Journal of Applied Science and Engineering, Vol. 21, No. 3, pp. 331–342 (2018) *DOI: 10.6180/jase.201809_21(3).0004*

MTLBO: A Multi-objective Multi-course Teaching-Learning-based Optimization Algorithm

Wenwen Lin[1]*, Lei Wang[1], Guangdong Tian[2] and Yuejun Zhang[1]

[1]*Faculty of Mechanical Engineering & Mechanics, Ningbo University, Ningbo 315211, P.R. China*
[2]*Transportation College, Jilin University, Changchun 130022, P.R. China*

Abstract

This paper presents a multi-objective evolutionary algorithm, which is called multi-course teaching-learning-based optimization algorithm. It consists of four phases: teacher, leaner, teacher training, and exchange student phases. Three novel modifications are introduced into the basic teaching-learning-based optimization algorithm. It helps diversify evolutionary population, enhance stochastic search ability, and avoid premature convergence. Eight widely used benchmark problems are employed in order to investigate the performance of the proposed algorithm. The experimental results indicate that it finds better Pareto optimal solutions than some other state-of-the-art algorithms for the majority of problems.

Key Words: Teaching-learning-based Optimization, Multi-objective Optimization, Evolutionary Algorithm, Heuristic Algorithm

1. Introduction

Two or more conflicting objective functions are concerned in multi-objective problems (MOPs). Unlike single objective optimization problems, many optimal solutions, which are commonly known as Pareto-optimal solutions, are found in a MOP. MOPs are frequently encountered in real world, such as production design [1], scheduling problems [2], and supply chain optimization problems [3].

The goal of multi-objective optimization methods is to obtain a set of Pareto-optimal solutions. Multi-objective evolutionary algorithms (MOEAs) tend to generate approximate Pareto-optimal solutions in reasonable time. Over the past two decades, researchers proposed a number of representative MOEAs: niched Pareto genetic algorithm [4], strength Pareto evolutionary algorithm [5], Pareto archived evolution strategy [6], non-dominated sorting genetic algorithm (NSGA-Π) [7], improved Pareto archived evolution strategy [8], micro-genetic algorithm [9], multi-objective evolutionary algorithm based on decomposition [10], A regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) [11], and multi-objective self-adaptive differential evolution algorithm with objective-wise learning strategies [12]. Recently, more evolutionary algorithms were proposed, such as imperialist competition algorithm (ICA) [13], particle swarm optimization (PSO) [14], artificial bee colony algorithm [15], teaching-learning-based optimization algorithm (TLBO) [16]. Necessary additional operators were introduced into those newly proposed algorithms, and then the modified algorithms were applicable for MOPs. For example, Hedayatzadeh et al. developed a multi-objective artificial bee colony algorithm (MOABC), in which a grid-based approach was adopted to maintain and access the Pareto front [17]. Ali and Khan proposed an attributed multi-objective comprehensive learning particle swarm optimizer (A-MOCLPSO) [18]. Zou et al. introduced a multi-objective teaching-learning-based optimization algorithm (MOTLBO) [19]. The

*Corresponding author. E-mail: linwenwen@nbu.edu.cn

experimental results showed that those new natural-inspired MOEAs outperformed the representative ones.

These aforementioned MOEAs proved their effectiveness and efficiency in solving MOPs. However, Wolpert and Macready analyzed "no free lunch" for optimization, and proved that any specific algorithm cannot override the others for all classes of problems [20]. It is the motivation of researchers to develop new MOEAs. The aim of this work is to propose a multi-course teaching-learning-based optimization algorithm (MTLBO) for MOPs. Compared with the existing work, we have three outstanding contributions: (1) A removing dominated solution methodology is employed in MTLBO in order to accelerate non-dominated set construction process; (2) A multiple teacher mechanism is employed in order to keep the diversity of evolutionary population and improve solutions' performance comprehensively; (3) An adaptive teaching factor and two more phases are introduced in order to enhance the algorithm's stochastic search ability and avoid premature convergence. The performance of MTLBO is validated via widely used numerical test functions. The performance analysis indicates that it provides better results than representative MOEAs and new naturally-inspired MOEAs.

The remainder of this paper is organized as follows. Some preliminary concepts are introduced in section 2, while basic TLBO is briefly described in section 3. Section 4 gives MTLBO. Finally, sections 4 and 5 outline the computation results and the conclusions respectively.

2. Problem Statement

In order to conveniently describe our algorithm, we present the following preliminaries.

2.1 Multi-objective Optimization

In general, an n-objective optimization model is defined as follows:

$$\text{minimize:} f_1(x), f_2(x), \cdots, f_n(x)$$
$$\text{subject to:} x \in S$$

where n objectives i.e. $f_1(x)$, $f_2(x)$, …, $f_n(x)$ are minimized, x is a feasible solution, and S is the feasible solution space. For the sake of concise notation, we use $f(x)$ to represent the vector of those objective functions of solution x. Solution x_1 is said to dominate solution x_2, when $f_i(x_1) < f_i(x_2)$, $\forall i = \{1, 2, \ldots, n\}$, and $\exists i \in \{1, 2, \ldots, n\}$ such that $f_i(x_1) < f_i(x_2)$. When x_1 and x_2 do not dominate each other, we say that x_1 is Pareto-equivalent to x_2. The true Pareto-optimal Set is a set of solutions that are Pareto-equivalent to others in the set and cannot be dominated by any other solutions in S. Solutions in such a set are called non-dominated solutions. For a MOP, the optimization goal is to find the entire Pareto-optimal set or its subset.

2.2 Teaching–learning-based optimization

The basic TLBO was first proposed by Rao et al. in 2011 [21]. As a naturally inspired population-based evolutionary algorithm, it mimics the teaching and learning activities in a class room. It consists of two phases: teacher and learner phases. Evolutionary population is considered as a class of students, and the best student among the class is selected as the teacher. Students learn from their teacher in the teacher phase, and then learn through interactions with their peers in the learner phase. Researchers have used it to solve many engineering optimization problems since its conception, such as constrained mechanical design optimization problems [21], data clustering problems [22], and process parameter optimization problems [23]. We next give its two steps in detail.

2.2.1 Teacher Phase

Generally, students would gain more knowledge when their teacher has better performance. Thus, the student with the best objective function, which must be the most knowledgeable one in the class, is selected as a teacher. In this phase, students improve themselves by learning from their teacher. However, some students may have high learning abilities, while others may have tough time to learn. Besides, in order to cover more students, the teacher should arrange teaching schedules according to the average level of the class. Let M_i denotes the average level of a class and T_i the teacher at any iteration i. T_i is designated as the ideal new mean M_{new} at iteration i+1, and M_i is moved towards M_{new}. Then, each student is updated as follows:

$$X_{new,i} = X_{old,i} + X_{\Delta i} \tag{1}$$

where $X_{\Delta i}$ characterizes the difference between the current mean M_i and new mean M_{new}. $X_{\Delta i}$ is given by as follows:

$$X_{\Delta i} = r_i(M_{new} - T_F M_i) \tag{2}$$

where r_i is a random number between 0 and 1, and T_F is a teaching factor that mimics the influence of teacher on students. T_F is set to 1 or 2, and is determined as follows:

$$T_F = \text{round}[1 + \text{rand}(0,1)] \tag{3}$$

If $X_{new,i}$ owns a better objective function than $X_{old,i}$, then $X_{new,i}$ is accepted while $X_{old,i}$ is discarded.

2.2.2 Learner Phase

In this phase, students improve themselves through interactions with their peers. For each student X_i of the class, we have the following steps:

Step 1. Randomly select one student X_j that is different from X_i.

$$X_{new,j} = X_j + r_i \times (X_i - X_j) \tag{4}$$

$$X_{new,i} = X_i + r_i \times (X_j - X_i) \tag{5}$$

Step 2. If X_i has a better objective function, X_j is updated by Eq. (4); otherwise, X_i is updated by Eq. (5).

Step 3. If $X_{new,i}$ owns a better objective function than $X_{old,i}$, $X_{new,i}$ is accepted while $X_{old,i}$ is discarded.

3. Multi-course TLBO

Researchers have TLBO modified to solve MOPs. For example, Zou et al. proposed a multi-objective TLBO [19], where a fast non-dominated sorting technique and crowding distance assignment method were adopted to obtain approximate Pareto-optimal front of numerical MOPs. Rao and Patel modified the basic TLBO algorithm for multi-objective optimization of the heat exchangers [1]. All objectives were assigned with different weighting factors, and were combined into a single normalized objective function. This paper proposes an MTLBO to deal with MOPs. Its four phases are introduced next.

3.1 Non-dominated Set Construction

A removing dominated solution (RDS) method is put forward to construct non-dominated set. This method only identifies the first non-dominated front, while the fast non-dominated sorting technique sorts solutions in evolutionary population to different non-dominated fronts. Thus, the proposed method has lower computational complexity. The non-dominated set has its size is empty initially. For each solution X_i in the population, following manipulation is applied: When the non-dominated set is empty, X_i is added into the set. Otherwise, X_i is compared with each solution X_j in the non-dominated set: If X_i dominates X_j, X_j is abandoned; if X_i cannot be dominated by any solution in the non-dominated set, we add it into the set.

The crowding distance of each solution X_i in the non-dominated set is calculated by using a crowding distance assignment method. Assume that extreme solutions are solutions with the smallest or largest function values. Figure 1 presents an example of crowding distance calculation for a two-objective optimization problem. Solid dots are non-dominated solutions. Among them, A and E are extreme solutions, and others are intermediate ones. Let i_{distance} denotes the crowding distance of any solution X_i in the non-dominated set. For intermediate solutions, i_{distance} is equal to the sum of the absolute difference between two function values of its adjacent solutions. For example, B and D are adjacent solutions of C. Therefore, $i_{distance}$ of point C is equal to the sum of d_1 and d_2. Finally, assign infinite values to $i_{distance}$ of extreme solutions. For details of crowding distance assignment method, the readers can refer to reference [24].

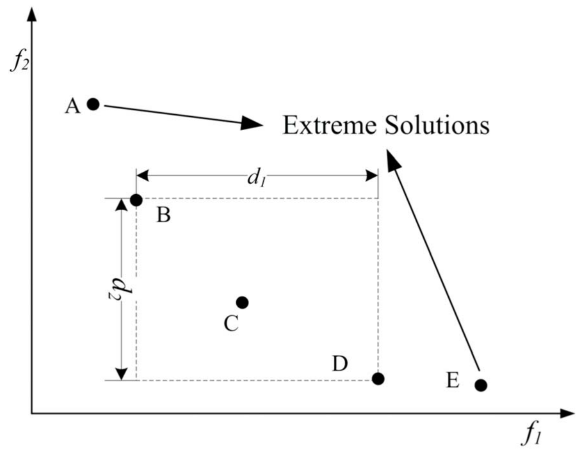

Figure 1. An example of crowding distance calculation.

3.2 Teacher Phase

In general, the basic courses of students in middle school are English, Mathematics, Physics, and Chemistry, and so on. Each course requires one teacher. Multiple courses are similar to multiple objectives in MOPs, and course scores are similar to objective function values. The solution (i.e. extreme solution) with the best function values of the ith objective is considered as the teachers of the ith course. Besides extreme solutions, solutions with the largest crowding distance among the rest solutions in the non-dominated set are considered as teachers to keep the diversity of population and improve solutions' performance comprehensively. Therefore, the first modification in MTLBO is achieved by introducing more than one teacher for students. Solutions in a non-dominated set are sorted in descending order of crowding distance. Then, solutions with the largest crowding distance are selected as teachers. Other solutions in the population are improved by those teachers simultaneously in this phase. The basic TLBO is simpler than other evolutionary algorithms as it does not require any algorithm parameters, which are tuned before or during algorithms execution, and their values have significant influences on the algorithm's performances. However, one algorithm parameter, namely teacher count, should be specified in MTLBO.

In basic TLBO algorithm, the teaching factor is either one or two, which indicates that students learn either 0% or 100% from the teacher. The basic TLBO algorithm decides the value of the teaching factor randomly with equal probability (Eq. (3)). But in actual cases, students' learning capability is randomly distributed in the range of 0% and 100%, and is mainly affected by the teaching capability. The ratio gap between students and the current teacher of a course is used to indicate the teaching capability in the multi-course TLBO. Thus, the teaching factor T_F is re-defined as follows:

$$T_F = \begin{cases} 2, \text{ if } \dfrac{M_i}{M_{new}} > 2 \\ 1, \text{ if } \dfrac{M_i}{M_{new}} < 1 \\ \dfrac{M_i}{M_{new}}, \textit{otherwise} \end{cases} \tag{6}$$

where M_i is the mean of the class in iteration i, and M_{new} is the current teacher of a course. The value of teaching factor affects the algorithm's convergence and exploration capability. Quicker convergence and worse exploration capability would result from a larger teaching factor. Thus, T_F in Eq. (6) automatically adjusts the influence of teachers on students.

Then a selection operator is used to determine the acceptance of $X_{new,i}$ or $X_{old,i}$; If $X_{new,i}$ dominates $X_{old,i}$, $X_{new,i}$ is added into the population and $X_{old,i}$ is abandoned. Otherwise if $X_{new,i}$ is Pareto equal to $X_{old,i}$, a coin will be flipped for decision.

3.3 Learner Phase

The learner phase in the multi-course TLBO algorithm is slightly different from that of the basic TLBO algorithm and the detailed procedures are described as the following:

Step 1. Randomly select a learner X_j that is different from X_i;

Step 2. If X_i dominates X_j, X_j is updated using Eq. (4); if X_j dominates X_i, X_i is updated using Eq. (5). If X_i is Pareto equal to X_j, a coin is flipped to randomly select one equation from Eqs. (4) and (5) for updating the old solutions.

Step 3. The selection operator in the teaching phase is applied for determination of whether the new or old learner to be accepted.

3.4 Teacher Training Phase

The rest solutions in the non-dominated set are considered as candidate trainers. Each teacher T_i improves the performance from a randomly selected trainer. T_i is updated by as follows.

$$T_{new,i} = T_{old,i} + r_i \times (T_{trainer} - T_i) \tag{7}$$

where r_i is a random number in (0, 1). A selection operator is used to choose an solution from $T_{new,i}$ and $T_{old,i}$. If $T_{new,i}$ dominates $T_{old,i}$, $T_{new,i}$ is accepted and $X_{old,i}$ is abandoned. If they are not dominated with each other, a coin is flipped to determine which one to be accepted.

3.5 Exchange Student Phase

An exchange student is a student who studies at one of his or her school's partner institution. Local students can improve themselves by mutual interactions with the exchange student coming from another school; so does the exchange student. In this phase, a mutation operator introduces an exchange student into the current class. Three different solutions are randomly selected from the non-dominated set. Let X_1, X_2, and X_3 denote those selected solutions. The mutation process is illustrated as follows.

$$R = |f(X_1)| + |f(X_2)| + |f(X_3)| \tag{8}$$

$$R_1 = \frac{|f(X_1)|}{R} \tag{9}$$

$$R_2 = \frac{|f(X_2)|}{R} \tag{10}$$

$$R_3 = \frac{|f(X_3)|}{R} \tag{11}$$

$$X_{exchange} = \left(\frac{X_1 + X_2 + X_3}{3}\right) + (R_1 - R_2)(X_2 - X_1) + (R_2 - R_3)(X_3 - X_2) + (R_3 - R_1)(X_1 - X_3) \tag{12}$$

where $|f(X)|$ is the objective function vector norm. Then, a coin is flipped to determinate whether the exchange student is accepted or not. If so, a student is randomly selected from the population and abandoned.

The flowchart of MTLBO is shown in Figure 2.

4. Experiment and Discussion

A series of simulation experiments have been conducted to examine the performance of MTLBO. Its results are compared with those that are provided in [17–19]. Algorithms are compiled in C++ in a personal com-

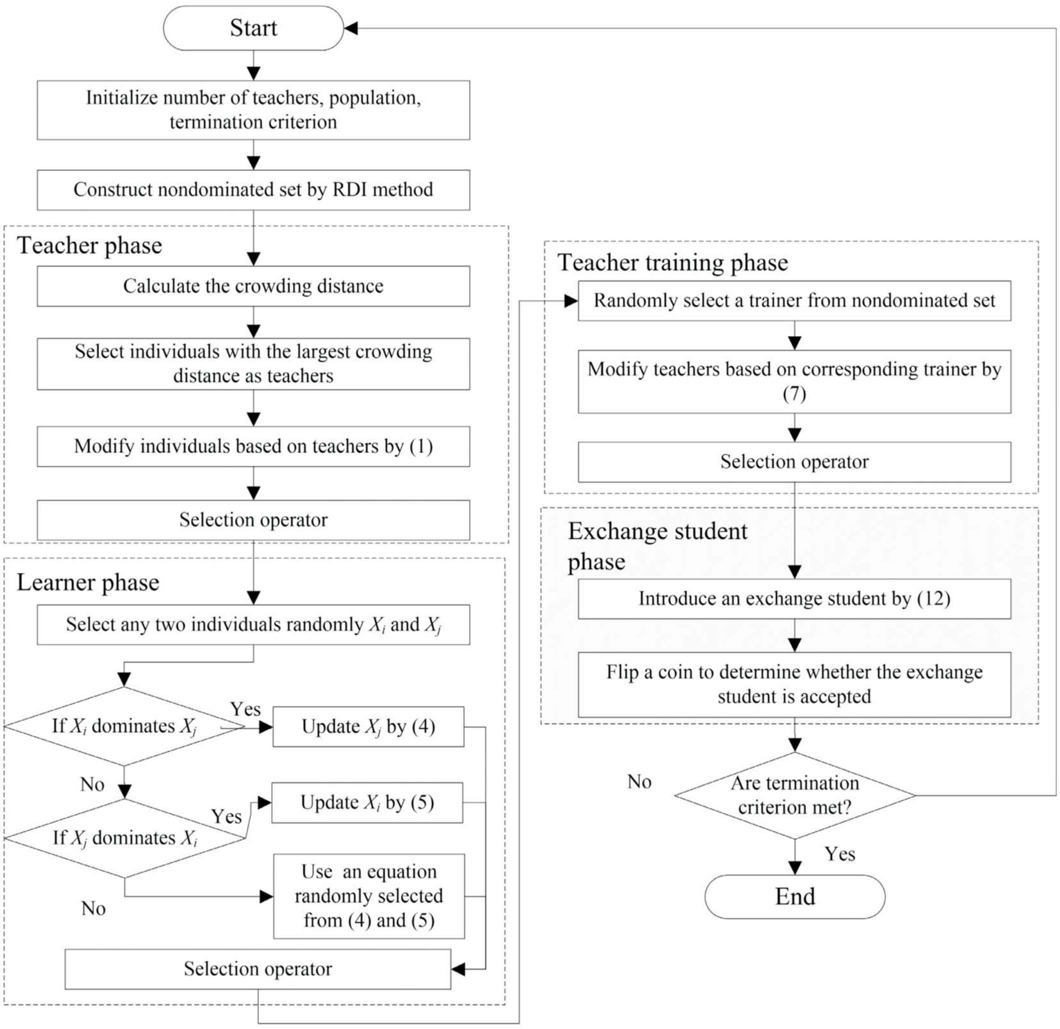

Figure 2. The flowchart of MTLBO algorithm.

puter with a Windows 7 operation system, a dual-core Pentium CPU, and 2 GB RAM.

4.1 Performance Criteria

The global optimum of a MOP is a non-dominated set of solutions, which is called as the true Pareto front; the non-dominated set obtained by an evolutionary algorithm is called as the approximate Pareto front. A set of 1001 uniformly spaced Pareto-optimal solutions are chosen from the true Pareto front in our tests. Let Γ denote the set of chosen solutions and $\tilde{\Gamma}$ denote the approximate Pareto front obtained. In order to compare the proposed algorithm with recently proposed algorithms, we consider the following standard performance criteria.

4.1.1 Generational Distance

Generational distance was proposed by Van Veldhuizen and Lamont in [25]. It is a measure of error between Γ and $\tilde{\Gamma}$, and measures the extent of convergence of $\tilde{\Gamma}$ to Γ. It is defined as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{|\tilde{\Gamma}|} d_i^2}}{|\tilde{\Gamma}|} \tag{13}$$

where $|\tilde{\Gamma}|$ = the number of solutions in $\tilde{\Gamma}$, and d_i is the Euclidean distance of solution i in $\tilde{\Gamma}$ to the nearest Pareto-optimal solution in Γ. The equation to determine d_i is computed as follows.

$$d_i = \min_{j=1}^{|\Gamma|} |f_i - f_j| \tag{14}$$

where $|\Gamma|$ is the number of solutions in Γ, f_i is the objective function value vector of solution i in $\tilde{\Gamma}$, and f_j = the objective function value vector of solution j in Γ. A lower value of GD indicates that the corresponding $\tilde{\Gamma}$ has a lower approximation error. When each solution in $\tilde{\Gamma}$ lies exactly in each in Γ, $GD = 0$.

4.1.2 Spacing Metric

Spacing metric desires to measure the diversity of solutions in $\tilde{\Gamma}$ [26]. This criterion provides a value that is evaluated by calculating the Euclidean distance of approximate Pareto-optimal solutions to its nearest solution in $\tilde{\Gamma}$. It is computed as follows:

$$SP = \sqrt{\frac{1}{|\tilde{\Gamma}|-1} \sum_{i=1}^{|\tilde{\Gamma}|} (\delta_i - \bar{\delta})^2} \tag{15}$$

where δ_i is the Euclidean distance of solution i to its nearest solution in $\tilde{\Gamma}$, and $\bar{\delta}$ is the average value of all δ_i. They are defined as follows:

$$\delta_i = \min_{j=1, j\neq i}^{|\tilde{\Gamma}|} |f_i - f_j| \tag{16}$$

$$\bar{\delta} = \frac{1}{|\tilde{\Gamma}|} \sum_{i=1}^{|\tilde{\Gamma}|} \delta_i \tag{17}$$

A lower SP value indicates that the corresponding $\tilde{\Gamma}$ is more evenly spread.

4.1.3 Diversity Metric

Besides of spacing metric, diversity metric, as proposed in [24], is used to measure the extent of spread in $\tilde{\Gamma}$. Let $\hat{d}_i$ denote Euclidean distance between two consecutive solutions in $\tilde{\Gamma}$, and d_a and d_b denote the Euclidean distances between extreme solutions of $\tilde{\Gamma}$ and Γ, respectively. Thereafter, diversity metric is defined by

$$DM = \frac{d_a + d_b + \sum_{i=1}^{|\tilde{\Gamma}|-1} |\hat{d}_i - \bar{d}|}{d_a + d_b + (|\tilde{\Gamma}|-1)\bar{d}} \tag{18}$$

4.2 Test Instances

To investigate the performance of MTLBO, eight benchmark problems, namely SCH1, DEB, FON, KUR ZDT1, ZDT3, ZDT6, and Poloni, are tested. Table 1 shows objective functions, variables, variable bounds, and the true Pareto-optimal solutions of those benchmark problems. They are widely used by other studies, e.g. [17,18, 19,27]. Their true Pareto fronts are publicly available at http://www.cs.cinvestav.mx/~emoobook/ [28].

4.3 Result Comparison

Three parameters should be determined before the implementation of MTLBO: number of teachers n_t, the population size s, and maximum number of function evaluations n_e or maximum number of iterations n_i. In this paper, n_t is set to 3, while n_e and n_i are largely depends on

Table 1. Benchmark problems (D = the number of variable in a problem)

Problem	Definition (minimization)	Variable bounds
SCH1	$f_1(x) = x^2, f_2(x) = (x-2)^2$	$x = [-10^3, 10^3]$
DEB	$f_1(x) = x_1$ $f_2(x) = (1 + x_2)(1 - g^2 - g \cdot \sin 8\pi x_1)$ $g(x) = x_1 / (1 + 10x_2)$	$x_i \in [0, 1], D = 2$
FON	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^{D} x_i - 1/\sqrt{3}\right)^2$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{D} x_i + 1/\sqrt{3}\right)^2$	$x_i \in [-4, 4], D = 3$
KUR	$f_1(x) = \sum_{i=1}^{D}\left(-10 \cdot \exp\left(-0.2 * \sqrt{x_i^2 + x_{i+1}^2}\right)\right)$ $f_2(x) = \sum_{i=1}^{D}\left(\lvert x_i\rvert^{0.8} + 5\sin x_i^3\right)$	$x_i \in [-5, 5], D = 3$
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \left[1 - \sqrt{x_1 / g(x)}\right]$ $g(x) = 1 + 9 \cdot \left(\sum_{i=2}^{D} x_i\right)/(D-1)$	$x_i \in [0, 1], D = 30$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \begin{bmatrix} 1 - \sqrt{x_1 / g(x)} - \\ x_1 / g(x) \cdot \sin 10\pi x_1 \end{bmatrix}$ $g(x) = 1 + 9 \cdot \left(\sum_{i=2}^{D} x_i\right)/(D-1)$	$x_i \in [0, 1], D = 30$
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \cdot \sin^6(6\pi x_1)$ $f_2(x) = g(x) \cdot [1 - (f_1(x) / g(x))^2]$ $g(x) = 1 + 9 \cdot \left(\sum_{i=2}^{D} x_i\right)/(D-1)$	$x_i \in [0, 1], D = 10$
Poloni	$f_1(x, y) = -[1 + (A_1 - B_1)^2 + (A_2 + B_2)^2]$ $f_2(x, y) = -[(x + 3)^2 + (y + 1)^2]$ $A_1 = 0.5 \cdot \sin 1 - 2 \cdot \cos 1 + \sin 2 - 1.5 \cdot \cos 2$ $A_2 = 1.5 \cdot \sin 1 - 1.5 \cdot \cos 1 + 2 \cdot \sin 2 - 0.5 \cdot \cos 2$ $B_1 = 0.5 \cdot \sin x - 2 \cdot \cos x + \sin y - 1.5 \cdot \cos y$ $B_2 = 1.5 \cdot \sin x - \cos x + 2 \cdot \sin y - 0.5 \cdot \cos y$	$-\pi \le x, y \le \pi$

the compared researches. They are given in Table 2.

Table 2. Algorithm parameters

Researches	s	n_e	n_i	# of runs
[17]	100	5,000	/	30
[18]	100	/	250	50
[19]	100	50,000 (ZDT6) 10,000 (other problems)	/	20

4.3.1 Comparison with Results in [17]

s and n_e of MTLBO are set to 100 and 5,000, respectively. MTLBO runs for 30 independent times. Then, the average and standard deviation of generational distance and spacing metric of $\tilde{\Gamma}$ are compared with those that are provided in [17]. All statistics are demonstrated in Table 3 and 4, and the best results are highlighted in bold. MTLBO produces best generational distances for ZDT1, ZDT3, and ZDT6 problems, while the MOABC in [17] produces best results of generational distance for KUR and Poloni problems (Table 3). Table 3 demonstrates the results of spacing metric of each algorithm. When consider spacing metric only, NSGA-Π outperforms the others for ZDT1, MTLBO outperforms the others for ZDT3 and ZDT6, and MOABC produces the best results for KUR and Poloni.

4.3.2 Comparison with results in [18]

In this comparison, s and n_i of MTLBO are set to 100

Table 3. Comparison of performance criteria (The results of NSGA-Π, MOPSO, and MOABC are due to [17])

Criteria	Problem	Statistics	NSGA-Π	MOPSO	MOABC	MTLBO
GD	ZDT1	Ave.	0.000294	0.055971	0.001593	**0.000203**
		Std. Dev.	0.000000	0.006906	0.000285	0.000119
	ZDT3	Ave.	0.0785023	0.074503	0.072869	**0.003953**
		Std. Dev.	0.0004601	0.006954	0.006359	0.003998
	ZDT6	Ave.	0.007125	0.041443	0.000158	**3.706e-05**
		Std. Dev.	0.000000	0.019230	0.000493	2.567e-06
	KUR	Ave.	0.003423	0.003053	**0.002219**	0.011953
		Std. Dev.	0.001040	0.000316	0.000453	0.006701
	Poloni	Ave.	0.001391	0.085586	**0.001129**	0.002108
		Std. Dev.	0.000000	0.006537	5.65e-05	0.001487
SP	ZDT1	Ave.	**0.002873**	0.102113	0.007187	0.004887
		Std. Dev.	0.000000	0.008997	0.001124	0.000665
	ZDT3	Ave.	0.062151	0.057493	0.042270	**0.009383**
		Std. Dev.	0.000294	0.004622	0.003581	0.003912
	ZDT6	Ave.	0.006427	0.191229	0.000927	**0.000510**
		Std. Dev.	0.000000	0.120909	0.000108	0.000211
	KUR	Ave.	0.040401	0.033112	0.014155	**0.012464**
		Std. Dev.	0.015577	0.005068	0.002979	0.021167
	Poloni	Ave.	0.011519	0.864664	**0.007154**	0.025672
		Std. Dev.	0.000000	0.071576	0.001087	0.598768

Table 4. Comparison of performance criteria (The results of NSGA-Π, MOPSO, MOPSO, and A-MOCLPSO are due to [18])

Criteria	Problem	Statistics	NSGA-Π	MOPSO	A-MOCLPSO	MTLBO
GD	SCH1	Ave.	8.016e-03	1.178e-01	9.006e-03	**2.877e-04**
		Std. Dev.	4.159e-03	6.209e-01	1.567e-03	1.179e-04
	KUR	Ave.	5.856e-01	1.960e-01	4.298e-01	**4.733e-03**
		Std. Dev.	4.408e-01	4.486e-02	9.809e-02	1.783e-03
	ZDT1	Ave.	4.797e-02	3.156e-01	2.016e-01	**1.196e-04**
		Std. Dev.	4.985e-02	1.379e-01	9.165e-02	3.832e-05
	ZDT3	Ave.	4.560e-02	3.308e-01	1.805e-02	**3.196e-04**
		Std. Dev.	4.538e-02	2.233e-01	2.864e-02	2.278e-05
	ZDT6	Ave.	3.322e-01	8.938e-01	9.785e-04	**3.541e-05**
		Std. Dev.	2.507e-01	8.310e-01	4.681e-04	1.545e-06
DM	SCH	Ave.	4.083e-01	1.231e+00	6.829e-01	**3.821e-01**
		Std. Dev.	3.003e-02	1.486e-01	5.619e-02	2.225e-02
	KUR	Ave.	5.861e-01	1.070e+00	6.628e-01	**4.429e-01**
		Std. Dev.	9.294e-02	7.409e-02	7.738e-02	3.122e-02
	ZDT1	Ave.	3.584e-01	9.835e-01	5.791e-01	**3.653e-01**
		Std. Dev.	4.420e-02	6.828e-02	8.672e-02	2.972e-02
	ZDT3	Ave.	**5.823e-01**	8.841e-01	7.652e-01	6.607e-01
		Std. Dev.	4.523e-02	6.228e-02	1.238e-01	2.195e-02
	ZDT6	Ave.	1.061e+00	1.193e+00	9.025e-01	**3.433e-01**
		Std. Dev.	1.484e-01	1.202e-01	1.077e-01	3.565e-02

and 250, respectively, as shown in Table 2. Average and standard deviation of generational distances, diversity metric and computational time are calculated after 50 independent runs. Table 4 and Figure 3 shows those results of MTLBO and the algorithms tested in [18]. MTLBO produces best results on the average of generational distance for each benchmark problem, as shown in Table 4; it generates more uniformly distributed results for all problems except for ZDT3, as shown in Table 4. Figure 3 indicate that MTLBO is the second fastest algorithm.

4.3.3 Comparison with Results in [19]

As shown in Table 2, s and n_e of MTLBO are set to 100 and 10,000, respectively. However, n_e is set to 50,000 in the case of ZDT 6. MTLBO runs for 20 independent times, and the average and average deviation of results are shown in Table 5. MTLBO produces best results for problems SCH1, DEB, FON, ZDT1, ZDT6, while MOTLBO in [19] generates the best generational distance for ZDT3 (Table 5). In the case of spacing metric, MTLBO outperforms the others for all benchmark problems except ZDT1. Figure 4 show that the MTLBO is the fastest algorithm.

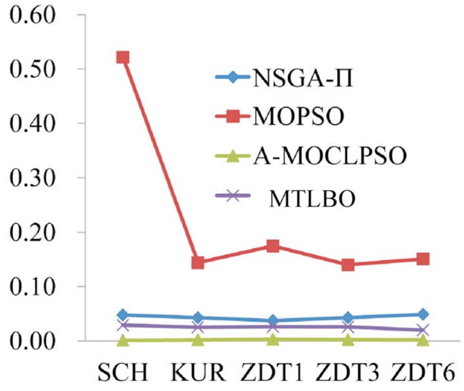

Figure 3. Computational times (in s) for one generation. (The results of NSGA-Π, MOPSO, and A-MOCLPSO are due to [18]).

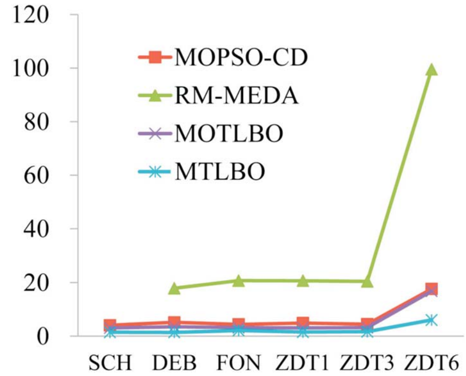

Figure 4. Computational times (in s). (The results of MOPSO-CD, RM-MEDA, MOTLBO are due to [19]).

Table 5. Comparison of performance criteria (The results of NSGA-Π, MOPSO-CD, MOTLBO are due to [19])

Criteria	Problem	Statistics	MOPSO-CD	RM-MEDA	MOTLBO	MTLBO
GD	SCH	Ave.	0.000389	/	0.000358	**0.000242**
		Ave. Dev.	0.000024	/	0.000014	7.133e-05
	DEB	Ave.	0.000384	0.000401	0.000399	**0.000308**
		Ave. Dev.	0.000032	0.000115	0.000021	1.411e-05
	FON	Ave.	0.002820	0.002921	0.002691	**0.000181**
		Ave. Dev.	0.000160	0.000288	0.000100	1.895e-05
	ZDT1	Ave.	0.000274	0.002867	0.000625	**9.919e-05**
		Ave. Dev.	0.000034	0.000180	0.000106	4.137e-05
	ZDT3	Ave.	0.001724	0.002173	**0.000962**	0.000976
		Ave. Dev.	0.000353	0.000280	0.000482	0.000811
	ZDT6	Ave.	0.063544	0.019598	0.006778	**3.543e-05**
		Ave. Dev.	0.013625	0.003118	0.012506	1.636e-06
SP	SCH	Ave.	0.027766	/	0.027800	**0.0230787**
		Ave. Dev.	0.004302	/	0.003712	0.0013841
	DEB	Ave.	0.005691	0.009392	0.010104	**0.0051291**
		Ave. Dev.	0.000596	0.002597	0.002490	0.0004759
	FON	Ave.	0.005545	0.009379	0.005598	**0.0046501**
		Ave. Dev.	0.000873	0.000882	0.000692	0.0005316
	ZDT1	Ave.	0.006334	0.011011	**0.006252**	0.0067297
		Ave. Dev.	0.000340	0.001581	0.000868	0.0020853
	ZDT3	Ave.	0.007426	0.011890	0.013946	**0.0076661**
		Ave. Dev.	0.001191	0.001691	0.004921	0.0019576
	ZDT6	Ave.	0.178535	0.008927	0.014462	**0.0039490**
		Ave. Dev.	0.103466	0.002258	0.002258	0.0003147

4.4 Discussion

A detail comparison of MTLBO with other evolutionary algorithms is presented in the last section. As we can see in Tables 3–5, MTLBO produces better results for the majority of problems than its peers. Figure 5 shows the total comparison count of each performance metric and how many times MTLBO provides the best results with respect to a particular metric. We can conclude that the number of times MTLBO provides the best results is much greater than its peer as shown in Figure 5.

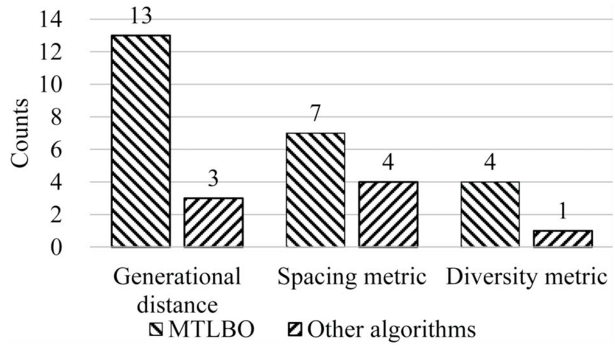

Figure 5. An overall comparison of performance metric.

5. Conclusion

A multi-course teaching-learning-based optimization algorithm is proposed for multi-objective optimization problems in this paper. The procedure of the proposed algorithm consists of four phases: teacher, student, teacher training, and exchange student phases. The removing dominated solution method constructs a non-dominated set from the current population. Then, the algorithm selects three solutions with the largest crowding distance in the non-dominated set as teachers. Thereafter, an adaptive teaching factor is adopted to mimic the dynamic influences of teachers on students. In the teacher training phase, trainers are randomly selected from the non-dominated set, and improve the performance of teachers; while in the exchange student phase, a mutation operator is used to produce an exchange student. Eight benchmark problems are tested to investigate the efficiency and effectiveness of the proposed algorithm. The results are compared with previous researches [17–19]. MTLBO produces the better results of generational distance, spacing metric, and diversity metric for the majority of benchmark problems (Figure 5). Besides, the proposed algorithm is the second fastest among all algorithms. Those comparison results indicate that MTLBO has a better overall performance than representative MOEAs and MOEAs proposed recently. For future works, MTLBO can also be used for some engineering optimization, such as metal cutting parameter optimization and scheduling.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 51705263 and 51705262), Zhejiang Provincial Natural Science Foundation of China (No. LQ16G010002), and Ningbo Natural Science Foundation (No. 2017A610141).

References

[1] Venkata Rao, R. and Patel, V., "Multi-objective Optimization of Heat Exchangers Using a Modified Teaching-learning-based Optimization Algorithm," *Applied Mathematical Modelling*, Vol. 37, No. 3, pp. 1147–1162 (2012). doi: 10.1016/j.apm.2012.03.043

[2] Pasupathy, T., Rajendran, C. and Suresh, R. K., "A Multi-objective Genetic Algorithm for Scheduling in Flow Shops to Minimize the Makespan and Total Flow Time of Jobs," *The International Journal of Advanced Manufacturing Technology*, Vol. 27, No. 7–8, pp. 804–815 (2006). doi: 10.1007/s00170-004-2249-6

[3] Jamshidi, R., Fatemi Ghomi, S. M. T. and Karimi, B., "Multi-objective Green Supply Chain Optimization with a New Hybrid Memetic Algorithm Using the Taguchi Method," *Scientia Iranica*, Vol. 19, No. 6, pp. 1876–1886 (2012). doi: 10.1016/j.scient.2012.07.002

[4] Horn, J., Nafpliotis, N. and Goldberg, D. E., "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on Evolutionary Computation, Orlando, FL, USA, June 27–29, Vol. 1, pp. 82–87 (1994). doi: 10.1109/ICEC.1994.350037

[5] Zitzler, E. and Thiele, L., "Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271 (1999). doi: 10.1109/4235.797969

[6] Knowles, J. and Corne, D., "The Pareto Archived Evolution Strategy: a New Baseline Algorithm for Pareto Multiobjective Optimisation," Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, July 6–9, Vol. 1, pp. 98–105 (1999).

[7] Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II," *Lecture Notes in Computer Science*, Vol. 1917, No. 2000, pp. 849–858 (2000). doi: 10.1007/3-540-45356-3_83

[8] Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L. and Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) (2001). doi: 10.3929/ethz-a-004284029

[9] Coello, C. A. C. C. and Pulido, G. T., "A Micro-genetic Algorithm for Multiobjective Optimization," International Conference on Evolutionary Multi-Criterion Optimization, Zurich, Switzerland, Mar 7–9, pp. 126–140 (2001).

[10] Zhang, Q. and Li, H., "MOEA/D: a Multiobjective Evolutionary Algorithm Based on Decomposition," *Evolutionary Computation, IEEE Transactions on*, Vol. 11, No. 6, pp. 712–731 (2007).

[11] Zhang, Q., Zhou, A. and Jin, Y., "RM-MEDA: a Regularity Model-based Multiobjective Estimation of Distribution Algorithm," *Evolutionary Computation, IEEE Transactions on*, Vol. 12, No. 1, pp. 41–63 (2008). doi: 10.1109/TEVC.2007.892759

[12] Huang, V. L., Zhao, S. Z., Mallipeddi, R., Suganthan, P. N. and Ieee, "Multi-objective Optimization Using Self-adaptive Differential Evolution Algorithm," IEEE Congress on Evolutionary Computation, Trondheim, Norway, May 18–21, pp. 190–194, (2009). doi: 10.1109/CEC.2009.4982947

[13] Atashpaz-Gargari, E. and Lucas, C., "Imperialist Competitive Algorithm: an Algorithm for Optimization Inspired by Imperialistic Competition," IEEE Congress on Evolutionary Computation, Singapore, Sept. 25–28, pp. 4661–4667 (2007). doi: 10.1109/CEC.2007.4425083

[14] Poli, R., Kennedy, J. and Blackwell, T., "Particle Swarm Optimization," *Swarm Intelligence*, Vol. 1, No. 1, pp. 33–57 (2007).

[15] Karaboga, D. and Basturk, B., "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm," *Journal of Global Optimization*, Vol. 39, No. 3, pp. 459–471 (2007). doi: 10.1007/s10898-007-9149-x

[16] Rao, R., Savsani, V. and Vakharia, D., "Teaching-learning-based Optimization: an Optimization Method for Continuous Non-linear Large Scale Problems," *Information Sciences*, Vol. 183, No. 1, pp. 1–15 (2012). doi: 10.1016/j.ins.2011.08.006

[17] Hedayatzadeh, R., Hasanizadeh, B., Akbari, R. and Ziarati, K., "A Multi-objective Artificial Bee Colony for Optimizing Multi-objective Problems," 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, Vol. 5, pp. V5-277–V5-281. doi: 10.1109/ICACTE.2010.5579761

[18] Ali, H. and Khan, F. A., "Attributed Multi-objective Comprehensive Learning Particle Swarm Optimization for Optimal Security of Networks," *Applied Soft Computing*, Vol. 13, No. 9, pp. 3903–3921 (2013). doi: 10.1016/j.asoc.2013.04.015

[19] Zou, F., Wang, L., Hei, X., Chen, D. and Wang, B., "Multi-objective Optimization Using Teaching-learning-based Optimization Algorithm," *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 4, pp. 1291–1300 (2013). doi: 10.1016/j.engappai.2012.11.006

[20] Wolpert, D. H. and Macready, W. G., "No Free Lunch Theorems for Optimization," *Evolutionary Computation, IEEE Transactions on*, Vol. 1, No. 1, pp. 67–82 (1997). doi: 10.1109/4235.585893

[21] Rao, R., Savsani, V. and Vakharia, D., "Teaching–learning-based Optimization: a Novel Method for Constrained Mechanical Design Optimization Problems," *Computer-Aided Design*, Vol. 43, No. 3, pp. 303–315 (2011). doi: 10.1016/j.cad.2010.12.015

[22] Satapathy, S. C. and Naik, A., "Data Clustering Based on Teaching-learning-based Optimization," SEMCCO 2011: *Swarm, Evolutionary, and Memetic Computing*, pp. 148–156: Springer (2011).

[23] Venkata Rao, R. and Kalyankar, V., "Parameter Optimization of Modern Machining Processes Using Teaching–learning-based Optimization Algorithm," *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 1, pp. 524–531 (2012). doi: 10.1016/j.engappai.

2012.06.007

[24] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197 (2002). doi: 10.1109/4235.996017

[25] Van Veldhuizen, D. A. and Lamont, G. B., "Evolutionary Computation and Convergence to a Pareto Front," Late Breaking Papers at the Genetic Programming 1998 Conference, pp. 221–228 (1998).

[26] Schott, J. R., *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Master's Desertation, Massachusetts Institute of Technology, Cambridge, USA. (1995).

[27] Deb, K., *Multi-objective Optimization Using Evolutionary Algorithms*, Wiley (2001).

[28] Coello, C. A. C., Lamont, G. B. and Van Veldhuisen, D. A., *Evolutionary Algorithms for Solving Multi-objective Problems*, Springer (2007).

Manuscript Received: Dec. 12, 2017
Accepted: Jan. 27, 2018