00	8.95E−15	1.63E+00	9.30E+00	5.19E+01
C08	Mean OFV	8.47E+00	1.72E+01	8.01E+00	8.94E+00	6.09E−01	6.21E+00	4.11E+02
	Std Dev	4.09E+00	2.64E+01	4.65E+00	3.98E+00	1.43E+00	9.30E+00	9.36E+02
C09	Mean OFV	0.00E+00	3.56E+05	2.98E+04	2.13E+06 Δ	1.99E+10	2.72E+08	2.02E+12
	Std Dev	0.00E+00	1.00E+06	1.49E+05	1.04E+07	9.97E+10	3.12E+02	1.81E+12
C10	Mean OFV	1.92E−01	1.31E+06	4.55E+01	1.35E+05 Δ	4.97E+10	1.04E+09	1.75E+12
	Std Dev	9.62E−01	6.59E+06	2.89E+01	1.61E+06	2.49E+11	5.84E+04	2.58E+12
C11	Mean OFV	−1.51E−03	−1.52E−03	−4.95E−03	−7.7E−02 Δ	−1.61E−01Δ	7.43E−01	−1.23E+00Δ
	Std Dev	1.30E−05	4.83E−14	1.72E−02	2.85E−02	6.60E−01	2.65E+00	3.04E+00
C12	Mean OFV	−2.39E+01	−5.01E+01	−1.99E−01	−6.14E+02 Δ	−2.34E+00	−1.49E+01	−1.80E+02Δ
	Std Dev	1.14E+02	1.41E+02	2.50E−06	2.74E+02	2.43E+01	1.67E+02	2.76E+02
C13	Mean OFV	−6.52E+01	−6.84E+01	−6.37E+01	−5.79E+01	−6.53E+01	−6.61E+01	−6.57E+01
	Std Dev	1.78E+00	4.42E−14	2.34E+00	4.09E+00	2.58E+00	2.23E+00	2.50E+00
C14	Mean OFV	0.00E+00	3.19E−01	9.53E+01	8.18E−09	3.19E−01	4.93E+04	8.00E+10
	Std Dev	0.00E+00	1.10E+00	4.01E+02	1.64E−08	1.10E+00	3.41E+03	2.37E+11
C15	Mean OFV	3.54E+00	3.73E+01	4.83E+13	1.20E+02	2.99E+00	8.92E+08	2.57E+13
	Std Dev	4.97E+00	9.47E+01	4.24E+13	3.48E+02	3.31E+00	4.12E+04	2.86E+13
C16	Mean OFV	2.27E−01	4.27E−12	8.27E−02	6.82E−05	5.99E−03	8.30E−01	8.35E−02
	Std Dev	3.11E−01	2.13E−11	1.38E−01	1.49E−04	1.33E−02	3.71E−01	9.11E−02
C17	Mean OFV	3.91E−01	7.68E+00	4.93E−34	4.37E−02	3.80E−01	6.65E−01	3.24E+00
	Std Dev	6.71E−01	3.71E+01	1.71E−33	1.12E−01	4.53E−01	1.03E+00	6.83E+00
C18	Mean OFV	0.00E+00	2.59E+00	0.00E+00	5.75E+00	2.32E−01	2.39E+00	3.47E+02
	Std Dev	0.00E+00	1.20E+01	0.00E+00	2.64E+02	9.96E−01	1.93E+00	3.71E+02

Table 5
Results of the multiple-problem Wilcoxon's test for ITLBO andother six selected methods on 18 test functions with 10D from IEEE CEC2010.

Algorithm	R^+	R^-	p-value	α=0.1	α=0.05
ICTLBO	127.0	26.0	1.500E−02	Yes	Yes
MS–$(\mu+\lambda)$–CDE	131.5	39.5	4.572E−02	Yes	Yes
CMODE	154.0	17.0	1.579E−03	Yes	Yes
Co-CLPSO	128.5	42.5	6.324E−02	Yes	No
RGA	160.5	10.5	3.738E−04	Yes	Yes
EABC	152.0	1.0	3.052E−05	Yes	Yes

by a method over 25 independent runs were taken as the comparison criteria. Due to the fact that we can only obtain the average and standard deviation of objective function values of ICTLBO, MS-$(\mu+\lambda)$-CDE, Co-CLPSO, RGA, and EABC from their original papers, we first compared all methods in terms of average values. When a method achieved the best performance on the corresponding test function, the average value was highlighted with a gray background. To test statistical significance, the multi-problem Wilcoxon's test and the Friedman's test were implemented to compare the seven methods concurrently.

In terms of the test functions with 10D, the average and standard deviation of objective function values, the results of the multi-problem Wilcoxon's test, and the results of the multi-problem Friedman's test were summarized in Tables 4, 5, and Fig. 2 (a), respectively. Note that a "Δ" was marked when a method could not achieve a feasible solution at the end of some runs. As shown in Table 4, ITLBO achieved the best performance on 10 test functions. However, the six competitors, i.e., ICTLBO, MS-$(\mu+\lambda)$-CDE, CMODE, Co-CLPSO, RGA, and EABC revealed the best performance on four, five, one, five, zero, and zero test functions, respectively. As shown in Table 5, all R^+ values were bigger than the R^- values. It indicates that ITLBO was superior to its six competitors. The p-values were less than 0.1 and 0.05 in six cases and five cases, respectively. As described in Fig. 2 (a), ITLBO ranked the first in the Friedman's test. In summary, ITLBO was superior to the six competitors on 18 test functions with 10D from IEEE CEC2010.

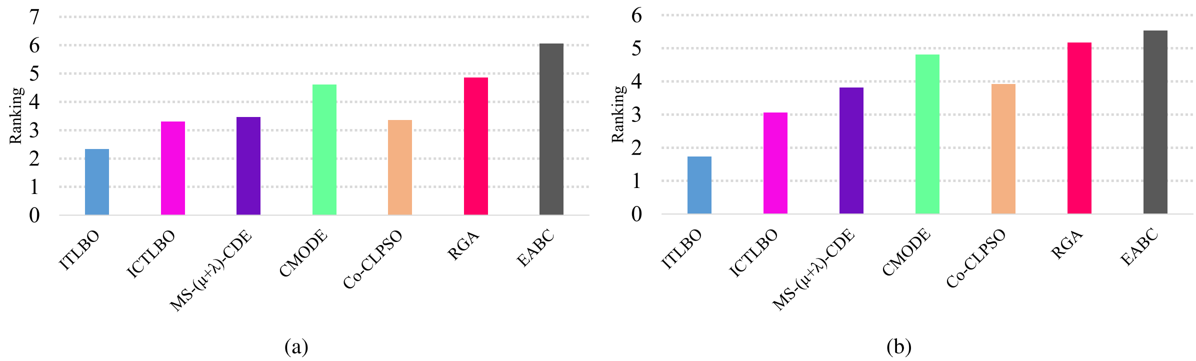

Fig. 2. Ranking of ITLBO and other six selected methods by the Friedman's test on 18 test functions from IEEE CEC2010: (a) 10D, (b) 30D.

Table 6
Experimental results of ITLBO and other six selected methods over 25 independent runs on 18 test functions with 30D from IEEE CEC2010.

Test function	Criteria	ITLBO	ICTLBO	MS-$(\mu+\lambda)$-CDE	CMODE	Co-CLPSO	RGA	EABC
C01	Mean OFV	−8.20E−01	−8.18E−01	−7.35E−01	−8.20E−01	−7.16E−01	−7.72E−01	−7.31E−01
	Std Dev	8.95E−04	2.97E−03	5.32E−02	−8.20E−01	5.03E−02	3.2E−02	4.88E−02
C02	Mean OFV	−2.03E+00	−3.83E−01	−1.27E+00	9.75E−01	−2.20E+00	−3.96E−01	2.56E+00
	Std Dev	7.64E−02	1.72E+00	4.21E−01	6.25E+01	1.93E−01	5.51E−01	9.43E−01
C03	Mean OFV	7.84E+01	2.13E+11	2.21E+13	2.18E+01	3.51E+01 Δ	3.57E+12 Δ	1.07E+13 Δ
	Std Dev	6.31E+01	2.13E+11	2.51E+13	1.25E+01	3.31E+01	2.68E+12	2.21E+12
C04	Mean OFV	1.69E−03	1.59E−01	6.85E+00	6.72E−04	1.13E−01 Δ	1.71E+01 Δ	2.15E+01 Δ
	Std Dev	1.14E−03	3.24E−01	9.94E+00	4.24E−04	5.63E−01	1.36E+01	6.22E+00
C05	Mean OFV	−4.82E+02	−5.98E+01	−3.57E+02	2.77E+02 Δ	−3.12E+02	1.87E+02	3.72E+02
	Std Dev	1.73E+00	8.86E+00	8.38E+01	2.03E+02	8.83E+01	7.90E+01	7.894E+01
C06	Mean OFV	−5.30E+02	−4.64E+02	−3.23+02	−4.96E+02 Δ	−2.45E+02	−1.97E+02 Δ	4.74E+02
	Std Dev	4.80E−01	9.20E+01	1.28E+02	2.15E+02	3.95E+01	9.39E+01	6.30E+01
C07	Mean OFV	1.59E−01	2.88E+01	1.59E−01	5.24E−05	1.12E+00	3.65E+01	1.33E+02
	Std Dev	7.97E−01	4.68E+01	7.97E−01	5.89E−05	1.83E+00	2.27E+01	2.06E+02
C08	Mean OFV	1.14E+01	1.01E+02	1.46E+03	3.68E−01	4.75E+01	2.88E+10	1.50E+02
	Std Dev	2.79E+01	1.22E+02	3.16E+03	2.62E−01	1.13E+02	4.23E+07	7.15E+01
C09	Mean OFV	2.86E+00	1.01E+07	2.51E+09	1.72E+13 Δ	1.48E+08	2.72E+08	1.61E+13
	Std Dev	1.43E+01	3.52E+07	3.75E+09	1.07E+13	2.45E+08	3.12E+04	9.29E+12
C10	Mean OFV	3.29E+01	6.01E+09	1.41E+09	1.60E+13 Δ	1.40E+09	6.99E+08	1.50E+13
	Std Dev	1.41E+01	2.31E+10	1.91E+09	7.00E+12	5.84E+09	6.13E+03	9.77E+12
C11	Mean OFV	−3.86E−04	−3.64E−04	2.69E−03	9.5E−03 Δ	2.82E−02 Δ	−1.68E−01 Δ	−5.89E−01 Δ
	Std Dev	1.14E−05	4.98E−05	7.12E−03	9.7E−03	3.21E−02	5.72E−01	6.49E−01
C12	Mean OFV	−1.98E−01	−1.99E−01	5.80E−01	−3.46E+00 Δ	−1.99E−01 Δ	−1.25E+02 Δ	5.07E+01
	Std Dev	2.39E−03	6.16E−05	3.10E+00	7.35E+02	1.18E−04	1.54E+02	3.70E+02
C13	Mean OFV	−5.05E+01	−6.81E+01	−5.96E+01	−3.89E+01	−6.08E+01	−6.34E+01	−6.49E+01
	Std Dev	1.18E+00	7.78E−01	2.60E+00	2.17E+00	1.12E+00	1.23E+00	1.38E+00
C14	Mean OFV	4.78E−01	8.02E+00	6.79E+05	9.31E+00	1.28E+00	8.78E+07	9.95E+03
	Std Dev	1.32E+00	8.69E+00	3.39E+06	2.46E+00	1.90E+00	3.13E+03	1.92E+04
C15	Mean OFV	2.38E+01	2.91E+01	1.20E+13	1.51E+13	5.11E+01	7.99E+09	3.79E+13
	Std Dev	2.51E+01	3.63E+01	2.20E+13	8.26E+12	9.18E+01	5.13E+04	3.44E+13
C16	Mean OFV	0.00E+00	0.00E+00	6.92E−03	6.30E−02	5.24E−16	1.05E+00	8.21E−01
	Std Dev	0.00E+00	0.00E+00	2.86E−02	2.72E−02	4.67E−16	3.99E−02	2.57E−01
C17	Mean OFV	9.65E−01	3.29E+01	2.65E−01	3.12E+02 Δ	1.39E+00	5.49E+01	2.68E+01
	Std Dev	1.73E+00	1.35E+02	2.32E−01	2.75E+02	4.26E+00	1.98E+01	1.63E+01
C18	Mean OFV	9.07E−17	8.82E−04	7.45E−01	7.36E+03	1.09E+01	4.40E+01	2.93E+02
	Std Dev	3.18E−16	3.22E−03	1.89E+00	3.12E+03	3.72E+01	1.61E+01	3.53E+02

Table 7
Results of the multiple-problem Wilcoxon's test for ITLBO and other six selected methods on 18 test functions with 30D from IEEE CEC2010.

Algorithm	R^+	R^-	p-value	α=0.1	α=0.05
ICTLBO	151.5	19.5	2.571E−03	Yes	Yes
MS-$(\mu+\lambda)$-CDE	148.0	5.0	1.526E−04	Yes	Yes
CMODE	141.0	12.0	1.068E−03	Yes	Yes
Co-CLPSO	160.5	10.5	3.738E−04	Yes	Yes
RGA	164.5	6.5	1.259E−04	Yes	Yes
EABC	163.0	8.0	1.907E−04	Yes	Yes

Table 8
Results of ITLBO and ITLBO-WoR on three test functions from IEEE CEC2010.

Test function	ITLBO feasible rate	ITLBO-WoR feasible rate
C11 with 10D	100%	28%
C12 with 10D	100%	0%
C11 with 30D	100%	92%

Table 9
Experimental results of ITLBO with different *K* over 25 independent runs on 18 test functions with 30D from IEEE CEC2010.

Test function	Criteria	$K=10$	$K=7$	$K=8$	$K=9$	$K=11$	$K=12$	$K=13$
C01	Mean OFV (feasible rate)	−8.20E−01	−8.20E−01 ≈	−8.20E−01 ≈	−8.19E−01 ≈	−8.20E−01 ≈	−8.20E−01 ≈	−8.20E−01 ≈
	Std Dev	8.95E−04	1.23E−03	1.86E−03	1.36E−03	1.19E−03	9.48E−04	1.56E−03
C02	Mean OFV (feasible rate)	−2.03E+00	−2.01E+00 ≈	−2.04E+00 ≈	−2.04E+00 ≈	−2.10E+00 ≈	−2.05E+00 ≈	−2.03E+00 ≈
	Std Dev	7.64E−02	6.34E−02	4.86E−02	6.46E−02	7.01E−02	5.31E−02	7.24E−02
C03	Mean OFV (feasible rate)	7.84E+01	5.49E+01 +	5.75E+01 +	5.21E+01 +	5.16E+01 +	8.15E+01 ≈	8.15E+01 ≈
	Std Dev	6.31E+01	3.89E+01	4.90E+01	4.46E+01	3.57E+01	5.63E+01	5.36E+01
C04	Mean OFV (feasible rate)	1.69E−03	68% −	1.24E−03 ≈	56% −	52% −	4.42E−03 ≈	76% −
	Std Dev	1.14E−03	*NA*	1.63E−03	*NA*	*NA*	1.20E−02	*NA*
C05	Mean OFV (feasible rate)	−4.82E+02	92% −	−4.82E+02 ≈	−4.82E+02 ≈	20% −	−4.81E+02 ≈	96% −
	Std Dev	1.73E+00	*NA*	2.02E+00	1.42E+00	*NA*	2.07E+00	*NA*
C06	Mean OFV(feasible rate)	−5.30E+02	96% −	−5.31E+02 ≈	−5.30E+02 ≈	80% −	−5.31E+02 ≈	−5.31E+02 ≈
	Std Dev	4.80E−01	*NA*	1.00E−01	3.74E−01	*NA*	1.25E−01	3.10E−01
C07	Mean OFV (feasible rate)	1.59E−01	4.06E−14 +	4.10E−14 +	1.59E−08 +	9.58E−01 ≈	1.59E−01 ≈	3.19E−01 ≈
	Std Dev	7.97E−01	2.02E−13	2.03E−13	7.73E−08	1.74E+00	7.97E−01	1.10E+00
C08	Mean OFV (feasible rate)	1.14E+01	5.85E+00 +	5.04E+00 +	5.61E+01 −	6.14E+01 −	7.80E+00 +	8.58E+01 −
	Std Dev	2.79E+01	2.02E+01	2.35E+01	1.39E+02	9.83E+01	2.66E+01	3.98E+02
C09	Mean OFV (feasible rate)	2.86E+00	5.76E+00 ≈	7.38E+00 ≈	8.97E+00 ≈	2.21E+01 −	6.08E+00 ≈	2.25E+01 −
	Std Dev	1.43E+01	2.79E+01	3.53E+01	2.86E+01	6.05E+01	3.04E+01	7.68E+01
C10	Mean OFV (feasible rate)	3.29E+01	4.03E+01 ≈	3.78E+01 ≈	4.92E+01 ≈	1.57E+02 −	3.99E+01 ≈	4.91E+01 ≈
	Std Dev	1.41E+01	4.03E+01	2.40E+01	8.87E+01	2.98E+02	4.13E+01	5.18E+01
C11	Mean OFV (feasible rate)	−3.86E−04	20% −	88% −	28% −	28% −	92% −	36% −
	Std Dev	1.14E−05	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*
C12	Mean OFV (feasible rate)	−1.98E−01	36% −	−1.63E−01 ≈	36% −	0% −	−1.75E−01 ≈	80% −
	Std Dev	2.39E−03	*NA*	1.79E−01	*NA*	*NA*	8.18E−02	*NA*
C13	Mean OFV (feasible rate)	−5.05E+01	−5.02E+01 ≈	−5.03E+01 ≈	−5.08E+01 ≈	−5.35E+01 ≈	−5.09E+01 ≈	−5.09E+01 ≈
	Std Dev	1.18E+00	8.27E−01	1.16E+00	1.03E+00	1.25E+00	1.24E+00	1.10E+00
C14	Mean OFV (feasible rate)	4.78E−01	4.78E−01 ≈	4.78E−01 ≈	6.38E−01 ≈	3.21E−01 ≈	9.57E−01 ≈	3.19E−01 ≈
	Std Dev	1.32E+00	1.32E+00	1.32E+00	1.49E+00	1.10E+00	1.74E+00	1.10E+00
C15	Mean OFV (feasible rate)	2.38E+01	2.07E+01 ≈	1.68E+01 ≈	2.66E+01 ≈	2.41E+01 ≈	1.97E+01 ≈	1.92E+01 ≈
	Std Dev	2.51E+01	1.33E+01	8.62E+00	2.85E+01	2.60E+01	1.76E+01	6.37E+00
C16	Mean OFV (feasible rate)	0.00E+00	0.00E+00 ≈	0.00E+00 ≈	0.00E+00 ≈	0.00E+00 ≈	0.00E+00 ≈	0.00E+00 ≈
	Std Dev	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
C17	Mean OFV (feasible rate)	9.65E−01	96% −	8.55E+01 −	3.73E−01 ≈	96% −	96% −	4.99E−01 ≈
	Std Dev	1.73E+00	*NA*	4.25E+02	1.32E+00	*NA*	*NA*	1.19E+00
C18	Mean OFV (feasible rate)	9.07E−17	5.80E−21 +	4.91E−18 ≈	5.68E−20 ≈	3.10E−14 −	6.05E−17 ≈	1.48E−19 ≈
	Std Dev	3.18E−16	2.33E−20	1.94E−17	2.76E−19	1.28E−13	1.45E−16	4.89E−19
	+	/	**4**	**3**	**2**	**1**	**1**	**0**
	−	/	**6**	**2**	**4**	**10**	**2**	**6**
	≈	/	**8**	**13**	**12**	**7**	**15**	**12**

In the case of the test functions with 30D, corresponding results were reported in Tables 6, 7, and Fig. 2 (b), respectively. As shown in Table 6, ITLBO was not worse than all other six competitors on 10 test functions. However, ICTLBO, MS-$(\mu+\lambda)$-CDE, CMODE, Co-CLPSO, RGA, and EABC revealed the best result on three, one, five, one, zero, and zero test functions, respectively. The R^+ and R^- values in Table 7 reflected that ITLBO was better than its competitors. Furthermore, the *p*-values were less than 0.05 in all cases. As shown in Fig. 2 (b), ITLBO ranked the first in the Friedman's test. In summary, ITLBO was superior to the six competitors on 18 test functions with 30D from IEEE CEC2010.

Taking all the above experimental results into consideration, ITLBO has the outstanding performance when tackling COPs.

4.4. Effectiveness of the restart strategy

As described in Section 3.3, the restart strategy plays a significant role in ITLBO. From the analyses, it is beneficial to tackling COPs with complicated constraints. In order to validate this statement experimentally, a method called ITLBO-WoR was implemented by removing the restart strategy from ITLBO. Then 36 test functions from IEEE CEC2010 were selected to evaluate ITLBO and ITLBO-WoR. Their performance were compared based on the feasible rate, that was, percentage of runs where at least one feasible solution was found. The experimental results of those test functions, in which at least one of the compared methods could not achieve a feasible solution at the end of some runs, were summarized in Table 8.

As shown in Table 8, ITLBO-WoR failed to achieve a feasible solution consistently on 3 test functions, which were, C11 with 10D, C12 with 10D, and C11 with 30D. More specifically, ITLBO-WoR could find a feasible solution of C11 with 10D, C12 with 10D, and C11 with 30D on only seven, zero, and 23 runs, respectively. However, ITLBO, which was aided by the restart strategy, could find a feasible solution consistently on these three test functions. The experimental results reflected that the proposed restart strategy can improve the ability to find a feasible solution.

4.5. Sensitivity of parameter K

As described in Section 3.1, when balancing diversity and convergence, the number of subpopulations K, is critical. Intuitively, a bigger K would result in more subpopulations and then introduce more diversity. In addition, too much diversity would disturb the process of finding the optimum while too little diversity would run the high risk of premature convergence. Hence, parameter K should be chosen properly to achieve the tradeoff between diversity and convergence. In view of this, parameter K was decided experimentally in this sub-section. For this purpose, seven ITLBO variants with different K values: $K = 7$, $K = 8$, $K = 9$, $K = 10$, $K = 11$, $K = 12$, and $K = 13$ were implemented. These seven variants were evaluated on the 18 test functions with 30D from IEEE CEC2010. Their performance was compared by the Wilcoxon's rank sum test at a 0.05 significance level.[1] When a method could not find a feasible solution on the considered test function over 25 runs consistently, the average value and standard deviation were replaced by the feasible rate and *NA*, respectively. Additionally, "$+$", "$-$", and "$\approx$" represent that a method was better than, worse than, and similar to ITLBO with $K = 10$, respectively.

As shown in Table 9, ITLBO with $K = 10$ outperformed those with $K = 7$, $K = 9$, $K = 11$, $K = 12$, and $K = 13$ on six, four, 10, two, and six test functions, respectively. However, ITLBO with $K = 7$, $K = 9$, $K = 11$, $K = 12$, and $K = 13$ revealed better performance than that with $K = 10$ on four, two, one, one, and zero functions, respectively. Hence, ITLBO with $K = 10$ was better than those with $K = 7$, $K = 9$, $K = 11$, $K = 12$, and $K = 13$ as a whole. It seems that ITLBO with $K = 8$ was better than that with $K = 10$. However, ITLBO with $K = 8$ cannot find a feasible solution of C11 consistently. As we know, seeking a feasible solution is vital to a COEA. Taking all of this into consideration, $K = 10$ was chosen in this paper. It is interesting to find that it seems to exist no explicit rules to tune K. It may be that the performance of ITLBO would be impaired if the population could not be divided into K subpopulations equally.

Remark 2. for the sake of paper length, some further discussions are referred to the supplementary file.

5. Conclusions

This paper has extended the outstanding global optimizer, i.e., TLBO, to solve COPs. An efficient subpopulation based teacher phase and a ranking-differential-vector-based learner phase have been proposed to balance diversity and convergence. A dynamic weighted sum has also been formulated to achieve the tradeoff between constraints and objective function. Furthermore, a simple yet powerful restart strategy has been designed to cope with complicated constraints. Extensive experiments on two benchmark test suites have validated that: (1) ITLBO outperformed or at least had competitive performance against two constrained TLBOs as well as some other state-of-the-art COEAs, and (2) each component of ITLBO was effective and significant.

Compared with the basic TLBO, ITLBO involves two algorithm-specific parameters, which are, the number of subpopulations K, and the threshold of the restart strategy μ. Due to the presentation of the sorting procedure, the computation time complexity of ITLBO is slightly higher than that of TLBO. In the future, effort will be put on reducing algorithm-specific parameters as well as improving the efficiency of ITLBO. Furthermore, the two tradeoffs will be utilized to extend other global optimizers to the constrained ones. To be specific, a diversity operator and a convergence operator need to be designed, respectively. When these two operators are designed, the tradeoff between constraints and objective function should be considered. Furthermore, ITLBO will be further enhanced by DE [7] or group counseling optimizer [5] to solve real-life engineering problems such as wireless sensor networks [19], filter design problems [8], antenna array design problems [4], and unit commitment problems [37].

Acknowledgment

This work was supported in part by the General Research Fund Project from Research Grant Council of Hong Kong SAR under Grant City U: 11205615 and in part by the Project from the City University of Hong Kong under Grant 7004666.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2018.04.083.

[1] The Wilcoxon's rank sum test is implemented by the MATLAB function "ranksum".

References

[1] S. Banerjee, D. Maity, C.K. Chanda, Teaching learning based optimization for economic load dispatch problem considering valve point loading effect, Int. J. Electr. Power Energy Syst. 73 (2015) 456–464.
[2] A. Baykasoğlu, A. Hamzadayi, S.Y. Köse, Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: flow shop and job shop scheduling cases, Inf. Sci. (Ny) 276 (2014) 204–218.
[3] K. Bhattacharjee, A. Bhattacharya, S.H.N. Dey, Teaching-learning-based optimization for different economic dispatch problems, Sc. Iranica. Trans. D Comput. Sci. Eng. Electr. 21 (3) (2014) 870.
[4] S. Biswas, D. Bose, S. Das, S. Kundu, Decomposition-based evolutionary multi-objective optimization approach to the design of concentric circular antenna arrays, Prog. Electromagn. Res. B 52 (2013) 185–205.
[5] S. Biswas, M.A. Eita, S. Das, A.V. Vasilakos, Evaluating the performance of group counseling optimizer on CEC 2014 problems for computational expensive optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 1076–1083.
[6] S. Biswas, S. Kundu, D. Bose, S. Das, Cooperative co-evolutionary teaching-learning based algorithm with a modified exploration strategy for large scale global optimization, in: Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing, Springer, 2012, pp. 467–475.
[7] S. Biswas, S. Kundu, S. Das, A.V. Vasilakos, Teaching and learning best differential evolution with self adaptation for real parameter optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2013, pp. 1115–1122.
[8] D. Bose, S. Biswas, A.V. Vasilakos, S. Laha, Optimal filter design using an improved artificial bee colony algorithm, Inf. Sci. (Ny) 281 (2014) 443–461.
[9] A. Caponio, G.L. Cascella, F. Neri, N. Salvatore, M. Sumner, A fast adaptive memetic algorithm for online and offline control design of PMSM drives, IEEE Trans. Syst. Man Cybern. Part B 37 (1) (2007) 28–41.
[10] A. Caponio, F. Neri, V. Tirronen, Super-fit control adaptation in memetic differential evolution frameworks, Soft Comput. 13 (8–9) (2009) 811.
[11] D. Chen, F. Zou, Z. Li, J. Wang, S. Li, An improved teaching–learning-based optimization algorithm for solving global optimization problem, Inf. Sci. (Ny) 297 (2015) 171–190.
[12] J. Cheng, G. Zhang, F. Neri, Enhancing distributed differential evolution with multicultural migration for global numerical optimization, Inf. Sci. (Ny) 247 (2013) 72–93.
[13] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. (CSUR) 45 (3) (2013) 35.
[14] S. Das, S. Biswas, S. Kundu, Synergizing fitness learning with proximity-based food source selection in artificial bee colony algorithm for numerical optimization, Appl. Soft. Comput. 13 (12) (2013) 4676–4694.
[15] S. Das, S. Biswas, B.K. Panigrahi, S. Kundu, D. Basu, A spatially informative optic flow model of bee colony with saccadic flight strategy for global optimization, IEEE Trans. Cybern. 44 (10) (2014) 1884–1897.
[16] S. Degertekin, M. Hayalioglu, Sizing truss structures using teaching-learning-based optimization, Comput. Struct. 119 (2013) 177–188.
[17] W. Gong, Z. Cai, D. Liang, Adaptive ranking mutation operator based differential evolution for constrained optimization, IEEE Trans. Cybern. 45 (4) (2015) 716–727.
[18] G. Iacca, F. Caraffini, F. Neri, Multi-strategy coevolving aging particle optimization, Int. J. Neural. Syst. 24 (01) (2014) 1450008.
[19] S. Kundu, S. Das, A.V. Vasilakos, S. Biswas, A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks, Soft Comput. 19 (3) (2015) 637–659.
[20] J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, C.C. Coello, K. Deb, Problem definitions and evaluation criteria for the CEC special session on constrained real-parameter optimization, J. Appl. Mech. 41 (8) (2006).
[21] J.J. Liang, Z. Shang, Z. Li, Coevolutionary comprehensive learning particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–8.
[22] R. Mallipeddi, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization, 24, Nanyang Technological University, Singapore, 2010.
[23] B. Mandal, P.K. Roy, Optimal reactive power dispatch using quasi-oppositional teaching learning based optimization, Int. J. Electr. Power Energy Syst. 53 (2013) 123–134.
[24] E. Mezura-Montes, C.A.C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm Evol. Comput. 1 (4) (2011) 173–194.
[25] E. Mezura-Montes, R.E. Velez-Koeppel, Elitist artificial bee colony for constrained real-parameter optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–8.
[26] F. Neri, C. Cotta, Memetic algorithms and memetic computing optimization: a literature review, Swarm Evol. Comput. 2 (2012) 1–14.
[27] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106.
[28] F. Neri, J. Toivanen, G.L. Cascella, Y.-S. Ong, An adaptive multimeme algorithm for designing HIV multidrug therapies, IEEE/ACM Trans. Comput. Biol. Bioinf. 4 (2) (2007) 264–278.
[29] T. Niknam, R. Azizipanah-Abarghooee, J. Aghaei, A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch, IEEE Trans. Power Syst. 28 (2) (2013) 749–763.
[30] A. Pan, L. Wang, W. Guo, Q. Wu, A diversity enhanced multiobjective particle swarm optimization, Inf. Sci. (Ny) 436 (2018) 441–465.
[31] I. Poikolainen, F. Neri, F. Caraffini, Cluster-based population initialization for differential evolution frameworks, Inf. Sci. (Ny) 297 (2015) 216–235.
[32] R. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (4) (2012) 535–560.
[33] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci. (Ny) 183 (1) (2012) 1–15.
[34] A. Saha, R. Datta, K. Deb, Hybrid gradient projection based genetic algorithms for constrained optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–8.
[35] W. Shao, D. Pi, Z. Shao, An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem, Appl. Soft. Comput. 61 (2017) 193–210.
[36] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–9.
[37] A. Trivedi, D. Srinivasan, S. Biswas, T. Reindl, A genetic algorithm–differential evolution based hybrid framework: case study on unit commitment scheduling problem, Inf. Sci. (Ny) 354 (2016) 275–300.
[38] G. Waghmare, Comments on a note on teaching–learning-based optimization algorithm, Inf. Sci. (Ny) 229 (2013) 159–169.
[39] B.-C. Wang, H.-X. Li, J.-P. Li, Y. Wang, Composite differential evolution for constrained evolutionary optimization, in: Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, doi:10.1109/TSMC.2018.2807785.
[40] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Trans. Evol. Comput. 16 (1) (2012) 117–134.
[41] Y. Wang, B.-C. Wang, H.-X. Li, G.G. Yen, Incorporating objective function information into the feasibility rule for constrained evolutionary optimization, IEEE Trans. Cybern. 46 (12) (2016) 2938–2952.
[42] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P. Suganthan, Ensemble of differential evolution variants, Inf. Sci. (Ny) 423 (2018) 172–186.
[43] K. Yu, X. Chen, X. Wang, Z. Wang, Parameters identification of photovoltaic models using self-adaptive teaching-learning-based optimization, Energy Convers. Manage. 145 (2017) 233–246.
[44] K. Yu, X. Wang, Z. Wang, Constrained optimization based on improved teaching–learning-based optimization algorithm, Inf. Sci. (Ny) 352 (2016) 61–78.

[45] K. Yu, X. Wang, Z. Wang, An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems, J. Intell. Manuf. 27 (4) (2016) 831–843.
[46] K. Yu, L. While, M. Reynolds, X. Wang, Z. Wang, Cyclic scheduling for an ethylene cracking furnace system using diversity learning teaching-learning-based optimization, Comput. Chem. Eng. 99 (2017) 314–324.
[47] K.Z. Zamli, F. Din, S. Baharom, B.S. Ahmed, Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites, Eng. Appl. Artif. Intell. 59 (2017) 35–50.
[48] Q. Zhang, W. Liu, X. Meng, B. Yang, A.V. Vasilakos, Vector coevolving particle swarm optimization algorithm, Inf. Sci. (Ny) 394 (2017) 273–298.
[49] L.M. Zheng, S.X. Zhang, K.S. Tang, S.Y. Zheng, Differential evolution powered by collective information, Inf. Sci. (Ny) 399 (2017) 13–29.
[50] F. Zou, L. Wang, X. Hei, D. Chen, D. Yang, Teaching–learning-based optimization with dynamic group strategy for global optimization, Inf. Sci. (Ny) 273 (2014) 112–131.