Soft Comput (2016) 20:1921–1943
DOI 10.1007/s00500-015-1613-9

METHODOLOGIES AND APPLICATION

SAMCCTLBO: a multi-class cooperative teaching–learning-based optimization algorithm with simulated annealing

Debao Chen · Feng Zou · Jiangtao Wang · Wujie Yuan

Published online: 14 February 2015

Abstract A variant of teaching–learning-based optimization algorithm (TLBO) with multi-classes cooperation and simulated annealing operator (SAMCCTLBO) is proposed in paper. To take full advantage of microteaching, the population is divided into several sub-classes, the mean of all learners in teacher phase of original TLBO is replaced by the mean solutions of different sub-classes, the modification might make the mean solutions improved quickly for the effect of microteaching is often better than teaching in big classes. With considering the limitation of learning ability of learner, the learners in different sub-classes only learn new knowledge from others in their sub-classes in learner phase of SAMCCTLBO, and all learners are regrouped randomly after some generations to improve the diversity of the sub-classes. The diversity of the whole class is improved by simulated annealing operator. The effectiveness of the proposed algorithm is tested on several benchmark functions, the results demonstrate that SAMCCTLBO has some good performances when compared with some other EAs.

Keywords Teaching–learning-based optimization · Particle swarm optimization · Multi-class cooperative · Simulated annealing

1 Introduction

Global optimization is a concerned research area in science and engineering for several decades. As many real-world optimization problems can be converted as global optimization problems, good optimization algorithms are always needed to derive the global optimal solution. Some traditional methods often fail while solving complex global optimization problems (Floudas and Gounaris 2009). Some meta-heuristics are developed to improve the performance of solving global optimization problems. Among these methods, particle swarm optimization algorithm (PSO) and its variants play an important role in solving global optimization problems for it is easy to be realized. Inertia weight PSO (WPSO) is introduced to balance the global and local search abilities. Linearly decreasing inertia weight PSO (LDWPSO) is proposed to improve the performance of WPSO (Shi and Eberhart 1998). PSO with a constriction factor (PSO-cf) was introduced by Clerc and Kennedy (2002). Some methods have been presented to improve the performance of PSO by designing different types of topologies. The conclusion that PSO with a small neighborhood might perform better on complex problems, while PSO with a large neighborhood would perform better on simple problems is drawn by Kennedy and Mendes (2002), Suganthan (1999). A PSO with dynamically adjusted neighborhoods is proposed by Suganthan (1999), in this method, the neighborhood of a particle gradually increases until it includes all particles. A unified particle swarm optimizer (UPSO) is introduced with combining the global version and local version together (Parsopoulos and Vrahatis 2004). Peram and Veeramachaneni developed the fitness-distance-ratio-based PSO (FDR-PSO) with near neighbor interactions (Peram et al. 2003). Liang et al. (2006) proposed a comprehensive learning particle swarm optimizer (CLPSO), all other particles' historical best information is used to update the velocity of a particle. Self-learning particles swarm optimizer (SLPSO), in which each particle has a set of four strategies to cope with different situations in the search space, and the cooperation of the four strategies is

Communicated by V. Loia.

D. Chen · F. Zou (✉) · J. Wang · W. Yuan
School of Physics and Electronic Information, Huai Bei Normal University, Huaibei 235000, China
e-mail: zfemail@163.com

implemented by an adaptive learning framework at the individual level, is proposed in Li et al. (2012). In addition, PSOs with several swarms are also studied for improving the performance of PSO algorithms. Cooperative particle swarm optimizer (CPSO) using multiple swarms to optimize different components of the solution vector cooperatively is introduced in Bergh and Engelbrecht (2004). Multi-swarm cooperative particle swarm optimizer (MCCPSO) based on master–slave model is presented in Niu et al. (2007); in the method, the slave swarms execute a single PSO and the master swarm evolves based on its own knowledge and the knowledge of the slave swarms. A multi-swarm PSO using charged particles in a partitioned search space is introduced in Dor et al. (2012), two kinds of swarms are used in this algorithm, the main swarm gathers the best particles of auxiliary ones and initialized several times. The auxiliary swarm is initialized in different areas, and an electrostatic repulsion heuristic is applied in each area to increase its diversity. A multi-swarm self-adaptive and cooperative particle swarm optimization (MSCPSO) using four sub-swarms to avoid the algorithm fall into local optimum is introduced in Zhang and Ding (2011). Moreover, some other EAs such as jDE (Brest et al. 2006), CMA-ES (Arnold and Hansen 2012) and JADE (Zhang and Sanderson 2009) are also used for solving global optimization problems.

In recent years, a new meta-heuristic algorithm, which is called teaching–learning-based optimization (TLBO), is proposed for global optimization problem (Rao et al. 2012). The algorithm simulates the teaching–learning process in a classroom, and each student represents a possible solution of the optimization problems. Details of the conceptual basis of TLBO were given by Waghmare (2013). Some results indicate that TLBO outperforms some meta-heuristics for constrained benchmark functions and non-linear numerical optimization problems (Rao et al. 2011, 2012). It has been extended to engineering optimization (Yu et al. 2014), such as constrained mechanical design optimization problems (Rao et al. 2011), engineering structure optimization problems (Vedat 2012), job-shop scheduling problem (Adil et al. 2014; Xu et al. 2015). It is also used for solving multi-objective optimization (Niknam et al. 2012; Rao and Patel 2011, 2013; Rao and Waghmare 2014; Zou et al. 2013), clustering problems (Naik et al. 2012; Suresh and Anima 2011), etc. Several variants of TLBO algorithm have started to improve the performance of TLBO (Hossein et al. 2011; Rao and Patel 2012). Similar to other natural computation methods, TLBO also might trap in local optimum when solving complex problems with multiple local optimal solutions.

To improve the performance of TLBO algorithm for solving global optimization problems, a multi-class cooperative teaching–learning-based optimization algorithm with simulated annealing operator (SAMCCTLBO) is proposed in this paper. A class is divided into several sub-classes, the students in each sub-class update their positions according to the teacher and the mean of their sub-classes in teacher phase, and they also learn knowledge from other students in their sub-classes. To improve the diversity of sub-classes, all individuals should be regrouped to form the new sub-classes after some evolutionary generations. The simulated annealing method is used to select some worse gens for the new population to increase the diversity of the whole class. The proposed algorithm is tested on some benchmark functions, and the results are compared with those of some other algorithms.

The rest of the paper is organized as follows: in Sect. 2, original TLBO algorithm is simply introduced. SAMCCTLBO is proposed in Sect. 3. Section 4 presents the test functions and the discussion of the experimental results. Some conclusions and the future research works are given in Sect. 5.

2 Teaching–learning-based optimization (TLBO) algorithm

Teaching–learning-based optimization (TLBO) algorithm is originally developed by Rao et al. (2012). It is based on the influence of a teacher on the output of learners in a class in terms of results or grades. It is also a population-based algorithm which simulates the teaching–learning process in the classroom. The teacher of the class is generally considered as a highly learned person who shares his or her knowledge with the learners. Generally, a good teacher benefits for improving the marks or grades of the students in his or her class. Moreover, learners also learn knowledge from others in their class to improve their marks and grades. For optimization, the solutions of the problem are represented by learners. The learner with the best fitness in current generation is chosen as the current teacher of the class. The learning process of TLBO is divided into two stages. The first stage is called the teacher phase and the second stage is called the learner phase. In teacher phase of TLBO, learners learn knowledge from the teacher to improve the average score of the class. In learner phase, learners learn knowledge from another random learner to improve their performance. The main stages of original TLBO are simply introduced as follows.

2.1 Teacher phase

In the teacher phase, all students learn knowledge from the teacher. The learner with the best fitness is chosen as the current teacher, which is represented by X_{teacher}. Suppose an objective function $f(x)$ with n-dimensional variables, the learner i can be represented as $X_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$. At any iteration g, the mean position of all learners in current iteration is calculated as $X_{\text{gmean}} = \frac{1}{m}\left[\sum_{i=1}^{m} x_{i1}, \sum_{i=1}^{m} x_{i2}, \ldots, \sum_{i=1}^{m} x_{in}\right]$, m is the number of learners in a class. For the ith learner, it updates its position as follows.

$$X_{i,\text{new}} = X_i + r_1(X_{\text{teacher}} - T_F X_{\text{gmean}}) \tag{1}$$

where $X_{i,\text{new}}$ and X_i are the new and the old positions of the ith learner, r_1 is a random number, uniformly distributed in [0, 1]. Value of $T_F = round[1 + rand(0, 1)\{2 - 1\}]$ can be either 1 or 2 which is shown in Eq. (2). If the new solution of the ith learner is better than the old one, the old position of the student will be replaced by the new one, or the position of the ith learner is not changed.

2.2 Learner phase

In the learner phase, learners lean knowledge from others. A learner randomly selects a learner which is different from him or her as the learning object. For jth learner X_j, it randomly selects the kth learner X_k which is different from him or her, the process of X_j learning knowledge from X_k is shown as follows.

If the fitness of X_j is better than that of X_k, then

$$X_{j,\text{new}} = X_j + r_j(X_j - X_k) \tag{2}$$

Else

$$X_{j,\text{new}} = X_j + r_j(X_k - X_j) \tag{3}$$

If the new position $X_{j,\text{new}}$ is better than the old one X_j, the old position X_j will be replaced by the new position $X_{j,\text{new}}$, otherwise, the position of jth learner is not changed. r_j is a random number, uniformly distributed in [0, 1]. The detailed algorithm can be found in Parsopoulos and Vrahatis (2004).

3 Multi-class cooperative TLBO with simulated annealing operator (SAMCCTLBO)

3.1 The main framework of SAMCCTLBO

Just like other population-based algorithms, due to the intrinsic randomness, TLBO suffers from premature convergence when solving complex optimization problems. A proper tradeoff between exploration and exploitation is necessary for the efficient and effective operation of a population-based stochastic optimization algorithm. The main motivation of our method is using cooperation of multi-swarms and simulated annealing method to prevent the whole population from getting trapped in a local optimum and improve the global performance of TLBO. The main framework of SAMCCTLBO is given as follows.

Step 1. Initialize the population of the classroom and randomly group the population into different sub-classes. Set the initial parameters of the algorithm.

Step 2. Calculates the mean position of different sub-classes and chooses the best learner among all sub-classes as the teacher.

Step 3. Executes the teacher phase of SAMCCTLBO with simulated annealing operator.

Step 4. Executes the learner phase of SAMCCTLBO with simulated annealing operator.

Step 5. Randomly regroups the population to form the new sub-classes.

Step 6. If the terminal condition is not satisfied, go to step 2, otherwise, output the best solution.

The main parts of the framework are introduced as follows.

3.2 The need for using multi-class cooperation and SA operator

In the teaching–learning process of a real classroom, a teacher can go to any place of the classroom to impart his or her knowledge, for example, solving the problem of any learner, mentoring the individuals in the class. The smaller size of class will benefit for quickly sharing the knowledge of the teacher, the average grade of the small class might be improved quicker than the big class. Microteaching might increase the convergence speed of the algorithm. Moreover, for a learner, he or she easily learns knowledge from other learners around himself or herself, and the good diversity of the local domain will benefit for local searching. Based on this idea, a multi-swarm TLBO is designed in our algorithm, the population in a class is divided into some sub-classes. In the teacher phase, the positions of learners in different sub-classes are renewed by the mean positions of the sub-classes and the position of the teacher of current generation. In the learner phase, the learners learn knowledge from other learners in their sub-classes.

According to the basic operator of TLBO, the good learners always maintained in the population whatever in teacher phase and learner phase, the searching process of it is greedy. These operators might make the diversity of the class decrease quickly with the development of evolution. When all learners are almost similar to the teacher, it is difficult to renew their position. The algorithm is easily trapped into local convergence. The regroup operator can change the diversity of the sub-classes, it cannot change the diversity of the whole class. Simulated annealing method is a very simple method to change the diversity of the population with introducing some worse learners according to a certain possibility. It is easily realized. In the paper, it is chosen to change the diversity of the population. The average position the population makes large effect for the original TLBO, the serve change of it might make large oscillation for the algorithm.

To maintain the stable performance of the algorithm, only some bits of the worse learner are introduced in the new population.

3.3 Teacher phase of SAMCCTLBO

During teacher phase of SAMCCTLBO, the individual with the best fitness is chosen as the teacher (X_{teacher}) of current generation. The learners in different sub-classes improve their performance by moving their positions towards the position of the teacher (X_{teacher}) with taking into account the current mean position $X_{C\text{mean}}$ of the learners in different sub-classes. Equation (4) shows how the learners in different sub-classes renew their positions.

$$X_{i,\text{new}} = X_i + r_1(X_{\text{teacher}} - T_F X_{C\text{mean}}) \quad (4)$$

where $X_{i,\text{new}}$ and X_i are the new and old positions of ith learner, respectively. r_1 is a random number, uniformly distributed in [0, 1]. X_{teacher} is the best position of the current generation, $X_{C\text{mean}}$ is the mean position of Cth sub-class. T_F is defined as shown in Eq. (2). If the new position $X_{i,\text{new}}$ is better than the old one X_i, $X_{i,\text{new}}$ is accepted, otherwise, if $X_{i,\text{new}}$ should be accepted according to the possibility which is determined by simulated annealing operator, a bit of the new position X_i will be randomly selected to take the place of the corresponding bit of $X_{i,\text{new}}$, and $X_{i,\text{new}}$ with the old bit is made to flow to the next phase. This operation can prevent the population from severe damaged. This operation is not adopted for the teacher in current generation.

3.4 Learner phase of SAMCCTLBO

In the learner phase of SAMCCTLBO, a learner often learns knowledge from other learners around them easily. The learners learn knowledge from other students in their sub-classes in SAMCCTLBO. For ith learner X_{Ci} in the Cth class, randomly select kth individual X_{Ck} in Cth sub-class which is different from X_{Ci}, X_{Ci} updates its position as follows.

If X_{Ci} is better than X_{Ck} according to their fitness, then

$$X_{Ci,\text{new}} = X_{Ci} + r_i(X_{Ci} - X_{Ck}) \quad (5)$$

Else

$$X_{Ci,\text{new}} = X_{Ci} + r_i(X_{Ck} - X_{Ci}) \quad (6)$$

where r_i is random number, uniformly distributed in [0, 1]. If the new position $X_{Ci,C\text{new}}$ is better than X_{Ci}, the new position $X_{Ci,C\text{new}}$ will be accepted, otherwise, the similar method with SA operator in teacher phase is also used to increase the diversity of the whole class.

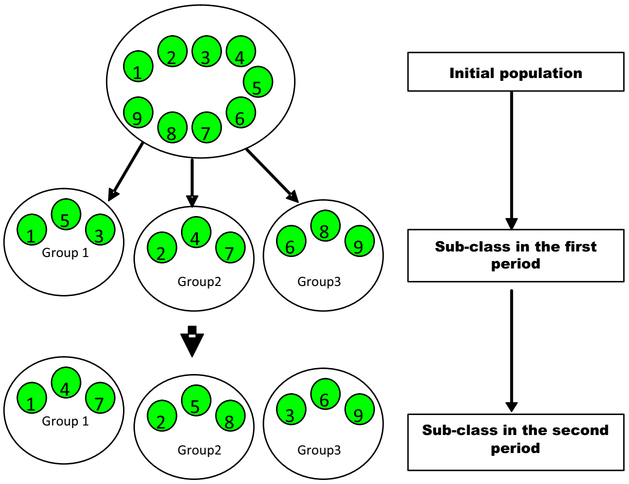

Fig. 1 The regroup process of SAMCCTLBO

3.5 Produce the new sub-classes (regroup)

When all learners complete the teacher and learner process, the new sub-classes of next generation should be generated. To improve the diversity of the sub-classes, the regroup operator is used in the given algorithm. In the real class, the positions of students are often changed after a period time to improve the effectiveness of teaching. Simulating this phenomenon, the learners in the class are regrouped after a given generation. This operator benefits for improving the diversities of the sub-classes, and the process of it is simple. A classroom with nine learners who are divided into three sub-classes is shown in Fig. 1 to interpret the process of forming the new sub-classes.

In Fig. 1, the initial population is composed by nine learners; in the first generation, the learners are grouped randomly; the first group contains learner 1, learner 3 and learner 5; the second group contains learner 2, learner 4 and learner 7, and the third group contains learner 6, learner 8 and learner 9. After some generations, the learners in the different sub-classes are integrated as a whole, and it is randomly regrouped again. As shown in Fig. 1, in the second generation, the learner 1, the learner 4 and the learner 7 are grouped in the first sub-class, the learner 2, the learner 5 and the learner 8 are grouped in the second sub-class, and the learner 3, the learner 6 and the learner 9 are grouped in the third sub-class. The diversity of the sub-classes is changed with change of the learners around them.

3.6 Simulated annealing operator

The detailed process of it can be found in Dekkers and Aarts (1991). In our method, to prevent the individual seriously damaged, a random bit chosen method is designed. The operator is simple, introduced as follows.

For the teacher, accept the better solution. For other learners, if the new solutions are better than the old ones, accept the new solutions. When the fitness value of the new learner is not better than the old one, calculate the possibility p according to Eq. (7).

$$p = \exp(-(\text{fitness}(X_{\text{new}}) - \text{fitness}(X_{\text{old}}))/T_k) \tag{7}$$

where $fitness(X_{\text{new}})$ and $fitness(X_{\text{old}})$ are the fitness of the new and the old positions of the individuals, respectively. T_{k} is the temperature of kth simulated annealing operator. If rand(.) $<$ p, a bit will be randomly selected from the old learner, and the correspondence bit of the new learner is replaced by it, the new learner with the old bit is made to flow to the next phase.

3.7 The steps of SAMCCTLBO algorithm

The detailed steps of SAMCCTLBO algorithm is described as follows.

Step 1: Set the maximal value $X_{\max}$ and minimal value $X_{\min}$ of variables, the maximal evolution generation *genmax*, the population size *popsize*, the class size C, the initial temperature T_0, the number of simulated annealing operator K, the temperature reduction coefficient λ. Initialize the initial population *pop* as follows.

$$\text{pop} = X_{\min} + r \times (X_{\max} - X_{\min}) \tag{8}$$

where r is a random number, uniformly distributed in [0, 1].

Step 2: Calculate the fitness value of all learners, and choose the best learner as the teacher of current generation.

Step 3: Divide the population into some sub-classes, the number of learners in each sub-class is shown in Eq. (9).

$$\text{Csize} = \text{popsize}/c \tag{9}$$

In general, popsize is the integer times of C. The size of all sub-classes is the same.

Step 4: Calculate the mean value $X_{C\text{mean}}$ of each sub-class and the learners in each sub-class implement teacher phase in different sub-classes according to Eq. (4). For the teacher, accept the better solution, and for other learners, execute simulated annealing operator according to part 3.6.

Step 5: For learners in each sub-class, randomly select a different learner in the same sub-class and implement the learner phase according to Eqs. (5, 6). The method of accepting new individual is the same as used in **Step 4**.

Step 6: when the generation satisfied the follow condition, the learners in the class will be regrouped.

$$\text{mod }(\text{gen}/M) = 0 \tag{10}$$

where gen is the current evolutionary generation, and M is the set period.

Step 7: If the number of simulated annealing operator is not arrived, $T_k = \lambda \times T_{k-1}$, if the terminal condition of the algorithm is not satisfied, the algorithm will go back to Step 4, or it is ended, output the best solutions.

3.8 The analysis of diversity and the number of class size

In the paper, two methods are designed to improve the diversity of the algorithm. The regrouping method is utilized to improve the diversity of the sub-classes. The simulated annealing operator is used to improve the diversity of the whole class. To show the efficiency of them, Rosenbrock function is simulated in this part. In the example, the size of the class is 4, the maximal evolutionary generation is 500, λ is 0.9, the initial temperature is 1,000, the population size is 20, the dimension of the function is 30, the period of regrouping is 10. The first example is the comparison of SAMCCTLBO with and without regrouping operator for the first sub-class. The second example is the comparison of multi-class cooperative teaching–learning-based optimization algorithm (MCCTLBO) with and without simulated annealing operator. A simple method with absolute distance is used in the paper to express the diversity of the population. It is shown in Eq. (11).

$$\text{Div(gen)} = m \times (m-1)/2 \sum_{i=1}^{m-1} \sum_{j=I+1}^{m} \sum_{l=1}^{D} |X(i,l) - X(j,l)| \tag{11}$$

where Div(gen) is the diversity function of gen generation, m is the population size, D is the dimension of the individuals, $X(i, l)$ and $X(j, l)$ are the lth dimensional variables of the ith and the jth individuals, respectively. The diversity of the first and the second examples is shown in Fig. 2.

The Fig. 2a displays that the diversity of SAMCCTLBO with regrouping operator is better than it without regrouping operator for the first class most of the time. Figure 2b shows that the diversity of SAMCCTLBO is better than MCCTLBO. The Fig. 2 also shows that the diversity of the improved algorithm is frequently changed.

The number of class size is also an important parameter for the proposed algorithm. The size of class should be larger than 2 and it is should satisfy Eq. (9). To show the influence of the size of sub-class for the algorithm, Rosenbrock function is also simulated with different sizes of sub-class. In the example, the population is 30, the other parameters of it is the same as those used in the above examples except the maximal evolutionary generation is 10,000. The results with different sizes (2, 3, 5, 6, 10, 15) of sub-class are shown in Table 1.

Table 1 shows that the fitness value decreases when the size is smaller than 5, and then increases when the size of sub-class is larger then 6. In our experiments, the median

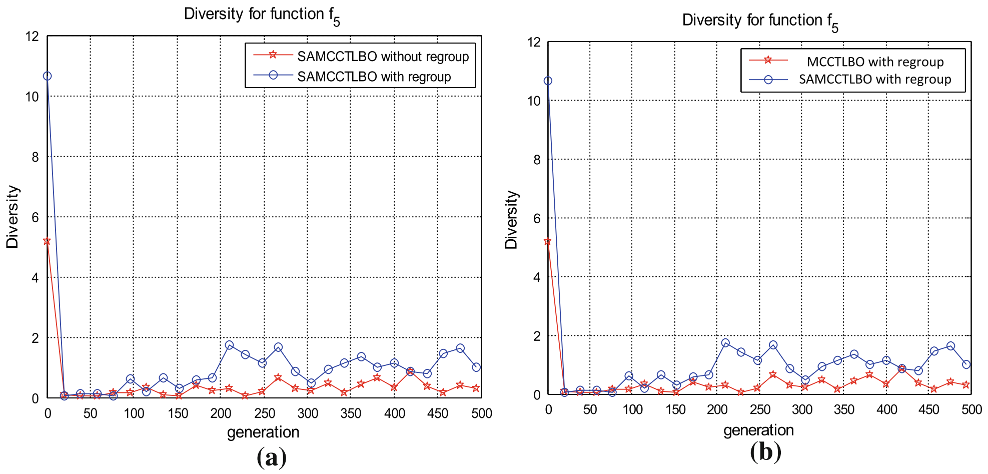

Fig. 2 The diversity examples for Rosenbrock function with different operator. **a** The result of SAMCCTLBO with and without regrouping operator, **b** the result of MCCTLBO and SAMCCTLBO

Table 1 The fitness value of Rosenbrock function with different size of class

Size	2	3	5	6	10	15
Fitness	8.78057	8.40195	6.43309	10.01520	15.09041	25.08271

value is chosen. For example, if the size of population is 30, the size of sub-classes is 5. If the size of population is 20, the size of sub-classes is 4.

3.9 The computation cost of algorithm

The fitness evaluation (FEs) often affects the computation cost of the algorithm, for large parts of EAs, the function is often evaluated once in a generation. But in TLBO, the function should be evaluated two times in a generation. For example, if the size of population is m, the maximal generation is genmax, the FEs of TLBO is $2 \times m \times$ genmax. For SAMCCTLBO algorithm, there are some worse individuals according to possibility p should be evaluated, the FEs is $2\times m\times$genmax$+2\times p\times m\times$genmax. The complexity of the proposed algorithm is larger than original TLBO if the size of population of them is the same. To balance the comparison, in our experiments, the size of population in SAMCCTLBO is smaller than that used in TLBO, it is smaller than half the size of population in other EAs with only one FEs in a generation.

4 Simulation results and discussions

Twenty-four of the well-known benchmark functions are used to evaluate the performance of SAMCCTLBO in this paper. The benchmark functions were also used in some other references (Liang et al. 2006; Sabat et al. 2011; Tang et al. 2007; Yao et al. 1999). To compare the performance of SAMCCTLBO with some other methods, UPSO (Parsopoulos and Vrahatis 2004), jDE (Brest et al. 2006), CMA_ES (Arnold and Hansen 2012; Wang et al. 2011), JADE (Zhang and Sanderson 2009) SaDE (Qin et al. 2009) fully informed particle swarm (FIPS) (Mendes et al. 2004) FDR-PSO (Peram et al. 2003) CLPSO (Liang et al. 2006) and TLBO (Rao et al. 2012) ETLBO (Rao and Patel 2012) are also simulated.

4.1 Parameter settings

For the purpose of reducing statistical errors, each function is independently simulated 50 runs, and their mean results are used in the comparison. The value of the function is defined as the fitness function. All the experiments are carried out on the same machine with a Celeron 2.26 GHz CPU, 512-MB memory system with Matlab software. All functions are simulated in 10 and 30 dimensions. The 24 functions are summarized in Table 2. "Range" is the lower and upper bounds for the variables. "Optima" is the theoretical global minimum solution. "Acceptance" is the acceptable solutions of the functions. For all PSOs and DEs, the evolutionary parameters are same as they are used in the references except that the population size is 50. Because the FEs of TLBO is more than 2 times of PSOs and DEs in a generation, the size of population of TLBO is 25; the size of population of SAMCCTLBO is 20. For ten-dimensional functions, the maximal fitness evaluations are 50,000 and 100,000 for 10- and 30-D functions, respectively. The size of class with SAMCCTLBO is 4.

4.2 Experimental results and comparisons

4.2.1 The results and the analysis of ten-dimensional functions

Table 3 displays the best solutions, the mean solutions, the standard deviations of the 50 independent runs of the ten

Table 2 24 Benchmark functions

Function	Formula	Range	Optima	Acceptance
Sphere	$F_1(x) = \sum_{i=1}^{D} x_i^2$	[−100,100]	0	1.0e−6
Quadric	$F_2(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	[−100,100]	0	1.0e−6
Sum square	$F_3(x) = \sum_{i=1}^{D} i x_i^2$	[−10,10]	0	1.0e−6
Zakharov	$F_4(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5 i x_i\right)^4$	[−10,10]	0	1.0e−6
Rosenbrock	$F_5(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$	[−2.048,2.048]	0	1.0e−1
Ackley	$F_6(x) = 20 - 20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$ $-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + e$	[−32.768,32.768]	0	1.0e−6
Rastrigin	$F_7(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[−5.12,5.12]	0	2.5e+0
Weierstrass	$F_8(x) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\max} \left[a^k \cos(2\pi b^k (x_i + 0.5))\right]\right)$ $-D\sum_{k=0}^{k\max} \left[a^k \cos(2\pi b^k \times 0.5)\right]$ $a = 0.5\ b = 3\ k\max = 20$	[−0.5,0.5]	0	1.0e−6
Griewank	$F_9(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[−600,600]	0	1.0e−1
Schwefel	$F_{10}(x) = 418.9829D + \sum_{i=1}^{D} (-x_i \sin\sqrt{abs(x_i)})$	[−500,500]	0	5.0e+2
Rotated sum square	$F_{11}(x) = \sum_{i=1}^{D} i y_i^2\ \ y = M \times x$	[−10,10]	0	1.0e−6
Rotated Zakharov	$F_{12}(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5 i y_i\right)^2 + \left(\sum_{i=1}^{D} 0.5 i y_i\right)^4$ $y = M \times x$	[−10,10]	0	1.0e−6
Rotated Rosenbrock	$F_{13}(x) = \sum_{i=1}^{D-1} \left[100(y_i^2 - y_{i+1})^2 + (y_i - 1)^2\right]$ $y = M \times x$	[−2.048,2.048]	0	1.5e+0
Rotated ackley	$F_{14}(x) = 20 - 20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right)$ $-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi y_i)\right) + e\quad y = M \times x$	[−32.768,32.768]	0	1.0e−6
Rotated Rastrigin	$F_{15}(x) = \sum_{i=1}^{D} (y_i^2 - 10\cos(2\pi y_i) + 10)$ $y = M \times x$	[−5.12,5.12]	0	5.0e+1
Rotated Weierstrass	$F_{16}(x) = \sum_{i=1}^{D} \left(\sum_{k=0}^{k\max} \left[a^k \cos(2\pi b^k (y_i + 0.5))\right]\right)$ $-D\sum_{k=0}^{k\max} \left[a^k \cos(2\pi b^k \times 0.5)\right]$ $a = 0.5\ b = 3\ \ k\max = 20\quad y = M \times x$	[−0.5,0.5]	0	1.0e−6
Rotated Griewank	$F_{17}(x) = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1\ \ y = M \times x$	[−600,600]	0	1.0e−2
Rotated Schwefel	$F_{18}(x) = 418.9829D + \sum_{i=1}^{D} (-y_i \sin\sqrt{abs(y_i)})$ $y = M \times x$	[−500,500]	0	2.4e+2
Shifted sphere	$F_{19}(x) = \sum_{i=1}^{D} z_i^2 - 450$ $z = x - o\ \ o:$ the shifted global optimum	[−100,100]	o	−450.1
Shifted Schwefel's Problem 2.21	$F_{20}(x) = \max_i \{\lvert z_i\rvert, 1 \le i \le D\} - 450$ $z = x - o + 1\ \ o:$ the shifted global optimum	[−100,100]	o	−450
Shifted Rosenbrock	$F_{21}(x) = \sum_{i=1}^{D-1} \left[100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2\right] + 390$ $z = x - o + 1\ \ o:$ the shifted global optimum	[−100,100]	o	393
Shifted Rastrigin	$F_{22}(x) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) - 330$ $z = x - o\ \ o:$ the shifted global optimum	[−5,5]	o	−319
Shifted Griewank	$F_{23}(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180$ $z = x - o\ \ o:$ the shifted global optimum	[−600,600]	o	179.9
Shifted ackley	$F_{24}(x) = 20 - 20\exp(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2})-$ $\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)\right) + e - 140$ $z = x - o\ \ o:$ the shifted global optimum	[−32,32]	o	−139.2

Table 3 Search result comparisons (the best solutions, average best solutions, standard deviation) among ten algorithms on 24 functions with 10 dimensions

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_1	Best	1.46E−46	1.55E−52	3.84E−46	**0.00E+00**	4.72E−56	6.86E−10	1.72E−44	3.22E−227	1.38E−207	3.85E−266
	Mean	1.67E−45	1.11E−47	3.56E−44	**0.00E+00**	1.79E−55	2.40E−09	7.36E−43	1.14E−225	1.47E−204	3.34E−259
	SD	2.03E−45	2.46E−47	7.55E−44	**0.00E+00**	1.44E−55	1.78E−09	7.97E−43	5.76E−226	3.74E−205	2.89E−264
f_2	Best	5.58E−19	3.34E−70	1.36E−22	**0.00E+00**	4.14E−23	1.88E−03	2.54E−13	1.87E−102	7.18E−103	4.13E−113
	Mean	2.59E−16	2.44E−64	6.50E−19	**0.00E+00**	1.12E−21	4.21E−03	6.75E−12	1.19E−100	2.52E−96	1.21E−107
	SD	3.95E−16	4.74E−64	1.45E−18	**0.00E+00**	1.74E−21	2.85E−03	1.37E−11	1.04E−100	4.28E−96	2.11E−107
f_3	Best	3.35E−47	1.41E−60	2.48E−47	**0.00E+00**	2.33E−58	6.27E−11	1.27E−45	6.33E−229	2.66E−208	2.75E−267
	Mean	1.22E−46	7.61E−53	1.26E−45	**0.00E+00**	6.49E−57	1.01E−10	3.40E−44	4.24E−226	2.02E−203	2.05E−260
	SD	7.24E−47	1.67E−52	2.13E−45	**0.00E+00**	8.49E−57	3.69E−11	4.22E−44	6.47E−226	5.21E−203	1.98E−260
f_4	Best	2.18E−20	1.49E−74	2.55E−29	**0.00E+00**	8.72E−32	2.68E−06	2.28E−20	4.42E−111	1.53E−110	2.75E−107
	Mean	1.41E−19	1.66E−63	1.99E−26	**0.00E+00**	6.16E−31	4.67E−06	9.85E−19	4.98E−108	3.22E−105	1.40E−105
	SD	7.21E−20	3.70E−63	4.37E−26	**0.00E+00**	7.11E−31	1.32E−06	1.85E−18	8.11E−108	4.19E−105	2.46E−105
f_5	Best	6.50E−05	**0.00E+00**	1.21E−04	3.89E−28	3.23E−01	4.60E+00	1.18E+00	1.08E+00	6.04E−01	4.27E+00
	Mean	2.40E−02	**0.00E+00**	2.39E−01	6.42E−28	1.19E+00	5.22E+00	1.80E+00	1.89E+00	1.19E+00	4.66E+00
	SD	2.41E−02	**0.00E+00**	3.59E−01	1.88E−28	1.64E+00	3.75E−01	4.20E−01	6.92E−01	5.11E−01	5.18E−01
f_6	Best	3.55E−15	3.55E−15	0.00E+00	3.55E−15	3.55E−15	1.16E−05	0.00E+00	3.55E−15	3.55E−15	**0.00E+00**
	Mean	3.55E−15	3.55E−15	2.84E−15	3.55E−15	3.55E−15	1.73E−05	2.84E−15	3.55E−15	3.55E−15	**0.00E+00**
	SD	0.00E+00	0.00E+00	1.59E−15	0.00E+00	0.00E+00	4.69E−06	1.59E−15	0.00E+00	0.00E+00	**0.00E+00**
f_7	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.98E+00	9.95E−01	1.45E−02	3.79E+00	6.00E−03	1.01E+00	**0.00E+00**
	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.96E+00	3.38E+00	1.99E+00	7.32E+00	3.19E+00	3.26E+00	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.56E+00	1.81E+00	1.59E+00	3.09E+00	2.16E+00	1.91E+00	**0.00E+00**
f_8	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.71E−04	**0.00E+00**	8.71E−04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	4.53E−09	0.00E+00	**0.00E+00**	5.94E−02	**0.00E+00**	1.18E−03	4.62E−02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	1.01E−08	0.00E+00	**0.00E+00**	6.47E−02	**0.00E+00**	2.36E−04	1.03E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	Best	**0.00E+00**	1.11E−16	**0.00E+00**	**0.00E+00**	4.43E−02	7.49E−02	9.86E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	**0.00E+00**	2.42E−11	4.60E−14	**0.00E+00**	6.64E−02	9.11E−02	2.86E−02	1.48E−03	4.92E−03	**0.00E+00**
	SD	**0.00E+00**	5.40E−11	1.03E−13	**0.00E+00**	1.57E−02	2.33E−02	1.82E−02	3.31E−03	1.10E−02	**0.00E+00**
f_{10}	Best	**1.27E−04**	**1.27E−04**	**1.27E−04**	4.15E+03	3.55E+02	2.44E+02	5.72E+02	6.52E+02	7.11E+02	3.55E+02
	Mean	**1.27E−04**	**1.27E−04**	**1.27E−04**	4.15E+03	8.13E+02	4.34E+02	7.52E+02	7.86E+02	8.93E+02	6.51E+02
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	3.49E+02	1.36E+02	3.16E+02	1.21E+02	1.73E+02	1.78E+02
f_{11}	Best	7.37E−44	2.56E−64	7.08E−43	**0.00E+00**	1.06E−55	1.17E−10	4.04E−43	4.07E−221	7.09E−203	3.74E−256
	Mean	4.51E−41	9.44E−52	2.37E−39	**3.46E−322**	4.04E−53	2.21E−10	1.89E−37	1.72E−212	2.47E−197	5.85E−246
	SD	9.69E−41	2.11E−51	2.61E−39	**0.00E+00**	8.96E−53	9.43E−11	4.24E−37	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{12}	Best	8.34E−22	8.79E−72	4.45E−28	**0.00E+00**	8.31E−34	1.79E−06	7.10E−20	1.15E−121	5.18E−113	8.43E−121
	Mean	4.32E−20	1.54E−66	1.99E−25	**7.40E−323**	7.76E−30	3.79E−06	8.78E−19	1.25E−111	4.39E−105	3.74E−104
	SD	6.88E−20	3.42E−66	3.93E−25	**0.00E+00**	1.73E−29	1.71E−06	1.22E−18	2.56E−111	9.67E−105	5.55E−104

Table 3 continued

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_{13}	Best	3.88E−01	0.00E+00	1.39E−02	**3.31E−28**	5.18E−01	5.08E+00	6.40E−01	2.58E+00	7.05E−02	4.86E+00
	Mean	1.18E+00	7.97E−01	3.54E+00	**8.46E−28**	1.10E+00	5.46E+00	2.21E+00	3.56E+00	1.90E+00	8.10E+00
	SD	1.58E+00	1.78E+00	3.01E+00	**3.07E−28**	7.29E−01	3.66E−01	1.38E+00	7.69E−01	1.62E+00	4.53E+00
f_{14}	Best	3.55E−15	0.00E+00	3.55E−15	3.55E−15	3.55E−15	1.69E−05	3.55E−15	0.00E+00	3.55E−15	**0.00E+00**
	Mean	3.55E−15	2.84E−15	3.55E−15	3.55E−15	3.55E−15	2.38E−05	3.55E−15	2.84E−15	3.55E−15	**0.00E+00**
	SD	0.00E+00	1.59E−15	0.00E+00	0.00E+00	0.00E+00	6.68E−06	0.00E+00	1.59E−15	0.00E+00	**0.00E+00**
f_{15}	Best	4.92E+00	5.79E−02	4.99E+00	2.98E+00	2.98E+00	3.54E+00	6.96E+00	1.99E+00	2.00E+00	**0.00E+00**
	Mean	6.76E+00	2.51E+00	7.03E+00	4.58E+00	5.57E+00	5.11E+00	1.01E+01	5.02E+00	6.74E+00	**7.97E−01**
	SD	1.35E+00	1.47E+00	1.71E+00	1.13E+00	2.06E+00	1.15E+00	2.38E+00	3.72E+00	4.41E+00	**1.08E+00**
f_{16}	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.77E−03	1.24E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	1.89E−11	1.85E-14	**0.00E+00**	8.94E−01	3.97E−02	3.43E−03	1.91E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	2.97E−11	4.13E−14	**0.00E+00**	1.72E+00	7.76E−02	1.27E−03	5.67E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{17}	Best	2.13E−03	2.52E−04	2.82E−10	**0.00E+00**	4.42E−02	2.02E−01	7.40E−03	1.22E−08	0.00E+00	0.00E+00
	Mean	2.10E−02	9.41E−03	1.55E−02	**0.00E+00**	9.69E−02	2.57E−01	2.64E−02	4.12E−03	2.97E−03	9.90E−03
	SD	1.25E−02	6.64E−03	2.02E−02	**0.00E+00**	6.22E−02	4.92E−02	1.26E−02	5.56E−03	6.60E−03	2.21E−02
f_{18}	Best	1.27E−04	1.18E+02	**1.50E−03**	2.61E+03	4.54E+02	3.55E+02	5.97E+02	3.36E+02	3.55E+02	5.91E+02
	Mean	1.65E+02	3.33E+02	**1.42E+02**	2.89E+03	8.51E+02	5.30E+02	8.31E+02	6.97E+02	7.33E+02	1.15E+03
	SD	1.64E+02	2.26E+02	**1.30E+02**	2.36E+02	3.75E+02	2.04E+02	1.74E+02	3.01E+02	2.96E+02	4.66E+02
f_{19}	Best	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**-4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**
	Mean	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.37E+02	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.56E+01	2.10E−09	**0.00E+00**	9.85E−14	1.06E−13	7.52E−14
f_{20}	Best	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**-4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**
	Mean	**−4.50E+02**	**−4.50E+02**	**-4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.49E+02**	**−4.50E+02**
	SD	8.27E−13	0.00E+00	9.78E−11	**6.36E−14**	**6.36E−14**	1.54E−04	2.60E−11	2.40E−011	1.96E−11	1.08E−11
f_{21}	Best	**3.90E+02**	**3.90E+02**	**3.90E+02**	**3.90E+02**	**3.90E+02**	3.96E+02	**3.90E+02**	**3.90E+02**	**3.92E+02**	**3.90E+02**
	Mean	**3.91E+02**	3.95E+02	3.92E+02	**3.91E+02**	3.96E+02	3.96E+02	3.92E+02	3.93E+02	3.95E+02	4.16E+02
	SD	**1.63E+00**	5.58E+00	2.29E+00	1.78E+00	1.10E+01	8.54E−01	1.79E+00	3.12E+00	2.93E+00	5.00E+01
f_{22}	Best	**−3.30E+02**	**−3.30E+02**	**−3.30E+02**	−2.81E+02	−3.28E+02	−3.28E+02	−3.25E+02	−3.22E+02	−3.24E+02	−3.20E+02
	Mean	**−3.30E+02**	**−3.30E+02**	**−3.30E+02**	−2.42E+02	−3.27E+02	−3.27E+02	−3.23E+02	−3.19E+02	−3.13E+02	−3.08E+02
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.50E+01	8.32E−01	1.27E+00	1.78E+00	1.91E+00	1.04E+01	1.03E+01
f_{23}	Best	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**
	Mean	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	−1.79E+02
	SD	**0.00E+00**	7.73E−11	7.88E−13	5.51E−03	4.46E−01	4.80E−02	1.95E−02	2.31E−02	8.29E−02	1.59E−01
f_{24}	Best	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**
	Mean	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	−1.25E+02	−1.38E+02	−1.40E+02	−1.40E+02	−1.39E+02	−1.39E+02	−1.39E+02
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.65E+00	2.42E+00	3.91E−06	**0.00E+00**	1.54E+00	7.84E−01	9.74E−01

algorithms on the 24 test functions. The best results among ten algorithms are shown in bold words.

Table 3 displays CMA_ES has the smallest best solutions, the smallest means and the smallest standard deviations for 10-D functions f_1, f_2, f_3, f_4. The performance in terms of the best solutions, the means and the standard deviations of SAMCCTLBO for functions f_1, f_2, f_3 are better than those of other algorithms except those with CMA_ES. For function f_4, the performance of TLBO in terms of three solutions is better than those of other algorithms except with CMA_ES. For function f_5, JADE has the smallest solutions, followed by it are CMA_ES, jDE, ETLBO and some other algorithms. For functions f_6, f_7, f_8, f_9, the solutions of SAMCCTLBO converge to theoretical optima. For function f_7, three DEs have the same solutions as it is with SAMCCTLBO. The performances of three TLBOs, SaDE and PSOFDR are the same. For function f_9, jDE and SAMCCTLBO have the same performance. Three DEs have the best performance and the solutions of them are the same. The solutions of SAMCCTLBO are better than those of other two TLBOs. For functions f_{11} and f_{12}, the solutions of CMA_ES are the best, and SAMCCTLBO ranked the second. CMA_ES has the best performance for function f_{13} and that of JADE followed by it. For functions f_{14} and f_{15}, the performance in terms of the three merits of SAMCCTLBO is the best among all ten algorithms. Three TLBOs and SaDE have the same best solutions and they all converge to theoretical optima for function f_{16}. CMA_ES and SaDE express the best performance for functions f_{17} and f_{18}, respectively. For functions f_{19} and f_{20}, all algorithms can converge to theoretical optima except the functions tested with PSOFDR. For function f_{21}, the best solutions of large parts of algorithms equal to theoretical optima, but the means of them cannot arrive the theoretical optima. jDE has the smallest standard deviations. For functions f_{22} and f_{24}, three DEs have the same best performances. All algorithms almost converge to theoretical optima for function f_{23}. Table 3 also shows that the performances of CMA_ES outperform those of other algorithms for 12 functions among the 24 test functions. The three merits of SAMCCTLBO generally outperform those of large part of algorithms except those with CMA_ES. The results also show that the performance of TLBO is globally improved by the improved methods.

To compare the computation cost and robustness of different algorithms, the average fitness evaluations (mFEs) and the successful ratios of the algorithms are shown in the Table 4. “mFEs” is the average FEs when the algorithm can converge to the acceptable solutions in all runs. If the algorithm is not convergent in all runs, the “mFEs” is expressed by “NaN”.

Table 4 displays that CMA_ES has the smallest mFEs for large part of functions except for functions $f_{7,}$ f_8, f_{10} and f_{18}. For functions $f_{7,}$ f_8 and f_{10}, SAMCCTLBO has the smallest mFEs. JADE has the smallest solution in terms of mFEs for function f_{18}. Except for CMA_ES, the mFEs of SAMCCTLBO are smaller than those of other algorithms for 17 functions. For function f_4 and f_{24}, the mFEs of TLBO is smaller than those of SAMCCTLBO. For functions $f_{5,}$ $f_{13,}$ $f_{18,}$ f_{21}, the mFEs of JADE are less than those of SAMCCTLBO. For functionsf_{22} and $f_{23,}$ jDE and ETLBO algorithm have better performance in terms of mFEs than SAMCCTLBO. Table 4 also indicates that the successful ratios of SAMCCTLBO for 15 functions are 100%. The successful ratios of SAMCCTLBO for nine functions are lower than some other algorithms. Three DEs have the higher successful ratios for five shift functions. The best solutions are shown in Table 4 with bold words.

To determine whether the results obtained by SAMCCTLBO are statistically different from the results generated by other algorithms, the nonparametric t tests are conducted between the best results of SAMCCTLBO and those achieved by the other algorithms for all functions. The t values and p values on every function of this two-tailed test with a significant level of 0.05 are shown in Table 5. Rows “Better”, “Same”, and “worse” represent the number of functions that SAMCCTLBO performs significantly better than, almost the same as, and significantly worse than the compared algorithm, respectively. The “better” are shown with bold words in the table. Table 5 indicates that the average excellent ratio $\left(\sum_{i=1}^{9} \text{better}(i)/24 \times 9\right)$ of SAMCCTLBO is 51.39 % for ten-dimensional functions. The Table 5 also shows the number of “better” is less than that of “worse” between and SAMCCTLBO and CMA_ES.

4.2.2 The results and analysis of 30−dimensional functions

The experiments conducted on ten-dimensional functions are repeated on the 30-dimensional functions, the best solutions, the mean solutions, the standard deviations of the 50 independent runs of the ten algorithms for the 24 test functions are shown in Table 6. The mean FEs (mFEs) and the successful ratios are shown in Table 7. The t tests results are shown in Table 8. To display the convergence process of different algorithms, the results of the front four functions are given in Fig. 3. Figure 3 shows the convergent process of ten algorithms for functions $f_{1,}$ f_2, f_3 and f_4. The figures are only used to show the convergent process of the ten algorithms, the detailed solutions can be found in the tables. Figure 3 displays that SAMCCTLBO has the fast convergence speed for the functions f_1 and f_3. CMA_ES has the fast convergence speed for the functions f_2 and f_4.

Table 6 displays that SAMCCTLBO can find the theoretical optima for 12 functions f_1, f_3, f_6, f_7, f_8, f_9, f_{11}, f_{14}, f_{15}, f_{16}, f_{17} and f_{19}. For functions f_1, f_3, f_8, f_9, f_{16} and f_{17}, the merits in terms of the best solutions, the mean solu-

Table 4 Search result comparisons (the mFEs and successful ratios) among ten algorithms on 24 functions with 10 dimensions

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_1	mFEs	10,699.0	7,610.4	10,516.2	**943.8**	22,616.6	38,187.8	10,749.8	2,304.2	2,431.0	1,797.4
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
f_2	mFEs	24,437.8	11,727.0	22,901.8	**1,631.6**	3,1612.2	NaN	3,0001.8	4,829.0	4,970.4	3,921.4
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_3	mFEs	8,736.4	6,488.2	9,154.4	**1141.8**	20,810.8	31,290.2	9,396.2	2,005.6	2,120.0	1,612.2
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
f_4	mFEs	20,401.6	10,896.6	18,301.6	**1,686.6**	26,270.2	NaN	23,013.6	4,758.8	4,983.6	5,006.4
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_5	mFEs	44,585.4	14,191.4	41,531.0	**3,698.8**	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	**100.0**	**100.0**	40.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0
f_6	mFEs	15,646.4	12,045.2	15,984.8	**1,799.2**	27,756.2	NaN	16,689.4	3,498.2	3,760.6	2,940.2
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_7	mFEs	10,065.8	9,309.0	13,376.4	NaN	27,101.0	38,640.0	NaN	28,737.0	32,564.0	**5393.4**
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	20.0	60.0	0.0	40.0	40.0	**100.0**
f_8	mFEs	30,597.2	20,025.0	21,157.8	NaN	30,666.8	NaN	24,040.5	5,022.4	5,544.4	**4,570.4**
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	**100.0**	0.0	80.0	**100.0**	**100.0**	**100.0**
f_9	mFEs	10,844.4	10,433.8	13,348.6	**49.4**	25,536.4	39,972.0	7,837.2	3,368.4	3,613.8	1,981.8
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.0	**100.0**	**100.0**	**100.0**	**100.0**
f_{10}	mFEs	4,890.6	4,777.0	6,871.6	NaN	13,226.0	37,864.0	NaN	NaN	NaN	**3888.0**
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	20.0	80.0	0.0	0.0	0.0	20.0
f_{11}	mFEs	9,685.6	6,863.8	10,529.0	**1,173.4**	21,623.2	33,621.4	10,192.2	2,112.4	2,189.8	1664.2
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
f_{12}	mFEs	20,920.2	10,551.6	18,597.6	**1,630.8**	25,619.0	NaN	22,954.6	4,869.6	4,970.2	4,616.2
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_{13}	mFEs	40,277.8	12,374.5	30,203.5	**3,151.6**	44,325.8	NaN	28,041.0	NaN	27,326.0	NaN
	Ratios (%)	**80.0**	**80.0**	40.0	**100.0**	**80.0**	0.0	40.0	0.0	40.0	0.0
f_{14}	mFEs	16,117.2	11,955.6	16,303.0	**1,994.0**	28,079.6	NaN	17,156.4	3,444.0	3,921.4	2,974.8
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**

Table 4 continued

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_{15}	mFEs	48,588.0	32,664.4	48,790.0	**508.3**	15,889.0	36,136.3	NaN	17,496.7	22,370.7	4,978.8
	Ratios (%)	20.0	**100.0**	20.0	80.0	40.0	60.0	0.0	60.0	60.0	**100.0**
f_{16}	mFEs	36,281.8	24,884.6	29,454.6	**3,955.0**	33,352.0	NaN	NaN	5,583.6	6,094.2	4,825.6
	Ratios (%)	**100.0**	**100.0**	**100.0**	20.0	60.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_{17}	mFEs	33,970.0	27,367.0	39,268.0	**237.0**	NaN	NaN	20,474.0	13,635.0	9,375.5	3,243.8
	Ratios (%)	20.0	40.0	60.0	**100.0**	0.0	0.0	20.0	80.0	80.0	80.0
f_{18}	mFEs	249,02.3	**16,475.0**	29,596.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	60.0	40.0	**80.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{19}	mFEs	18,676.2	14,403.2	18,813.0	**2,943.4**	33,490.7	NaN	1,9302.4	4,369.6	4,218.7	3,146.3
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
F_{20}	mFEs	21,346.5	29,277.8	19,207.8	3,102.4	35,479.2	37,814.6	23,498.4	4,517.4	4,289.6	**3,179.8**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
F_{21}	mFEs	33,876.0	20,641.0	29,962.3	**4,987.3**	34,456.0	NaN	20,715.0	25,758.3	46,051.5	35,516.3
	Ratios (%)	**80.0**	40.0	60.0	**80.0**	**80.0**	0.0	**80.0**	60.0	40.0	60.0
f_{22}	mFEs	**4,948.2**	5328.6	9,006.8	NaN	13,652.6	22,183.4	17,650.4	6,538.0	7,228.0	34599.0
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	40.0	40.0	20.0
f_{23}	mFEs	9,674.6	10,696.6	12,566.6	**1,502.0**	22,195.3	37,911.3	7,757.6	NaN	6,011.3	44,185.0
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	60.0	60.0	100.0	0.0	80.0	20.0
f_{24}	mFEs	4,598.4	3,256.0	5,190.6	**1,441.0**	14,688.7	11,724.6	5,425.0	2,701.0	2,602.0	28,337.5
	Ratios (%)	**100.0**	**100.0**	**100.0**	20.0	60.0	100.0	100.0	40.0	20.0	40.0

Table 5 Comparisons between SAMCCTLBO and other algorithms on t tests for ten-dimensional functions

F	Value	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO
f_1	t	4.50918	2.46860	2.58428	−25.26786	6.79673	7.35101	5.05536	10.08729	14.99352
	p	**0.00003**	**0.01653**	**0.01230**	0.00000	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_2	t	3.58790	2.82068	2.45566	−3.13272	3.51824	8.09943	2.69040	6.28062	3.22813
	p	**0.00069**	**0.00655**	**0.01708**	0.00271	**0.00085**	**0.00000**	**0.00931**	**0.00000**	**0.00205**
f_3	t	9.23532	2.49391	3.24357	−25.99146	4.18674	15.03286	4.41309	37.03897	14.99186
	p	**0.00000**	**0.01551**	**0.00196**	0.00000	**0.00010**	**0.00000**	**0.00004**	**0.00000**	**0.00000**
f_4	t	10.70762	2.45104	2.49095	−3.10939	4.73898	19.43356	2.92203	−3.09829	2.05508
	p	**0.00000**	**0.01728**	**0.01562**	0.00290	**0.00001**	**0.00000**	**0.00495**	**0.00300**	**0.04438**
f_5	t	−48.97256	−49.27969	−38.44611	−49.27969	−11.03763	4.82612	−23.52345	−17.56732	−26.11156
	p	0.00000	0.00000	0.00000	0.00000	0.00000	**0.00001**	0.00000	0.00000	0.00000
f_6	t	12.63008	12.63008	9.79796	12.63008	12.63008	20.14298	9.79796	12.63008	12.63008
	p	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_7	t	0.00000	0.00000	0.00000	8.36661	10.25139	6.84657	12.98712	8.09647	9.35286
	p	0.00000	0.00000	0.00000	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_8	t	2.44949	0.00000	0.00000	5.03308	0.00000	27.27761	2.44949	0.00000	0.00000
	p	**0.01735**	0.00000	0.00000	**0.00000**	0.00000	**0.00000**	**0.01735**	**0.00000**	**0.00000**
f_9	t	0.00000	2.45274	2.44949	0.00000	23.13698	21.46243	8.57771	2.44949	2.44949
	p	0.00000	**0.01721**	**0.01735**	0.00000	**0.00000**	**0.00000**	**0.00000**	**0.01735**	**0.01735**
f_{10}	t	−20.08312	−20.08312	−20.08312	107.87204	2.26201	−5.32743	1.52297	3.43087	5.33001
	p	0.00000	0.00000	0.00000	**0.00000**	**0.02746**	0.00000	0.13320	**0.00111**	**0.00000**
f_{11}	t	2.54772	2.44954	4.97525	−38.88564	2.47058	12.83788	2.45014	14.85935	25.67125
	p	**0.01352**	**0.01734**	**0.00001**	0.00000	**0.01645**	**0.00000**	**0.01732**	**0.00000**	**0.00000**
f_{12}	t	3.43802	2.47313	2.77135	−3.69113	2.45420	12.15183	3.92908	−3.69113	−3.20924
	p	**0.00109**	**0.01634**	**0.00749**	0.00049	**0.01714**	**0.00000**	**0.00023**	0.00049	0.00217
f_{13}	t	−7.90037	−8.21868	−4.58524	−9.79833	−8.36066	−3.17464	−6.81673	−5.41660	−7.05839
	p	0.00000	0.00000	0.00002	0.00000	0.00000	0.00240	0.00000	0.00000	0.00000
f_{14}	t	12.63008	9.79796	12.63008	12.63008	12.63008	19.50026	12.63008	9.79796	12.63008
	p	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**

Table 5 continued

F	Value	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO
f_{15}	t	14.62848	4.07077	13.80220	9.79697	9.59081	11.13515	17.24507	5.61646	6.84152
	p	**0.00000**	**0.00014**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_{16}	t	3.48370	2.44949	0.00000	2.83962	2.80487	14.77582	18.43684	0.00000	0.00000
	p	**0.00095**	**0.01735**	0.00000	**0.00622**	**0.00684**	**0.00000**	**0.00000**	0.00000	0.00000
f_{17}	t	2.38545	−0.11511	1.01688	−2.44949	7.21573	25.05783	3.54499	−1.38749	−1.64363
	p	**0.02035**	0.90876	0.31343	0.01735	**0.00000**	**0.00000**	**0.00078**	0.17060	0.10566
f_{18}	t	−10.93637	−8.64806	−11.42464	18.27546	−2.74298	−6.68839	−3.51400	−4.48133	−4.13828
	p	0.00000	0.00000	0.00000	**0.00000**	0.00809	0.00000	0.00086	0.00004	0.00011
f_{19}	t	−8.28079	−8.28079	−8.28079	−8.28079	2.83448	7.61568	−8.28079	0.00000	2.39046
	p	0.00000	0.00000	0.00000	0.00000	**0.00631**	**0.00000**	0.00000	0.00000	**0.02010**
F_{20}	t	−18.26875	−18.26875	−18.26875	−18.26875	−18.26875	−18.21528	−18.26875	−0.38950	3.00093
	p	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.69833	**0.00396**
F_{21}	t	−2.69582	−2.29154	−2.60307	−2.74197	−2.17509	−2.12080	−2.65113	−2.52170	−2.28677
	p	0.00917	0.02558	0.01171	0.00811	0.03371	0.03823	0.01033	0.01445	0.02588
f_{22}	t	−11.63078	−11.63078	−11.63078	13.46894	−9.91023	−9.84374	−7.71760	−5.62554	−1.72826
	p	0.00000	0.00000	0.00000	**0.00000**	0.00000	0.00000	0.00000	0.00000	0.08926
f_{23}	t	−20.67229	−20.67229	−20.67229	−20.57499	−3.64532	−17.08604	−19.47761	−14.04476	−15.67893
	p	0.00000	0.00000	0.00000	0.00000	0.00057	0.00000	0.00000	0.00000	0.00000
f_{24}	t	−7.82959	−7.82959	−7.82959	8.85987	0.79168	−7.82950	−7.82959	0.03870	−0.43920
	p	0.00000	0.00000	0.00000	**0.00000**	0.43177	0.00000	0.00000	0.96926	0.66215
Better		12	11	10	9	15	14	14	12	14
Worse		10	11	10	14	7	10	9	7	4
Same		2	2	4	1	2	0	1	5	6

Table 6 Search result comparisons (the best solutions, average best solutions, standard deviation) among ten algorithms on 24 functions with 30 dimensions

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_1	Best	5.71E−40	4.45E−77	4.36E−41	2.72E−259	3.95E−50	1.09E−03	2.16E−33	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	4.12E−38	7.54E−75	8.28E−39	4.25E−255	1.50E−47	1.23E−03	1.86E−32	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	8.77E−38	7.50E−75	1.02E−38	3.12E−255	2.17E−47	1.76E−04	2.85E−32	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_2	Best	1.39E−02	6.82E−13	3.83E−02	**1.33E−230**	1.62E−02	3.46E+02	4.11E+00	9.24E−92	1.28E−89	4.68E−128
	Mean	1.53E−01	6.02E−12	1.16E−01	**6.14E−226**	1.21E−01	4.19E+02	1.14E+01	3.66E−87	1.21E−83	2.15E−117
	SD	1.19E−01	6.62E−12	8.21E−02	**2.32E−226**	2.10E−01	1.06E+02	1.11E+01	8.17E−87	2.69E−83	4.81E−117
f_3	Best	6.24E−41	4.66E−78	5.51E−43	2.28E−250	1.59E−49	1.04E−04	4.46E−34	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	8.04E−40	1.86E−62	7.09E−40	3.73E−247	2.63E−47	1.49E−04	2.21E−33	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	9.15E−40	4.16E−62	1.44E−39	1.02E−247	5.70E−47	3.16E−05	1.75E−33	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_4	Best	1.69E−05	2.99E−16	8.40E−06	**2.74E−228**	2.13E−06	4.52E+00	1.75E−01	9.97E−57	5.72E−57	1.17E−72
	Mean	2.76E−04	4.75E−13	5.22E−05	**8.14E−225**	2.95E−05	5.28E+00	5.44E−01	1.77E−54	7.13E−52	8.04E−69
	SD	2.59E−04	8.51E−13	6.65E−05	**7.34E−225**	3.09E−05	4.49E−01	4.83E−01	2.18E−54	1.44E−51	1.10E−68
f_5	Best	1.46E+01	1.26E−07	2.34E+01	**1.07E−26**	1.60E+01	2.61E+01	2.15E+01	2.19E+01	2.26E+01	2.03E+01
	Mean	1.63E+01	1.55E−03	2.45E+01	**1.14E−26**	1.66E+01	2.62E+01	2.18E+01	2.32E+01	2.31E+01	2.17E+01
	SD	1.09E+00	3.46E−03	1.03E+00	**6.61E−28**	5.06E−01	7.54E−02	2.45E−01	7.39E−01	3.95E−01	1.18E+00
f_6	Best	7.11E−15	3.55E−15	3.55E−15	7.11E−15	7.11E−15	7.51E−03	3.55E−15	3.55E−15	3.55E−15	**0.00E+00**
	Mean	7.11E−15	4.97E−15	3.55E−15	7.82E−15	1.21E−14	9.02E−03	6.39E−15	3.55E−15	3.55E−15	**0.00E+00**
	SD	0.00E+00	1.95E−15	0.00E+00	1.59E−15	4.77E−15	1.97E−03	1.59E−15	0.00E+00	0.00E+00	**0.00E+00**
f_7	Best	0.00E+00	0.00E+00	0.00E+00	1.69E+01	2.79E+01	3.73E+01	5.97E+01	8.95E+00	8.95E+00	**0.00E+00**
	Mean	0.00E+00	0.00E+00	0.00E+00	2.27E+01	3.32E+01	5.37E+01	7.30E+01	1.77E+01	1.19E+01	**0.00E+00**
	SD	0.00E+00	0.00E+00	0.00E+00	8.60E+00	5.91E+00	1.14E+01	9.59E+00	8.37E+00	3.52E+00	**0.00E+00**
f_8	Best	1.30E−04	0.00E+00	0.00E+00	3.18E−01	5.14E−04	1.19E−01	4.83E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	7.48E−02	0.00E+00	0.00E+00	2.47E+00	6.30E−01	1.36E−01	8.48E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	1.60E−01	0.00E+00	0.00E+00	1.61E+00	7.97E−01	1.13E−02	2.90E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.51E−03	0.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	0.00E+00	2.96E−03	0.00E+00	0.00E+00	1.03E−02	2.00E−02	1.48E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	0.00E+00	4.05E−03	0.00E+00	0.00E+00	5.88E−03	2.03E−02	3.31E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{10}	Best	**3.82E−04**	**3.82E−04**	**3.82E−04**	1.25E+04	2.79E+03	4.01E+03	3.30E+03	3.02E+03	4.17E+03	3.82E+03
	Mean	**3.82E−04**	**3.82E−04**	**3.82E−04**	1.25E+04	3.70E+03	4.62E+03	4.51E+03	4.29E+03	4.47E+03	4.59E+03
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	9.53E+02	5.17E+02	9.34E+02	8.25E+02	2.98E+02	5.27E+02
f_{11}	Best	7.88E−27	2.06E−50	6.67E−23	2.86E−250	2.48E−38	4.38E−04	3.35E−20	0.00E+00	0.00E+00	**0.00E+00**
	Mean	6.72E−11	4.56E−37	6.31E−07	5.27E−247	1.95E−24	2.14E−03	6.49E−13	0.00E+00	5.10E−322	**0.00E+00**
	SD	1.50E−10	8.95E−37	1.41E−06	0.00E+00	4.35E−24	2.00E−03	1.45E−12	0.00E+00	0.00E+00	**0.00E+00**
f_{12}	Best	3.60E−05	2.58E−14	1.45E−05	**5.54E−228**	4.55E−07	2.48E+00	1.95E−01	4.51E−60	7.05E−58	4.65E−73
	Mean	1.13E−03	4.30E+00	7.02E−05	**1.31E−224**	1.00E−05	3.37E+00	5.15E−01	7.71E−52	5.31E−54	2.23E−68
	SD	1.84E−03	9.61E+00	9.92E−05	**1.02E−225**	9.39E−06	5.87E−01	3.01E−01	1.72E−51	1.12E−53	4.98E−68

Table 6 continued

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_{13}	Best	1.52E+01	5.79E+00	2.49E+01	**1.36E−26**	2.12E+01	2.54E+01	2.05E+01	1.46E+01	2.77E+01	1.63E+01
	Mean	1.92E+01	1.08E+01	4.76E+01	**1.56E−26**	4.80E+01	2.66E+01	2.43E+01	2.32E+01	4.66E+01	2.43E+01
	SD	2.27E+00	3.42E+00	2.68E+01	**2.01E−26**	3.46E+01	8.07E−01	3.25E+00	6.46E+00	2.16E+01	5.42E+00
f_{14}	Best	7.11E−15	3.55E−15	3.55E−15	7.11E−15	7.11E−15	7.42E−03	7.11E−15	3.55E−15	3.55E−15	**0.00E+00**
	Mean	7.11E−15	5.68E−15	3.55E−15	8.53E−15	2.68E−01	9.91E−03	7.11E−15	3.55E−15	3.55E−15	**0.00E+00**
	SD	0.00E+00	1.95E−15	0.00E+00	1.95E−15	5.99E−01	2.21E−03	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
f_{15}	Best	3.70E+01	2.10E+01	7.19E+01	1.09E+01	2.39E+01	1.01E+02	4.70E+01	1.82E+00	8.95E+00	**0.00E+00**
	Mean	4.60E+01	2.56E+01	8.35E+01	1.51E+01	3.48E+01	1.27E+02	6.81E+01	1.13E+01	1.22E+01	**0.00E+00**
	SD	9.28E+00	5.44E+00	8.90E+00	3.75E+00	7.87E+00	1.85E+01	1.27E+01	8.27E+00	2.48E+00	**0.00E+00**
f_{16}	Best	0.00E+00	1.28E−03	0.00E+00	2.01E+00	8.02E−01	3.17E−01	1.59E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	1.14E+00	2.40E−01	8.14E−02	3.14E+00	1.81E+00	4.34E−01	1.77E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	2.56E+00	4.57E−01	1.63E−01	1.28E+00	1.26E+00	9.22E−02	1.74E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{17}	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.76E−03	0.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	0.00E+00	1.51E−06	0.00E+00	0.00E+00	1.57E−02	1.89E−02	1.48E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	0.00E+00	3.37E−06	0.00E++00	0.00E+00	1.20E−02	1.54E−02	3.31E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{18}	Best	2.31E+03	**1.53E+03**	3.33E+03	6.66E+03	2.77E+03	3.86E+03	4.59E+03	4.89E+03	3.76E+03	4.36E+03
	Mean	2.65E+03	**2.19E+03**	4.26E+03	7.46E+03	4.16E+03	4.45E+03	5.16E+03	5.51E+03	5.14E+03	5.36E+03
	SD	**2.63E+02**	4.73E+02	7.48E+02	7.91E+02	9.34E+02	7.31E+02	6.08E+02	6.00E+02	1.10E+03	6.02E+02
f_{19}	Best	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.23E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**
	Mean	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**9.63E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**
	SD	2.84E−14	4.02E−14	**0.00E+00**	6.36E−14	2.23E+03	2.52E−04	**0.00E+00**	7.50E−11	8.55E−13	1.13E−09
f_{20}	Best	−4.41E+02	**−4.50E+02**	**4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.49E+02	**−4.50E++02**	−3.98E+02	−4.07E+02	−4.47E+02
	Mean	−4.32E+02	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	**−4.50E+02**	−4.48E+02	**−4.50E+02**	−3.89E+02	−3.97E+02	−4.46E+02
	SD	5.96E+00	3.82E−01	4.82E−03	**5.68E−14**	1.68E−01	3.38E−01	8.44E−02	1.12E+01	9.05E+00	5.73E−01
f_{21}	Best	4.10E+02	**3.90E+02**	4.15E+02	**3.90E+02**	4.08E+02	4.27E+02	4.12E+02	3.92E+02	4.04E+02	4.70E+02
	Mean	4.44E+02	**3.91E+02**	4.61E+02	3.92E+02	3.94E+07	4.87E+02	5.90E+02	4.55E+02	5.84E+02	5.59E+02
	SD	3.17E+01	**1.91E+00**	2.58E+01	2.18E+00	8.68E+07	4.99E+01	2.71E+02	6.15E+01	1.17E+02	9.25E+01
f_{22}	Best	**−3.30E+02**	**−3.30E+02**	**−3.30E+02**	−8.03E+01	−3.02E+02	−2.74E+02	−2.54E+02	−2.37E+02	−2.25E+02	−2.47E+02
	Mean	**−3.30E+02**	+ **−3.30E+02**	**−3.30E+02**	−6.06E+01	−2.85E+02	−2.49E+02	−2.31E+02	−2.11E+02	−2.08E+02	−2.11E+02
	SD	**2.84E−14**	**2.84E−14**	4.02E−14	1.77E+01	1.74E+01	1.67E+01	1.53E+01	2.59E+01	2.11E+01	2.06E+01
f_{23}	Best	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	**−1.80E+02**	−1.71E+02	**−1.80E+02**	**−1.80E+02**	−1.80E+02	**−1.80E+02**	**−1.80E+02**
	Mean	**−1.80E+02**	**−1.80E−02**	**−1.80E+02**	**−1.80E+02**	−1.61E+02	**−1.80E+02**	**−1.80E+02**	−1.79E+02	**−1.80E+02**	−1.77E+02
	SD	2.01E−14	3.31E−03	**0.00E+00**	5.51E−03	9.79E+00	1.36E−02	1.78E−11	1.66E+00	1.21E−01	2.50E+00
f_{24}	Best	**−1.40E+02**	**−1.40E+02**	**−1.40E−02**	−1.21E+02	−1.37E+02	**−1.40E+02**	**−1.40E+02**	−1.29E+02	−1.32E+02	−1.37E+02
	Mean	**−1.40E+02**	**−1.40E+02**	**−1.40E+02**	−1.20E+02	−1.33E+02	**−1.40E+02**	**−1.40E+02**	−1.26E+02	+ −1.30E+02	−1.32E+02
	SD	2.01E−14	2.01E−14	**0.00E+00**	6.95E−03	3.66E+00	2.12E−03	1.42E−14	2.00E+00	2.02E+00	3.68E+00

Table 7 Search result comparisons (the mFEs and successful ratios) among ten algorithms on 24 functions with 30 dimensions

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_1	mFEs	25562.8	13676.0	23132.8	2723.8	54889.4	NaN	29057.2	2878.2	3282.0	**1940.2**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_2	mFEs	NaN	64834.0	NaN	10125.0	NaN	NaN	NaN	11026.0	11969.8	**8726.8**
	Ratios (%)	0.0	**100.0**	0.0	**100.0**	0.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_3	mFEs	23283.4	12579.0	21071.2	4413.6	53611.0	NaN	26761.4	2617.0	3014.0	**1872.0**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_4	mFEs	NaN	56963.4	NaN	**12067.0**	NaN	NaN	NaN	22162.6	22951.8	15617.2
	Ratios (%)	0.0	**100.0**	0.0	**100.0**	0.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_5	mFEs	NaN	81055.4	NaN	**37802.0**	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	0.0	**100.0**	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0
f_6	mFEs	35882.4	20299.0	34284.2	4886.8	64052.4	NaN	44618.6	4153.6	4782.4	**2966.2**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_7	mFEs	36920.2	31948.8	54627.6	NaN	NaN	NaN	NaN	NaN	NaN	**28049.8**
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**
f_8	mFEs	NaN	47746.2	50243.8	NaN	NaN	NaN	NaN	5875.8	6641.0	**4386.2**
	Ratios (%)	0.0	**100.0**	**100.0**	0.0	0.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_9	mFEs	14596.8	8118.0	13851.2	**216.4**	45732.8	80487.4	17058.0	1766.6	1866.4	1564.4
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
f_{10}	mFEs	22328.4	**18890.8**	28140.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{11}	mFEs	42661.8	21333.4	39072.0	4310.0	60809.8	NaN	44663.2	2715.2	3094.0	**1942.4**
	Ratios (%)	**100.0**	**100.0**	80.0	**100.0**	**100.0**	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_{12}	mFEs	NaN	58785.8	NaN	**11873.2**	97758.0	NaN	NaN	22688.2	21819.0	15965.0
	Ratios (%)	0.0	80.0	0.0	**100.0**	20.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_{13}	mFEs	NaN	NaN	NaN	**18303.2**	NaN	NaN	NaN	NaN	NaN	NaN

Table 7 continued

F	*M*	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
	Ratios (%)	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0
f_{14}	mFEs	36994.6	20576.6	34295.6	4988.2	64338.0	NaN	47776.2	4133.2	4716.8	**2982.6**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	80.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
f_{15}	mFEs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	17568.0	NaN	**3651.6**
	Ratios (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	**100.0**
f_{16}	mFEs	80102.5	NaN	69579.7	NaN	NaN	NaN	NaN	6101.0	6803.2	**4441.0**
	Ratios (%)	80.0	0.0	60.0	0.0	0.0	0.0	0.0	**100.0**	**100.0**	**100.0**
f_{17}	mFEs	19163.4	13535.2	17925.2	776.2	46694.0	96523.0	25862.0	2166.8	3151.6	**1513.2**
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	20.0	40.0	**100.0**	**100.0**	**100.0**	**100.0**
f_{18}	mFEs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{19}	mFEs	60591.8	29867.3	46554.8	**2418.6**	NaN	65793.4	60408.2	4356.8	4318.4	57611.8
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	**0.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
F_{20}	mFEs	NaN	46741.8	**47320.4**	6534.8	72540.6	NaN	64031.2	NaN	NaN	NaN
	Ratios (%)	0.0	**100.0**	**100.0**	**100.0**	20.0	0.0	90.0	0.0	0.0	0.0
F_{21}	mFEs	NaN	84039.3	NaN	**41667.7**	NaN	NaN	NaN	75857.0	NaN	NaN
	Ratios (%)	0.0	**80.0**	0.0	60.0	0.0	0.0	0.0	20.0	0.0	0.0
f_{22}	mFEs	29513.2	**25302.4**	44266.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
f_{23}	mFEs	14777.0	**8743.6**	15545.0	3574.4	NaN	78815.8	17700.0	26067.0	24150.5	18796.4
	Ratios (%)	**100.0**	**100.0**	**100.0**	**100.0**	0.0	100.0	100.0	60.0	80.0	20.0
f_{24}	mFEs	**11428.2**	6786.6	11552.0	NaN	NaN	48026.0	17030.6	NaN	NaN	NaN
	Ratios (%)	**100.0**	**100.0**	**100.0**	0.0	0.0	**100.0**	**100.0**	0.0	0.0	0.0

Table 8 Comparisons between SAMCCTLBO and other algorithms on *t* tests for 30-dimensional functions

F	Value	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO
f_1	t	2.57129	5.50795	4.45724	37.12632	3.77675	38.43113	3.57305	0.00000	0.00000
	p	**0.01272**	**0.00000**	**0.00004**	**0.00000**	**0.00038**	**0.00000**	**0.00072**	0.00000	0.00000
f_2	t	7.03724	4.98208	7.71826	-2.44954	3.14993	21.56487	5.62208	2.45045	2.44950
	p	**0.00000**	**0.00001**	**0.00000**	0.01734	**0.00258**	**0.00000**	**0.00000**	**0.01731**	**0.01735**
f_3	t	4.81351	2.44949	2.70314	9.41477	2.52646	25.90861	6.94182	0.00000	0.00000
	p	**0.00001**	**0.01735**	**0.00900**	**0.00000**	**0.01427**	**0.00000**	**0.00000**	0.00000	0.00000
f_4	t	5.84780	3.05511	4.29932	−3.98649	5.23322	64.50603	6.17275	4.45585	2.71095
	p	**0.00000**	**0.00340**	**0.00007**	0.00019	**0.00000**	**0.00000**	**0.00000**	**0.00004**	**0.00881**
f_5	t	−18.42574	−100.52920	9.59665	−100.53682	−21.88638	20.79546	0.40295	5.67158	5.86811
	p	0.00000	0.00000	**0.00000**	0.00000	0.00000	**0.00000**	0.68847	**0.00000**	**0.00000**
f_6	t	28.87625	14.00000	6.46547	26.94439	13.88044	25.11662	22.04541	7.16872	8.57405
	p	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_7	t	0.00000	0.00000	0.00000	14.44894	30.79073	25.82012	41.69293	11.59501	18.58996
	p	0.00000	0.00000	0.00000	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_8	t	2.56703	0.00000	0.00000	8.41672	4.33267	65.85690	16.01288	0.00000	0.00000
	p	**0.01286**	0.00000	0.00000	**0.00000**	**0.00006**	**0.00000**	**0.00000**	0.00000	0.00000
f_9	t	0.00000	4.00000	0.00000	0.00000	9.63654	5.40856	2.44949	0.00000	0.00000
	p	**0.00000**	**0.00018**	0.00000	0.00000	**0.00000**	**0.00000**	**0.01735**	0.00000	0.00000
f_{10}	t	−47.64324	−47.64324	−47.64324	81.66111	−4.46464	0.23555	−0.41527	−1.64795	−1.06261
	p	0.00000	0.00000	0.00000	**0.00000**	0.00004	0.81461	0.67948	0.10477	0.29236
f_{11}	t	2.44997	2.78909	2.44949	6.30239	2.44988	5.84077	2.44978	0.00000	21.95203
	p	**0.01733**	**0.00714**	**0.01735**	**0.00000**	**0.01733**	**0.00000**	**0.01733**	0.00000	**0.00000**
f_{12}	t	3.37504	2.44949	3.87964	−2.45297	5.85749	31.48361	9.38690	2.46115	2.60392
	p	**0.00132**	**0.01735**	**0.00027**	0.01726	**0.00000**	**0.00000**	**0.00000**	**0.01685**	**0.01169**
f_{13}	t	−4.72521	−11.47745	4.68882	−24.53145	3.70567	2.37694	0.02122	−0.72218	5.49871
	p	0.00002	0.00000	**0.00002**	0.00000	**0.00047**	**0.02078**	0.98314	0.47309	**0.00000**

Table 8 continued

F	Value	jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO
f_{14}	t	34.43507	16.00000	14.99946	24.00000	2.44949	24.61280	22.11727	8.88527	6.62198
	p	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.01735**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_{15}	t	27.16030	25.75582	51.37298	22.09328	24.24880	37.51085	29.43220	7.48939	27.03602
	p	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
f_{16}	t	2.44949	2.88169	2.73567	13.45741	7.88851	25.77564	55.42342	0.00000	0.00000
	p	**0.01735**	**0.00554**	**0.00825**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	0.00000	0.00000
f_{17}	t	0.00000	2.44950	0.00000	0.00000	7.19689	6.73403	2.45081	0.00000	0.00000
	p	0.00000	**0.01735**	0.00000	0.00000	**0.00000**	**0.00000**	**0.01729**	0.00000	0.00000
f_{18}	t	−22.59255	−22.66691	−6.27127	11.54989	−5.93337	−5.25960	−1.31864	0.93816	−0.98731
	p	0.00000	0.00000	0.00000	**0.00000**	0.00000	0.00000	0.19247	0.35205	0.32759
f_{19}	t	−3.63490	−3.63490	−3.63490	−3.63435	3.46663	29.78762	−3.63490	−3.34053	−3.62610
	p	0.00059	0.00059	0.00059	0.00059	**0.00100**	**0.00000**	0.00059	0.00147	0.00061
F_{20}	t	12.55015	−29.76563	−40.13484	−40.20145	−36.72735	−21.91895	−38.34012	27.96451	29.57267
	p	**0.00000**	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	**0.00000**	**0.00000**
F_{21}	t	−6.47201	−9.94760	−5.62903	−9.93356	2.48410	−3.78654	0.59512	−5.14075	0.92603
	p	0.00000	0.00000	0.00000	0.00000	**0.01590**	0.00036	0.55408	0.00000	0.35827
f_{22}	t	−31.60625	−31.60625	−31.60625	30.34430	−15.05010	−7.83376	−4.15320	0.02403	0.61823
	p	0.00000	0.00000	0.00000	**0.00000**	0.00000	0.00000	0.00011	0.98091	0.53884
f_{23}	t	−6.75186	−6.74861	−6.75186	−6.74644	8.77967	−6.72977	−6.75186	−4.14115	−6.58151
	p	0.00000	0.00000	0.00000	0.00000	**0.00000**	0.00000	0.00000	0.00011	0.00000
f_{24}	t	−11.43332	−11.43332	−11.43332	17.61847	−0.83801	−11.41896	−11.43332	7.96114	3.29749
	p	0.00000	0.00000	0.00000	**0.00000**	0.40547	0.00000	0.00000	**0.00000**	**0.00167**
Better		13	12	11	12	18	17	14	10	12
Worse		9	10	9	10	5	6	5	3	2
Same		2	2	4	2	1	1	5	11	10

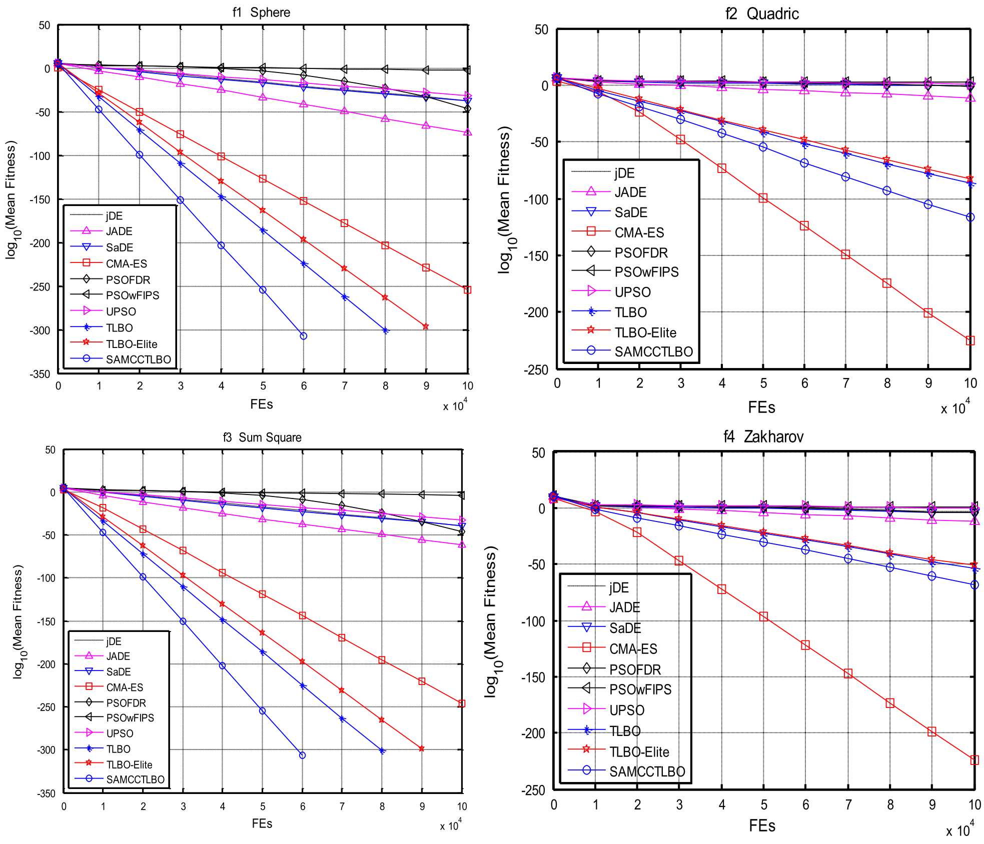

Fig. 3 The representative convergence curves of the ten algorithms

tions, the standard deviations of three TLBOs are the same. The solutions of them are better than those of other algorithms. CMA_ES has the best performance in terms of the three merits for functions f_2, f_4, f_5,f_{12} and f_{13}. For functions f_{10}, f_{22}, f_{23} and f_{24}, three DEs almost have the same performance except that the standard deviations of SaDE is larger than the other DEs for function f_{22}, and smaller than the others for function f_{23} and f_{24}. For function f_{18}, all algorithms cannot converge to the acceptable solution, the performances of JADE are better than that of other algorithms. The table also indicates that the performances of DEs for shift functions are generally better than those of other algorithms which are used in the paper.

Table 7 shows the mFEs and the successful ratios of different algorithms in the paper. The table indicates that the mFEs of SAMCCTLBO are less than those of other algorithms for functions f_1, f_2, f_3, f_6, f_7, f_8, f_{11}, f_{14}, f_{15}, f_{16} and f_{17}. The performance in terms of mFEs of CMA_ES is better than those of some other algorithms for functions f_4, f_5, f_9, f_{12}, f_{13}, f_{19}, f_{21}. JADE has the smallest mFEs for functions f_{10}, f_{22} and f_{23}. For function f_{20}, SaDE has the smallest mFEs. SAMCCTLBO and CMA_ES can converge to acceptable solutions with 100 % successful ratios for 15 functions among the 24 test functions. JADE can reach the acceptable solutions with 100 % successful ratios for 17 functions. The performance in terms of successful ratios for other algorithms is relative lower than these three algorithms. Table 8 displays that the average excellent ratio of SAMCCTLBO is 55.99 % for 30-dimensional functions.

4.2.3 The results and the analysis of 200−dimensional functions

To test the performance of the proposed algorithm in dealing with higher dimensional problems, some 200−dimensional functions f_2, f_4, f_5, f_{12} and f_{13} are simulated in this section. The training parameters of these experiments are same as those for 30-dimensional functions. Because the algorithms cannot converge to acceptable solutions for large part of functions, only the average solutions, the standard deviations of the 50 independent runs are listed in Table 9.

Table 9 displays that SAMCCTLBO has better performance in terms of the mean and the standard deviations than those of other algorithms for functions f_2, f_4, f_5 and f_{12}. The performance of CMA_ES is better than those of other algorithms for function f_{13}, and the performances of TLBOs are worse than those of DEs.

Table 9 Search result comparisons (the average best solutions, standard deviation) among ten algorithms on 5 functions with 200 dimensions

F		jDE	JADE	SaDE	CMA_ES	PSOFDR	PSOwFIPS	UPSO	TLBO	ETLBO	SAMCCTLBO
f_2	Mean	7.05E+04	1.34E+04	1.62E+04	1.04E+01	1.63E+05	2.18E+05	2.54E+05	3.41E−38	6.92E−34	**1.31E−51**
	SD	3.38E+04	9.76E+03	1.48E+04	6.54E+00	1.07E+05	1.24E+05	1.37E+05	1.05E−38	4.78E−34	**1.22E−51**
f_4	Mean	2.32E+03	1.85E+03	1.58E+03	3.82E+01	1.93E+03	1.38E+03	5.07E+03	2.12E+02	1.31E+02	**1.42E−03**
	SD	1.71E+03	1.06E+03	9.27E+02	1.04E+01	1.53E+03	1.11E+03	2.35E+03	1.05E+02	9.77E+01	**1.08E−03**
f_5	Mean	2.44E+02	1.94E+02	1.95E+02	2.06E+02	2.02E+02	7.22E+02	1.96E+02	1.96E+02	1.95E+02	**1.93E+02**
	SD	1.22E+02	1.03E+02	1.15E+02	1.12E+02	1.07E+02	2.48E+02	1.14E+02	1.07E+02	1.36E+02	**1.93E+02**
f_{12}	Mean	1.84E+03	3.04E+03	1.76E+03	5.56E+01	1.68E+03	1.87E+03	6.16E+03	1.40E+02	1.82E+02	**4.04E**−01
	SD	1.02E+03	2.17E+03	1.32E+03	3.36E+01	9.75E+02	1.03E+03	3.27E+03	1.01E+02	1.37E+02	**2.16E−01**
f_{13}	Mean	6.21E+02	5.67E+02	5.58E+02	**8.11E+00**	2.05E+03	1.22E+03	6.50E+02	3.58E+03	3.34E+03	6.91E+03
	SD	2.33E+02	1.98E+02	2.47E+02	**3.74E+00**	1.23E+03	1.07E+03	3.42E+02	2.17E+03	1.54E+03	4.27E+03

To take full advantage of microteaching, the population of the SAMCCTLBO algorithm is divided into several sub-classes, the learning of each learner is now restricted within a certain sub-classes area and might make the mean solutions improved quickly so as to fully utilize the whole sub-classes space and avoid over-congestion around local optima. With considering the limitation of learning ability of learner of the same sub-classes, all learners are regrouped randomly after a certain generation to improve the diversity of the sub-classes by simulated annealing operator. By analyzing the results of different algorithms for 24 benchmark functions considered, one may conclude that SAMCCTLBO has some advantages compared with some other algorithms. First, for 10-, 30- and 200-dimensional functions, the solution accuracies of SAMCCTLBO are generally superior, at least not worse to those of all variants of TLBO; at the same time, they outperform those of the largest part of some other algorithms except those with CMA_ES. Second, the convergence speeds of SAMCCTLBO are smaller than those of other algorithms for 17 functions among 10-dimensional functions and 11 functions among 30-dimensional functions. Third, the successful ratios of SAMCCTLBO are 100 % for 15 functions among the 24 test functions, and are lower than the part of some other algorithms for other nine functions, when the algorithm can converge to the acceptable solutions in all runs for 10- and 30-dimensional functions. Finally, statistical analysis indicates that SAMCCTLBO performs significantly better than the compared algorithm for the largest functions among the 24 test functions and especially for 30-dimensional functions. Although the proposed SAMCCTLBO algorithm has good performance on most of the functions, it does not perform the best for all benchmark functions in all different dimensions. According to the theorem of "no free lunch" (Wolpert and Macready 1997) one algorithm cannot offer better performance than all the others on every aspect or on every kind of problem. This is also observed in our experimental results. For example, some algorithms have good performance in terms of the mean and the standard deviations for large part of functions, but the mFEs or the successful ratios are worse. In summary, the proposed SAMCCTLBO algorithm has good performance, and are especially superior, at least not worse to all variants of TLBO for rotated benchmark functions and high-dimensional benchmark functions.

5 Conclusions

This paper presents a multi-class TLBO algorithm with combining the operator of simulated annealing algorithm; the mean value of sub-class is used in teacher phase to update the position of student in each sub-class. All learners also learn knowledge from others in their sub-class. Compare to original TLBO, there is no removing duplicate process in SAMCCTLBO, the diversity of different sub-classes is maintained by regrouped of learners, and the diversity of whole population is improved by simulated annealing operator. The results of experiments indicate that the performance of SAMCCTLBO is not the best for all benchmark functions, but it has good performance of solution accuracy and convergence speed for rotated benchmark functions and high-dimensional benchmark functions. Because TLBO is a young algorithm, it is not widely used in many practical problems. Future work will emphasize on extensive study of the applications in more complex practical optimization problems to fully investigate the properties and evaluate the performance of SAMCCTLBO.

Acknowledgments This work is partially supported by Natural Science Foundation of Anhui Province, China, (Grant No. 1308085MF82), National Natural Science Foundation of China (Grants No. 41475017, No. 61403304, No. 61304082), Sci-tech talents cultivation Fund projects of HuaiBei city (Grant No. 20110304). The authors would like to thank Prof. Suganthan who provided some code of PSOs and the code provided by Y. Wang, Z. Cai and Q. Zhang (http://dces.essex.ac.uk/staff/qzhang/mypublication.htm).

References

Adil B, Alper H, Simge YK (2014) Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: flow shop and job shop scheduling cases. Inf Sci 276:204–218

Arnold DV, Hansen N (2012) A (1+1)-CMA-ES for constrained optimisation. GECCO, Philadelphia

Bergh FVD, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evolut comput 8(3):225–239

Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evolut Comput 10(6):646–657

Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73

Dekkers A, Aarts E (1991) Global optimization and simulated annealing. Math Program 5:367–393

Dor AE, Clere M, Siarry P (2012) A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization. Comput Optim Appl 53(1):271–295

Floudas CA, Gounaris CE (2009) A review of recent advances in global optimization. J Global Optim 45:3–38

Hossein H, Taher N, Seyed IT (2011) A Modified TLBO algorithm for placement of AVRs considering DGs, 26th international power system conference, 2011, pp 1–8

Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Proceedings of IEEE congress Ecol. Comput. Honolulu, HI, pp 1671–1676

Li CH, Yang SX, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. IEEE Trans Syst Man Cybernet Part B 42(3):627–646

Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evolut Comput 10(3):281–294

Mendes R, Kenedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. IEEE Trans Evolut Comput 8(3):868–873

Naik A, Satapathy SC, Parvathi K (2012) Improvement of initial cluster center of c-means using teaching–learning based optimization. 2nd international conference on communication, computing and security. Proc Technol 6:428–435

Niknam T, Golestaneh F, Sadeghi MS (2012) θ-Multi-objective teaching–learning-based optimization for dynamic economic emission dispatch. IEEE Syst J 6(2):341–352

Niu B, Zhu YL, Hw XX, Henry W (2007) MCPSO: a multi-swarm cooperative particle swarm optimizer. Appl Math Comput 185:1050–1062

Parsopoulos KE, Vrahatis MN (2004) UPSO—a unified particle swarm optimization scheme, in lecture series on computational sciences, pp 868–873

Peram T, Veeramachaneni K, Mohan CK (2003) Fitness-distance-ratio based particle swarm optimization. In: Proceedings of swarm intelligence symposium, 2003, pp 174–181

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evolut Comput 13(2):398–417

Rao RV, Patel V (2011) Multi-objective optimization of combined Brayton and inverse Brayton cycles using advanced optimization algorithms. Eng Optim 44(8):965–983

Rao RV, Patel V (2012) An elitist teaching–learning-based optimization algorithm for solving complex constrained optimization problems. Int J Ind Eng Comput 3:535–560

Rao RV, Patel V (2013) Multi-objective optimization of heat exchangers using a modified teaching–learning-based optimization algorithm. Appl Math Model 37(3):1147–1162

Rao RV, Savasni VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15

Rao RV, Savasni VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aid Des 43(3):303–315

Rao RV, Savsani VJ, Balic J (2011) Teaching–learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Eng Optim 44(12):1447–1462

Rao RV, Waghmare GG (2014) A comparative study of a teaching–learning-based optimization algorithm on multi-objective unconstrained and constrained functions. J King Saud Univ Comput Inf Sci 26:332–346

Sabat SL, Ali L, Udgata SK (2011) Integrated learning particle swarm optimizer for global optimization. Appl Soft Comput 11:574–584

Suganthan PN (1999) Particle swarm optimizer with neighborhood operator. In: Proceedings of conference Evol. Comput, Washington, DC, pp 1958–1962

Suresh CS, Anima N (2011) Based data clustering, on teaching–learning-based optimization SEMCCO, 2011 Part II. LNCS 7077 (2011):148–156

Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of IEEE Congr. Evol. computation, 1998:69–73

Tang K, Yao X, Suganthan PN et al (2007) Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Nature Inspired Computation and Applications Laboratory, USTC, China

Vedat T (2012) Design of planar steel frames using teaching–learning based optimization. Eng Struct 34:225–232

Waghmare G (2013) Comments on "a note on teaching–learning-based optimization algorithm". Inf Sci 229:159–169

Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evolut Comput 14(1):55–66

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1(1):67–82

Xu Y, Wang L, Wang SY, Liu M (2015) An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. Neurocomputing 148:260–268

Yu K, Wang X, Wang Z (2014) An improved teaching–learning-based optimization algorithm for numerical and engineering optimization problems [J]. J Intel Manufact. doi:10.1007/s10845-014-0918-3

Yao X, Liu Y, Lin GM (1999) Evolutionary programming made faster. IEEE Trans Evolut Comput 3(2):82–102

Zhang JQ, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evolut comput 13(5):945–958

Zhang JZ, Ding XM (2011) A multi-swarm self-adaptive and cooperative particle swarm optimization. Eng Appl Artif Intel 24:958–967

Zou F, Wang L, Lei XH, Chen DB et al (2013) Multi-objective optimization using teaching–learning-based optimization algorithm. Eng Appl Artif Intel 26:1291–1300