Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **4**, November **2023**

Two-Stage Guided Constraint Differential Evolution Algorithm

Tien-Wen Sung*

College of Computer Science and Mathematics
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, Fujian, 350118, China
tienwen.sung@gmail.com

Qiaoxin Liang

College of Computer Science and Mathematics
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, Fujian, 350118, China
773590899@qq.com

Chuanbo Hong

College of Computer Science and Mathematics
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, Fujian, 350118, China
935619954@qq.com

Zeming Huang

College of Computer Science and Mathematics
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, Fujian, 350118, China
1216681301@qq.com

Wei Li

School of Electronic, Electrical Engineering and Physics
Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technology
Fuzhou, Fujian, 350118, China
1159696071@qq.com

Trinh-Dong Nguyen

Department of Software Engineering
University of Information Technology
Ho Chi Minh City, Vietnam
dongnt@uit.edu.vn

*Corresponding author: Tien-Wen Sung

Received February 13, 2023, revised May 26, 2023, accepted August 11, 2023.

Abstract. *The development of population intelligence has shown a great trend of a linear surge in recent years, and a large number of intelligent algorithms inspired by biology have been studied. Among them, differential evolution (DE) algorithms have attracted the attention of many researchers because of their simplicity, high efficiency, powerful search capability, etc. DE has strong exploitability and can approach the global optimum faster, but it cannot avoid the phenomenon of search stagnation and premature convergence. To address the drawbacks of DE, the two-stage bootstrap constrained differential evolution algorithm (TSGCDE) is proposed in this paper. In TSGCDE, instead of the traditional one population, the population is divided into two as a way to increase the diversity of the population. Divided into two stages, each stage searches for optimal solutions in different directions in different populations, and the optimal solutions and the bootstrap solutions generated by the populations in the two stages are bounded by each other's bootstrap, which makes the DE have a more powerful optimal-seeking ability and easier to jump out of the local optimum. To validate the performance of TSGCDE, we conducted experiments on CEC2013 and CEC2014 benchmarks with some well-known algorithm variants, and the experimental results show that our proposed algorithm is competitive.*
Keywords: Differential evolution, Swarm intelligence, Adaptive parameter control

---

1. **Introduction.** As networks and technologies continue to mature and evolve, optimization problems are everywhere in every neighborhood. Optimization problems have gradually become one of the indispensable research methods and fundamental theories in existing science and technology. Finding the optimal state is a basic criterion for the current society. The development of population intelligence has shown a great trend of linear soaring, and more and more intelligent algorithms inspired by biology are being studied. For biological evolution, the principle of survival of the fittest to the environment is always maintained [1, 2].

In the early stage, various intelligent optimization algorithms were proposed by scholars. Particle Swarm Optimization (PSO) [3] was first proposed by Kennedy and Eberhart in 1995 to simulate the behavior of a flock of birds foraging for food. In the optimization process, each optimization problem is treated as a bird, which is called a "particle". Among all the particles, each particle is endowed with a memory function, which can clearly remember the best location to search for, and each particle has a speed to determine the direction and distance of the flight. In the process of flight, the speed is dynamically adjusted by its own flight experience and combined with the flight experience of its companions, and finally, the current position is judged by the set fitness function. The Ant Colony Algorithm (ACA) [4] was first proposed by Marco Dorigo in 1992. The algorithm simulates ant foraging behavior. When ants search for a food source, they not only release a pheromone in their pathway to being perceived by other ants but also can perceive the pheromone released by their peers at the same time. The distance of a path is mainly determined by the concentration of pheromone, where a low concentration of pheromone means that the path chosen is longer, while a higher concentration of pheromone means that the path has a shorter distance. For ants, they usually have a higher probability to choose the path with higher concentration. At the same time, ants on that path will release a certain amount of pheromone in order to increase the pheromone concentration in that path. This whole process is a kind of positive feedback. In the process of continuous feedback, ants will find an optimal foraging route to minimize the foraging distance. Simulated Annealing (SA) [5] was applied to combinatorial optimization by Kirkpatrick et al. in 1983, and the idea originated from the solid annealing process. First, the temperature of a solid is raised to a certain level, and then slowly lowered. During the whole process of temperature increase, in order to let the internal energy of the solid gradually increase, the internal disorderly particles will be formed. As the temperature decreases, these disordered particles will become ordered again, and the whole process will be an equilibrium process, with the ultimate goal of minimizing the internal energy. The Metropolis process is repeated at a certain temperature with the objective function satisfying the Boltzmann probability distribution. The Artificial Bee Colony (ABC) algorithm [6] is an intelligent algorithm first proposed by Karaboga. ABC is an algorithm proposed to imitate honey bees for honey harvesting and is mainly divided into three kinds of bee colonies: hiring bees, observation bees, and scouting bees. When the hired bee is collecting honey, it will keep searching for better nectar sources, and when it finishes its task and returns to the hive, it will share the information it has found by dancing to the observation bees. The observer bee probabilistically selects one of the many known nectar sources and follows one of the hired bees to collect nectar, while the observer bee searches for a better nectar source in the neighborhood of that source. When the hired bee and the observation bee can no longer search for a better nectar source in the corresponding neighborhood, the nectar source is abandoned and the hired bee is transformed into a scout bee, searching for a new nectar source in large strides. Genetic Algorithm (GA) [7] is an optimization model that simulates the Darwinian theory of biological evolution and was first proposed by John holland. GA focuses on modeling the evolutionary process of organisms, and it divides the problem solving process into selection, crossover and variation. A probabilistic approach to finding the best solution is used.

The Differential Evolution (DE) algorithm [8] is one of the more competitive evolutionary algorithms, which was first proposed in 1995 by Storn et al. DE can be dynamically tracked according to the current search situation. And the optimization is performed by variation, crossover and selection of the difference vector between parents. During evolution, DE evaluates the best parameter vector for each generation and records its minimization process, which can obtain better convergence [9, 10, 11]. As a simple and efficient global optimization algorithm, DE has powerful search and optimization capabilities, but for some specific optimization problems, search stagnation and premature convergence are still unavoidable. To solve these problems, this paper proposes a new DE algorithm, divided into two stages, instead of the traditional population, the population is divided into two, to increase the diversity of the population. The optimal solution is searched in different directions in the two phases of the population, and the optimal solution and the bootstrap solution generated in the two phases of the population are bounded by each other, which makes the DE have a more powerful ability to find the optimal solution. For the proposed new DE algorithm in this paper, the experimental tests are mainly analyzed and compared by the CEC2013 benchmark set [12] and the CEC2014 benchmark set [13], and the new DE algorithm is evaluated and tested in an all-around way by experiments. This article has the following key points:

(1) The mutation operation is divided into two stages, conducted in two regions, to better ensure the diversity of the population, which can be explored to a larger range and find better results.

(2) A two-stage bootstrap constraint model is established to make the two-stage populations generate the optimal solution and the bootstrap solution respectively with mutual bootstrap constraints, which enhances the information exchange between excellent individuals and makes it easier to jump out of the local optimal solution and thus find the global optimal solution.

(3) In this paper, CEC2013 and CEC2014 benchmark sets are used to test and compare and analyze some well-known ABC variants and DE variants with our new DE algorithm. Through the analysis of the experimental results, it can be seen that the proposed new DE variant has great advantages.

In the following chapter descriptions, the following steps are mainly followed. Chapter 2 of the article provides a detailed description of the development of DE algorithms from ancient times to the present. In the third chapter of the article, the main innovations of our proposed new DE algorithm are described. In the fourth chapter of the paper, the main performance of the proposed new DE algorithm is evaluated in all aspects, and these evaluations are mainly achieved by two test sets, CEC2013 and CEC2014. In the last part of the paper, a systematic description of the whole paper is summarized.

2. **Related Works.** The DE algorithm is one of the most widely used stochastic real-parameter optimization algorithms. The DE algorithm mainly uses random selection and differences in the proportions of individuals from different populations to disturb contemporary population individuals, so it is not necessary to apply a separate probability distribution to generate offspring [14, 15]. DE has powerful search and optimization capabilities. In the DE algorithm, there are three control parameters, which are population size NP, scaling factor F, and crossover probability CR. In practical applications, the DE algorithm develops several deformation situations, and the general conventions that define these deformation forms are given below. Where in DE/x/y/z, x indicates that the vector of mutants is random or optimal, y indicates the number of difference vectors utilized, and z indicates the scheme of a certain crossover. The mutation strategy can be written in the following forms: DE/rand/1/z, DE/best/1/z, DE/target-to-best/1/z, DE/best/2/z, and DE/rand/2/z. The specific form is shown in Table 1. The two crossover schemes are described by DE/x/y/bin (binomial crossover) and DE/x/y/ exp (exponential crossover) [16, 17].

Table 1. Mutation forms for DE algorithm

Number	DE/x/y/z	Equation
1	DE/rand/1/z	$v_i^{G+1} = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G)$
2	DE/rand/1/z	$v_i^{G+1} = x_{best}^G + F \cdot (x_{r1}^G - x_{r2}^G)$
3	DE/target-to-best/1/z	$v_i^{G+1} = x_i^G + F_1 \cdot (x_{best}^G - x_i^G) + F_2 \cdot (x_{r1}^G - x_{r2}^G)$
4	DE/best/2/z	$v_i^{G+1} = x_{best}^G + F_1 \cdot (x_{r1}^G - x_{r2}^G) + F_2 \cdot (x_{r3}^G - x_{r4}^G)$
5	DE/rand/2/z	$v_i^{G+1} = x_{best}^G + F_1 \cdot (x_{r1}^G - x_{r2}^G) + F_2 \cdot (x_{r3}^G - x_{r4}^G)$

The operation procedure of the standard DE/best/1 algorithm is divided into several steps, the first step for population initialization, the second for mutation, the third for crossover, followed by the fourth step for selection, and the last step for boundary condition processing.

2.1. **Initialization.** Initialization of populations is generally a necessary step to establish initial points and facilitate optimal search. DE utilizes NP vectors of real-valued parameters of dimension D and treats them as a population for each generation, where the individuals are represented as in Equation 1.

$$x_i^G (i = 1, 2, \ldots, NP) \tag{1}$$

In Equation 1, NP is the population size; G is the number of generations evolved, and i denotes the sequence of individuals. During the minimization process, NP is kept constant. The initialized population is as in Equation 2.

$$x_{ji}^G = \mathrm{rand}[0, 1] * (x_{maxj} - x_{minj}) + x_{minj} \tag{2}$$

In Equation 2, $j = 1.2, \cdots, D$; x_{maxj} and x_{minj} are bounds on the parameter variables; rand [0,1] is a uniformly distributed random number between 0 and 1.

2.2. **Mutation.** For all target vectors x_i^G, The basic DE algorithm randomly select2 vectors from the NP, x_{r1}^G, x_{r2}^G and x_{best}^G is the current local optimum, and the selected vector is subjected to a variation operation on the population to obtain the variation v_i^{G+1} as in Equation 3.

$$v_i^{G+1} = x_{best}^G + F * (x_{r1}^G - x_{r2}^G) \tag{3}$$

In Equation 3, the two randomly selected ordinal numbers $r1$ and $r2$ are two ordinal numbers that are not equal to each other, and $r1$ and $r2$ are also different from each other with the target vector ordinal number i. The scaling factor $F \in [0, 2]$, is a constant real factor whose main role is to control the scaling of the deviation vector. If the value of F is too small, it may lead to "premature" algorithm, and the process of increasing F can well prevent the algorithm from falling into a local optimum. However, when F is larger than 1.0, the convergence of the population will become worse as appropriate. Storn et al. [8] suggested that the initial value of F is set to 0.5 and the effective range of F is in the range of 0.4-1.0.

2.3. **Crossover.** The crossover operation exists so that the diversity of the interference parameter vector is increased. For the $G + 1st$ generation population, the crossover operation is performed between the original individuals and the individuals after the mutation operation according to a certain probability, resulting in a new crossover vector u_{ji}^{G+1}, the trial vector u_i^{G+1}, as in Equation 4 and Equation 5.

$$u_i^{G+1} = (u_{1i}^{G+1}, u_{2i}^{G+1}, \ldots, u_{Di}^{G+1}) \tag{4}$$

$$u_{ji}^{G+1} = \begin{cases} v_{ji}^{G+1}, \text{rand}\, b(j) \le CR \text{ or } j = rnbr(j) \\ x_{ji}^G, \text{rand}\, b(j) > CR \text{ or } j \ne rnbr(j) \end{cases} \tag{5}$$

In Equation 5, $i = 1, 2, \ldots, NP$; $j = 1.2, \ldots, D$; $rnbr(j)$ is the jth estimate of the random number generator between 0 and 1; CR denotes the crossover probability, which takes values in the range of [0,1], and a larger CR indicates a higher probability of crossover, and a larger CR usually leads to faster convergence. $rnbr(j)$ denotes a randomly chosen sequence, $rnbr(j) \in (1.2, \ldots, D)$, which is mainly used to ensure that u_{ji}^{G+1} obtains at least one parameter from v_{ji}^{G+1}.

2.4. **Selection and Boundary Condition Processing.** The DE algorithm compares x_i^G with u_i^{G+1} rows according to the greedy criterion, from which the child with the smaller value in the fitness function $f(t)$ is selected for retention as the next generation for differential evolution, i.e., Equation 6.

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, & f\left(u_i^{G+1}\right) \le f\left(x_i^G\right) \\ x_i^G, & f\left(u_i^{G+1}\right) > f\left(x_i^G\right) \end{cases} \tag{6}$$

In the process of differential evolution, it is necessary to ensure that the parameter values of the newly generated individuals are in the feasible domain range. One way to do this is to perform absorption of the boundaries, i.e., to set the values of the individuals that are not in the feasible domain range to the adjacent boundary values.

2.5. **DE Algorithm and Its Variants.** The DE algorithm has attracted the attention of many researchers because of its more powerful optimization and search capabilities, and researchers have investigated different aspects of the problem and proposed a series of improvement methods. Brest et al. [18] proposed in 2005 the jDE algorithm, a variant of DE with adaptive control parameters, which compiles the F and CR parameters in DE into individuals and updates them using an evolutionary approach. The mutation, crossover and selection operations of new individuals are affected. Zhang and Sanderson [19] developed and improved the jDE in 2008 and proposed the JADE algorithm. In the JADE algorithm, the two parameters F and CR are adaptively controlled and automatically updated to the appropriate values, more efficiently avoiding the optimization problem in relation to the parameter settings. The parameters obey semi-fixed distributions during the evolution and are more optimally updated dynamically according to the resulting children among them. Tanabe and Fukunaga [20] proposed the SHADE algorithm in 2013, which uses the historical memory of control parameter settings to guide the selection of future control parameters and is an advanced variant of the DE algorithm. Then Tanabe and Fukunaga [21] proposed the LSHADE algorithm in 2014, which further extended SHADE by further improving the adaptation scheme of the control parameters and making the population size decreasing according to a linear function. Brest et al. [22] proposed the iLSHADE algorithm to further optimize SHADE by calculating the next generation of historical memory values based on the historical memory values of the previous generation. for these DE variants, all have some drawbacks, there are good F with poor for these DE variants, there is either a combination of good F with poor CR or poor F with good CR, making the resulting test vector better, but the poor parameters self-adapting. For this deficiency, Meng et al. [23] proposed the LPALMDE algorithm, which divides the control parameters into different groups and updates them independently in their respective groups, which is a new adaptation for the control parameters.

Meng et al. [24] proposed the Quasi-Affine TRansformation Evolutionary (QUATRE) algorithm in 2016. The QUATRE algorithm was first proposed on top of the original DE algorithm, which improved the original DE algorithm for the deficiency of position bias and used co-evolution for iterative updates, giving the QUATRE algorithm a greater competitive advantage over the DE algorithm. Meng et al. [25] proposed CS-DE algorithm in 2021, CS-DE proposes two mutation strategies, one mutation strategy for storing poorer solutions and the other for storing historical solutions. Elliptic linear decay is used in CS-DE to reduce the population size. Sutton et al. [26] proposed the RBDE algorithm. In the RBDE algorithm, the better individual will be selected, the probability of selection is determined by the goodness of the individual itself, and the children of the algorithm are optimally selected based on the selection of the parent. Das and Abraham [27] proposed the DEGL algorithm in which the parameter control is in the form of random generation and the quality control parameters are easily retained.

3. **Proposed Approach.** The traditional DE algorithm has a multidimensional search strategy and has a strong development performance. In the DE algorithm, there are three control parameters, which are population size NP, scaling factor F and crossover probability CR. In the whole search process, the DE algorithm can approach the global optimum relatively quickly. Due to the inferior search performance of the algorithm in the late iteration, it is easy to converge prematurely. For some specific optimization problems, the DE algorithm cannot avoid the occurrence of search stagnation and premature convergence [28]. In order to solve these problems, this paper improves DE/best/1 and proposes a new DE algorithm, in which the search process is divided into two phases, i.e., instead of the traditional one population, the population is divided into two, so as to

increase the diversity of the population. The optimal solution is searched in different directions in the two phases, and the optimal solution and the bootstrap solution generated in the two phases are constrained to attract each other, which makes it easier for the DE algorithm to go beyond the local optimum to obtain the global optimum solution, thus making the DE more powerful in finding the optimal solution.

3.1. **Update of Control Parameters F and CR.** For the traditional DE/best/1 algorithm, a fixed F and CR is used, which will greatly limit the optimal value of the resulting individuals. It is shown that using adaptively varying F and CR will produce better individuals, which will be more likely to survive and produce offspring. Inspired by JADE, an adaptive form is used for the control constants F and CR, where F and CR are updated as shown in Equation 7 and Equation 8.

$$\begin{cases} F_i^{G+1} = \operatorname{rand} c_i\left(\theta_F, 0.1\right) \\ \theta_F = \operatorname{mean}\left(C_F^G\right) * k + \theta_F * (1-k) \\ \operatorname{mean}\left(C_F^G\right) = \frac{\sum_{F\in C_F} F^2}{\sum_{F\in C_F} F} \end{cases} \tag{7}$$

$$\begin{cases} CR_i^{G+1} = \operatorname{rand} d_i\left(\theta_{CR}, 0.1\right) \\ \theta_{CR} = \operatorname{mean}\left(C_{CR}^G\right) * k + \theta_{CR} * (1-k) \end{cases} \tag{8}$$

In Equation 7 and Equation 8, F_i^{G+1} is the Corsi distribution generated according to the position parameter θ_F and the scale parameter 0.1. In the experiment, the initial value of θ_F is set to 0.5. Where k is a positive number between 0 and 1, mean$(\cdot)$ is used to calculate the arithmetic mean. C_F^G is the set of all successful mutation factors in generation G. CR_i^{G+1} is a normal distribution generated based on a mean of θ_{CR} and a standard deviation of 0.1, C_{CR}^G is the set of all successful crossover probabilities in generation G. During the experiment, we set the initial value of θ_{CR} to 0.5. and all $CR \in [0,1]$while the new $F \in [0,1]$. In the initialization operation stage, the same as Equation 2.

3.2. **New Mutation Strategy.** In the new variation strategy, there are two stages of finding the optimum, in which the population is divided into two fixed subpopulations. In these two stages, the optimal solution population NP_1 will be generated in the first stage population and the population NP_2 will be generated in the second stage with the bootstrap solution, where the magnitude of the number of individuals in these two populations will be derived from Equation 9.

$$\begin{cases} N_{NP_1} = NP * \mu \\ N_{NP_2} = NP * (1-\mu) \end{cases} \tag{9}$$

In Equation 9, N_{NP_1} is the number of optimal solution populations generated in the first stage, and NP is the total population size. μ is the parameter that determines the size of both populations. N_{NP_2} is the number of bootstrap populations generated in the second stage. μ is the parameter that determines the size of the two populations. μ is particularly important in the algorithm we design because its size determines the difference in population size between the two stages. As you know, the size of the population usually has a serious impact on the performance of the designed algorithm, so it is important to choose the most suitable μ value for the new algorithm we design. In the subsequent experiments, the specific value of μ will be explained in detail.

For global optimization algorithms, finding the global optimal solution is the primary task. In the exploration update, the global optimal solution may have the neighborhood of the local optimal solution and the guide solution, which indicates that there is mutual information between the local optimal solution and the neighborhood of the guide solution, and the information interaction between individuals is stronger in QUATRE-DEG [29]. Inspired by QUATRE-DEG, a model based on two-stage bootstrap constraint is established, which can make the optimal solution generated in two stages respectively with The mutual bootstrap constraint between the optimal solutions generated in each of the two stages and the bootstrap solution, thus enhancing the information exchange between the best individuals, which will make the original DE algorithm easier to jump out of the local optimum and thus better find the global optimal solution. The two-stage bootstrap constraint model for the global optimal solution x_{best1} generated in the first stage and the bootstrap solution x_{best2} generated in the second stage is shown in Equation 10.

$$\begin{cases} GC_{12} = C * \left(\frac{Ub-Lb}{2} * \frac{x_{\text{best }2} - x_{\text{best }1}}{d_{12}} * h(\varphi)\right) \\ \varphi = \frac{d_{12}}{D} + 2 \\ C = C_{\max} - \frac{C_{\max} - C_{\min}}{M} * G \\ h(\varphi) = \exp\left(-\frac{\varphi}{\delta_1}\right) * \delta_2 - \exp(-\varphi) \end{cases} \tag{10}$$

In Equation 10, GC_12 is the bootstrap force between the first-stage global optimal solution and the second-stage bootstrap solution. where Lb and Ub are the lower and upper bounds of the search space, respectively. d_{12} is the Euclidean distance between x_{best1} and x_{best2}. D represents dimensionality. C is the decreasing coefficient, by which C can avoid over aggregation of populations and better reduce the probability of the algorithm falling into local optimum. M is the total number of iterations and G is the current generation. In the experiment, we set $C_{max} = 1$, $C_{minx} = 0.00001$, and set δ_1 and δ_2 in $h(\varphi)$ to the values of 1.5 and 0.5, respectively. According to the study, in the $h(\varphi)$ function, the function value is 0 when φ =2.079, and when φ gradually increases, the function value increases first and then decreases. The maximum value of the function is 0.02 and the minimum value is 0. A bootstrap constraint model influenced by distance is established by using this feature of $h(\varphi)$. When x_{best1} and x_{best2} are far away from each other, the guiding force is smaller, conversely, the closer the distance, the greater the guiding force. Using the bootstrap constraint model, a new variational strategy is proposed as shown in Equation 11.

$$\begin{cases} v_{NP_{1i}^G}^{G+1} = x_{\text{best1}}^G + F * \left(x_{r1}^G - x_{r2}^G\right) - GC_{12}^G \\ v_{NP_{2i}}^{G+1} = x_{\text{best2}}^G + F * \left(x_{r3}^G - x_{r4}^G\right) + GC_{12}^G \end{cases} \tag{11}$$

In Equation 11, v_{NP1i}^{G+1} is the variance vector generated in the optimal solution population in the first stage. And v_{NP2i}^{G+1} is the variance vector generated in the second stage bootstrap solution population. Where x_{best1}^G is the optimal solution vector and x_{best2}^G is the bootstrap solution vector, x_{r1}^G and x_{r2}^G are two mutually unequal vectors randomly selected from the NP. GC_{12}^G is the attraction between the optimal solution vector and the guiding solution vector. There are different operations for each subpopulation in the variation process. According to the distance, the two operations are shown as negative correlation and positive correlation respectively. This strategy can achieve better information interaction in the search process. The global suboptimal is used as a guiding constraint when the search process is centered on the global optimal. At the same time, the global optimum is used as a guiding constraint when the search is centered on the global suboptimal. The populations in the two phases interact with each other, and it is easier to go beyond the local optimum to find the global optimum solution in the search process. In the next crossover operation and selection operation, the same as the original DE/best/1, the same as Equation 5 and Equation 6.

3.3. **Algorithm of TSGCDE.** The pseudo-code of Algorithm of TSGCDE is shown in Algorithm 1.

Algorithm 1 TSGCDE

Initialization: preset population size NP, dimension D, maximum number of iterations M, parameters θ_F, θ_{CR}, μ, C_{max}, C_{min}, δ_1, δ_2 and boundary Lb, Ub.
According to Equation (2), the initial population NP is randomly generated, the fitness values of all individuals in the initial population are calculated, and after ranking the fitness, the first-stage optimal individual $Xbest1$ and the second-stage bootstrap individual $Xbest2$ are obtained.
According to Equation (9), the optimal individual and the bootstrap individual are divided into two spaces to generate the first stage global optimal solution vector x^G_{best1} and the second stage bootstrap solution vector x^G_{best2}.
for g=1 to M **do**
  for i=1 to NP **do**
    According to Equation (7)(8), the F and CR values are generated, $CR \in [0, 1]$, $F \in [0, 1]$.
  end for
  for i=1 to μ*NP **do**
    In the process of generating new individuals, ensure that all individuals are not equal to each other, and generate $r1$, $r2$ matrices of 1 to μ*NP in order.
  end for
  According to Equation (10), the relative position d_{12} of the optimal individual in the first stage and the guide individual in the second stage is calculated.
  Calculate the guiding force GC^G_{12}. according to Equation (10).
  for i= 1 to μ*NP **do**
    According to Equation (11) mutant individuals generate the optimal population of individuals $v^{G+1}_{NP_{1i}}$.
  end for
  for i= 1 to NP-μ*NP **do**
    In the process of generating new individuals, ensure that all individuals are not equal to each other and generate $r3$, $r4$ matrices of 1 to NP-μ*NP in order.
    According to Equation (11) mutant individuals generate the optimal population of individuals $v^{G+1}_{NP_{2i}}$.
  end for
  Merge $v^{G+1}_{NP_{1i}}$ and $v^{G+1}_{NP_{2i}}$ into a new population V.
  Crossover operation according to Equation (5)
  Boundary processing.
  Calculate the fitness of each individual.
  Select operation according to Equation (6).
  Update the current optimal individual $Xbest1$ and the current bootstrap individual $Xbest2$.
  Generate a new one-stage global optimal matrix x^G_{best1} and a two-stage global suboptimal matrix x^G_{best2} .
end for
Output global most individual x_{best1}, global optimal value $f(x_{best1})$

4. **Experiment and Analysis.** We used CEC2013 and CEC2014 benchmark sets to verify the algorithm performance, and carried out optimization experiments on 58 test functions. In these benchmark sets, the argument range is set to [-100, 100]. All the functions are considered as black-box problems. 28 test functions are included in the CEC2013 test set. In CEC2013, fa1-fa5 are unimodal functions, fa6-fa20 are basic multimodal functions, and fa21-fa28 are composition functions. fb1-fb3 are unimodal functions, fb4-fb16 are simple multimodal functions, fb17-fb22 are hybrid functions, and fb23-fb30 are composition functions. The algorithm experiments were all carried out on the simulation platform MATLAB2021a, and the running device processor was 11th Gen Intel(R) Core(TM) i5-11300H..

4.1. **Parameter Settings.** The μ parameter is particularly important in our algorithm, which determines the size of the two-stage population, and this will have a large impact on the performance of the algorithm. In general, the population size of the first-stage optimal solution is not allowed to be smaller than the second stage, so this way is more able to find the global optimal solution. For μ, the smaller its value the more populations are allocated to the second stage, and the larger its value the more populations are allocated to the first stage. Therefore, in this experiment, the values of μ were set to 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, and the mean and variance of the run results were compared by running these algorithms 51 times each in dimension 30D. The results are shown in Table 2. We can see from the table that our algorithms are competitively combined optimal when μ is 0.6, so in the next experiments, setting μ to 0.6.

TABLE 2. Optimization performance of 30D with different μ values under CEC2013 test set.

TSGCDE	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{a1}	2.55E-06/5.30E-07	2.39E-06/4.91E-07	2.44E-06/6.07E-07	2.49E-06/5.66E-07	2.40E-06/6.29E-07
f_{a2}	6.34E+05/3.09E+06	7.76E+05/3.74E+06	1.88E+06/6.72E+06	1.29E+06/4.79E+06	1.15E+06/4.05E+06
f_{a3}	5.72E+07/8.63E+07	6.24E+07/1.18E+08	8.13E+07/1.78E+08	9.48E+07/1.74E+08	1.47E+08/3.17E+08
f_{a4}	1.48E+04/1.59E+04	1.27E+04/1.60E+04	1.33E+04/1.62E+04	1.26E+04/1.60E+04	1.23E+04/1.51E+04
f_{a5}	1.10E-06/4.22E-07	1.02E-06/3.97E-07	1.07E-06/3.37E-07	1.02E-06/3.37E-07	1.00E-06/3.41E-07
f_{a6}	1.21E+01/1.40E+01	1.15E+01/6.96E+00	1.23E+01/1.24E+01	1.26E+01/8.85E+00	1.16E+01/6.80E+00
f_{a7}	2.79E+01/1.46E+01	2.77E+01/1.46E+01	2.80E+01/1.30E+01	3.28E+01/1.55E+01	3.19E+01/1.73E+01
f_{a8}	2.09E+01/4.91E-02	2.08E+01/3.28E-01	2.09E+01/5.03E-02	2.09E+01/5.07E-02	2.09E+01/5.62E-02
f_{a9}	2.69E+01/2.81E+00	2.64E+01/3.86E+00	2.68E+01/2.72E+00	2.44E+01/5.01E+00	2.61E+01/3.87E+00
f_{a10}	5.16E-02/3.14E-02	4.89E-02/3.37E-02	4.79E-02/3.25E-02	4.73E-02/2.76E-02	4.99E-02/3.53E-02
f_{a11}	3.69E+00/4.04E+00	2.24E+00/3.89E+00	1.39E+00/2.74E+00	7.41E-01/2.08E+00	1.35E+00/2.61E+00
f_{a12}	4.80E+01/1.15E+01	4.54E+01/1.13E+01	4.97E+01/1.18E+01	4.92E+01/1.24E+01	4.94E+01/1.24E+01
f_{a13}	9.42E+01/2.79E+01	9.39E+01/2.02E+01	9.49E+01/2.64E+01	9.59E+01/2.67E+01	1.05E+02/2.35E+01
f_{a14}	4.55E+01/6.53E+01	6.28E+01/9.60E+01	4.70E+01/6.53E+01	2.32E+01/4.61E+01	3.72E+01/8.50E+01
f_{a15}	4.43E+03/1.10E+03	4.36E+03/9.14E+02	4.56E+03/1.21E+03	4.28E+03/9.69E+02	4.36E+03/1.02E+03
f_{a16}	1.68E+00/8.78E-01	1.48E+00/9.55E-01	1.52E+00/9.22E-01	1.64E+00/9.45E-01	1.42E+00/9.60E-01
f_{a17}	3.05E+01/2.19E-01	3.04E+01/2.83E-04	2.86E+01/7.23E+00	3.04E+01/7.57E-04	3.04E+01/2.01E-02
f_{a18}	1.16E+02/4.59E+01	1.12E+02/4.48E+01	1.15E+02/3.40E+01	1.13E+02/3.74E+01	1.22E+02/2.92E+01
f_{a19}	1.60E+00/2.25E-01	1.54E+00/1.79E-01	1.54E+00/1.90E-01	1.56E+00/1.79E-01	1.62E+00/2.13E-01
f_{a20}	1.12E+01/7.06E-01	1.14E+01/7.78E-01	1.15E+01/6.65E-01	1.14E+01/6.55E-01	1.13E+01/7.81E-01
f_{a21}	2.83E+02/7.32E+01	3.04E+02/7.68E+01	2.78E+02/6.75E+01	2.87E+02/7.29E+01	2.89E+02/7.18E+01
f_{a22}	1.42E+02/8.21E+01	1.52E+02/9.49E+01	1.29E+02/6.78E+01	1.33E+02/7.05E+016	1.30E+02/6.56E+01
f_{a23}	4.40E+03/9.45E+02	4.16E+03/7.06E+02	4.37E+03/7.69E+02	4.12E+03/8.70E+02	4.12E+03/7.15E+02
f_{a24}	2.43E+02/1.43E+01	2.40E+02/1.53E+01	2.45E+02/1.56E+01	2.42E+02/1.49E+01	2.44E+02/1.31E+01
f_{a25}	2.63E+02/1.17E+01	2.63E+02/1.07E+01	2.62E+02/1.08E+01	2.64E+02/1.24E+01	2.62E+02/1.14E+01
f_{a26}	2.56E+02/7.34E+01	2.51E+02/7.28E+01	2.56E+02/7.35E+01	2.57E+02/7.50E+01	2.44E+02/6.83E+01
f_{a27}	8.41E+02/1.45E+02	8.80E+02/1.31E+02	8.47E+02/1.38E+02	8.30E+02/1.38E+02	8.15E+02/1.52E+02
f_{a28}	3.21E+02/1.48E+02	2.96E+02/2.80E+01	3.62E+02/2.51E+02	3.23E+02/1.63E+02	3.44E+02/2.22E+02
Best Mean std	3/3	10/11	4/4	6/4	5/6

The TSGCD algorithm is mainly an improvement of the original DE/best/1. TThe exploration strategy of ABC algorithm makes it competitive with other algorithms in the search process, and TSGCD increases the exploration ability of DE. TSGCD algorithm compares and analyzes some classical ABC variants and DE variants. Specifically, DE/rand/1, RBDE, DEGL, DE/best/1, MABC [30], and IABC [31]. We conducted 51 independent experiments for all algorithms with dimensions set to 30 D, 50 D, and 100 D. The mean and std (standard deviation) of the fitness error values of these test functions were collected for comparison. In the comparison results, "+", "=", and "-" indicate that our algorithm is "better", "similar", or "worse" than other algorithms, respectively [32]. The values of the parameters involved in all the algorithms used are always consistent with the initial recommended values to avoid ambiguities, as shown in Table 3.

Table 3. Parameter settings for experiment

Name	Initial parameter settings
DE/rand/1	NP=100,α =β=F=0.7, CR=0.5, $neighborhood\ size = 0.1 * NP$
RBDE	NP=100,F=0.7, CR=0.5, β=3.0
DEGL	NP=100,α =β=F=0.7, CR=0.5, $neighborhood\ size = 0.1 * NP$
DE/best/1	NP=100,F=0.7,CR=0.5
MABC	NP=50, $limit = D * NP/2$
IABC	NP=50, $limit = D * NP/2, MR = 0.3$
TSGCDE	NP=100,F=0.5,CR=0.9,θ_F=$\theta_C R$=0.5,$C_m ax$=1,$C_m inx$=0.00001,δ_1=1.5,δ_2=0.5, μ=0.6

4.2. **Experimental Analysis in the CEC2013 Test Set.** The results of the experiments comparing the dimensions of 30D, 50D and 100D in the CEC2013 test set are shown in Tables 4-6, respectively. From Table 4, it can be seen that in 30D real number optimization, the TSGCDE algorithm has 25 benchmark functions outperforming DE/rand/1, 26 benchmark functions outperforming RBDE, 21 benchmark functions outperforming DEGL, 1 function similar to DEGL, 17 benchmark functions outperforming DE/best/1, 1 function similar to DE/ best/1 is similar, 18 benchmark functions are better than MABC, and 16 benchmark functions are better than IABC. from Table 5, we can see that in 50D real number optimization, TSGCDE algorithm has 25 benchmark functions better than DE/rand/1, 25 benchmark functions better than RBDE, 1 function similar to RBDE, 21 benchmark functions are better than DEGL, 1 function is similar to DEGL, 20 benchmark functions are better than DE/best/1, 16 benchmark functions are better than MABC, 1 benchmark function is similar to MABC, and 17 benchmark functions are better than IABC. from Table 6, we can see that in 100D real number optimization, TSGCDE algorithm has 26 benchmark functions are better than DE/rand/1, 2 functions are similar to DE/rand/1, 26 benchmark functions are better than RBDE, 2 functions are similar to RBDE, 20 benchmark functions are better than DEGL, 3 functions are similar to DEGL, 22 benchmark functions are better than DE/best/1, 2 functions are similar to DE/best/1, 16 benchmark functions outperform MABC, 2 benchmark functions are similar to MABC, 16 benchmark functions outperform IABC, and 2 benchmark functions are similar to IABC. Overall, our algorithm outperforms these ABC variants as well as the DE variants.

4.3. **Experimental Analysis in the CEC2014 Test Set.** The results of the experiments comparing the dimensions of 30D, 50D and 100D in the CEC2014 test set are shown in Tables 7-9, respectively. From Table 7, it can be seen that in 30D real number optimization, the TSGCDE algorithm has 24 benchmark functions outperforming DE/rand/1, 2 functions similar to DE/rand/1, 25 benchmark functions outperforming RBDE, 2 functions similar to RBDE, 19 benchmark functions outperforming DEGL, 1 function similar to DEGL similar, 19 benchmark functions outperform DE/best/1, 2 functions are similar to DE/best/1, 19 benchmark functions outperform MABC, 1 function is similar to MABC, 17 benchmark functions outperform IABC, 2 functions are similar to IABC. From Table 8, we can see that in 50D real number optimization, TSGCDE algorithm has 27 benchmark functions better than DE/rand/1, 1 function similar to DE/rand/1, 27 benchmark functions better than RBDE, 1 function similar to RBDE, 20 benchmark functions better than DEGL, 1 function similar to DEGL Similar to DE/best/1, 20 benchmark functions outperform DE/best/1, 1 function is similar to DE/best/1, 19 benchmark functions outperform MABC, 18 benchmark functions outperform IABC. As can be seen from Table

Table 4. Comparison on 30D optimization under CEC2013 test suite with different algorithms μ.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{a1}	2.39E-06/ 4.91E-07	1.01E-10/ 3.01E-11(-)	5.43E-07/ 1.52E-07(-)	1.96E-13/ 7.90E-14(-)	1.20E-13/ 2.27E-13(-)	2.32E-13/ 2.52E-29(-)	3.88E-13/ 1.05E-13(-)
f_{a2}	7.76E+05/ 3.74E+06	1.01E+08/ 1.91E+07(+)	1.10E+08/ 2.11E+07(+)	3.23E+06/ 1.45E+06(+)	3.62E+07/ 3.57E+07(+)	1.11E+07/ 1.02E+07(+)	9.24E+06/ 3.32E+06(+)
f_{a3}	6.24E+07/ 1.18E+08	7.23E+08/ 1.90E+08(+)	1.73E+09/ 3.36E+08(+)	3.00E+08/ 4.31E+08(+)	3.90E+07/ 3.90E+07(-)	4.67E+08/ 1.66E+08(+)	2.83E+08/ 3.10E+08(+)
f_{a4}	1.27E+04/ 1.60E+04	4.12E+04/ 5.60E+03(+)	4.97E+04/ 6.35E+03(+)	6.37E+03/ 2.83E+03(-)	1.57E+04/ 1.57E+04(+)	7.72E+04/ 3.14E+03(+)	8.08E+04/ 1.05E+04(+)
f_{a5}	1.02E-06/ 3.97E-07	5.24E-06/ 1.25E-06(+)	6.59E-04/ 1.38E-04(+)	1.76E-13/ 5.71E-14(-)	1.14E-13/ 1.14E-13(-)	2.96E-13/ 1.27E-29(-)	5.37E-13/ 6.46E-14(-)
f_{a6}	1.15E+01/ 6.96E+00	2.32E+01/ 6.26E+00(+)	5.13E+01/ 1.28E+01(+)	2.21E+01/ 2.36E+01(+)	1.87E+01/ 1.87E+01(+)	2.65E+01/ 9.54E+00(+)	1.84E+01/ 4.21E+00(+)
f_{a7}	2.77E+01/ 1.46E+01	5.24E+01/ 4.88E+00(+)	6.28E+01/ 5.05E+00(+)	5.64E+01/ 2.01E+01(+)	1.61E+01/ 1.61E+01(-)	1.10E+02/ 1.27E+01(+)	1.05E+02/ 1.57E+01(+)
f_{a8}	2.64E+01/ 3.86E+00	2.09E+01/ 5.79E-02(+)	2.10E+01/ 4.79E-02(+)	2.09E+01/ 5.02E-02(+)	2.09E+01// 2.09E+01(+)	2.10E+01/ 4.49E-02(+)	2.09E+01/ 4.37E-02(+)
f_{a9}	2.64E+01/ 3.86E+00	3.92E+01/ 1.21E+00(+)	3.89E+01/ 1.17E+00(+)	2.84E+01/ 2.70E+00(+)	2.59E+01/ 2.59E+01(-)	2.86E+01/ 1.08E+01(+)	2.88E+01/ 2.06E+00(+)
f_{a10}	4.89E-02/ 3.37E-02	1.58E+02/ 2.72E+01(+)	2.62E+02/ 4.06E+01(+)	1.40E-01/ 1.10E-01(+)	1.38E-02/ 1.38E-02(-)	1.78E+00/ 1.15E-02(+)	1.45E+00/ 3.42E-01(+)
f_{a11}	2.24E+00/ 3.89E+00	1.26E+02/ 9.44E+00(+)	1.32E+02/ 7.82E+00(+)	5.63E+01/ 1.48E+01(+)	6.68E+01/ 6.68E+01(+)	5.57E-14/ 3.25E+01(-)	6.13E-14/ 1.54E-14(-)
f_{a12}	4.54E+01/ 1.13E+01	2.14E+02/ 1.20E+01(+)	2.21E+02/ 9.84E+00(+)	8.88E+01/ 2.21E+01(+)	1.92E+02/ 1.92E+02(+)	1.33E+02/ 1.01E+01(+)	1.55E+02/ 2.71E+01(+)
f_{a13}	9.39E+01/ 2.02E+01	2.15E+02/ 9.96E+00(+)	2.18E+02/ 1.31E+01(+)	1.41E+02/ 2.47E+01(+)	1.87E+02/ 1.87E+02(+)	2.04E+02/ 1.09E+01(+)	2.29E+02/ 3.00E+01(+)
f_{a14}	6.28E+01/ 9.60E+01	4.70E+03/ 2.27E+02(+)	4.76E+03/ 2.23E+02(+)	3.56E+03/ 3.74E+02(+)	3.40E+03/ 3.40E+03(+)	5.77E-01/ 7.52E+02(-) /	3.09E-01/ 2.39E-01(-)
f_{a15}	4.36E+03/ 9.14E+02	7.21E+03/ 2.75E+02(+)	7.26E+03/ 2.52E+02(+)	4.74E+03/ 4.90E+02(+)	7.13E+03/ 7.13E+03(+)	3.94E+03/ 3.48E+02(-)	3.81E+03/ 4.08E+02(-)
f_{a16}	1.48E+00/ 9.55E-01 /	2.44E+00/ 3.08E-01(+)	2.46E+00/ 2.57E-01(+)	1.82E+00/ 2.62E-01(+)	2.42E+00/ 2.42E+00(+)	1.39E+00/ 2.80E-01(-)	1.26E+00/ 2.58E-01(-)
f_{a17}	3.04E+01/ 2.83E-04	1.72E+02/ 9.02E+00(+)	1.75E+02/ 8.02E+00(+)	1.06E+02/ 1.33E+01(+)	1.43E+02/ 1.43E+02(+)	3.05E+01/ 1.02E+01(-)	2.97E+01/ 3.47E+00(+)
f_{a18}	1.12E+02/ 4.48E+01	2.49E+02/ 1.30E+01(+)	2.63E+02/ 1.13E+01(+)	1.49E+02/ 1.61E+01(+)	2.20E+02/ 2.20E+02(+)	1.78E+02/ 1.08E+01(+)	1.97E+02/ 2.14E+01(+)
f_{a19}	1.54E+00/ 1.79E-01	1.59E+01/ 1.11E+00(+)	1.64E+01/ 1.17E+00(+)	9.22E+00/ 2.91E+00(+)	1.27E+01/ 1.27E+01(+)	9.89E-01/ 8.56E-01(-)	8.38E-01/ 1.90E-01(-)
f_{a20}	1.14E+01/ 7.78E-01	1.26E+01/ 1.69E-01(+)	1.28E+01/ 2.20E-01(+)	1.14E+01/ 4.17E-01(=)	1.24E+01/ 1.24E+01(+)	1.39E+01/ 3.07E-01(+)	1.40E+01/ 5.16E-01(+)
f_{a21}	3.04E+02/ 7.68E+01	2.57E+02/ 5.22E+01(-)	3.07E+02/ 7.15E+01(+)	3.07E+02/ 5.85E+01(+)	2.93E+02/ 2.93E+02(-)	3.08E+02/ 7.90E+01(-)	2.00E+02/ 3.86E+01(+)
f_{a22}	1.52E+02/ 9.49E+01	5.05E+03/ 2.73E+02(+)	5.21E+03/ 3.20E+02(+)	3.36E+03/ 4.99E+02(+)	2.97E+03/ 2.97E+03(+)	1.01E+02/ 9.78E+02(-)	9.49E+01/ 3.54E+01(-)
f_{a23}	4.16E+03/ 7.06E+02	7.36E+03/ 3.01E+02(+)	7.46E+03/ 2.81E+02(+)	5.28E+03/ 5.90E+02(+)	7.23E+03/ 7.23E+03(+)	4.75E+03/ 3.17E+02(+)	4.68E+03/ 5.60E+02(+)
f_{a24}	2.40E+02/ 1.53E+01	2.66E+02/ 4.85E+00(+)	2.79E+02/ 4.15E+00(+)	2.35E+02/ 1.26E+01(-)	2.27E+02/ 2.27E+02(-)	2.84E+02/ 1.28E+01(+)	2.84E+02/ 6.13E+00(+)
f_{a25}	2.63E+02/ 1.07E+01	3.05E+02/ 6.36E+00(+)	3.08E+02/ 3.07E+00(+)	2.71E+02/ 1.31E+01(+)	2.49E+02/ 2.49E+02(-)	2.96E+02/ 7.27E+00(+)	2.98E+02/ 5.56E+00(+)
f_{a26}	2.51E+02/ 7.28E+01	2.08E+02/ 1.48E+00(-)	2.09E+02/ 1.49E+00(-)	2.02E+02/ 1.56E+01(-)	2.51E+02/ 2.51E+02(=)	2.01E+02/ 6.51E+01(-)	2.01E+02/ 2.30E-01(-)
f_{a27}	8.80E+02/ 1.31E+02	1.20E+03/ 4.91E+01(+)	1.24E+03/ 3.56E+01(+)	6.30E+02/ 9.34E+01(-)	6.14E+02/ 6.14E+02(-)	5.18E+02/ 1.04E+02(-)	4.69E+02// 2.10E+02(-)
f_{a28}	2.96E+02/ 2.80E+01	3.00E+02/ 2.11E-04(+)	3.00E+02// 1.34E-02(+)	3.69E+02/ 3.83E+02(+)	3.00E+02/ 3.00E+02(+)	3.00E+02/ 0.00E+00(+)	2.97E+02/ 1.93E+01(+)

TABLE 5. Comparison on 50D optimization under CEC2013 test suite with different algorithms.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{a1}	1.11E-05/ 1.28E-06	1.84E+00/ 3.12E-01(+)	2.42E+01/ 3.11E+00(+)	3.48E-13/ 1.23E-13(-)	2.27E-13/ 0.00E+00(-)	7.53E-13/ 1.24E-13(-)	9.94E-13/ 1.36E-13(-)
f_{a2}	3.13E+06/ 1.50E+07	4.49E+08/ 6.06E+07(+)	4.55E+08/ 5.86E+07(+)	3.60E+06/ 1.46E+06(+)	1.42E+08/ 3.51E+07(+)	2.56E+07/ 9.90E+06(+)	2.35E+07/ 6.93E+06(+)
f_{a3}	2.45E+09/ 2.90E+09	3.42E+10/ 3.78E+09(+)	4.07E+10/ 4.16E+09(+)	8.69E+08/ 1.15E+09(-)	4.28E+09/ 4.74E+09(+)	6.47E+09/ 5.84E+09(+)	2.84E+09/ 2.61E+09(+)
f_{a4}	3.05E+04/ 3.64E+04	8.65E+04/ 9.58E+03(+)	9.79E+04/ 7.65E+03(+)	1.63E+04/ 3.76E+03(-)	4.52E+04/ 5.02E+03(+)	1.57E+05/ 1.43E+04(+)	1.61E+05/ 1.70E+04(+)
f_{a5}	6.52E-06/ 1.27E-06	4.72E+00/ 6.07E-01(+)	1.68E+01/ 1.37E+00(+)	3.88E-13/ 8.86E-14(-)	2.32E-13/ 3.18E-14(-)	8.54E-13/ 8.91E-14(-)	1.09E-12/ 1.09E-13(-)
f_{a6}	4.67E+01/ 6.67E+00	4.99E+01/ 5.67E-01(+)	7.05E+01/ 4.61E+00(+)	6.00E+01/ 2.35E+01(+)	4.54E+01/ 1.25E+00(-)	4.68E+01/ 1.62E+00(+)	4.45E+01/ 1.49E+00(-)
f_{a7}	6.60E+01/ 2.15E+01	1.31E+02/ 8.74E+00(+)	1.43E+02/ 7.41E+00(+)	8.40E+01/ 1.29E+01(+)	7.13E+01/ 1.55E+01(+)	1.65E+02/ 1.68E+01(+)	1.48E+02/ 1.42E+01(+)
f_{a8}	2.11E+01/ 4.14E-02	2.12E+01/ 3.31E-02(+)	2.11E+01/ 3.60E-02(=)	2.11E+01/ 3.19E-02(=)	2.12E+01// 3.02E-02(+)	2.11E+01/ 3.72E-02(=)	2.12E+01/ 2.92E-02(+)
f_{a9}	5.41E+01/ 5.24E+00	7.35E+01/ 1.27E+00(+)	7.31E+01/ 1.50E+00(+)	5.60E+01/ 4.62E+00(+)	6.87E+01/ 4.99E+00(+)	5.91E+01/ 2.75E+00(+)	5.84E+01/ 2.71E+00(+)
f_{a10}	7.29E-02/ 3.39E-02	1.66E+03/ 1.52E+02(+)	2.10E+03/ 2.05E+02(+)	3.04E-01/ 5.06E-01(+)	5.12E-01/ 4.14E-01(+)	1.67E+00/ 3.52E-01(+)	1.94E+00/ 4.04E-01(+)
f_{a11}	7.06E+00/ 7.46E+00	3.34E+02/ 1.52E+01(+)	3.41E+02/ 1.38E+01(+)	1.37E+02/ 3.42E+01(+)	2.68E+02/ 2.24E+01(+)	1.77E-13/ 3.14E-14(-)	1.55E-13/ 3.02E-14(-)
f_{a12}	1.23E+02/ 3.26E+01	4.86E+02/ 2.05E+01(+)	5.14E+02/ 1.90E+01(+)	2.36E+02/ 4.73E+01(+)	4.09E+02/ 1.84E+01(+)	4.44E+02/ 7.27E+01(+)	4.72E+02/ 6.17E+01(+)
f_{a13}	2.34E+02/ 5.67E+01	4.92E+02/ 1.42E+01(+)	5.11E+02/ 1.65E+01(+)	3.46E+02/ 4.67E+01(+)	4.07E+02/ 1.84E+01(+)	5.44E+02/ 5.10E+01(+)	5.79E+02/ 4.87E+01(+)
f_{a14}	8.03E+01/ 1.09E+02	1.06E+04/ 2.84E+02(+)	1.06E+04/ 2.92E+02(+)	8.33E+03/ 6.56E+02(+)	9.74E+03/ 5.75E+02(+)	9.32E+00/ 1.74E+01(-)	2.69E+00/ 1.79E+00(-)
f_{a15}	8.21E+03/ 1.33E+03	1.42E+04/ 3.42E+02(+)	1.42E+04/ 3.81E+02(+)	1.07E+04/ 7.73E+02(+)	1.40E+04/ 3.76E+02(+)	8.44E+03/ 8.51E+02(+)	8.30E+03/ 6.43E+02(+)
f_{a16}	2.45E+00/ 1.18E+00	3.51E+00/ 3.40E-01(+)	3.44E+00/ 3.36E-01(+)	2.68E+00/ 3.34E-01(+)	3.43E+00/ 3.03E-01(+)	2.03E+00/ 3.44E-01(-)	1.85E+00/ 3.08E-01(-)
f_{a17}	5.09E+011/ 3.70E-01	1.72E+02/ 1.17E+01(+)	4.55E+02/ 1.68E+01(+)	2.95E+02/ 2.68E+01(+)	3.46E+02/ 1.51E+01(+)	5.08E+01/ 6.38E-02(-)	5.11E+01/ 8.60E-02(+)
f_{a18}	2.39E+02/ 9.71E+01	5.63E+02/ 1.88E+01(+)	5.94E+02/ 1.59E+01(+)	3.81E+02/ 3.50E+01(+)	4.48E+02/ 2.06E+01(+)	4.72E+02/ 5.57E+01(+)	5.06E+02/ 5.57E+01(+)
f_{a19}	4.20E+00/ 6.45E-01	4.79E+01/ 3.51E+00(+)	6.77E+01/ 5.75E+00(+)	3.21E+01/ 6.95E+00(+)	2.96E+01/ 1.41E+00(+)	2.33E+00/ 2.36E-01(-)	82.01E+00/ 2.74E-01(-)
f_{a20}	2.09E+01/ 8.63E-01	2.27E+01/ 1.92E-01(+)	2.28E+01/ 1.97E-01(+)	2.11E+01/ 4.58E-01(+)	2.25E+01/ 1.99E-01(+)	2.42E+01/ 4.89E-01(+)	2.44E+01/ 2.45E-01(+)
f_{a21}	7.70E+02/ 4.19E+02	5.83E+02/ 4.46E+02(-)	1.29E+03/ 3.73E+02(+)	8.90E+02/ 2.88E+02(+)	7.29E+02/ 4.10E+02(-)	6.74E+02/ 4.29E+02(-)	2.04E+02/ 8.20E+00(-)
f_{a22}	1.31E+02/ 1.10E+02	1.09E+04/ 3.76E+02(+)	1.12E+04/ 4.32E+02(+)	8.53E+03/ 6.96E+02(+)	9.46E+03/ 5.65E+02(+)	5.38E+01/ 8.10E+01(-)	4.08E+01/ 4.74E+01(-)
f_{a23}	9.26E+03/ 2.64E+03	1.45E+04/ 3.95E+02(+)	1.45E+04/ 4.02E+02(+)	1.18E+04/ 7.95E+02(+)	1.40E+04/ 4.15E+02(+)	9.90E+03/ 1.06E+03(+)	1.04E+04/ 8.51E+02(+)
f_{a24}	2.91E+02/ 2.14E+01	3.66E+02/ 4.39E+00(+)	3.71E+02/ 4.26E+00(+)	2.92E+02/ 1.35E+01(+)	2.78E+02/ 2.34E+01(-)	3.66E+022/ 6.77E+00(+)	3.65E+02/ 7.22E+00(+)
f_{a25}	3.27E+02/ 2.31E+01	4.08E+02/ 3.94E+00(+)	4.13E+02/ 4.37E+00(+)	3.47E+02/ 1.93E+01(+)	2.99E+02/ 1.18E+01(-)	3.92E+02/ 7.99E+00(+)	4.00E+02/ 7.26E+00(+)
f_{a26}	4.04E+02/ 6.33E+01	3.56E+02/ 7.07E+01(-)	3.49E+02/ 5.10E+01(-)	2.57E+02/ 8.55E+01(-)	3.91E+02/ 7.80E+01(-)	2.04E+02/ 9.12E-01(-)	2.03E+02/ 5.50E-01(-)
f_{a27}	1.45E+03/ 2.47E+02	2.10E+03/ 3.69E+01(+)	2.13E+03/ 4.01E+01(+)	1.24E+03/ 1.93E+02(-)	1.38E+03/ 3.37E+02(-)	1.59E+03/ 5.93E+02(+)	1.54E+03/ 6.38E+02(+)
f_{a28}	6.97E+02/ 9.11E+02	4.13E+02/ 1.47E+00(-)	4.48E+02// 3.65E+00(-)	8.84E+02/ 1.23E+03(+)	8.29E+02/ 1.09E+03(+)	4.00E+02/ 4.93E-06(-)	4.00E+02/ 2.32E-02(-)

Table 6. Comparison on 100D optimization under CEC2013 test suite with different algorithms.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{a1}	1.80E-04/ 1.60E-05	1.16E+04/ 7.93E+02(+)	1.66E+04/ 1.05E+03(+)	1.58E-09/ 5.95E-09(-)	3.75E-09/ 2.53E-09(-)	2.66E-12/ 1.84E-13(-)	4.49E-12/ 6.80E-13(-)
f_{a2}	8.70E+06/ 2.15E+07	2.09E+09/ 1.88E+08(+)	2.13E+09/ 2.04E+08(+)	1.51E+07/ 9.33E+06(+)	6.33E+08/ 1.20E+08(+)	9.76E+07/ 2.67E+07(+)	8.56E+07/ 1.91E+07(+)
f_{a3}	6.41E+09/ 9.36E+09	2.72E+11/ 4.10E+10(+)	6.89E+11/ 3.22E+11(+)	2.84E+10/ 1.36E+10(+)	4.64E+10/ 2.09E+10(+)	1.86E+10/ 1.22E+10(+)	1.11E+10/ 8.12E+09(+)
f_{a4}	8.11E+04/ 1.03E+05	2.38E+05/ 1.54E+04(+)	2.53E+05/ 1.41E+04(+)	6.05E+04/ 1.12E+04(-)	1.49E+05/ 1.87E+04(+)	3.76E+05/ 2.41E+04(+)	3.78E+05/ 2.65E+04(+)
f_{a5}	2.13E-04/ 2.68E-05	7.08E+02/ 4.25E+01(+)	9.63E+02/ 5.49E+01(+)	8.93E-05/ 2.62E-04(-)	5.24E-05/ 3.07E-05(-)	5.20E-12/ 7.61E-13(-)	1.48E-07/ 4.04E-08(-)
f_{a6}	2.22E+02/ 4.37E+01 /	1.98E+03/ 1.72E+02(+) /	2.51E+03/ 1.62E+02(+) /	2.26E+02/ 4.77E+01(+) /	2.48E+02/ 3.43E+01(+) /	2.47E+02/ 3.31E+01(+) /	2.27E+02/ 3.17E+01(+)
f_{a7}	1.27E+02/ 2.51E+01	2.56E+02/ 2.08E+01(+)	1.43E+02/ 7.41E+00(+)	8.40E+01/ 1.29E+01(+)	7.13E+01/ 1.55E+01(+)	1.65E+02/ 1.68E+01(+)	1.48E+02/ 1.42E+01(+)
f_{a8}	2.13E+01/ 4.29E-02	2.13E+01/ 2.76E-02(=)	2.13E+01/ 3.00E-02(=)	2.13E+01/ 2.22E-02(=)	2.13E+01// 2.83E-02(=)	2.13E+01/ 3.31E-02(=)	2.13E+01/ 3.50E-02(=)
f_{a9}	1.35E+02/ 5.89E+00	1.63E+02/ 1.60E+00(+)	1.62E+02/ 2.23E+00(+)	1.44E+02/ 5.57E+00(+)	1.60E+02/ 2.74E+00(+)	1.39E+02/ 3.37E+00(+)	1.40E+02/ 3.93E+00(+)
f_{a10}	1.27E+02/ 1.37E-01	9.01E+03/ 6.04E+02(+)	1.02E+04/ 6.53E+02(+)	1.72E+01/ 2.17E+01(+)	1.62E+01/ 5.41E+00(+)	1.78E+00/ 2.65E-01(+)	2.11E+00/ 4.52E-01(+)
f_{a11}	6.74E+01/ 5.53E+01	1.04E+03/ 2.66E+01(+)	1.08E+03/ 2.38E+01(+)	5.86E+02/ 9.29E+01(+)	7.74E+02/ 2.83E+01(+)	6.54E-13/ 4.74E-14(-)	1.14E-01/ 2.59E-01(-)
f_{a12}	3.97E+02/ 9.99E+01	1.37E+03/ 3.29E+01(+)	1.42E+03/ 3.18E+01(+)	7.35E+02/ 1.01E+02(+)	1.00E+03/ 3.92E+01(+)	1.54E+03/ 1.40E+02(+)	1.64E+03/ 1.50E+02(+)
f_{a13}	6.95E+02/ 1.04E+02	1.36E+03/ 3.65E+01(+)	1.42E+03/ 3.63E+01(+)	1.10E+03/ 7.19E+01(+)	1.01E+03/ 3.57E+01(+)	1.79E+03/ 1.19E+02(+)	1.84E+03/ 1.06E+02(+)
f_{a14}	1.99E+02/ 1.57E+02	2.71E+04/ 4.27E+02(+)	2.70E+04/ 4.85E+02(+)	2.35E+04/ 1.20E+03(+)	2.66E+04/ 6.00E+02(+)	1.79E+01/ 2.59E+01(-)	7.64E+01/ 8.53E+01(-)
f_{a15}	2.43E+04/ 6.03E+03	3.09E+04/ 5.22E+02(+)	3.09E+04/ 5.43E+02(+)	2.51E+04/ 1.01E+03(+)	3.10E+04/ 6.28E+02(+)	1.87E+04/ 1.54E+03(-)	1.84E+04/ 1.14E+03(-)
f_{a16}	3.24E+00/ 1.15E+00	4.17E+00/ 2.11E-01(+)	4.27E+00/ 2.15E-01(+)	3.68E+00/ 2.01E-01(+)	4.23E+00/ 2.23E-01(+)	2.67E+00/ 3.42E-01(-)	2.58E+00/ 3.15E-01(-)
f_{a17}	1.17E+02/ 1.06E+01	1.98E+03/ 6.08E+01(+)	2.23E+03/ 7.06E+01(+)	1.02E+03/ 1.03E+02(+)	9.21E+02/ 2.69E+01(+)	1.02E+02/ 9.53E-02(-)	1.08E+02/ 5.25E-01(-)
f_{a18}	7.64E+02/ 2.29E+02	2.12E+03/ 6.41E+01(+)	2.36E+03/ 8.51E+01(+)	1.18E+03/ 9.28E+01(+)	1.09E+03/ 3.79E+01(+)	1.67E+03/ 1.15E+02(+)	1.83E+03/ 1.15E+02(+)
f_{a19}	2.40E+01/ 4.32E+00	1.53E+04/ 3.31E+03(+)	2.59E+04/ 5.16E+03(+)	1.35E+02/ 3.04E+01(+)	7.91E+01/ 3.76E+00(+)	7.09E+00/ 6.91E-01(-)	6.21E+00/ 6.45E-01(-)
f_{a20}	5.00E+01/ 1.35E-11	5.00E+01/ 9.57E-13(=)	5.00E+01/ 6.08E-10(=)	5.00E+01/ 2.21E-07(=)	5.00E+01/ 0.00E+00(=)	5.00E+01/ 0.00E+00(=)	5.00E+01/ 0.00E+00(=)
f_{a21}	3.19E+02/ 4.18E+01	4.37E+03/ 1.55E+02(+)	4.92E+03/ 1.46E+02(+)	4.03E+02/ 1.70E+01(+)	3.22E+02/ 4.62E+01(+)	3.86E+02/ 5.81E+01(+)	4.78E+02/ 7.21E+01(+)
f_{a22}	2.34E+02/ 1.36E+02	2.79E+04/ 5.55E+02(+)	2.79E+04/ 6.03E+02(+)	2.38E+04/ 1.04E+03(+)	2.64E+04/ 6.77E+02(+)	9.91E+01/ 6.33E+01(-)	2.19E+02/ 6.59E+01(-)
f_{a23}	2.10E+04/ 5.03E+03	3.20E+04/ 5.33E+02(+)	3.24E+04/ 5.49E+02(+)	2.79E+04/ 1.44E+03(+)	3.17E+04/ 6.04E+02(+)	2.39E+04/ 1.82E+03(+)	2.38E+04/ 1.48E+03(+)
f_{a24}	4.66E+02/ 4.45E+01	5.82E+02/ 7.54E+00(+)	5.94E+02/ 7.12E+00(+)	4.66E+02/ 2.39E+01(=)	4.91E+02/ 4.93E+01(+)	5.92E+02/ 9.52E+00(+)	5.95E+02/ 1.04E+01(+)
f_{a25}	5.23E+02/ 4.70E+01	6.81E+02/ 5.99E+00(+)	6.87E+02/ 7.59E+00(+)	5.47E+02/ 2.97E+01(+)	4.94E+02/ 4.19E+01(-)	6.64E+02/ 1.05E+01(+)	6.85E+02/ 1.33E+01(+)
f_{a26}	6.14E+02/ 3.98E+01	7.07E+02/ 4.86E+00(+)	7.07E+02/ 4.43E+00(+)	5.27E+02/ 2.38E+01(-)	6.83E+02/ 2.49E+01(+)	2.22E+02/ 6.33E+01(-)	2.20E+02/ 6.54E+01(-)
f_{a27}	3.40E+03/ 4.69E+02	4.44E+03/ 6.29E+01(+)	4.46E+03/ 5.75E+01(+)	2.80E+03/ 2.10E+02(-)	3.79E+03/ 4.82E+02(+)	3.88E+03/ 8.86E+02(+)	4.09E+03/ 5.34E+02(+)
f_{a28}	4.18E+03/ 1.18E+03	7.64E+03/ 6.63E+02(+)	8.33E+03// 3.73E+02(+)	8.85E+03/ 1.98E+03(+)	4.04E+03/ 1.14E+03(-)	4.68E+03/ 1.30E+03(+)	4.50E+03/ 1.19E+03(+)

9, in 100D real number optimization, TSGCDE algorithm has 30 benchmark functions better than DE/rand/1, 30 benchmark functions outperform RBDE, 20 benchmark functions outperform DEGL, 1 function is similar to DEGL, 23 benchmark functions outperform DE/best/1, 1 function is similar to DE/best/1, 18 benchmark functions outperform MABC, and 18 benchmark functions outperform IABC. In summary, our algorithm is better competitive with ABC variants and DE variants.

Table 7. Comparison on 30D optimization under CEC2014 test suite with different algorithms.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{b1}	6.31E+05/ 3.13E+06	8.96E+07/ 1.66E+07(+)	1.03E+08/ 1.91E+07(+)	4.97E+05/ 9.21E+05(-)	1.87E+07/ 6.30E+06(+)	7.14E+06/ 4.87E+06(+)	5.87E+06/ 4.60E+06(+)
f_{b2}	4.48E-02/ 1.12E-01	1.63E+03/ 3.76E+02(+)	7.62E+04/ 1.73E+04(+)	2.90E-14/ 3.98E-15(-)	2.56E-14/ 8.54E-15(-)	2.58E+02/ 8.57E+02(+)	3.52E+02/ 7.33E+02(+)
f_{b3}	6.52E-01/ 1.36E+00	2.55E+01/ 5.16E+00(+)	1.64E+02/ 2.66E+01(+)	5.02E-09/ 1.95E-08(-)	7.15E-10/ 4.13E-10(-)	4.82E+02/ 5.49E+02(+)	5.82E+02/ 6.12E+02(+)
f_{b4}	6.95E+00/ 1.89E+01	1.21E+02/ 5.29E+00(+)	1.48E+02/ 7.02E+00(+)	4.03E+01/ 3.93E+01(+)	6.74E+01/ 3.25E+01(+)	6.28E+01/ 2.53E+01(+)	3.76E+01/ 3.06E+01(+)
f_{b5}	2.03E+01/ 3.65E-02	2.09E+01/ 4.64E-02(+)	2.09E+01/ 6.46E-02(+)	2.08E+01/ 7.16E-02(+)	2.09E+01/ 6.10E-02(+)	2.02E+01/ 5.29E-02(-)	2.02E+01/ 4.39E-02(-)
f_{b6}	4.51E+00/ 2.98E+00	3.02E+01/ 9.51E-01(+)	3.09E+01/ 1.21E+00(+)	6.33E+00/ 2.34E+00(+)	1.51E+00/ 1.60E+00(-)	1.35E+01/ 1.49E+00(+)	1.32E+01/ 1.56E+00(+)
f_{b7}	5.83E-03/ 7.93E-03	4.03E-02/ 7.55E-02(+)	3.99E-01/ 1.49E-01(+)	8.78E-03/ 1.49E-02(+)	4.01E-03/ 5.10E-03(-)	1.57E-04/ 7.73E-04(-)	2.85E-06/ 4.16E-06(-)
f_{b8}	1.44E+00/ 1.96E+00	1.20E+02/ 7.37E+00(+)	1.22E+02/ 6.26E+00(+)	4.69E+01/ 9.84E+00(+)	4.71E+01// 3.15E+01(+)	1.23E-13/ 3.09E-14(-)	1.47E-13/ 5.23E-14(-)
f_{b9}	3.95E+01/ 5.64E+00	1.96E+02/ 1.03E+01(+)	1.94E+02/ 1.41E+01(+)	6.58E+01/ 1.35E+01(+)	1.76E+02/ 1.14E+01(+)	4.70E+01/ 7.75E+00(+)	4.83E+01/ 6.88E+00(+)
f_{b10}	5.12E+01/ 7.59E+01	3.92E+03/ 1.86E+02(+)	3.96E+03/ 2.15E+02(+)	2.39E+03/ 5.04E+02(+)	3.31E+02/ 4.43E+02(+)	5.18E-01/ 8.10E-01(-)	3.04E-01/ 2.24E-01(-)
f_{b11}	2.20E+03/ 3.05E+02	6.55E+03/ 2.21E+02(+)	6.50E+03/ 3.00E+02(+)	3.92E+03/ 5.56E+02(+)	6.36E+03/ 2.93E+02(+)	1.85E+03/ 3.64E+02(-)	1.77E+03/ 2.77E+02(-)
f_{b12}	4.15E-01/ 6.13E-02	2.02E+00/ 2.13E-01(+)	2.04E+00/ 2.48E-01(+)	1.23E+00/ 1.96E-01(+)	1.99E+00/ 2.58E-01(+)	2.29E-01/ 4.97E-02(-)	2.10E-01/ 4.62E-02(-)
f_{b13}	2.85E-01/ 5.59E-02	4.97E-01/ 5.71E-02(+)	5.14E-01/ 5.73E-02(+)	2.79E-01/ 4.69E-02(-)	3.58E-01/ 5.22E-02(+)	1.72E-01/ 3.33E-02(-)	2.11E-01/ 3.23E-02(-)
f_{b14}	3.43E-01/ 1.72E-01	2.87E-01/ 3.67E-02(-)	2.93E-01/ 4.72E-02(-)	2.25E-01/ 4.16E-02(-)	3.51E-01/ 1.65E-01(+)	1.80E-01/ 3.34E-02(-)	1.76E-01/ 1.97E-02(-)
f_{b15}	4.63E+00/ 7.54E-01	1.87E+01/ 1.21E+00(+)	1.96E+01/ 1.15E+00(+)	1.07E+01/ 2.95E+00(+)	1.63E+01/ 1.14E+00(+)	4.81E+00/ 9.13E-01(+)	4.99E+00/ 6.91E-01(+)
f_{b16}	9.77E+00/ 7.64E-01	1.25E+01/ 1.92E-01(+)	1.25E+01/ 2.11E-01(+)	1.14E+01/ 3.26E-01(+)	1.21E+01/ 2.82E-01(+)	9.48E+00/ 3.97E-01(-)	9.37E+00/ 3.80E-01(-)
f_{b17}	4.69E+05/ 6.60E+05	2.37E+06/ 7.69E+05(+)	2.76E+06/ 6.81E+05(+)	8.19E+04/ 8.50E+04(-)	5.85E+05/ 3.19E+05(+)	1.73E+06/ 1.10E+06(+)	1.69E+06/ 1.08E+06(+)
f_{b18}	1.76E+02/ 1.58E+02	3.31E+04/ 1.75E+04(+)	7.45E+04/ 3.15E+04(+)	2.99E+02/ 8.72E+01(+)	8.02E+02/ 3.68E+02(+)	6.37E+03/ 6.57E+03(+)	3.33E+03/ 5.09E+03(+)
f_{b19}	5.80E+00/ 1.51E+00	1.04E+01/ 7.47E-01(+)	1.13E+01/ 6.39E-01(+)	1.21E+01/ 1.15E+01(+)	6.46E+00/ 1.42E+00(+)	7.34E+00/ 8.15E-01(+)	6.71E+00/ 7.10E-01(+)
f_{b20}	2.20E+03/ 2.00E+03	4.49E+02/ 9.35E+01(-)	9.27E+02/ 2.71E+02(-)	9.68E+01/ 2.67E+01(-)	9.29E+01/ 1.44E+01(-)	5.99E+03/ 3.23E+03(+)	7.40E+03/ 3.61E+03(+)
f_{b21}	1.01E+05/ 1.79E+05	1.95E+05/ 6.89E+04(+)	3.11E+05/ 9.86E+04(+)	1.02E+04/ 9.40E+03(-)	1.64E+04/ 7.39E+03(-)	4.12E+05/ 4.24E+05(+)	2.27E+05/ 1.73E+05(+)
f_{b22}	1.45E+02/ 9.33E+01	2.22E+02/ 6.46E+01(+)	2.39E+02/ 6.67E+01(+)	1.92E+02/ 6.40E+01(+)	1.36E+02/ 1.07E+02(-)	2.67E+02/ 1.19E+02(+)	2.64E+02/ 1.03E+02(+)
f_{b23}	3.15E+02/ 3.24E-08	3.15E+02/ 1.04E-04(=)	3.15E+02/ 2.21E-03(=)	3.15E+02/ 4.41E-13(=)	3.15E+02/ 4.02E-13(=)	3.16E+02/ 7.82E-01(+)	3.15E+02/ 1.61E-01(=)
f_{b24}	2.33E+02/ 7.09E+00	2.09E+02/ 2.69E+00(-)	2.22E+02/ 2.65E+00(-)	2.28E+02/ 4.70E+00(-)	2.35E+02/ 5.17E+00(+)	2.26E+02/ 1.28E+00(-)	2.25E+02/ 6.20E+00(-)
f_{b25}	2.05E+02/ 1.50E+00	2.23E+02/ 2.47E+00(+)	2.25E+02/ 3.44E+00(+)	2.09E+02/ 3.08E+00(+)	2.07E+02/ 2.01E+00(+)	2.09E+02/ 1.52E+00(+)	2.08E+02/ 1.37E+00(+)
f_{b26}	1.00E+02/ 6.78E-02	1.00E+02/ 4.85E-02(=)	1.00E+02/ 4.71E-02(=)	1.22E+02/ 4.14E+01(+)	1.00E+02/ 4.79E-02(=)	1.00E+02/ 3.35E-02(=)	1.00E+02/ 4.56E-02(=)
f_{b27}	4.12E+02/ 5.53E+01	4.03E+02/ 4.21E+01(-)	6.87E+02/ 7.14E+01(+)	4.48E+02/ 6.88E+01(+)	3.83E+02/ 4.90E+01(-)	4.14E+02/ 5.25E+00(+)	4.11E+02/ 3.91E+00(-)
f_{b28}	8.54E+02/ 4.40E+01	9.82E+02/ 1.94E+01(+)	1.00E+03/ 2.04E+01(+)	9.63E+02/ 1.66E+02(+)	8.55E+02/ 7.60E+01(+)	8.70E+02/ 6.15E+01(+)	8.68E+02/ 6.65E+01(+)
f_{b29}	1.24E+03/ 3.91E+02	1.12E+04/ 2.70E+03(+)	1.78E+04/ 3.83E+03(+)	3.81E+05/ 1.91E+06(+)	1.67E+05/ 1.18E+06(+)	1.67E+03/ 5.04E+02(+)	1.29E+03/ 2.77E+02(+)
f_{b30}	2.29E+03/ 1.03E+03	5.71E+03/ 1.09E+03(+)	8.14E+03/ 1.10E+03(+)	2.06E+03/ 5.82E+02(-)	1.89E+03/ 6.90E+02(-)	4.06E+03/ 1.61E+03(+)	2.88E+03/ 8.45E+02(+)
+/=/-		24/2/4	25/2/3	19/1/10	19/2/9	19/1/10	17/2/11

Table 8. Comparison on 50D optimization under CEC2014 test suite with different algorithms.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{b1}	6.20E+05/ 2.27E+06	3.89E+08/ 5.87E+07(+)	4.40E+08/ 6.02E+07(+)	1.87E+06/ 7.31E+05(+)	9.41E+07/ 2.63E+07(+)	1.67E+07/ 7.11E+06(+)	1.29E+07/ 4.37E+06(+)
f_{b2}	5.58E+03/ 6.36E+03	2.82E+08/ 7.43E+07(+)	8.30E+08/ 1.52E+08(+)	4.23E+03/ 5.18E+03(-)	2.22E+03/ 2.79E+03(-)	7.16E+03/ 9.11E+03(+)	5.36E+03/ 7.20E+03(-)
f_{b3}	5.11E+03/ 2.86E+03	9.11E+04/ 9.67E+03(+)	1.03E+05/ 8.36E+03(+)	6.79E+03/ 2.95E+03(+)	3.34E+04/ 5.15E+03(+)	9.71E+03/ 3.65E+03(+)	7.72E+03/ 2.95E+03(+)
f_{b4}	8.64E+01/ 1.95E+01	2.01E+02/ 2.16E+01(+)	3.27E+02/ 2.80E+01(+)	9.64E+01/ 4.10E+01(+)	9.46E+01/ 1.05E+01(+)	9.90E+01/ 1.52E+01(+)	8.70E+01/ 2.20E+01(+)
f_{b5}	2.05E+01/ 3.13E-02	2.11E+01/ 3.26E-02(+)	2.11E+01/ 3.83E-02(+)	2.10E+01/ 5.93E-02(+)	2.11E+01/ 4.37E-02(+)	2.03E+01/ 4.08E-02(-)	2.03E+01/ 4.74E-02(-)
f_{b6}	1.87E+01/ 7.65E+00	6.08E+01/ 1.75E+00(+)	6.19E+01/ 1.75E+00(+)	2.10E+01/ 3.45E+00(+)	4.13E+00/ 2.52E+00(-)	2.95E+01/ 2.71E+00(+)	2.91E+01/ 2.57E+00(+)
f_{b7}	5.20E-03/ 5.76E-03	1.10E+00/ 1.62E-02(+)	1.99E+00/ 1.39E-01(+)	1.32E-02/ 1.55E-02(+)	1.50E-03/ 3.35E-03(-)	6.47E-03/ 6.91E-03(+)	2.60E-03/ 2.28E-03(-)
f_{b8}	2.30E+00/ 3.01E+00	2.97E+02/ 1.60E+01(+)	3.01E+02/ 1.16E+01(+)	1.05E+02/ 2.52E+01(+)	2.42E+02/ 2.53E+01(+)	4.28E-13/ 4.87E-14(-)	4.39E-13/ 5.58E-14(-)
f_{b9}	1.11E+02/ 1.46E+01	4.32E+02/ 1.47E+01(+)	4.37E+02/ 1.42E+01(+)	1.69E+02/ 2.83E+01(+)	3.87E+02/ 1.65E+01(+)	1.16E+02/ 1.81E+01(+)	1.18E+02/ 1.45E+01(+)
f_{b10}	7.88E+01/ 1.02E+02	9.64E+03/ 2.92E+02(+)	9.72E+03/ 3.29E+02(+)	6.77E+03/ 5.99E+02(+)	7.19E+03/ 1.63E+03(+)	2.63E+00/ 2.30E+00(-)	5.50E+00/ 1.84E+00(-)
f_{b11}	5.72E+03/ 3.68E+02	1.31E+04/ 4.05E+02(+)	1.32E+04/ 3.50E+02(+)	9.05E+03/ 8.15E+02(+)	1.29E+04/ 4.49E+02(+)	4.42E+03/ 5.18E+02(-)	4.14E+03/ 3.62E+02(-)
f_{b12}	5.79E-01/ 6.27E-02	3.32E+00/ 2.92E-01(+)	3.29E+00/ 2.79E-01(+)	2.09E+00/ 3.06E-01(+)	3.23E+00/ 2.84E-01(+)	2.85E-01/ 5.32E-02(-)	2.71E-01/ 4.42E-02(-)
f_{b13}	4.25E-01/ 7.14E-02	7.08E-01/ 6.70E-02(+)	7.24E-01/ 5.56E-02(+)	4.87E-01/ 7.70E-02(+)	5.57E-01/ 6.63E-02(+)	2.53E-01/ 3.66E-02(-)	3.22E-01/ 3.70E-02(-)
f_{b14}	4.51E-01/ 2.33E-01	4.20E-01/ 1.20E-01(-)	4.22E-01/ 8.80E-02(-)	3.13E-01/ 3.76E-02(-)	4.73E-01/ 2.70E-01(+)	2.42E-01/ 3.03E-02(-)	2.43E-01/ 2.41E-02(-)
f_{b15}	1.28E+01/ 2.32E+00	7.20E+01/ 8.48E+00(+)	1.67E+02/ 3.21E+01(+)	3.62E+01/ 7.47E+00(+)	3.46E+01/ 1.67E+00(+)	1.22E+01/ 1.59E+00(-)	1.39E+01/ 1.80E+00(+)
f_{b16}	1.88E+01/ 7.85E-01	2.24E+01/ 2.07E-01(+)	2.24E+01/ 2.22E-01(+)	2.11E+01/ 3.91E-01(+)	2.21E+01/ 2.15E-01(+)	1.79E+01/ 3.92E-01(-)	1.79E+01/ 3.44E-01(-)
f_{b17}	1.49E+06/ 2.16E+06	1.97E+07/ 4.95E+06(+)	2.31E+07/ 4.35E+06(+)	5.36E+05/ 4.74E+05(-)	6.34E+06/ 1.98E+06(+)	5.60E+06/ 2.72E+06(+)	3.75E+06/ 2.12E+06(+)
f_{b18}	1.46E+03/ 1.10E+03	1.64E+05/ 6.69E+04(+)	3.97E+05/ 1.33E+05(+)	1.15E+03/ 1.02E+03(-)	2.00E+03/ 1.39E+03(+)	1.75E+03/ 1.60E+03(+)	2.13E+03/ 1.35E+03(+)
f_{b19}	1.74E+01/ 1.20E+01	3.13E+01/ 2.50E+00(+)	4.03E+01/ 3.70E+00(+)	3.66E+01/ 2.72E+01(+)	1.49E+01/ 1.81E+00(-)	2.65E+01/ 8.50E+00(+)	2.00E+01/ 5.27E+00(+)
f_{b20}	1.66E+04/ 9.33E+03	2.99E+04/ 6.90E+03(+)	3.85E+04/ 8.22E+03(+)	1.69E+03/ 1.00E+03(-)	9.12E+03/ 3.42E+03(-)	2.97E+04/ 9.79E+03(+)	2.66E+04/ 9.46E+03(+)
f_{b21}	1.12E+06/ 2.32E+06	8.88E+06/ 2.22E+06(+)	9.42E+06/ 2.12E+06(+)	3.37E+05/ 1.86E+05(-)	2.52E+06/ 1.21E+06(+)	3.28E+06/ 2.18E+06(+)	2.53E+06/ 1.32E+06(+)
f_{b22}	6.89E+02/ 1.81E+02	1.33E+03/ 1.71E+02(+)	1.37E+03/ 1.63E+02(+)	6.59E+02/ 1.77E+02(-)	1.23E+03/ 1.93E+02(+)	8.38E+02/ 2.28E+02(+)	7.55E+02/ 1.67E+02(+)
f_{b23}	3.44E+02/ 2.00E-08	3.44E+02/ 1.05E-02(=)	3.44E+02/ 3.60E-02(=)	3.44E+02/ 4.37E-13(=)	3.44E+02/ 4.59E-13(=)	3.47E+02/ 2.73E+00(+)	3.46E+02/ 1.96E+00(+)
f_{b24}	2.80E+02/ 3.97E+00	2.90E+02/ 1.84E+00(+)	2.99E+02/ 1.87E+00(+)	2.79E+02/ 5.99E+00(-)	2.70E+02/ 3.16E+00(-)	2.64E+02/ 3.98E+00(-)	2.58E+02/ 6.39E-01(-)
f_{b25}	2.12E+02/ 4.81E+00	2.82E+02/ 8.23E+00(+)	2.87E+02/ 8.48E+00(+)	2.22E+02/ 8.44E+00(+)	2.23E+02/ 7.62E+00(+)	2.18E+02/ 2.02E+00(+)	2.17E+02/ 1.61E+00(+)
f_{b26}	1.23E+02/ 6.39E+01	1.01E+02/ 5.50E-02(-)	1.01E+02/ 6.52E-02(-)	1.75E+02/ 4.39E+01(+)	1.40E+02/ 6.08E+01(+)	1.00E+02/ 5.93E-02(-)	1.00E+02/ 6.28E-02(-)
f_{b27}	7.23E+02/ 1.07E+02	1.61E+03/ 4.77E+01(+)	1.69E+03/ 3.75E+01(+)	9.01E+02/ 1.07E+02(+)	4.60E+02/ 6.12E+01(-)	1.07E+03/ 2.03E+02(+)	1.09E+03/ 1.83E+02(+)
f_{b28}	1.27E+03/ 1.82E+02	1.56E+03/ 3.52E+01(+)	1.60E+03/ 5.81E+01(+)	1.76E+03/ 4.30E+02(+)	1.19E+03/ 1.75E+02(-)	1.34E+03/ 1.54E+02(+)	1.35E+03/ 5.27E+01(+)
f_{b29}	1.41E+03/ 3.67E+02	2.00E+05/ 6.13E+04(+)	3.32E+05/ 7.80E+04(+)	1.26E+03/ 3.42E+02(-)	1.78E+04/ 1.55E+04(+)	1.40E+04/ 1.47E+04(+)	1.80E+03/ 4.07E+02(+)
f_{b30}	1.05E+04/ 1.37E+03	4.90E+04/ 1.31E+04(+)	7.53E+04/ 1.65E+04(+)	1.20E+04/ 1.20E+03(+)	9.33E+03/ 7.87E+02(-)	1.19E+04/ 2.13E+03(+)	1.06E+04/ 8.91E+02(+)

Table 9. Comparison on 100D optimization under CEC2014 test suite with different algorithms.

	TSGCDE	DE/rand/1	RBDE	DEGL	DE/best/1	MABC	IABC
f	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**	**Mean/std**
f_{b1}	1.03E+07/ 2.95E+07	2.92E+09/ 3.34E+08(+)	3.08E+09/ 2.64E+08(+)	2.20E+07/ 8.99E+06(+)	7.57E+08/ 1.94E+08(+)	1.11E+08/ 4.00E+07(+)	9.35E+07/ 2.15E+07(+)
f_{b2}	2.23E+04/ 2.99E+04	2.55E+10/ 2.31E+09(+)	3.35E+10/ 2.07E+09(+)	1.78E+04/ 1.69E+04(-)	2.76E+04/ 3.61E+04(+)	2.02E+04/ 2.10E+04(-)	1.51E+04/ 1.98E+04(-)
f_{b3}	1.37E+04/ 4.86E+03	3.38E+05/ 1.84E+04(+)	3.67E+05/ 2.07E+04(+)	7.01E+04/ 1.60E+04(+)	1.98E+05/ 2.35E+04(+)	2.06E+04/ 6.26E+03(+)	2.20E+04/ 4.96E+03(+)
f_{b4}	2.12E+02/ 4.40E+01	3.10E+03/ 2.77E+02(+)	3.96E+03/ 3.07E+02(+)	3.07E+02/ 5.67E+01(+)	2.18E+02/ 3.12E+01(+)	2.49E+02/ 2.95E+01(+)	2.43E+02/ 3.40E+01(+)
f_{b5}	2.05E+01/ 3.49E-01	2.13E+01/ 2.82E-02(+)	2.13E+01/ 2.75E-02(+)	2.13E+01/ 3.06E-02(+)	2.13E+01/ 2.71E-02(+)	2.06E+01/ 4.14E-02(+)	2.06E+01/ 3.99E-02(+)
f_{b6}	6.13E+01/ 1.37E+01	1.46E+02/ 2.53E+00(+)	1.48E+02/ 2.18E+00(+)	7.53E+01/ 5.22E+00(+)	2.48E+01/ 6.36E+00(-)	8.36E+01/ 4.38E+00(+)	8.06E+01/ 3.63E+00(+)
f_{b7}	1.82E-02/ 9.51E-03	1.67E+02/ 1.09E+01(+)	2.27E+02/ 1.47E+01(+)	1.55E-02/ 4.26E-02(-)	9.99E-04/ 2.09E-03(-)	1.63E-02/ 1.05E-02(-)	1.48E-02/ 8.38E-03(-)
f_{b8}	1.20E+01/ 1.71E+01	8.50E+02/ 1.59E+01(+)	8.58E+02/ 2.16E+01(+)	3.24E+02/ 4.82E+01(+)	7.36E+02/ 3.36E+01(+)	7.80E-02/ 2.70E-01(-)	3.88E-02/ 1.27E-01(-)
f_{b9}	3.73E+02/ 7.07E+01	1.09E+03/ 2.21E+01(+)	1.10E+03/ 2.98E+01(+)	4.61E+02/ 1.03E+02(+)	9.70E+02/ 3.33E+01(+)	4.50E+02/ 4.90E+01(+)	4.54E+02/ 3.93E+01(+)
f_{b10}	1.78E+02/ 2.07E+02	2.56E+04/ 4.16E+02(+)	2.56E+04/ 4.30E+02(+)	1.96E+04/ 1.28E+03(+)	2.40E+04/ 9.97E+02(+)	1.36E+01/ 3.22E+01(-)	2.53E+01/ 1.78E+01(-)
f_{b11}	1.70E+04/ 1.15E+03	3.05E+04/ 5.36E+02(+)	3.05E+04/ 6.65E+02(+)	2.34E+04/ 1.29E+03(+)	3.06E+04/ 6.01E+02(+)	1.31E+04/ 1.11E+03(-)	1.23E+04/ 8.80E+02(-)
f_{b12}	1.03E+00/ 9.97E-02	4.19E+00/ 2.03E-01(+)	4.27E+00/ 2.14E-01(+)	3.14E+00/ 2.40E-01(+)	4.23E+00/ 1.97E-01(+)	5.49E-01/ 7.11E-02(-)	5.28E-01/ 5.37E-02(-)
f_{b13}	5.55E-01/ 6.32E-02	9.76E-01/ 7.82E-02(+)	1.02E+00/ 9.09E-02(+)	5.60E-01/ 5.73E-02(+)	7.40E-01/ 7.06E-02(+)	3.84E-01/ 4.25E-02(-)	4.09E-01/ 3.28E-02(-)
f_{b14}	1.37E-01/ 1.56E-02	1.11E+02/ 7.05E+00(+)	1.16E+02/ 6.66E+00(+)	1.60E-01/ 1.69E-02(+)	2.67E-01/ 6.73E-02(+)	1.40E-01/ 1.04E-02(+)	1.38E-01/ 1.23E-02(+)
f_{b15}	6.37E+01/ 2.38E+01	1.31E+05/ 3.55E+04(+)	2.56E+05/ 6.24E+04(+)	1.40E+02/ 3.20E+01(+)	8.49E+01/ 2.82E+00(+)	4.63E+01/ 5.17E+00(-)	4.61E+01/ 5.77E+00(-)
f_{b16}	4.27E+01/ 1.27E+00	4.68E+01/ 2.28E-01(+)	4.68E+01/ 2.95E-01(+)	4.51E+01/ 4.36E-01(+)	4.67E+01/ 2.94E-01(+)	4.08E+01/ 5.03E-01(-)	4.08E+01/ 4.41E-01(-)
f_{b17}	3.39E+06/ 9.90E+06	2.36E+08/ 3.29E+07(+)	2.42E+08/ 4.49E+07(+)	3.16E+06/ 1.13E+06(-)	7.56E+07/ 1.67E+07(+)	2.69E+07/ 1.01E+07(+)	2.29E+07/ 8.18E+06(+)
f_{b18}	2.26E+03/ 2.20E+03	3.11E+06/ 2.91E+06(+)	1.82E+07/ 1.03E+07(+)	1.61E+03/ 1.50E+03(-)	3.52E+03/ 3.45E+03(+)	4.17E+03/ 4.00E+03(+)	2.40E+03/ 2.43E+03(+)
f_{b19}	1.04E+02/ 1.18E+01	1.43E+02/ 2.73E+00(+)	1.65E+02/ 6.76E+00(+)	1.21E+02/ 3.38E+01(+)	1.02E+02/ 5.35E+00(-)	1.06E+02/ 1.02E+01(+)	9.35E+01/ 1.48E+01(-)
f_{b20}	4.62E+04/ 3.37E+04	1.90E+05/ 2.81E+04(+)	2.17E+05/ 3.83E+04(+)	2.57E+04/ 9.37E+03(-)	8.07E+04/ 2.35E+04(+)	1.02E+05/ 2.13E+04(+)	9.98E+04/ 1.79E+04(+)
f_{b21}	6.43E+06/ 1.11E+07	1.04E+08/ 1.84E+07(+)	1.12E+08/ 2.07E+07(+)	3.09E+06/ 1.35E+06(-)	3.52E+07/ 1.06E+07(+)	1.77E+07/ 9.18E+06(+)	1.62E+07/ 5.66E+06(+)
f_{b22}	2.35E+03/ 3.37E+02	4.50E+03/ 2.56E+02(+)	4.54E+03/ 2.53E+02(+)	2.18E+03/ 4.12E+02(-)	4.34E+03/ 2.55E+02(+)	2.57E+03/ 3.69E+02(+)	2.37E+03/ 2.92E+02(+)
f_{b23}	3.48E+02/ 7.84E-03	4.08E+02/ 4.53E+00(+)	4.33E+02/ 6.16E+00(+)	3.48E+02/ 4.32E-04(-)	3.48E+02/ 4.09E-02(+)	3.58E+02/ 5.20E+00(+)	3.50E+02/ 6.22E-01(+)
f_{b24}	4.19E+02/ 9.43E+00	5.95E+02/ 7.99E+00(+)	6.25E+02/ 1.02E+01(+)	4.10E+02/ 1.22E+01(-)	3.86E+02/ 4.82E+00(-)	3.65E+02/ 1.67E+00(-)	3.56E+02/ 8.40E-01(-)
f_{b25}	2.57E+02/ 1.80E+01	6.29E+02/ 2.83E+01(+)	6.50E+02/ 2.70E+01(+)	2.58E+02/ 1.83E+01(+)	3.32E+02/ 3.18E+01(+)	2.66E+02/ 6.52E+00(+)	2.60E+02/ 4.17E+00(+)
f_{b26}	1.95E+02/ 2.38E+01	4.51E+02/ 1.11E+02(+)	4.75E+02/ 9.14E+01(+)	2.00E+02/ 1.48E-01(+)	2.63E+02/ 2.97E+01(+)	1.67E+02/ 5.40E+01(-)	1.74E+02/ 4.99E+01(-)
f_{b27}	1.72E+03/ 1.76E+02	3.67E+03/ 6.67E+01(+)	3.76E+03/ 5.94E+01(+)	2.19E+03/ 2.17E+02(+)	8.87E+02/ 1.44E+02(-)	2.21E+03/ 5.49E+02(+)	2.25E+03/ 4.26E+02(+)
f_{b28}	3.14E+03/ 6.38E+02	3.51E+03/ 9.14E+01(+)	3.62E+03/ 2.17E+02(+)	4.87E+03/ 1.02E+03(+)	2.31E+03/ 3.10E+02(-)	2.93E+03/ 4.37E+02(-)	3.33E+03/ 6.12E+02(+)
f_{b29}	2.11E+03/ 7.75E+02	8.12E+05/ 2.11E+05(+)	1.86E+06/ 3.77E+05(+)	1.89E+03/ 4.06E+02(-)	1.96E+04/ 8.71E+03(+)	3.08E+04/ 1.85E+04(+)	3.55E+03/ 8.93E+02(+)
f_{b30}	1.16E+04/ 6.08E+03	2.29E+06/ 4.13E+05(+)	3.77E+06/ 6.75E+05(+)	1.26E+04/ 3.36E+03(+)	1.31E+04/ 4.56E+03(+)	7.94E+04/ 2.31E+04(+)	5.79E+04/ 1.61E+04(+)

4.4. **Analysis of Convergence Rate.** The convergence ability of TSGCDE algorithm is evaluated in this paper, compare the convergence curves of representative benchmark functions selected from the CEC2013 test set and CEC2014 test set, respectively, and set the dimension to 30D, as shown in Figure 1-4, respectively. In the CEC2013 test set, unimodal functions: fa2, fa4, basic multimodal functions: fa6, fa7, fa9, fa10, fa12, fa13, fa18, fa20, composition functions: fa23, fa24, fa25, fa28. In the CEC2014 test set, unimodal function: fb1, simple multimodal function: fb4, fb9, fb15, hybrid functions: fb17, fb18, fb19, fb21, fb22, composition function: fb25, fb27, fb28, fb29, fb30. From Figure 1-2, we can see that the convergence ability of functions fa2, fa4, fa9, fa12, fa13, fa18, fa20, and fa23 is more competitive than other algorithms, and the convergence speed in function fa28 is similar to the other functions. As can be seen in Figure 3-4, the convergence speed in functions fb1, fb4, fb17, fb18, fb21, and fb22 is much better than the other algorithms, and the convergence speed in functions fb9, fb15, fb19, fb25, fb27, fb28, fb29 and fb30 is slightly better than the other algorithms. Overall, our TSGCDE algorithm achieves the best results compared to the other algorithms and it is still competitive in terms of convergence speed.

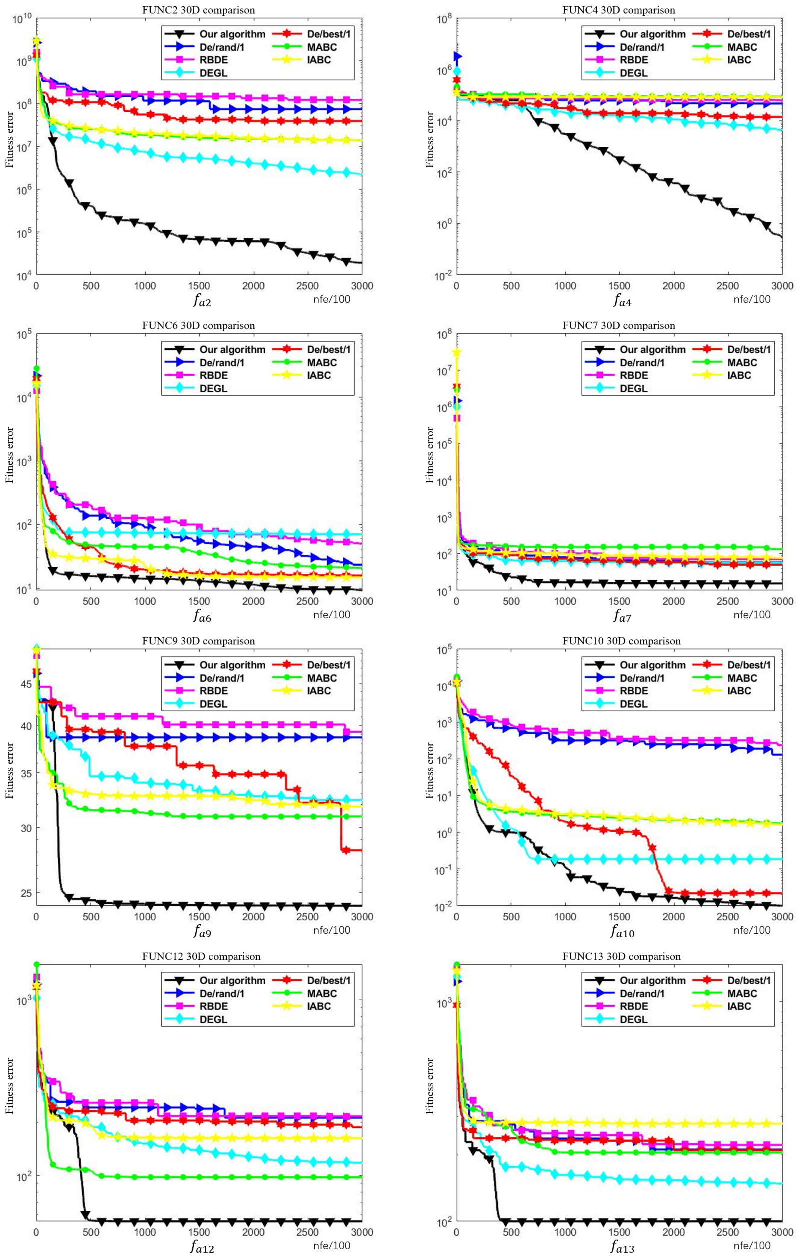


FIGURE 1. Comparison of convergence curves of benchmark functions for the CEC2013 test set (a)

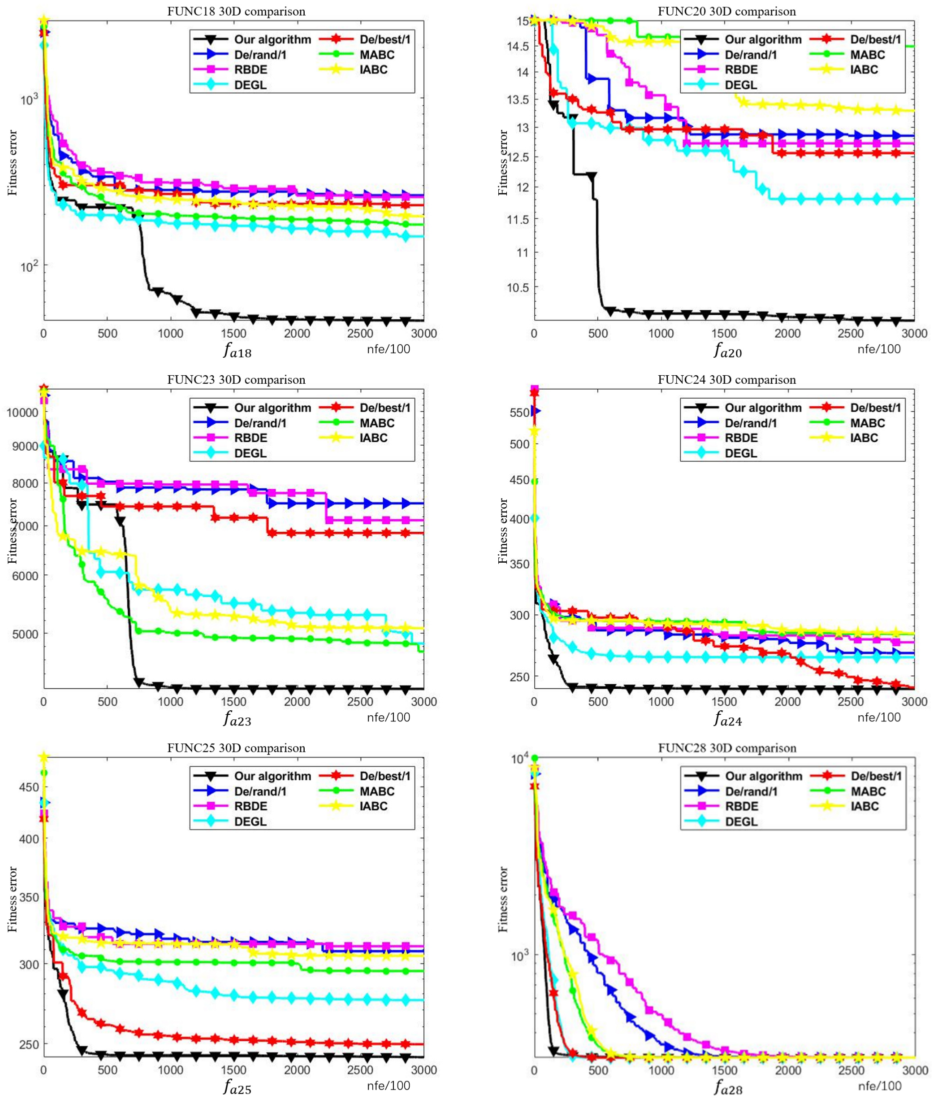
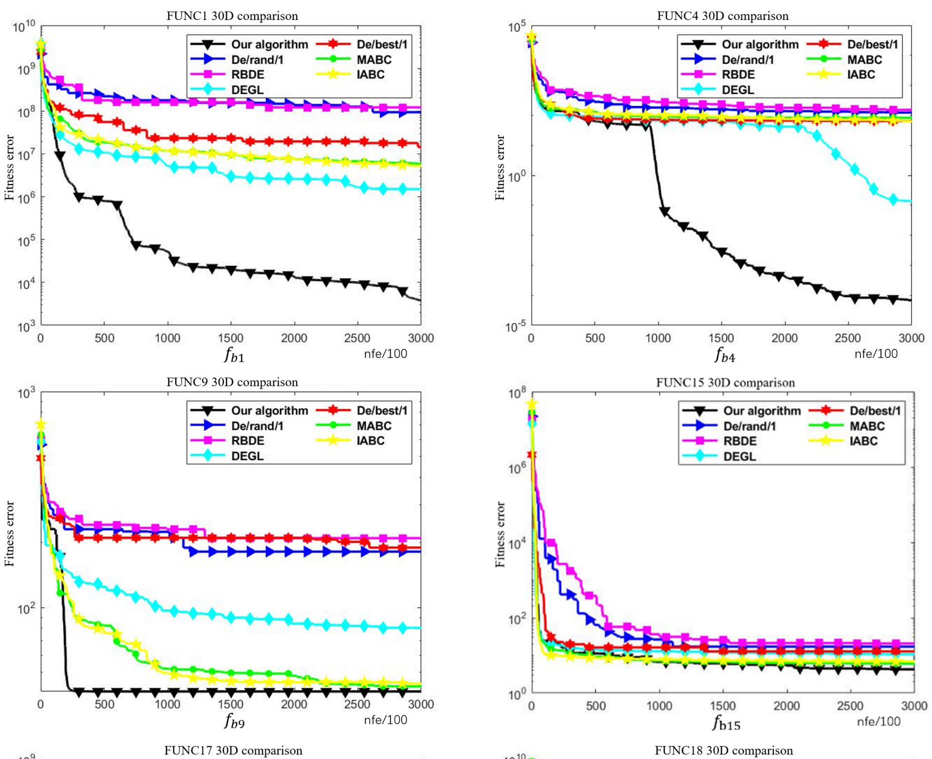


FIGURE 2. Comparison of convergence curves of benchmark functions for the CEC2013 test set (b)



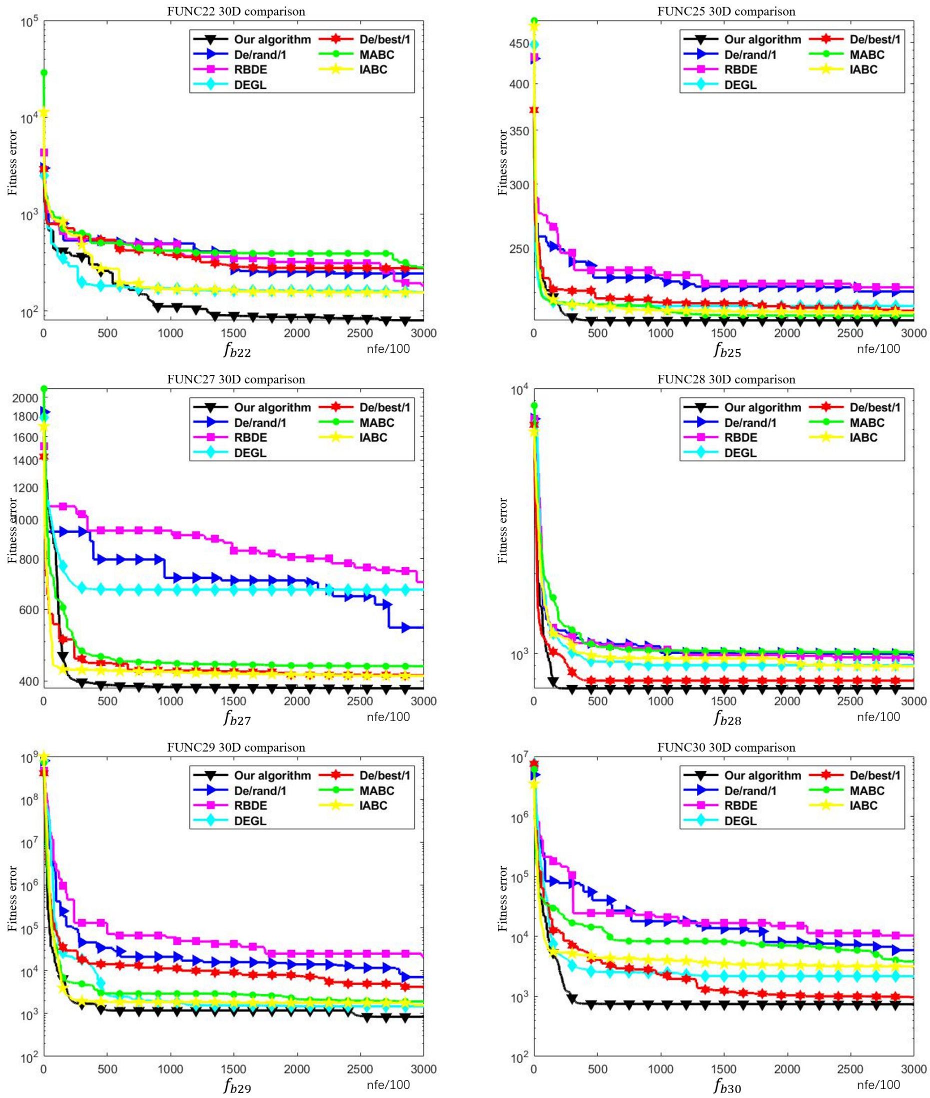


FIGURE 4. Comparison of convergence curves of benchmark functions for the CEC2014 test set (b)

5. **Conclusions.** The development of intelligent optimization algorithms has shown a major trend of linear soaring in recent years, and a large number of intelligent algorithms inspired by biology have been studied [33, 34], the DE algorithm has attracted the attention of many researchers because of its simplicity, high efficiency, and possession of powerful search capability. As a simple and efficient global optimization algorithm, DE possesses powerful search ability and optimization capability, but it still cannot avoid the phenomenon of search stagnation and premature convergence for some specific optimization problems. The population of DE algorithm is too homogeneous, and this paper proposes Two-stage guided constraint differential evolution algorithm to improve the traditional DE/best/1. Dividing the population into two stages for the optimal search increases the diversity of the population. The optimal solutions are searched in different directions in the two-stage population, and the optimal solutions generated by the two-stage population and the guided solutions are guided constraint to each other, which makes the DE more easily to jump out of the local optimum. The performance of TS-GCDE is verified on CEC2013 and CEC2014 test sets, and the proposed algorithm is

compared with ABC variant, traditional DE algorithm and its variants. Experimental results show that this algorithm has a good competitive advantage in basic performance. In subsequent studies, a deeper experiment using TSGCDE in conjunction with some practical problems will be considered.

Acknowledgment. This work was partially supported by the Fujian University of Technology under Grant GY-Z18183 and GY-Z20016, and National Natural Science Foundation of China under Grant 61773415

REFERENCES

[1] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555–104564, 2020.

[2] L. Kang, R.-S. Chen, Y.-C. Chen, C.-C. Wang, X. Li, and T.-Y. Wu, "Using cache optimization method to reduce network traffic in communication systems based on cloud computing," *IEEE Access*, vol. 7, pp. 124397–124409, 2019.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in Proceedings of ICNN'95-international Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948

[4] Z. Chen and R.-L. Wang, "Ant colony optimization with different crossover schemes for global optimization," *Cluster Computing*, vol. 20, no. 2, pp. 1247–1257, 2017.

[5] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983

[6] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Computers Engineering Department, Engineering Faculty, Erciyes University*, vol. 200, pp. 1-10, 2005.

[7] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion Wesley*, vol. 1989, no. 102, 36, 1989

[8] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, 341, 1997

[9] S. Ajithapriyadarsini, P. M. Mary, and M. W. Iruthayarajan, "Automatic generation control of a multi-area power system with renewable energy source under deregulated environment: adaptive fuzzy logic-based differential evolution (de) algorithm," *Soft Computing*, vol. 23, no. 22, pp. 12087–12101, 2019.

[10] M. Ghiasi, T. Niknam, M. Dehghani, P. Siano, H. Haes Alhelou, and A. Al- Hinai, "Optimal multi-operation energy management in smart microgrids in the presence of ress based on multi-objective improved de algorithm: Cost-emission based optimization," *Applied Sciences*, vol. 11, no. 8, 3661, 2021.

[11] Z. Sun, Y. Ling, H. Qu, F. Xiang, Z. Sun, and F. Wu, "An adaptive de algorithm based fuzzy logic anti-swing controller for overhead crane systems," *International Journal of Fuzzy Systems*, vol. 22, pp. 1905–1921, 2020.

[12] J. J. Liang, B. Qu, P. N. Suganthan, and A. G. Hernandez-Diaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, no. 34, pp. 281–295, 2013.

[13] J. J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, no. 2, 2013.

[14] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differen- tial evolution–an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[15] Z. Sun, Y.-W. Ling, H.-B. Qu, F. Xing, Z.-X. Sun and F. Wu, "An adaptive DE algorithm based fuzzy logic anti-swing controller for overhead crane systems," *International Journal of Fuzzy Systems*, vol. 22, pp. 1905-1921, 2020.

[16] K. Price, R. M. Storn, and J. A. Lampinen, "Differential evolution: a practical approach to global optimization," *Springer Science & Business Media*, 2006, vol. 141, no. 2

[17] Z. Meng and J.-S. Pan, "Quasi-affine transformation evolution with external archive (quatre-ear): an enhanced structure for differential evolution," *Knowledge-Based Systems*, vol. 155, pp. 35–53, 2018

[18] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[19] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[20] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," *in 2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 71–78.

[21] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," *in 2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[22] J. Brest, M. S. Maučec, and B. Bošković, "Il-shade: Improved l-shade algorithm for single objective real-parameter optimization," *in 2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 1188–1195.

[23] Z. Meng, J.-S. Pan, and L. Kong, "Parameters with adaptive learning mech- anism (palm) for the enhancement of differential evolution," *Knowledge-Based Systems*, vol. 141, pp. 92–112, 2018.

[24] Z. Meng, J.-S. Pan, and H. Xu, "Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.

[25] Z. Meng, Y. Zhong, and C. Yang, "Cs-de: Cooperative strategy based differential evolution with population diversity enhancement," *Information Sciences*, vol. 577, pp. 663–696, 2021.

[26] A. M. Sutton, M. Lunacek, and L. D. Whitley, "Differential evolution and non-separability: using selective pressure to focus search," *in Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, 2007, pp. 1428–1435.

[27] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[28] T.-W. Sung, B. Zhao, and X. Zhang, "An adaptive dimension differential evolution algorithm based on ranking scheme for global optimization," *PeerJ Computer Science*, vol. 8, e1007, 2022.

[29] T.-W. Sung, B. Zhao, and X. Zhang, "Quasi-Affine Transformation Evolutionary with Double Excellent Guidance," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–15, 2021.

[30] W.-F. Gao , S.-Y. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.

[31] Y. Cao, Y. Lu, X. Pan, and N. Sun, "An improved global best guided artificial bee colony algorithm for continuous optimization problems," *Cluster Computing*, vol. 22, pp. 3011–3019, 2019.

[32] T.-W. Sung, P.-W. Tsai, T. Gaber, and C.-Y. Lee, "Artificial intelligence of things (aiot) technologies and applications," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–2, 2021.

[33] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, and C.-M. Chen, "Gaussian mutation–spider monkey optimization (gm-smo) model for remote sensing scene classification," *Remote Sensing*, vol. 14, no. 24, p. 6279, 2022.

[34] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, "A genetic algorithm for the waitable time-varying multi-depot green vehicle routing problem," *Symmetry*, vol. 15, no. 1, 124, 2023.