Received July 14, 2020, accepted July 26, 2020, date of publication August 3, 2020, date of current version August 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013886

Two-Stage Adaptive Constrained Particle Swarm Optimization Based on Bi-Objective Method

QIAN FENG [1,2], QING LI [1,3], HENG WANG [1,3], (Member, IEEE), YONGFENG FENG [1], AND YICHEN PAN [1]

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] College of Mechanical Engineering, North China University of Science and Technology, Tangshan 063009, China
[3] Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing 100083, China

Corresponding author: Qing Li (liqing@ies.ustb.edu.cn)

This work was supported in part by the Funds of National Science of China, China, under Grant 61673098.

ABSTRACT For the sake of better balancing the relationship between diversity and convergence when handling constrained optimization problems, a two-stage adaptive constrained particle swarm optimization algorithm based on bi-objective method (TABC-PSO) is proposed. In accordance with different phases of the constraint process, the target-constraint space derived from the angle is partitioned adaptively, and simultaneously the global best particle is selected and the external archive set is safeguarded. In the first stage, the whole space is divided adaptively in term of the angular distribution of individual, and the feasible region is explored comprehensively. In the second stage, local regions are adaptively compartmentalized and in-depth exploitation is carried out. Primary and secondary external archive sets are established to maintain population diversity and accelerate convergence. The two phases are switched adaptively in light of the storage status of the two external archive sets. We evaluated TABC-PSO algorithm on the benchmark functions in CEC 2006 and CEC 2017. The experimental results show that TABC-PSO algorithm compared with other state-of-the-art algorithms can be superior to applied to test functions with different types of constraints and possesses a competitive search capability.

INDEX TERMS Constrained optimization, particle swarm optimization algorithm, bi-objective optimization, adaptive.

I. INTRODUCTION

Optimization problems have been used more and more widely in many fields such as scientific research, industrial production, engineering technology, and economic management. Nevertheless, due to constraints in practical applications, some or all of the non-inferior solutions may be in the infeasible region. These optimization problems are called constrained optimization problems (COPs). The effect of dealing with COPs depends on the effective constraint processing mechanism on the one hand and the advanced search mechanism on the other. In recent years, as a group-based optimization method, evolutionary algorithm has been used to solve COPs with its advantages of fast convergence speed and high search efficiency [1].

In light of different constraint processing mechanisms, constraint optimization algorithms can be divided into three categories: penalty function method, multiobjective optimization method and hybrid method. Researchers combine these constraint processing methods with evolutionary algorithms, propose some constraint optimization methods based on evolutionary algorithms, and have achieved certain results.

Penalty function method is one of the simplest and most extensive constraint processing technology. The fitness function is constructed by adding a penalty term to the objective function, and the COP is transformed into an unconstrained optimization problem. The optimal feasible solution is obtained by adjusting the weight of the penalty function. Therefore, the weight adjustment between objective function and penalty term is the key to solve COPs. Penalty function method was first proposed by Courant [2].

The second kind of method is multiobjective constrained optimization. Different from the penalty function method, this type of method treats objective and constraint as two or more targets respectively, and then applies the multiobjective optimization method to deal with the COPs. The key to

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

this kind of problem lies in the criterion of determining the optimal solution. The most typical comparison criteria are the method according to feasibility rule, the method based on stochastic ranking, the method in term of ϵ constraint and the method due to repairment.

The feasibility rule methods [3] judge the pros and cons of solution in accordance with preset criteria, and the setting of the criteria is generally based on experience and preference for feasible solutions [4]. The ϵ constraint processing method [5] is essentially an extension of the feasibility rule. The ϵ level comparison method is used to judge the superiority of the solution. This method further improves the search effect by dividing the constraints.

Stochastic ranking methods [6] adopt random mechanism to balance the objective function and penalty items. The repairment method [7] is in light of a local search mechanism and uses repairment strategies to convert some representative infeasible solutions into feasible solutions.

The third type of method is hybrid approach. The method merges with different constraint processing mechanisms to perform constraint optimization. This method is mainly divided into two categories: the first is to divide the entire population into multiple sub-populations, which are optimized using different constraint mechanisms respectively [8]. The other is to divide the whole evolution process into multiple stages, each stage uses different constraint methods [9].

With the development of evolutionary computation, penalty function method is combined with evolutionary algorithms such as annealing algorithms [10], genetic algorithms [11], and also combined with collaborative algorithms to form collaborative penalty function methods [12].

Due to the fast convergence speed and high search efficiency of evolutionary algorithm, the union of multiobjective optimization method and evolutionary algorithm has achieved good results. Combining feasibility rule with difference algorithm, Mezure-Montes and Coello recommended two diversity mechanisms [13]. The differential evolution with dynamic parameters selection (DE-DPS) algorithm [14] and feasible rule with the incorporation of objective function information (FROFI) algorithm [15] have also achieved certain results. Inosculated with genetic algorithm, [16] advised an multiparent crossover (MPC) algorithm.

ϵ constraint processing method is incorporated with differential evolution algorithm [17], genetic algorithm [18] and particle swarm algorithm [19], and stochastic ranking method is combined with differential algorithm [20], which has achieved good optimization effect.

There are many ways to combine hybrid method with evolutionary algorithm. Hybrid differential evolution and adaptive trade-off model (ATM-HD) algorithm [21] incorporates multiple mutation strategies. Thereafter, Elsayed *et al.* [22] proposed a strategy that differential evolution is combined with two constraint processing mechanisms of feasibility rule and ϵ constraint processing method. Differential evolution algorithm [23] based on orthogonal design is used to deal with COPs. Datta and Deb united penalty function method, multiobjective method and NSGA-II, and put forward a constraint processing technique in conformity with individual penalty [24].

As a typical evolutionary algorithm, particle swarm optimization (PSO) shows good rapidity and convergence when dealing with COPs. With particle swarm algorithm as search mechanism, penalty function method based on fuzzy rules [25], penalty function with memory [26], adaptive penalty function method [27] has been applied for constraint optimization. Paper [28] proposed a new self-adaptive mechanism for adapting the PSO parameters. The parameter configuration of each individual in the population optimized by a PSO variant is adjusted through an adaptive mechanism.

In addition to the widespread penalty function method, PSO algorithm is also combined with other constraint processing methods, and has achieved good optimization results. Constrained multi-swarm particle swarm optimization without velocity (CMPSOWV) algorithm [29] introduced current swarm evolution and memory swarm evolution to strengthen the capacity of exploration and exploitation. The multi-swarm technique and a mutation scheme are incorporated to prevent the population diversity loss and premature convergence.

When dealing with COPs, we not only want to take advantage of the rapidity and convergence of PSO, but also need to effectively avoid premature convergence and maintain a certain population diversity. With respect to constrained optimization problems with a small proportion of feasible solutions, there are few or no feasible solutions in the initialization population. Accordingly, good diversity can promote the discovery and expansion of feasible regions in the initial stage. With the continuous deepening of constrained optimization process, the key task is gradually shifted to the thorough exploration of one or several local districts. Therefore, this article proposes a two-stage adaptive particle swarm constrained optimization algorithm based on bi-objective method. In the first phase, the bi-objective method is applied to constraint optimization in combination with adaptive angle region division and the establishment of the two external archive sets in the entire target-constraint (f-v) two-dimensional space. In the second phase, the global best individual (gbest) is selected from the primary external archive set (arc_p) and the secondary external archive set (arc_s) respectively, and the updated population is subjected to information interaction to guide the population individuals to approach the boundary of constraint conditions and strengthen local in-depth search. Compared with previous research results, the innovation points of our paper are as follows:

1) In the f-v space, the region is partitioned adaptively derived from angle, and the phase is divided according to the angle region distribution of particles. Furthermore, the bi-objective optimization method is applied as a constraint processing mechanism. In the first stage, the possible distribution areas of feasible solutions are searched comprehensively for the whole space; While

in the second stage, the optimal solution is mined deeply in accordance with the local area.

2) Two external archive sets are established: the arc_p and the arc_s. The external archive set is maintained by combining the distribution of particles in the angular region of the f-v space. The establishment of the arc_s is conducive to extracting useful information of some infeasible solutions or non-optimal solutions.
3) Double selection mechanism of gbest. Two external archive sets are respectively taken as "alternative sets" of optimal particles. The first phase is conducive to increasing exploration of unknown feasible regions. The second phase makes reasonable use of effective information of infeasible solutions to avoid premature convergence to local optimization.

The structure of the paper is roughly divided into several parts: Section I mainly introduces the constrained optimization technology and the research progress combined with evolutionary algorithm, especially PSO constrained optimization algorithm. Simultaneously, the main innovations of phased particle swarm constrained optimization based on bi-objective method are put forword. Section II recommends the basic theories of constrained optimization, normalization and particle swarm optimization, and explains the research motivation of this article through the analysis of some advanced PSO constraint algorithms. Section III is the interpretation and flow of TABC-PSO algorithm. Section IV applies TABC-PSO algorithm to solve the typical standard constraint function problem. The experimental results are compared with other advanced constraint optimization algorithms and the performance of the algorithm is evaluated through nonparametric test. Section V draws some conclusions and looks forward to the future work.

II. MOTIVATION AND RELATED WORK

A. CONSTRAINED OPTIMIZATION PROBLEM

1) DEFINITION OF CONSTRAINED OPTIMIZATION

The single-objective optimization problem can be expressed as follows:

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } g_i(x) \le 0, \quad i = 1, \ldots, p \\ &\qquad h_i(x) = 0, \quad i = p+1, \ldots, m \\ &\qquad l_k \le x_k \le u_k, \quad k = 1, \ldots, n \end{aligned} \tag{1}$$

The total number of constraints is m, $g_i(x) \le 0 (i = 1, \ldots, p)$ is p inequality constraints, $h_i(x) = 0 (i = p+1, \ldots, m)$ is m-p equality constraints. $x = (x_1, \ldots, x_n) \in X \subseteq R^n$ is decision variables in n-dimensional decision space X. The value range of the k-th element x_k of the decision variable x is $[l_k, u_k]$. In the decision space, the solution that can satisfy all constraints is called feasible solution, and the space formed by all feasible solutions is called feasible region $\Omega \subseteq R^n$.

In practice, equality constraints are usually converted into inequality constraints. Therefore, the degree of constraint violation of individual x on the i-th constraint condition is expressed as:

$$G_i(x) = \begin{cases} \max\{g_i(x), 0\}, & 1 \le i \le p \\ \max\{|h_i(x)| - \delta, 0\}, & p+1 \le i \le m \end{cases} \tag{2}$$

Among them, δ is the tolerance parameter of equality constraint, which can be set on the basis of the required precision, usually 0.0001. Therefore, the total degree of constraint violation of individual x, that is, the degree of constraint violation, is expressed as:

$$v(x) = \sum_{i=1}^{m} G_i(x) \tag{3}$$

2) NORMALIZATION PROCESSING

In practical problems, there is usually a large gap between multiple objective functions. If they are directly aggregated, target values with small values are often ignored, which affects the balanced treatment among various objectives. Therefore, we need to normalize each target value. The objective function value obtained after normalization is more objective.

Similar to the objective function, many constraints often have great differences, which leads to large numerical constraints playing a leading role in the degree of constraint breach. Therefore, the constraint conditions need to be normalized.

The normalized constraint violation degree $v_{\text{norm}}(x)$ of an individual x is defined as the average value of each constraint violation standard value of the individual:

$$v_{\text{norm}}(x) = \frac{1}{m} \sum_{i=1}^{m} \frac{G_i(x_j)}{G_i^{\max}}, \quad j \in \{1, 2, \ldots, N\} \tag{4}$$

where N is the population size, $G_i^{\max}(i = 1, 2, \ldots, m)$ is the maximum value for each constraint.

Similarly, the dissimilarities between the objective function and the constraint conditions can also be balanced by standardization in the process of constraint optimization. After standardization, the impact of target value and constraint default on optimization is compromised. The normalized objective function $f_{\text{norm}}(x)$ of individual x is:

$$f_{\text{norm}}(x) = \frac{f(x_j) - f_{\min}}{f_{\max} - f_{\min}}, \quad j \in \{1, 2, \ldots, N\} \tag{5}$$

Among them, $f_{\min}$ and $f_{\max}$ are the minimum and maximum values of the objective function of all individuals in the current population, respectively.

3) INTRODUCTION OF PARTICLE SWARM ALGORITHM

Particle Swarm Optimization [30] is a heuristic swarm intelligence algorithm inspired by the behavior of bird swarms in the biological world. Due to the simple and efficient implementation of particle swarm optimization, it has been widely used in unconstrained optimization problems [31], constrained optimization problems [32], and other practical applications.

The speed and position of the i-th ($i = 1,2, \ldots, N$, N is the population size) particle are determined by the information of the global best particle (gbest) and the individual best particle (pbest). The update formula is as follows:

$$\begin{cases} v_d(t) = \omega v_d(t-1) + c_1 r_1 \left(pbest_d - x_d(t-1)\right) \\ \qquad\quad + c_2 r_2 \left(gbest_d - x_d(t-1)\right) \\ x_d(t) = x_d(t-1) + v_d(t) \end{cases} \tag{6}$$

where t is the iteration number; ω is the inertial weight; c_1 and c_2 are two learning factors with a uniform distribution in the range [0,2]; r_1 and r_2 are two random variables with a uniform distribution in the range [0,1]. $pbest_d$ and $gbest_d$ are the d-th decision variable of the personal best solution and the global best solution for particle i, respectively.

B. RESEARCH MOTIVATION OF ALGORITHM

In Part I of this article, we have already introduced a variety of methods to deal with constrained optimization. The most prominent advantage of applying multiobjective optimization method to constrained optimization problem is that it can bypass the design of penalty function, not only does not need to set penalty coefficient, but also can effectively avoid the ''premature'' of algorithm. Furthermore, the purpose of multiobjective optimization is to obtain non-dominated solution set with both constraints and objectives, and its good diversity provides great choice for searching the optimal solution.

Zhou *et al.* [33] put forward a double-objective constrained optimization algorithm based on Pareto strength and minimum generation gap (MGG) model by taking advantage of Pareto-dominance in multiobjective optimization. The improved particle swarm optimization algorithm MOPSO (mod), which is put forward by Venter and Haftka [34] and takes the bi-objective method as the constraint processing mechanism, and has achieved certain effects on inequality constraint processing. The double-objective method adopted in article [35] is presented to solve nonlinear constrained programming problems (NLCPs). Considering maintaining a certain infeasible solution ratio, a new fitness function is designed to measure the degree of violation of constraint targets.

Due to its simple principle and fast convergence speed, evolutionary algorithms such as PSO are not only used to deal with relatively simple single-objective optimization problems, but also have been well applied in multiobjective optimization and constraint optimization. A hybrid particle swarm optimization (CPSO-Shake) algorithm [36] based on double population strategy and chattering mechanism is proposed. The two populations search different regions respectively. When the infeasible solution reaches a certain proportion, shake mechanism is applied to improve the diversity of the population and effectively alleviate the problem of early convergence. A parallel boundary search particle swarm optimization (PBSPSO) algorithm is developed in paper [37]. A cooperation mechanism of the two branches is established, and each branch adopts different methods for global search and local boundary search respectively.

If the degree of constraint violation is used to select excellent particles, it is helpful for individuals to approach the feasible region. However, the disadvantage brought by rapid convergence is that it converges to a few or even a single ''optimal'' individual, resulting in the loss of diversity. When the number of feasible regions in the decision space is multiple or the shape is special, the algorithm is easy to fall into local optimization. For the purpose of solve this problem, we take the best individual of the two archive sets as the ''leader'' respectively, and the number of particles in the external archive set is balanced by the way of angular region distribution.

Since the optimal solution is often located on the constraint boundary, relying on the feasible solution alone obviously has certain limitations. Therefore, we hope to reasonably use the information of feasible and infeasible solutions to accelerate the approach to the constraint boundary. For this motive, we will not only conduct in-depth exploration on the region where the current optimal feasible solution is located, but also use the infeasible solution information near the region to accelerate the approach to the boundary of the feasible region.

A MOPSO algorithm based on adaptive angle region division [38] is proposed. In the target space, the method adaptively divides angular regions and strengthens the guidance and search of ''low density'' regions, which is beneficial to maintaining the diversity of the population and obtaining uniformly distributed non-dominated solution sets. Inspired by this algorithm, different adaptive angle region partition methods are adopted in the f-v space for two stages. The first phase place special emphasis on the search of global feasible regions, and the second phase stress a particular aspect on the in-depth mining of local areas. Simultaneously, in term of the distinct emphasis of the two stages, secondary external archive set is established, and dissimilar external archive set maintenance mechanisms and gbest selection mechanisms are designed.

Compared to the algorithms already existed, the bi-objective method proposed in this article divides the target-constraint space based on angle.

1) Selection mechanism of optimal particles: not only sorting according to Pareto dominance, but also combining distribution of particles in angular region to prevent falling into local optimization;
2) Archive storage mechanism: two external archive sets are established to preserve some infeasible solutions in the angle areas and obtain better diversity;
3) Archive maintenance method: redundant particles are deleted according to the number of particles in the angle districts, which is different from the method in view of crowding degree or grids and is conducive to probe of infeasible areas;
4) In the later stage of optimization, the local angle region is divided adaptively to strengthen the exploration of

the local region and promote the convergence of the algorithm.

III. TABC-PSO ALGORITHM

We expect to give consideration to both constraints and objectives, which will not only help prevent the algorithm from falling into local optimization, but also help maintain good diversity in stage I. In the whole f-v space, the search intensity is increased, and through comprehensive search, widely distributed feasible solutions are obtained. In this article, we use the multiobjective particle swarm optimization (MOPSO) algorithm as the search mechanism and adaptively adjust the division of regions based on angles in the whole f-v space. According to the regional distribution of the solution, the exploration of feasible regions will be further expanded.

In order to speed up the convergence, in stage II, combining with the adaptive division of local regions in the f-v space, a more in-depth search is carried out for the local regions; The gbest is selected and the "useful" information of some infeasible solutions is reasonably utilized. An adaptive mechanism is adopted to switch between the two phases. Next, we will introduce our algorithm in detail in two stages.

A. ESTABLISHMENT AND MAINTENANCE MECHANISMS OF TWO EXTERNAL ARCHIVE SETS

In stage I of the bi-objective particle swarm constrained optimization algorithm, we expect to obtain the non-dominated solution set that is uniform and covers all feasible regions. The method of angle region division is helpful to guide individuals to search in light of the distribution of particles in the f-v space, so that the non-dominated solution set formed by the target value and the constraint value is more uniform.

In general, when bi-objective optimization is applied to constraint optimization, feasible solutions are selected according to the non-dominated ranking. In this process, some infeasible solutions with low non-dominated ranking will be deleted, and these individuals may even be closer to the constraint boundary than feasible solutions if the degree of violation of constraints is very low. Therefore, these solutions are often of some value. If they can be used, they will be helpful to approach the constraint boundary.

In the f-v space shown in Fig. 1, in conformity with the comparison of Pareto non-dominance levels, particles a, b, c and d will be stored in the external archive set, while particles a_1 and c_1 will be deleted because the non-dominance level is lower than particles a and c. This will lead to the area where particles a_1 and c_1 are located losing the opportunity to be explored, which is not conducive to the maintenance of population diversity. Nonetheless, the establishment of the arc_s will save the highest non-dominated particles in each angle region. The preservation of these particles, such as a_1 and c_1, enables the next update of these particles, which is beneficial to the development of unknown feasible regions, and also helps to maintain the diversity of the population and prevent the algorithm from falling into local optimization.

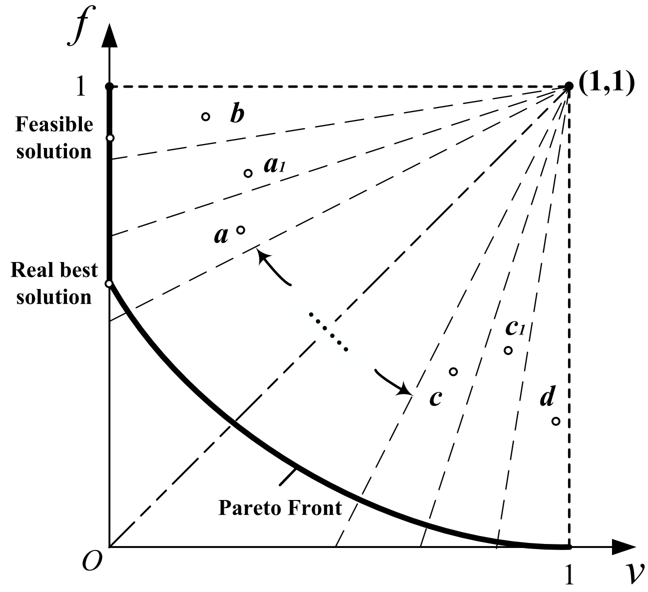


FIGURE 1. Schematic diagram of target-constraint space adaptive angle region division in the first stage.

For the sake of utilizing particles that are not stored in the external archive set but may contain some useful information, we recommend the arc_p and the arc_s. The arc_p is employed to store the feasible solution of the highest non-dominant level in the global or angular region. The arc_s is used to store infeasible solutions of the highest non-dominant level in the global or angular region. Combined with the distribution of angle regions, useful information of some infeasible solutions is extracted.

We set both the number upper limit of particles stored in the arc_p and the number upper limit of particles stored in the arc_s to 50. The greater the upper limit, the better the diversity. However, if the upper limit is too large, on the one hand, it will affect the calculation efficiency and increase the complexity; on the other hand, it will lead to the first stage consuming too long time and affect the convergence. After dividing the adaptive region, there is only one particle in each angle region saved by the two external archive sets. The principle for deleting redundant particles is: the arc_p retains particles with small target value, while the arc_s retains particles with low degree of constraint violation.

The number of particles in each angular region is taken as the standard to measure "sparse" or "dense" regions. With the increase of the number of particles in the two external archive sets, the division of angle regions is continuously refined. When the external archive set does not reach the upper limit, in order to ensure good diversity, it is necessary to augment the guidance of individuals in the "sparse" area where there are fewer particles. When dealing with COPs, emphasis is placed on the guidance of particles with low degree of constraint violation and located in "sparse" region. When the number of particles in the primary external archive set and the secondary external archive set exceeds the upper limit, there is only one particle in each angular region. The selection of gbest and pbest and the maintenance of the two external archive sets are accomplished synchronously in the entire process of angle region adaptive partition, the selection of gbest and pbest, and the maintenance of two external archive sets are carried out.

B. GBEST SELECTION MECHANISM

For the reason that the MOPSO algorithm is developed from the single-objective particle swarm optimization (SOPSO) algorithm, the obtained optimization result is the highest non-dominated solution set rather than the unique solution in the multiobjective optimization problem. The selection of gbest is an obstacle point in the process of multiobjective optimization. Under normal circumstances, when PSO is regarded as a search mechanism for constraint optimization, the selection principle of gbest is: if the feasible solution does not exist, the individual with the lowest degree of constraint violation is selected; if there are some feasible solutions, the particle with the smallest target value of the feasible solution is selected; when feasible solution and infeasible solution coexist, feasible solution takes precedence. This approach lays particular emphasis on feasible solutions, but counts against the development of unknown feasible areas and the maintenance of population diversity.

In this article, we propose a new gbest selection mechanism combined with adaptive angle region division in the target constraint space. Both the degree of constraint violation and the density of particle distribution are thought over.

1) SELECTION OF GBEST IN THE FIRST STAGE

a: IF $P_{arc\_p} = 0$, THERE IS NO FEASIBLE SOLUTION

The particle x_A in arc_s that satisfies the condition of $\frac{f(x_A)-1}{v(x_A)-1} \leq 1$ and has the smallest constraint value in the ‘‘sparse’’ angle region is selected as gbest;

If the condition is not met, the individual x_C with the smallest constraint value is selected as gbest.

b: IF $P_{arc\_p}>0$, THERE IS AT LEAST ONE FEASIBLE SOLUTION

The feasible solution x_A with the minimum target value in arc_p is selected as gbest_1;

The infeasible solution x_B satisfying the conditions: $\frac{f(x_B)-1}{v(x_B)-1} \leq \frac{f(x_A)-1}{v(x_A)-1}$ in arc_s and having the smallest constraint value is selected as gbest_2.

$P_{\mathrm{arc\_p}}$ is the number of particles in arc_p, $P_{\mathrm{arc\_s}}$ is the number of particles in arc_s.

gbest_1 and gbest_2 are used as gbest respectively to update the population. The updated population carries out information exchange and maintains the external archive set according to Part A in Section III.

2) SELECTION MECHANISM OF GBEST IN THE SECOND STAGE

In the decision space, when there is a feasible solution, we will choose the particle with low target value as the ‘‘leader’’ in the feasible solution to guide the individual update. The optimal solution is always on the boundary of feasible region, while the feasible solution with the lowest target value is not necessarily on the boundary of the feasible region. Consequently, in order to guide the population to come close to the boundary, we expect to take advantage of some information of infeasible solutions.

In general, we only use the feasible solution information to draw near the constraint boundary. If the feasible solution ratio is large, it is likely to be limited to the feasible area and lack of opportunity to approach the border. However, the feasible solution and the infeasible solution have the opportunity to near the constraint boundary in the process of approaching each other, regardless of the distribution of feasible regions. The optimal solution is usually the compromise solution obtained after balancing the feasible solution and the infeasible solution. For this reason, the reservation of infeasible solutions, especially infeasible solutions with small degree of constraint violation, if they can be reasonably utilized, is propitious to accelerating the particles to close the constraint boundary, thus improving the convergence efficiency.

Based on this idea, the selection of gbest not only takes into account the minimum target value point in the feasible solution, but also the ‘‘useful information’’ of the individual is very valuable and should be utilized if the infeasible solution with smaller target value appears in the same or adjacent angle area.

Generally speaking, the particle with the smallest target value in the arc_p is selected as gbest. If there is a particle with smaller target value in the arc_s, which is located in two adjacent angular areas of gbest, this particle will be selected as gbest. As shown in Fig. 2, the individual gb_1 with

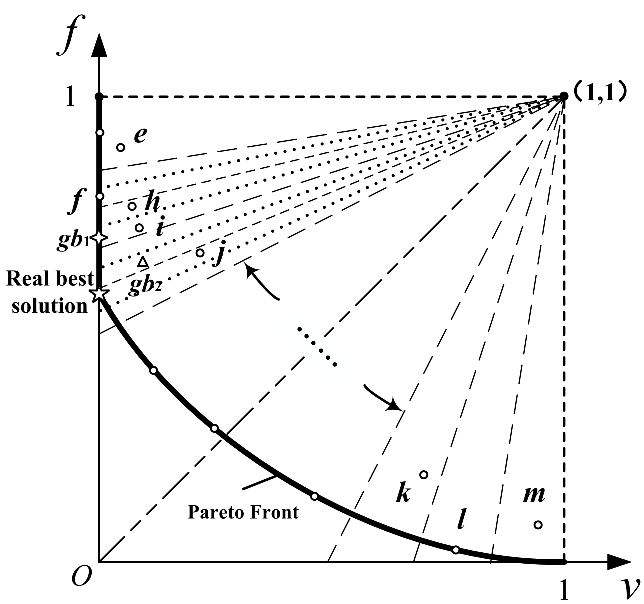


FIGURE 2. Schematic diagram of local adaptive angle region division in the second stage.

Algorithm 1 Framework of TABC-PSO in the First Stage

Input: Number of particles in arc_p $P_{\text{arc\_p}}$, number of particles in arc_s $P_{\text{arc\_s}}$, number of iterations n, number of initial divisions D, number of divisions i, number of archive particles in both arc_p and arc_s P_{arc}

Termination conditions: 1) $P_{\text{arc}} \geq 100$

2)There still no new feasible areas after 10 iterations of continuous updating

Termination while 1) or 2)

Step1: Global adaptive angle region division

1) Angle regions are partitioned in f-v space in the light of $P_{\text{arc}} = P_{\text{arc\_s}} + P_{\text{arc\_p}}$

While $\frac{(i-1)P}{i_{\max}} < P_{\text{arc}} \leq \frac{iP}{i_{\max}}$, divide the target space into $D \cdot i$ regions

(if $D \cdot i \geq 100$, order $D \cdot i = 100$) regions

2) Count $P_{\text{arc\_p}}$, $P_{\text{arc\_s}}$, P_{arc} and the number of particles in each angel region in arc_p and arc_s

Step2: gbest selection

1) if $P_{\text{arc\_p}} = 0$

go to 3)

else

go to 4)

end

2) if $P_{\text{arc\_p}}$>0

go to 5)

else

go to 6)

end

3)The particle x_{A} in arc_s that satisfies the condition: $\frac{f(x_{\text{A}})-1}{v(x_{\text{A}})-1} \leq 1$ and has the smallest constraint value in the "sparse" angle region is selected as gbest;

4) The individual x_{C} in arc_s with the smallest constraint value is selected as gbest;

5) The feasible solution with the minimum target value in arc_p is selected as gbest_1;

6) The infeasible solution x_{B} satisfying the conditions: $\frac{f(x_{\text{B}})-1}{v(x_{\text{B}})-1} \leq \frac{f(x_{\text{A}})-1}{v(x_{\text{A}})-1}$ in arc_s and having the smallest constraint value is selected as gbest_2.

Step3: Maintenance of arc_p and arc_s

If the number of particles in the region is >1, the excess particles in arc_p with large target values are deleted; while the redundant particles in arc_s with high degree of constraint violation are deleted.

Output: a group of solutions

the smallest target value in arc_p is usually regarded as a gbest, but if there is an individual gb_2 with a smaller target value in two adjacent angular regions of gb_1 (in arc_s), then gb_2 will be regarded as gbest. This diversity gbest selection mechanism can not only guide other particles to approach the constraint boundary, but also promote the in-depth search of local areas.

We propose the second stage of gbest selection mechanism: the minimum target particle gb_1 in arc_p is adopted as gbest_1. If there is a smaller target value individual gb_2 in arc_s than gb_1 in the same area or two adjacent areas, then the particle gb_2 is taken as gbest_2. gb_1 and gb_2 are respectively applied as gbest to guide population update, and the updated population is maintained again according to the update mechanism of the external archive set. If there is no one in arc_s that meets the requirements, only gb_1 is selected as the only gbest. By this means, the information in population obtained by gb_1 and gb_2 are respectively taken as gbest is exchanged, the approach of particles to the constraint boundary is accelerated and the search efficiency is improved.

C. ANGLE BASED ADAPTIVE REGION PARTITION MECHANISM

1) STAGE I: THE WHOLE TARGET-CONSTRAINT SPACE

The significance of self-adaptive division of angle areas lies in: in the initial stage, the number of particles in the external archive set is small, the division area is too thin, the particle distribution is "sparse" or there are many areas without particles, it is difficult to guide the particle update of these areas in a targeted manner, which is close to random selection. The angle area is adaptively divided in the company of the increase of the number of particles in the external archive set, which makes the angle area gradually reduce. This is more conducive to the directivity of guidance and further search of particles in the "sparse" area. Meanwhile, aiming at the "no particle" region, the search is enhanced through the guidance of particles in adjacent angle regions, which is beneficial to the development of new feasible regions. Consequently, we make use of the method of adaptive angle region division to strengthen the local search of "sparse" regions and "blank" regions, which is contribute to the development and exploration of feasible regions, can avoid premature

Algorithm 2 Framework of TABC-PSO in the Second Stage

Input: Number of particles in arc_p $P_{\text{arc\_p}}$, number of particles in arc_s $P_{\text{arc\_s}}$,
number of archive particles in both arc_p and arc_s P_{arc}

Termination conditions: Reach the maximum number of iterations

Step1: Initial angel of local adaptive region division

1) particle x_K with the minimum target value in arc_p in f-v space is selected as gbest_1

2) angel region of particle x_K (gbest_1) is determined

While $\tan\left(\frac{\pi}{2}\cdot\frac{p}{50}\right) \le \frac{f(x_K)-1}{v(x_K)-1} < \tan\left(\frac{\pi}{2}\cdot\frac{(p+1)}{50}\right)$

x_K is in the pth angel region

Step2: Local adaptive angle region division

1) if particle x_L in arc_s: $f(x_L) < f(x_K)$

if $\tan\left(\frac{\pi}{2}\cdot\frac{p}{50}\right) \le \frac{f(x_L)-1}{v(x_L)-1} < \tan\left(\frac{\pi}{2}\cdot\frac{(p+1)}{50}\right)$.

divide the pth angel region equally

else if $f(x_L) < f(x_K)$

if $\tan\left(\frac{\pi}{2}\cdot\frac{(p+1)}{50}\right) \le \frac{f(x_L)-1}{v(x_L)-1} < \tan\left(\frac{\pi}{2}\cdot\frac{(p+2)}{50}\right)$.

divide the $(p+1)$th angel region equally

else if $f(x_L) < f(x_K)$

if $\tan\left(\frac{\pi}{2}\cdot\frac{(p+2)}{50}\right) \le \frac{f(x_L)-1}{v(x_L)-1} < \tan\left(\frac{\pi}{2}\cdot\frac{(p+3)}{50}\right)$.

divide the $(p+2)$th angel region equally

go to 2)

else if

go to 3)

2) if x_L is selected as gbest_2

3) if gbest_1 is selected as gbest

Step3: Maintenance of arc_p and arc_s

If the number of particles in the region is >1, the excess particles in arc_p with large target values are deleted; while the redundant particles in arc_s with high degree of constraint violation are deleted.

Output: a group of solutions

convergence more effectively, and also maintain the diversity of the population.

When the angle region is divided in the target constraint space, if (0,0) is still used as the reference point to divide the angle region, due to the final requirements must satisfy the constraints, the search of the optimal individual will concentrate on the target axis of $v(x) = 0$. No matter how small the angle division is, the entire target value axis $f(x)$ coordinate axis only belongs to one angle region. In this way, at most 2 solutions (one is in the arc_p and the other is in the arc_s) corresponding to each angular region in the external archive set can be saved, which greatly reduces the in-depth search of the $f(x)$ axis. Here, we consider the point (1,1) as the reference point for angle regionalization. Under the circumstances, the entire $f(x)$ coordinate axis will be finely searched with the increase of angle regions, which is contribute to the exploration of the optimal solution in such important local angle regions, as well as diversity maintenance.

Usually the optimal solution is located at the edge of the feasible region. That is to say, the optimal solution is the particle with the minimum target value that just does not violate the constraint conditions. Accordingly, the optimal solution is the region of $v(x) = 0$ in the f-v space, as shown in Fig. 1. The conventional multiobjective optimization algorithm is to delete redundant individuals through calculating the congestion degree, so as to maintain the number of individuals in the external archive set, thus making the non-dominated solution set have better uniformity. This method does not reinforce the local search of the target value axis $f(x)$, which is such an important region. For this hence, we propose that the boundary points (1,1) are applied as the benchmark to divide the angle region adaptively in the f-v space.

In stage I, in the whole f-v space, in pace with the increasing number of particles in the region, the divided regions gradually increase until the upper limit of the region is reached and the division is stopped.

$$\frac{(i-1)P}{i_{\max}} < P_{\text{arc}} \le \frac{iP}{i_{\max}} \tag{7}$$

where D is the number of regions initially divided, i is the number of divisions, and Parc is the number of particles stored in the arc_s and arc_p, as shown in Equation 8, where the maximum capacity of the external archive set is set to 100.

$$P_{\text{arc}} = P_{\text{arc\_s}} + P_{\text{arc\_p}} \tag{8}$$

In the first stage, the algorithm of adaptive angle region division, external archive set maintenance and gbest selection is expanded as Algorithm1.

TABLE 1. Characteristics of the benchmark functions in CEC 2006.

Function	D	Type of Function	$\rho(\%)$	LI	NI	LE	NE	a	$f(x^*)$
g01	13	Quadratic	0.0111	9	0	0	0	6	-15.0000000000
g02	20	Nonlinear	99.9971	0	2	0	0	1	-0.8036191042
g03	10	Polynomial	0.0000	0	0	0	1	1	-1.0005001000
g04	5	Quadratic	51.1230	0	6	0	0	2	-30665.5386717834
g05	4	Cubic	0.0000	2	0	0	3	3	5126.4967140071
g06	2	Cubic	0.0066	0	2	0	0	2	-6961.8138755802
g07	10	Quadratic	0.0003	3	5	0	0	6	24.3062090681
g08	2	Nonlinear	0.8560	0	2	0	0	0	-0.0958250415
g09	7	Polynomial	0.5121	0	4	0	0	2	680.6300573745
g10	8	Linear	0.0010	3	3	0	0	0	7049.2480205286
g11	2	Quadratic	0.0000	0	0	0	1	1	0.7499000000
g12	3	Quadratic	4.7713	0	1	0	0	0	-1.0000000000
g13	5	Nonlinear	0.0000	0	0	0	3	3	0.0539415140
g14	10	Nonlinear	0.0000	0	0	3	0	3	-47.7648884595
g15	3	Quadratic	0.0204	0	0	1	1	2	961.7150222899
g16	5	Nonlinear	0.0000	4	34	0	0	4	-1.9051552586
g17	6	Nonlinear	0.0000	0	0	0	4	4	8853.5396748064
g18	9	Quadratic	0.0000	0	13	0	0	0	-0.8660254038
g19	15	Nonlinear	33.4761	0	5	0	0	0	32.6555929502
g21	7	Linear	0.0000	0	1	0	5	6	193.7245100700
g23	9	Linear	0.0000	0	2	3	1	6	-400.0551000000
g24	2	Linear	79.6556	0	2	0	0	2	-5.5080132716

TABLE 2. Parameter settings of compared algorithms in solving CEC 2006.

Algorithm	Parameter Settings
ISR	Parameter is used to scale the step length $\gamma = 0.6, 0.85, 1.1$, the population of individuals is 15, 30, 60.
SMES	$\mu = 100$, $\lambda = 300$, the number of generations is set to 800, number of objective function evaluations = 240000, initial ϵ_0 is set to 0.001, 0.01 or 3.0.
ATMES	The number of generations is set to 800, the number of fitness function evaluations (FFEs) is equal to 240000, the initial ϵ_0 is set to 3, the value of $\breve{e}$ is specified as 1.0168.
PSO+	GA: population is 30, iterations is 50, mutation is $0.5 - 0.9$, crossover is $0.8 - 0.1$, steady state is 0.2, independent runs is 17; PSO: population is 30, iterations is 50, inertia is $0.9 - 0.3$, independent runs is 17.
AHPSOMO	Tolerance delta = 1E-05, the number of individual particles popsize = 100, the total number of iterations totalGen = 1500, scaling factor $F = 0.7$, crossover probability $CR = 1.0$.
CVI-PSO	Population $N = 50$, $c_1 = c_2 = 2.0$, inertia weight $\omega = 1.0 - 0.5$.

2) STAGE II: AREA OF LOCAL ANGLE

When the first stage reaches the switching condition, it enters the second phase. Switching conditions: when the number of particles in arc_p reaches the upper limit or no new feasible areas have been found after continuous updating for 10 generations.

When the first phase is switched to the second phase, the bias point of constraint optimization will also change. In the second stage, local search of the optimal solution is emphasized. We know that the feasible solution is a point on the $f(x)$ axis. Consequently, it is necessary to strengthen the search for the region where the feasible solution lies on the $f(x)$ axis. Here, we further divide the region where the particle with the smallest target value in arc_p is located and the two adjacent regions along the direction of target value reduction. The above three regions are equally divided again, and each region is divided into two regions, so that more particles will have the opportunity to be stored in arc_p and arc_s. Once a feasible solution with a smaller target value is found, we will continue to divide the region where

TABLE 3. Performance comparison between TABC-PSO with six peer algorithms on g01-g13 of CEC 2006 benchmark functions.

Function	Criteria	ISR	SMES	ATMES	PSO+	AHPSOMO	CVI-PSO	TABC-PSO
g01 (-15.0000000000)	Best	**-15.0000**	**-15.0000**	**-15.0000**	**-15.000**	**-15**	**-14.9999999999998**	**-15.0000**
	Worst	**-15.0000**	**-15.0000**	**-15.0000**	-11	**-15**	**-14.9999999999998**	**-15.0000**
	mean	**-15.0000**	**-15.0000**	**-15.0000**	**-15.0**	**-15**	**-14.9999999999998**	**-15.0000**
	SD	5.80E-14	**0**	1.6E-14	**0**	**0**	0.45 E-15	**0**
g02 (-0.8036191042)	Best	-0.803515	**-0.803601**	-0.803388	0.7967	-0.80308	-0.80009774225808	-0.803573
	Worst	-0.726288	-0.751322	-0.756986	0.7743	-0.7856	-0.74694209848138	**-0.786309**
	mean	-0.781975	-0.785238	-0.790148	0.7967	-0.7944	-0.79087557689617	**-0.801007**
	SD	2.20E-02	1.70E-02	1.3E-02	1.59E-06	0.006182	0.010912	**1.43E-06**
g03 (-1.0005001000)	Best	-1.000	-1.000	-1.000	1.0000	-0.41692	-1.00000000000000	**-1.000500**
	Worst	**-1.000**	**-1.000**	**-1.000**	0.0939	-0.03979	-0.99999999999999	**-1.000499**
	mean	-1.000	-1.000	-1.000	0.6739	-0.2151	-0.99999999999999	**-1.000500**
	SD	8.20E-09	2.10E-04	5.9E-05	**1.16e-15**	0.15971	0.37E-15	4.36E-08
g04 (-30665.5386717834)	Best	-30,665.5390	-30,665.5390	-30,665.5390	-30665.5	-30665.5	-30665.8217102204	**-30,665.5387**
	Worst	-30,665.5390	-30,665.5390	-30,665.5390	-30665.5	-30665.5	-30665.8032411366	**-30,665.5387**
	mean	-30,665.5390	-30,665.5390	-30,665.5390	-30661.0	-30665.5	-30665.8209961102	**-30,665.5387**
	SD	1.10E-11	NA	7.4E-12	5.09E-11	**0**	0.003391	**0**
g05 (5126.4967140071)	Best	**5126.497**	5126.599	5126.498	5126.49	5126.507	5127.27766734565	**5126.4972**
	Worst	5142.472	5304.167	5135.256	5905.7	5248.512	5127.27766734565	**5130.6853**
	mean	5128.882	5174.492	5127.648	5126.49	5168.669	5127.27766734565	**5127.4096**
	SD	7.20E-13	5.00E+01	1.8E+00	1.66E-10	49.74829	**0.0**	1.05E-10
g06 (-6961.8138755802)	Best	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.8**	**-6961.81**	**-6961.81387558016**	**-6961.8139**
	Worst	-6350.262	-6952.482	**-6961.814**	-6783.1	**-6961.81**	**-6961.81387558016**	**-6961.8139**
	mean	-6875.940	-6961.284	**-6961.814**	-6961.7	**-6961.81**	**-6961.81387558016**	**-6961.8139**
	SD	1.90E-12	1.90E+00	4.6E-12	3.18E-1	1.02E-12	**0.0**	**0**
g07 (24.3062090681)	Best	24.307	24.327	**24.306**	24.310	**24.30621**	24.4738268493991	**24.3062**
	Worst	24.642	24.843	24.359	42.25	**24.30621**	29.5242543022246	**24.3062**
	mean	24.374	24.475	24.316	26.46	**24.30621**	26.5612953717409	**24.3062**
	SD	6.30E-05	1.30E-01	1.1E-02	8.52E-14	5.68E-07	1.642519	**0**
g08 (-0.0958250415)	Best	**-0.095825**	**-0.095825**	**-0.095825**	0.09582	**-0.09583**	-0.10545950508841	**-0.095825**
	Worst	**-0.095825**	**-0.095825**	**-0.095825**	0.08544	**-0.09583**	-0.10545950508841	**-0.095825**
	mean	**-0.095825**	**-0.095825**	**-0.095825** 0.09582	**-0.09583**	-0.10545950508841	**-0.095825**	
	SD	2.70E-07	**0**	2.8E-17	1.89E-4	1.20E-17	**0.0**	**0**
g09 (680.6300573745)	Best	**680.630**	680.632	**680.630**	**680.6300**	**680.6301**	680.635400792284	**680.6305**
	Worst	680.763	680.719	680.673	681.47	**680.6301**	680.863957829483	680.6359
	mean	680.656	680.643	680.639	681.6134	**680.6301**	680.755705150086	680.6382
	SD	3.20E-13	1.60E-02	1.0E-02	9.45E-1	**0**	0.079232	2.37E-05
g10 (7049.2480205286)	Best	7054.316	7051.903	7052.253	7050.2600	**7049.248**	7049.27658551720	**7049.2480**
	Worst	8835.655	7638.366	7560.224	7222.77	**7049.248**	7091.88085911943	**7049.2480**
	mean	7559.192	7253.047	7250.437	7050.26	**7049.248**	7053.21431059926	**7049.2480**
	SD	3.20E-03	1.40E+02	1.2E+02	8.13E-3	6.43E-05	10.615642	**0**
g11 (0.7499000000)	Best	**0.75**	**0.75**	**0.75**	**0.7500**	**0.74999**	**0.75000000000000**	**0.7500**
	Worst	**0.75**	**0.75**	**0.75**	1.00	**0.74999**	**0.75000000000000**	**0.7500**
	mean	**0.75**	**0.75**	**0.75**	**0.75**	**0.74999**	**0.75000000000000**	**0.7500**
	SD	1.10E-16	1.50E-04	3.4E-04	1.33E-15	**0**	**0.0**	**0**
g12 (-1.0000000000)	Best	**-1.000**	**-1.000**	**-1.000**	**1.0000**	**-1**	**-1.00000000000000**	**-1.0000**
	Worst	**-1.000**	**-1.000**	-0.994	0.52	**-1**	**-1.00000000000000**	**-1.0000**
	mean	**-1.000**	**-1.000**	**-1.000**	1.0000	**-1**	**-1.00000000000000**	**-1.0000**
	SD	1.20E-09	**0**	1.0E-03	1.27E-10	**0**	**0.0**	**0**
g13 (0.0539415140)	Best	0.053957	0.053986	0.053950	**0.0539**	0.439973	0.05555582974209	**0.05394**
	Worst	0.216915	0.468294	0.053999	0.0622	2.218274	0.09372815700166	**0.05394**
	mean	0.067543	0.166385	0.053959	**0.0539**	0.993015	0.06559074439931	**0.05394**
	SD	7.00E-02	1.80E-01	1.3E-05	4.13E-17	0.711186	0.010177	**0**

the particle is located along the direction where the target value decreases in the light of the same method. In this way, we will strengthen the search for the local angle region in a targeted way by enhancing the particle distribution. As shown in Fig. 2, if the current particle gb_1 with minimum target is gbest_1, we start from the angle area where gb_1

TABLE 4. Performance comparison between TABC-PSO with six peer algorithms on g14-g24 of CEC 2006 benchmark functions.

Function	Criteria	ISR	SMES	ATMES	PSO+	AHPSOMO	CVI-PSO	TABC-PSO
g14 (-47.7648884595)	Best	NA	NA	NA	-47.3639	-47.7615	-47.4530114308810	**-47.7647**
	Worst	NA	NA	NA	-47.36	-47.7615	-42.4277271159680	**-47.7647**
	mean	NA	NA	NA	-47.36	-47.7615	-44.4246904372610	**-47.7647**
	SD	NA	NA	NA	6.38e-14	3.12E-12	1.406368	**0**
g15 (961.7150222899)	Best	NA	NA	NA	961.6500	961.7152	961.715707147147	**961.71518**
	Worst	NA	NA	NA	961.65	965.1479	961.718781406392	**961.71518**
	mean	NA	NA	NA	961.65	962.4226	961.718595474823	**961.71518**
	SD	NA	NA	NA	1.72e-7	1.523719	6.87E-04	**0**
g16 (-1.9051552586)	Best	NA	NA	NA	-1.9030	-1.90516	**-1.9051549999999**	**-1.905155**
	Worst	NA	NA	NA	-1.9030	-1.90516	**-1.9051549999998**	**-1.905155**
	mean	NA	NA	NA	-1.9030	-1.90516	**-1.9051549999999**	**-1.905155**
	SD	NA	NA	NA	3.37e-8	2.48E-16	8.52E-15	**0**
g17 (8853.5396748064)	Best	NA	NA	NA	8942.0700	8935.134	8853.53989132959	**8853.5398**
	Worst	NA	NA	NA	8967.63	8960.543	8853.53989132959	**8853.5398**
	mean	NA	NA	NA	8942.07	8945.909	8853.53989132959	**8853.5398**
	SD	NA	NA	NA	3.59e-9	9.276019	3.70E-12	**0**
g18 (-0.8660254038)	Best	NA	NA	NA	-0.8642	-0.86602	-0.86463128789989	**-0.866025**
	Worst	NA	NA	NA	-0.8587	-0.86602	-0.64443684728892	**-0.866001**
	mean	NA	NA	NA	-0.8642	-0.86602	-0.80910925896692	**-0.866025**
	SD	NA	NA	NA	**7.77e-16**	2.55E-06	0.062701	1.03E-06
g19 (32.6555929502)	Best	NA	NA	NA	41.1469	32.6582	32.8270270899692	**32.65767**
	Worst	NA	NA	NA	86.39	32.6742	43.3749595963395	**32.66802**
	mean	NA	NA	NA	49.48	32.66636	35.0673371976379	**32.65795**
	SD	NA	NA	NA	**1.41e-14**	0.007955	2.286728	1.35E-06
g21 (193.7245100700)	Best	NA	NA	NA	189.3736	193.777	193.786925246823	**193.75194**
	Worst	NA	NA	NA	512.74	325.1039	193.787093370887	**193.78168**
	mean	NA	NA	NA	189.37	246.3077	193.786935225931	**193.77213**
	SD	NA	NA	NA	4.54e-13	71.9307	3.38E-05	**5.21E-06**
g23 (193.7245100700)	Best	NA	NA	NA	-401.09	-400.005	-400.000000000000	**-400.05509**
	Worst	NA	NA	NA	-2.38	-363.511	-400.000000000000	**-400.05509**
	mean	NA	NA	NA	-5.0265	-385.191	-400.000000000000	**-400.05509**
	SD	NA	NA	NA	4.44e-15	17.17549	**0.0**	**0**
g24 (-5.5080132716)	Best	NA	NA	NA	-5.5080	-5.50801	-5.50801327159532	**-5.508013**
	Worst	NA	NA	NA	-3.00	-5.50801	-5.50801327159532	**-5.508013**
	mean	NA	NA	NA	-5.5080	-5.50801	-5.50801327159529	**-5.508013**
	SD	NA	NA	NA	7.99e-15	**0**	9.46E-15	**0**

is located and divide the two adjacent angle areas together twice along the direction of target value reduction on the $f(x)$ axis with (1,1) as the reference point. This occasion, if particle gb_2 with smaller target value appears in arc_s, gb_1 and gb_2will be adopted as gbest_1 and gbest_2 to guide the update respectively. The removal method of redundant particles in the two external archive sets arc_p and arc_s remains unchanged. After the local area is intensively searched, if a feasible solution with a smaller target value is found, we will start with the angle area where the new feasible solution is, and continue to divide the area in the above way.

In the second stage, the algorithm of adaptive angle region division, external archive set maintenance and gbest selection is expanded as Algorithm2.

The advantages of the two-stage multiobjective optimization method based on adaptive angle region division are as follows:

First, it is applicable to all COPs using multiobjective optimization methods, and is not limited to problems. Secondly, according to the different stages of the constraint optimization process, different measures are taken: in the first stage, efforts are striven to explore feasible areas and strengthen the search of the whole space; In the second stage, the search intensity of local regions is enhanced for feasible regions. Third, the establishment of the external archive set and the preservation of non-inferior solutions in the angle region not only retain some useful information of infeasible solutions, but also contribute to the maintenance of diversity. Fourth, different gbest selection mechanisms are adopted in

TABLE 5. Parameter settings of compared algorithms in solving CEC 2017.

Algorithm	Parameter Settings
CMPSOWV	Population size $N = 100$, subpopulation size $N_k = 10$ with $k = 1, \ldots, 10$, acceleration coefficients $c_1 = c_2 = c_3 = 4.1/3$.
C^{2}oDE	The settings of MaxFEs is 2.4E+05, the population size $NP = 50$, the tolerance value δ for equality constraints was set to 10^{-4}, $F_{pool} = [0.6, 0.8, 1.0]$, $CR_{pool} = [0.1, 0.2, 1.0]$, p in the ϵ constrained method and μ in the restart scheme were set to 0.5 and 10^{-8}.
L-SHADE44	current-to-pbest/1 mutation $p = 0.11$, strategy competition $n_0 = 2$, $\delta = 1/(5 * K) = 1/20, K = 4$, historical circle memories $H = 6$, archive $maxsizeA = 2.6 * N$, decreasing of population size $N_{\text{init}} = 18 * D$, $N_{\min} = 4$, the tolerance for equalities is set to $\epsilon = 0.0001$.
L-S44+IDE	Stage I: the size of population $N_{\min} = 50$ the number of required feasible individuals $nfeas = 50$, $FES_{\text{avail}} = 0.99 * MaxFES$,Maximum value of function evaluation $MaxFES = 20000 * D$, $N_{\text{init}} = 18 * D$, $p = 0.11$, $K = 2.6 * N$, $H_m = 6$; Stage II: the population size $N = nfeas = 50$.
UDE	Parameter pool: $[F = 0.9, CR = 0.9]$, $[F = 0.5, CR = 0.5]$, population size $N_p = \min((10 * D), 500)$, D is dimension of problem, size of top sub-population $T = 0.5 * N_p$, p in rank-DE/current-to-pbest/1 = 5, learning period $L = 25$ generations, frequency of local search $f_L = 2$ generations, number of FES consumed by local search in every instance: $N_L = 10$, penalty value $penalty = 100$, mutation factor in local search $F_L = 0.9$.
L-S44-IEpsilon	The maximal population size $N_{\max} = 5D$, the minimal size $N_{\min} = 5$, the length of historic memory $H = 10$, parameters of strategy competition $K = 4$, $n_0 = 2$, $\delta = 1/(5K) = 0.05$, DE/current-to-pbest/1 parameter $p = 0.2$, parameters of IEpsilon $cp = 2$, $\theta = 20\%$, $T_c = 0.8MaxFEs$, $\alpha = 0.5$, $\tau = 0.1$.

different stages. The first stage is conducive to the exploration of unknown feasible regions. In the second stage, infeasible solutions are used to accelerate the approach to the constraint boundary.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. COMPARATIVE EXPERIMENTS OF CEC 2006

In order to verify the feasibility and effectiveness of the algorithm, we use benchmark test functions g01-g24 proposed by CEC 2006 [39] to test the algorithm. These test functions and constraints involve different dimensions and different types of functions, including linear, quadratic, cubic polynomial and non-linear, and can evaluate constraint optimization algorithms from different angles. The basic characteristics of these benchmark functions are shown in Table 1. D refers to the dimension of decision variables. Column 3 describes the types of test functions. ρ is the estimated ratio between the feasible region and the search space. The lower the ratio of feasible solutions is, the more difficult it is to search the feasible solution area. Columns 5-8 in table give the number and types of constraints for each objective function: LI is the number of linear inequality type constraints, NI is the number of nonlinear inequality type constraints, LE is the number of linear equality type constraints, and NE is the number of nonlinear equality type constraints. a refers to the number of active constraints when the optimal solution x^* is obtained. $f(x^*)$ gives the optimal solution obtained so far.

1) ALGORITHMS COMPARISONS USING CEC 2006

We chose two kinds of constrained optimization algorithms to compare with TABC-PSO algorithm to test the effectiveness of the algorithm. Among them, the first type of algorithm is the classical algorithm for constrained optimization: simple multi-agent evolution strategy (SMES) proposed by Mezura-Montes and Ceolo [13], and improved random sorting method (ISR) proposed by Runarsson *et al.* [6]. The third is the more classical ATMES algorithm [9], which is also a phased division and Pareto-based adaptive constrained optimization algorithm. This article designs corresponding tradeoff schemes to different stages to obtain an appropriate balance between objective and constraint.

The algorithm we advance is based on particle swarm optimization, so for the second kind of comparison algorithm, we choose the advanced constrained optimization algorithm with particle swarm algorithm as the search mechanism. AHPSOMO algorithm [40] initializes the individuals of population with good point set (GPS) theory and differential evolution (DE) algorithm is introduced for updating local optimal individuals. PSO+ algorithm [41] also uses particle swarm algorithm as the search mechanism, and ensures the diversity and convergence of the population by using feasible repair operators and improved particle update methods. In addition, the third comparison algorithm is CVI-PSO [42], in which the objective and the constraint are regarded as two objective values respectively, and the constraint optimization is carried out by using the double-objective method. Interval arithmetic is used as the evaluation mechanism of constraint degree,

TABLE 6. Performance comparison between TABC-PSO with six peer algorithms on C01-C14 of CEC 2017 benchmark functions for 10*D*.

Function	Criteria	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon	TABC-PSO
C01	mean	1.77E-02	**0**	**0**	**0**	**0**	**0**	**0**
	SD	1.60E-02	**0**	**0**	**0**	**0**	**0**	**0**
C02	mean	3.39E+00	**0**	**0**	**0**	**0**	**0**	**0**
	SD	8.22E-01	**0**	**0**	**0**	**0**	**0**	**0**
C03	mean	1.20E-02	**0**	31548.2	3.259E+05	73.6548	5.931E+01	**0**
	SD	1.41E-02	**0**	37019.6	2.575E+05	4.4228	4.990E+01	**0**
C04	mean	1.37E+01	**13.573**	13.6147	14.418	24.2371	**1.357E+01**	1.3582E+01
	SD	1.06E-01	**2.42E-14**	0.061549	1.1495	8.3633	6.160E-05	5.79E-05
C05	mean	4.70E-01	**0**	**0**	**0**	**0**	**0**	**0**
	SD	3.02E-01	**0**	**0**	**0**	**0**	**0**	**0**
C06	mean	7.61E+01	**0**	648.899	808.36	75.8562	1.690E+01	**0**
	SD	2.50E+00	**0**	283.706	545.03	13.2422	4.732E+01	**0**
C07	mean	-7.02E+01	**-433.52**	3.74362	-34	-68.3614	-1.205E+02	-2.3529E+02
	SD	3.72E+01	22.646	69.5716	57	79.3309	4.403E+01	**1.87E+01**
C08	mean	-1.21E-03	-0.00135	-0.00135	**0**	-0.00135	-1.348E-03	-1.3500E-03
	SD	2.28E-04	**0**	2.21E-19	**0**	**0**	4.426E-19	**0**
C09	mean	-4.47E-03	-0.00498	-0.00497	**0**	-0.00498	-4.975E-03	-4.9800E-03
	SD	7.70E-04	**0**	2.44E-05	**0**	**0**	2.656E-18	**0**
C10	mean	-5.09E-04	-0.00051	-0.00051	**0**	-0.00051	-5.096E-04	**0**
	SD	9.81E-06	1.56E-15	1.11E-19	**0**	1.42E-15	2.213E-19	**0**
C11	mean	-3.37E+0	-0.80078	-0.16882	**0**	-5.9956	-1.688E-01	**0**
	SD	9.63E+01	3.99E-12	2.83E-17	**0**	8.78E-09	7.986E-06	**0**
C12	mean	4.44E+00	**-0.80078**	3.99316	3.9879	4.73979	3.988E+00	3.9610E+00
	SD	5.41E-01	3.99E-12	0.002322	**0**	3.759476	1.582E-04	**0**
C13	mean	9.19E+00	0.47839	**0**	**0**	5.6324	2.816E+00	**0**
	SD	1.43E-01	1.3222	**0**	**0**	17.15859	1.247E+01	**0**
C14	mean	**1.13E+00**	2.3763	2.8819	3.0001	2.484	2.386E+00	2.1940E+00
	SD	1.34E-02	1.36E-15	0.204114	**0**	0.167566	4.937E-02	3.18E-03

and lexicographic method is used to deal with COPs. The parameter settings of all compared algorithms are adopted from the recommendation of their original literatures and summarized in Table 2.

The initial population number popsize = 100, the initial number of angle regions in the f-v space $D = 2, 3, 4, 5, 6$, the total number of iterations totalGen = 1500, number of particles in arc_p $P_{\text{arc\_p}} = 50$, number of particles in arc_s $P_{\text{arc\_s}} = 50$, number of iterations n, number of divisions i, number of archive particles in both arc_p and arc_s $P_{\text{arc}} = 100$. The population size is set at 100 for all test problems. The maximum number of iterations is set to 1000 generations. All tests have been run independently for 30 times.

2) TEST RESULTS OF CEC 2006

We compare the TABC-PSO algorithm with the above six methods. CEC 2006, as a benchmark function commonly used in literature, is used for testing. The experimental results of 22 benchmark functions in terms of best results, worst results, average results and standard deviation are put in Table 3 and Table 4 are compared with the six classical algorithms. The results of six algorithms ISR, SMES, ATMES and PSO+, AHPSOMO and CVI-PSO are directly extracted from their original papers. The best results of the six comparison algorithms are marked by boldface.

The comparison between TABC-PSO and other six algorithms shows that, except for the test functions g02 and g09, TABC-PSO algorithm is superior to all the comparison algorithms or is equal to the comparison algorithm in most cases. In addition, from the standard deviation data, the proposed method has zero standard deviation for 17 of the 22 test functions (g01, g03-g08, g10-g17, g23 and g24), so the algorithm has good robustness. Moreover, the TABC-PSO algorithm does not affect the optimization effect due to the reduction of the feasible solution ratio of the test function. Except for the test functions g02, g04, g12, g19 and g24, the feasible solution ratio of other test functions is very low, especially the feasible solution ratio of the test functions g05, g11, g13, g14, g16-g18, g21 and g23 is very small, so the probability of finding a feasible solution after population initialization is very small. However, at the stage when no feasible solution is found, the TABC-PSO algorithm enhances the development of possible areas of feasible solutions, which is conducive

TABLE 7. Performance comparison between TABC-PSO with six peer algorithms on C15-C28 of CEC 2017 benchmark functions for 10*D*.

Function	Criteria	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon	TABC-PSO
C15	mean	**1.05E+01**	8.5137	14.5456	11.278	7.5084	2.859E+00	2.0870E+00
	SD	4.76E+00	5.4776	3.77422	2.344	2.199144	1.485E+00	**1.03E+00**
C16	mean	6.28E+01	**0**	40.715	40.401	6.283	**0**	**0**
	SD	9.29E+00	**0**	6.72452	6.025	9.06E-16	**0**	**0**
C17	mean	9.85E-01	1.0443	0.911098	1	0.9823	5.276E-01	**2.3800E+01**
	SD	3.42E-02	0.24276	0.176694	**0**	0.085664	4.682E-01	6.54E-02
C18	mean	1.62E+03	709.44	2105.76	3166.3	615.644	**3.666E+01**	4.6731E+01
	SD	2.623+03	364.38	2114.28	2409.0	1017.693	**3.239E-01**	2.58E+00
C19	mean	6.49E-01	0.34599	1.45E-06	**0**	**0**	**0**	**0**
	SD	1.56E-01	0.26628	3.12E-07	**0**	**0**	**0**	**0**
C20	mean	8.01E-01	0.50682	**0.19298**	0.41561	1.2567	6.179E-01	2.8900E-01
	SD	1.28E-01	0.1463	0.057899	0.12388	0.359046	9.810E-02	**3.21E-02**
C21	mean	7.87E+00	5.6863	3.98811	**3.9879**	5.5889	3.990E+00	4.0160E+00
	SD	2.29E+00	3.9717	0.00052	**0**	4.630544	7.877E-04	5.38E-04
C22	mean	9.41E+01	**3.54E-27**	0.637901	0.1595	5.3135	1.116E+00	9.9700E-02
	SD	8.07E+01	**4.85E-28**	1.49162	0.7973	17.22583	1.827E+00	5.83E-01
C23	mean	2.82E+00	**2.3763**	2.54105	3.0212	2.4066	2.395E+00	**2.3760E+00**
	SD	4.51E-01	2.72E-16	0.248944	0.2093	0.083739	6.393E-02	**0**
C24	mean	5.68E+00	4.6181	8.63932	8.7650	5.6234	2.859E+00	**2.4090E+00**
	SD	4.66E-02	3.0781	0.906899	1.1033	1.919545	1.485E+00	**1.77E-02**
C25	mean	7.10E+01	**0**	38.7043	37.699	6.5344	1.005E+00	**0**
	SD	8.98E+00	**0**	5.1876	7.312	1.256637	1.749E+00	**0**
C26	mean	1.01E+00	**1.12**	1.0562	0.93738	0.8969	7.730E-01	1.0720E+00
	SD	**5.99E-03**	0.30124	0.322893	0.37243	0.206659	3.634E-01	1.03E-01
C27	mean	4.18E+03	2734.8	3546.97	7.94E+03	858.075	**3.660E+01**	2.7632E+02
	SD	4.20E+03	3114.7	4810.83	9.15E+03	3218.672	**1.787E-04**	4.53E+02
C28	mean	3.85E+01	20.976	19.6953	10.763	8.6854	2.876E+01	**9.9920E+00**
	SD	1.03E+01	15.662	11.5459	16.281	6.706656	6.493E+00	**3.58E+00**

to optimizing the test functions with low ratio of feasible solutions. The ratio of feasible solutions of test function g02 is very high, so the first stage of comprehensive search of feasible regions did not show absolute advantages. Overall, the proposed algorithm has a stable effect.

B. COMPARATIVE EXPERIMENTS OF CEC 2017

CEC 2017 [43] is a constraint function test set proposed in recent years, which has been tested in some algorithms. To further verify the performance of our proposed TABC-PSO algorithm, we applied this algorithm to CEC 2017 for testing. In 28 constraint problems of CEC 2017 constraint problem, C01-C03, C05, C13, C14, C17 and C20 are known as inseparable functions, while C04, C06-C12, C15-C16 and C18-C19 are considered as separable functions. In addition, C21-C28 are described as rotated problems.

1) ALGORITHMS COMPARISONS USING CEC 2017

CMPSOWV [34] has been introduced in Section II. C^2oDE [44] is a hybrid constraint method using a strategy pool and a restart mechanism. LSHADE44 [45] is a constrained optimization method based on feasibility rule and dynamically adjusting population size. L-S44+IDE [46] handles constraint problems by coordinating the application of two adaptive differential evolution algorithms. UDE [47] applies three test vector generation strategies and two parameter setting methods. L-S44-IEpsilon [48] is an improved ϵ constraint processing method based on the feasible solution ratio of the current population. The parameter settings of all compared algorithms are extracted from their original literatures and concluded in Table 5.

2) TEST RESULTS OF CEC 2017

We use TABC-PSO algorithm and six comparison algorithms to solve the benchmark function problem of CEC 2017. The mean value and SD value of $D = 10$ are shown in tables Table 6 and Table 7. It can be seen from the two tables that when TABC-PSO solved 28 constraint problems in CEC 2017, 16 functions obtained the best mean value, and 11 functions acquired the second-best mean value. This shows that our proposed algorithm is competitive with other six constraint processing algorithms.

The mean value and SD value of $D = 30$ are displayed in Table 8 and Table 9. Among them, 14 of the

TABLE 8. Performance comparison between TABC-PSO with six peer algorithms on C01-C14 of CEC 2017 benchmark functions for 30*D*.

Function	Criteria	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon	TABC-PSO
C01	mean	2.48E-01	1.58E-29	3.87E-30	**0**	7.34E-29	5.705E-30	**0**
	SD	3.56E-01	2.06E-29	6.10E-30	**0**	6.13E-29	8.052E-30	**0**
C02	mean	8.26E+02	1.17E-29	5.26E-30	**0**	7.39E-29	6.063E-30	**0**
	SD	3.53E+02	1.60E-29	8.39E-30	**0**	5.97E-29	8.640E-30	**0**
C03	mean	8.45E+01	74.78	355118	6.70E+06	73.25376	7.192E+03	**6.6373E+01**
	SD	1.18E+01	**0.31805**	446751	2.25E+06	4.048386	6.339E+03	8.97E+00
C04	mean	4.76E+02	**13.573**	**13.5728**	13.854	82.42193	1.357E+01	2.9815E+01
	SD	6.82E+01	6.89E-14	5.44E-15	0.7782	21.93918	3.626E-15	4.36E+00
C05	mean	3.01E-01	**0**	**0**	**0**	2.32E-17	6.556E-27	**0**
	SD	3.58E-01	**0**	**0**	**0**	9.92E-17	1.595E-26	**0**
C06	mean	3.04E+02	288.88	4071.08	5526.4	303.62	4.923E+02	**2.6733E+02**
	SD	5.38E+00	78.074	981.519	759.06	129.3956	1.083E+02	**5.01E+01**
C07	mean	**-3.84E+01**	-758.6	-109.428	-81.088	-598.134	-2.772E+02	-7.7389E+01
	SD	9.04E+01	231.68	88.7374	90.929	174.0513	5.858E+01	**7.93E+00**
C08	mean	**-2.62E-04**	-0.00028	-0.00028	-2.63E-04	-0.00028	-2.840E-04	**-2.6200E-04**
	SD	3.63E-05	1.20E-08	**0**	2.04E-05	1.45E-05	1.397E-09	**0**
C09	mean	2.45E-03	**-0.00267**	**-0.00267**	**-2.67E-03**	**-0.00267**	**-2.666E-03**	**-2.6700E-03**
	SD	3.42E-04	4.61E-16	1.33E-18	**0.00E+00**	4.23E-16	8.852E-19	**0**
C10	mean	**-1.29E-04**	**-0.0001**	**-0.0001**	**-9.78E-05**	**-1.00E-04**	**-1.028E-04**	**-1.0000E-04**
	SD	6.65E-05	5.67E-09	**0**	3.69E-06	5.20E-06	1.059E-10	**0**
C11	mean	-3.18E+02	-135.22	-0.87479	**-8.65E-01**	-28.3482	-7.690E+02	-1.0205E+01
	SD	1.05E+03	346.86	0.109524	**9.55E-02**	6.426222	5.721E+02	3.82E-01
C12	mean	6.18E+00	**3.9828**	3.98532	6.0679	18.67962	3.987E+00	**3.9830E+00**
	SD	2.57E+00	**0.000173**	0.001443	2.8379	9.459164	4.763E-03	5.78E-04
C13	mean	1.47E+02	**0**	6.14549	26	81.4897	5.163E+01	3.2730E+00
	SD	5.19E+01	**0**	4.72262	39	122.3796	3.952E+01	2.076E-01
C14	mean	2.50E+00	**1.4085**	1.8296	1.9086	1.525867	1.412E+00	**1.4090E+00**
	SD	2.88E-01	5.46E-16	0.083396	0.0496	0.165624	1.738E-02	**0**

28 test functions in CEC 2017 got the best mean value, and 8 got the second-best mean value. Therefore, compared with the other six constrained optimization algorithms, TABC-PSO algorithm still has a strong optimization effect to tackle the 28 benchmark functions. In addition, with the increase of dimensions, the difficulty of solving constraint problems increases greatly, which leads to a decline in the optimization effect of all algorithms. Although the advantages of TABC-PSO algorithm are slightly weakened, it is more advanced than the other six algorithms.

3) COMPARISON OF NON-PARAMETRIC STATISTICAL TEST RESULTS

Wilcoxon signed rank test is a nonparametric method, which is used to evaluate the statistical significance of previous results. It is employed to compare the significant differences between two paired samples. The statistical analysis results are evaluated in terms of the positive rank (R^+), the negative rank (R^-) and p-value for the pairwise performance comparisons.

The results of Wilcoxon signed rank test for the pairwise comparison between CMPSOWV with six peer algorithms are summarized in the table. Wilcoxon signed rank test results of the mean values of seven algorithms when $D = 10$ are shown in Table 10.

Table 11 represents the comparison of algorithms based on Wilcoxon signed rank test for 30D problem's mean result. R^+ shows that TABC-PSO algorithm has advantages in solving COPs in CEC 2017 compared with other six comparative algorithms. In all paired tests with $D = 10$ and $D = 30$, the positive rank is found to be higher than R^-, which means that the TABC-PSO algorithm can get better average than the other six algorithms. The p-value indicates the significant difference between the comparison pairs, and the smaller the p-value, the greater the significant difference between the comparison pairs. At $D = 10$, there are significant differences between TABC-PSO algorithm and the other six algorithms, while at $D = 30$, there is no significant difference between TABC-PSO algorithm and L-S44-IEpsilon and C^2oDE algorithm. This shows that with the increase of dimensions, the advantages of TABC-PSO algorithm decrease slightly. This is because the increase of dimensions will greatly affect the selection pressure. Of course, TABC-PSO algorithm has obvious advantages over most algorithms.

TABLE 9. Performance comparison between TABC-PSO with six peer algorithms on C15-C28 of CEC 2017 benchmark functions for 30*D*.

Function	Criteria	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon	TABC-PSO
C15	mean	2.31E+01	8.765	19.1951	12.912	9.142014	4.995E+00	**4.8560E+00**
	SD	8.40E+00	4.8398	3.61396	1.539	1.739695	1.740E+00	**9.08E-01**
C16	mean	1.51E+02	11.435	146.335	143.01	8.419332	**6.283E+00**	7.1290E+00
	SD	1.99E+01	4.9366	14.0327	12.75	2.934488	**5.859E-08**	3.64E+00
C17	mean	1.07E+00	1.0391	0.996949	1.0128	1.021389	**8.728E-01**	1.0160E+00
	SD	1.63E-01	0.052355	0.016725	**0.0100**	0.011378	1.179E-01	2.47E-01
C18	mean	5.67E+03	6052.5	2787.68	4009.6	3568.187	**3.652E+01**	1.0984E+02
	SD	7.08E+03	7038.6	4348.06	4731.7	1632.765	**4.573E-03**	4.38E+01
C19	mean	1.94E+00	1.5045	6.20E-06	**0**	3.65755	**0**	**0**
	SD	2.23E-01	0.62476	3.40E-07	**0**	4.160429	**0**	**0**
C20	mean	2.98E+00	2.3984	**1.41713**	2.24758	4.581578	1.935E+00	1.6860E+00
	SD	1.94E-01	0.36437	**0.099064**	0.19692	1.55839	1.663E-01	7.85E-01
C21	mean	**4.21E+01**	8.8485	22.5123	27.467	11.91034	1.062E+01	2.6087E+01
	SD	1.93E+01	4.1851	8.90794	9.027	9.760462	8.462E+00	**3.48E+00**
C22	mean	5.13E+06	**6.9553**	1268.44	844.0	94.14725	3.731E+03	1.6331E+01
	SD	2.24E+06	**23.624**	1407.55	1400.1	106.5929	6.529E+03	5.23E+01
C23	mean	2.49E+00	**1.4085**	1.71505	1.8564	1.434988	**1.409E+00**	**1.4090E+00**
	SD	1.30E-01	**1.78E-07**	0.126775	0.0685	0.070857	1.214E-04	4.62E-04
C24	mean	1.93E+01	6.2517	12.1579	13.917	8.387982	6.377E+00	**5.7950E+00**
	SD	1.62E+00	2.9077	1.04195	1.496	1.256637	1.701E+00	**1.35E+00**
C25	mean	**1.73E+00**	10.933	142.188	140.62	15.95915	2.551E+01	1.2908E+01
	SD	**1.05E-01**	4.0426	13.933	11.27	3.589994	8.191E+00	2.39E+00
C26	mean	1.03E+00	1.6216	0.997208	1.0123	1.019398	**9.552E-01**	9.8300E-01
	SD	1.81E-03	1.5118	0.016393	0.0106	0.011186	7.080E-02	**3.99E-02**
C27	mean	5.26E+03	30914	9317.6	5.23E+04	11876.39	**3.884E+01**	1.0984E+03
	SD	8.85E+01	15385	8788.42	4.68E+04	5924.8	**4.437E+00**	3.94E+02
C28	mean	1.39E+02	76.643	143.31	152.85	64.32294	1.560E+02	**5.8387E+01**
	SD	9.58E+00	35.347	24.4725	21.84	12.37468	2.212E+01	**9.27E+00**

TABLE 10. Wilcoxon signed rank test for the pairwise comparison between TABC-PSO with six peer algorithms for 10*D*.

TABC-PSO vs.	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon
R^+	368	189	246	220	227	203
R^-	38	51	30	11	26	73
p-value	0.000	0.047	0.001	0.000	0.001	0.048

TABLE 11. Wilcoxon signed rank test for the pairwise comparison between TABC-PSO with six peer algorithms for 30*D*.

TABC-PSO vs.	CMPSOWV	C^2oDE	LSHADE44	L-S44+IDE	UDE	L-S44-IEpsilon
R^+	343	199	261	248	287	203
R^-	35	126	64	28	64	148
p-value	0.000	0.326	0.003	0.001	0.001	0.485

V. CONCLUSION AND FUTURE WORK

This article proposes a single-objective constrained optimization algorithm with bi-objective method as constraint processing mechanism and PSO algorithm as search mechanism. On the basis of the distribution of feasible solutions in the target constraint space, the global and local angle districts are divided in stages adaptively. Through the establishment of primary and secondary external archive sets, the algorithm completes the selection of the optimal individuals and the maintenance of the archive sets. Compared with other classical and advanced constrained optimization algorithms, TABC-PSO algorithm is proved to be effective in standard test functions. As a multiobjective optimization method, TABC-PSO algorithm is beneficial to maintain good population diversity when dealing with complex single-objective constraint problems such as low proportion of feasible solutions and scattered distribution of feasible regions.

The future work is not only to simulate the test function, but also to apply the proposed algorithm to engineering practice.

REFERENCES

[1] C.-H. Lin, "A rough penalty genetic algorithm for constrained optimization," *Inf. Sci.*, vol. 241, pp. 119–137, Aug. 2013.

[2] R. Courant, "Variational methods for the solution of problems of equilibrium and vibration," *Bull. Amer. Math. Soc.*, vol. 49, pp. 1–23, Jul. 1943.

[3] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.

[4] K. Zielinski and R. Laur, "Constrained single-objective optimization using differential evolution," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 223–230.

[5] Z. Fan, H. Li, C. Wei, W. Li, H. Huang, X. Cai, and Z. Cai, "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.

[6] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man Cybern., C (Appl. Rev.)*, vol. 35, no. 2, pp. 233–243, May 2005.

[7] A. Bekasiewicz, S. Koziel, and Q. S. Cheng, "Reduced-cost constrained miniaturization of wideband antennas using improved trust-region gradient search with repair step," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 4, pp. 559–562, Apr. 2018.

[8] R. Mallipeddi, S. Mallipeddi, and P. N. Suganthan, "Ensemble strategies with adaptive evolutionary programming," *Inf. Sci.*, vol. 180, no. 9, pp. 1571–1581, May 2010.

[9] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 80–92, Feb. 2008.

[10] Z. Michalewicz and N. Attia, "Evolutionary optimization of constrained problems," in *Proc. 3rd Annu. Conf. Evol. Program.*, A. V. Sebald and L. J. Fogel, Eds. Singapore: World Scientific, 1994, pp. 98–108.

[11] J. Xiao, J. Xu, Z. Shao, C. Jiang, and L. Pan, "A genetic algorithm for solving multi-constrained function optimization problems based on KS function," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 4497–4501.

[12] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.

[13] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 1–17, Feb. 2005.

[14] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 689–707, Oct. 2014.

[15] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2938–2952, Dec. 2016.

[16] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "GA with a new multi-parent crossover for constrained optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 857–864.

[17] T. Takahama and S. Sakai, "Efficient constrained optimization by the ε constrained adaptive differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[18] S. Bandini and S. Manzoni, *AI*IA 2005: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2005.

[19] Y. AjithAbraham, T. MarioKöppen, and Y. AzumaOhuchi, *Soft Computing as Transdisciplinary Science and Technology*. Berlin, Germany: Springer, 2005.

[20] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Inf. Sci.*, vol. 178, no. 15, pp. 3043–3074, Aug. 2008.

[21] Y. Wang and Z. Cai, "Hybrid differential evolution and adaptive trade-off model to solve constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–5.

[22] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Integrated strategies differential evolution algorithm with a local search for constrained optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 2618–2625.

[23] Y. Wang, H. Liu, Z. Cai, and Y. Zhou, "An orthogonal design based constrained evolutionary optimization algorithm," *Eng. Optim.*, vol. 39, no. 6, pp. 715–736, Sep. 2007.

[24] R. Datta and K. Deb, "Individual penalty based constraint handling using a hybrid bi-objective and penalty function approach," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2720–2727.

[25] C. Saha, S. Das, K. Pal, and S. Mukherjee, "A fuzzy rule-based penalty function approach for constrained evolutionary optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2953–2965, Dec. 2016.

[26] C. Park and S.-H. Kim, "Penalty function with memory for discrete optimization via simulation with stochastic constraints," *Oper. Res.*, vol. 63, no. 5, pp. 1195–1212, Oct. 2015.

[27] R. Datta and K. Deb, "Uniform adaptive scaling of equality and inequality constraints within hybrid evolutionary-cum-classical optimization," *Soft Comput.*, vol. 20, no. 6, pp. 2367–2382, Jun. 2016.

[28] S. M. Elsayed, R. A. Sarker, and E. Mezura-Montes, "Self-adaptive mix of particle swarm methodologies for constrained optimization," *Inf. Sci.*, vol. 277, pp. 216–233, Sep. 2014.

[29] K. M. Ang, W. H. Lim, N. A. M. Isa, S. S. Tiang, and C. H. Wong, "A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112882.

[30] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73.

[31] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.

[32] J. Yang, X. Wang, and P. Bauer, "Extended PSO based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76328–76341, 2019.

[33] Y. Zhou, Y. Li, J. He, and L. Kang, "Multi-objective and MGG evolutionary algorithm for constrained optimization," in *Proc. Congr. Evol. Comput. (CEC)*, Dec. 2004, pp. 1–5.

[34] G. Venter and R. T. Haftka, "Constrained particle swarm optimization using a bi-objective formulation," *Struct. Multidisciplinary Optim.*, vol. 40, nos. 1–6, pp. 65–76, Jan. 2010.

[35] C.-A. Liu, "New multiobjective PSO algorithm for nonlinear constrained programming problems," in *Proc. Intell. Syst. Knowl. Eng. (ISKE)*, 2007, pp. 80–83.

[36] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving constrained optimization problems with a hybrid particle swarm optimization algorithm," *Eng. Optim.*, vol. 43, no. 8, pp. 843–866, Aug. 2011.

[37] Z. Liu, Z. Li, P. Zhu, and W. Chen, "A parallel boundary search particle swarm optimization algorithm for constrained optimization problems," *Struct. Multidisciplinary Optim.*, vol. 58, no. 4, pp. 1505–1522, Oct. 2018.

[38] Q. Feng, Q. Li, P. Chen, H. Wang, Z. Xue, L. Yin, and C. Ge, "Multi-objective particle swarm optimization algorithm based on adaptive angle division," *IEEE Access*, vol. 7, pp. 87916–87930, 2019.

[39] E. Mezuramontes, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Int. J. Comput. Assist. Radiol. Surgery*, no. 2, 2005.

[40] H. Hu, S. Tian, Q. Guo, and A. Ouyang, "An adaptive hybrid PSO multi-objective optimization algorithm for constrained optimization problems," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 6, Sep. 2015, Art. no. 1559009.

[41] M. Kohler, M. M. B. R. Vellasco, and R. Tanscheit, "PSO+: A new particle swarm optimization algorithm for constrained problems," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105865.

[42] I. Mazhoud, K. Hadj-Hamou, J. Bigeon, and P. Joyeux, "Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1263–1273, Apr. 2013.

[43] G. Wu, R. Mallipeddi, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," Nat. Univ. Defense Technol., Changsha, China, Tech. Rep., 2016.

[44] A. Trivedi and D. Srinivasan, "Empirical investigations into the composite differential evolution on CEC 2017 constrained optimization problems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 280–285.

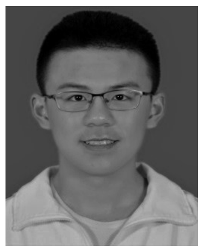
[45] R. Polakova, "L-SHADE with competing strategies applied to constrained optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1683–1689.
[46] J. Tvrdík and R. Poláková, "A simple framework for constrained problems with application of L-SHADE44 and IDE," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1436–1443.
[47] A. Trivedi, K. Sanyal, P. Verma, and D. Srinivasan, "A unified differential evolution algorithm for constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1231–1238.
[48] Z. Fan, Y. Fang, W. Li, Y. Yuan, Z. Wang, and X. Bian, "LSHADE44 with an improved constraint-handling method for solving constrained single-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

QIAN FENG received the B.S. degree from the School of Information Engineering, Inner Mongolia University of Science and Technology, in 2008. She is currently pursuing the Ph.D. degree with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. She is currently an Associate Professor with the North China University of Science and Technology, Tangshan, China. Her research interests include swarm intelligent optimization and simulation of process control systems.

QING LI received the B.Sc. and Ph.D. degrees in control theory and its applications from the University of Science and Technology Beijing, Beijing, China, in 1996 and 2000, respectively. He was a Visiting Scholar with Ryerson University, Toronto, ON, Canada, from 2006 to 2007. He is currently a Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. His research interests include intelligent control and intelligent optimization.

HENG WANG (Member, IEEE) received the bachelor's and Ph.D. degrees from the College of Information Science and Engineering, Northeastern University, China, in 2003 and 2008, respectively. He was a Postdoctoral Research Fellow with the Centre for Autonomous Systems, University of Technology Sydney, from 2010 to 2011, and a Lecturer with the College of Electronic Information and Control Engineering, Beijing University of Technology, China, from 2008 to 2014. From 2014 to 2015, he was a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore, and from 2015 to 2018, he was an Associate Professor with the College of Information Science and Engineering, Northeastern University. He is currently a Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. His current research interests include fault detection, robust control, and mobile robots simultaneous localization and mapping.

YONGFENG FENG is currently pursuing the master's degree in control science and engineering with the University of Science and Technology Beijing, Beijing, China. His current research interests include machine learning and intelligent optimization.



YICHEN PAN is currently pursuing the bachelor's degree with the University of Science and Technology Beijing, China. His current research interest includes fuzzy control.