

Research Article

Quasi-Affine Transformation Evolutionary with Double Excellent Guidance

Tien-Wen Sung , Baohua Zhao , and Xin Zhang 

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

Correspondence should be addressed to Tien-Wen Sung; tienwen.sung@gmail.com

Received 5 February 2021; Revised 29 March 2021; Accepted 1 April 2021; Published 19 April 2021

Academic Editor: Chi-Hua Chen

Copyright © 2021 Tien-Wen Sung et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Quasi-Affine Transformation Evolutionary (QUATRE) algorithm is a swarm-based collaborative optimization algorithm, which has drawn attention from researchers due to its simple structure, easy implementation, and powerful performance. However, it needs to be improved regarding the exploration, especially in the late stage of evolution, and the problem of easy falling into a local optimal solution. This paper proposes an improved algorithm named Quasi-Affine Transformation Evolutionary with double excellent guidance (QUATRE-DEG). The algorithm uses not only the global optimal solution but also the global suboptimal solution to guide the individual evolution. We establish a model to determine the guiding force by the distance between the global optimal position and the suboptimal position and propose a new mutation strategy through the double population structure. The optimization of population structure and the improvement of operation mechanisms bring more exploration for the algorithm. To optimize the algorithm, the experiments on parameter settings were made to determine the size of the subpopulation and to achieve a balance between exploration and development. The performance of QUATRE-DEG algorithm is evaluated under CEC2013 and CEC2014 test suites. Through comparison and analysis with some ABC variants known for their strong exploration ability and advanced QUATRE variants, the competitiveness of the proposed QUATRE-DEG algorithm is validated.

1. Introduction

Optimization problems have been a hot research field since optimization solutions are crucial to various engineering practices. Scientific theories have indicated that appropriate optimization methods applied in different fields [1–7] will bring significant benefits. Accordingly, powerful and efficient algorithms have drawn the attention of researchers in different application fields.

Inspired by the laws of the biological community and natural world, various intelligent optimization algorithms have been proposed. At the early stages, evolutionary algorithms were first developed, such as the Genetic Algorithm (GA) [8] and Differential Evolution (DE) algorithm [9]. Genetic Algorithm is a global optimization algorithm developed by imitating the genetic and evolutionary characteristics of organisms. The core idea of the algorithm is “survival of the fittest in natural selection.” The other algo-

rithm, Differential Evolution, is also an efficient evolutionary algorithm. In its process of searching, the better individuals are continuously selected with the cooperation and competition among particles in the population to find the optimal solution. At the later stages, numerous swarm intelligence algorithms inspired by biology emerged. Particle Swarm Optimization (PSO) [10] is an optimization algorithm proposed by observing the living habits of birds. The algorithm imitates the behavior of continuous migration and clustering in the process of searching foods. It has the characteristics of simple structure and easy implementation. The artificial bee colony (ABC) algorithm [11] is a swarm intelligence algorithm firstly proposed by Karaboga in 2005, inspired by the bee colony behavior of honey source searching. Compared with the premature convergence of PSO, the ABC algorithm is relatively mature in the application of swarm intelligence because of its strong searching ability. In recent years, with the development of computational intelligence, many novel

global optimization algorithms have been proposed, such as Multiverse Optimizer (MVO) [12] algorithm, grasshopper optimization algorithm (GOA) [13], and Quasi-Affine Transformation Evolutionary (QUATRE) algorithm [14]. These algorithms also have a simple structure and excellent performance.

It is well known that particle exploitability and exploratory are considered of equivalent importance in swarm intelligence algorithms, and they should be treated equally in the process of updating particle position. QUATRE algorithm is evolved from DE algorithm. It solves the position deviation problem of DE algorithm and has a stronger development ability. But it is slightly insufficient in search performance. In this paper, a new QUATRE-DEG algorithm is proposed, which divides the population into two subpopulations to increase the diversity of the population and uses the positions of the global optimal individual and the global suboptimal individual to establish the guiding force model, which improves the original mutation strategy and enhances the searching ability of the algorithm. The performance of the proposed QUATRE-DEG algorithm is verified by the experiments on CEC2013 benchmark set [15] and CEC2014 benchmark set [16]. The specific contributions of this paper are as follows:

- (1) The two-population structure of mutation in two central regions is proposed, which maintains the diversity of particles, so that more benchmark environments can be explored
- (2) The gravity model based on the relationship between the global optimal solution and the global suboptimal solution is established and applied to the new operation strategy, so as to enhance the information interaction between the excellent individuals
- (3) In the benchmark set of CEC2013 and CEC2014, some famous ABC variants and advanced QUATRE variants are compared and analyzed with our QUATRE-DEG algorithm. The results show that the performance of the new QUATRE-DEG algorithm is competitive

The next few sections are arranged as follows. The second section reviews the development of QUATRE and ABC algorithms. The third section mainly introduces the QUATRE-DEG algorithm proposed in this paper. In the fourth section, the performance of the QUATRE-DEG algorithm is evaluated by using CEC2013 and CEC2014 benchmarks, respectively. The final section provides the summary.

2. Related Works

QUATRE is a relatively new global optimization algorithm first proposed in 2016. The term QUATRE is the abbreviation of Quasi-Affine Transformation Evolution. The algorithm is based on the DE algorithm and uses particle cooperative evolution in the update process. It replaces the parameter CR in the DE algorithm with a new collaborative search matrix M , which solves the position deviation prob-

lem in the DE algorithm. It is a powerful and simple optimization algorithm. Its operation mechanism is shown in the following formula:

$$\begin{cases} B = X_{g_{best,G}} + F \cdot (X_{r1,G} - X_{r2,G}) \\ X \leftarrow \bar{M} \otimes X + M \otimes B, \end{cases} \quad (1)$$

$$\begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \dots & \\ 1 & 1 & \dots & 1 \end{bmatrix} \sim \begin{bmatrix} & & 1 & \\ & & & 1 \\ & & \dots & \\ 1 & 1 & \dots & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & & & \\ \dots & & & \\ 1 & 1 & \dots & 1 \\ & & 1 & 1 \end{bmatrix} = M, \quad (2)$$

where F is an adjustable mutation size factor. In the original QUATRE algorithm, its optimal value is 0.7. $X_{g_{best,G}}$ is a global optimal matrix, and all row vectors consist of global optimal individuals. $X_{r1,G}$ and $X_{r2,G}$ are all randomly generated matrices. They are generated by randomly replacing the row vector of X , but the total elements remain unchanged. The M is a cooperative search matrix, which is the key to the evolutionary performance of the algorithm. The required M matrix can be generated in three steps. Firstly, the lower triangular matrix is initialized, and the elements in the matrix are 0 or 1. Then, all the elements in the row vector of the matrix are randomly displaced. Finally, the M matrix can be obtained by randomly displacing the row vector in the matrix. \bar{M} is the binary inverse operation of all elements in M , and \otimes represents the component multiplication.

The QUATRE algorithm has the characteristics of fewer control parameters, simple to use, and powerful performance, which attracts more and more scholars to study it. Several successful variants have been proposed [17–21], including the competition-based QUATRE algorithm (C-QUATRE) [22] which employs a pairing competition. The winning individuals are directly retained to the next generation, while the losers need to evolve like the original QUATRE algorithm. It effectively improves the performance of the algorithm. Like the DE algorithm, the QUATRE has the disadvantages of premature convergence and easy to fall into a local optimal solution. The diversity of particles can effectively reduce premature convergence, but the convergence of particles is also important, so finding the balance between particle convergence and diversity is the key. A new sorting-based QUATRE algorithm (S-QUATRE) [23] was proposed, which not only keeps the convergence of particles but also enhances the diversity of particles. Its main improvement scheme is to divide the population into better and worse groups by sorting strategy and to evolve the individuals in the worse group, which improves the search performance of the algorithm. In addition, the performance of the algorithm is greatly affected by the disturbance strategy and the parameters of the disturbance. For example, in QUATRE, a larger disturbance factor F will enhance the algorithm in search, while a smaller one will pay more attention to development. In [24], the BP-QUATRE algorithm is proposed and the original population is divided into different subpopulations by a new sorting

- 1: Initialization: preset maximum number of iterations G , the number of individuals SN , dimension D , and parameters a_1, a_2, z .
- 2: Generate the initial population X , to calculate the fitness of all individuals and get the optimal individual x_{gbest} and the suboptimal individual x_{sbest} .
- 3: According to Eq. (3), individuals are divided into two fixed subpopulations X_1 and X_2 and generate the corresponding global optimal matrix $X_{gbest,G}$ and global suboptimal matrix $X_{sbest,G}$.
- 4: for $g = 1$ to G
- 5: Randomly generate the matrix M according to Eq. (2), then generate the \bar{M} matrix.
- 6: Calculate the Euclidean distance d_{gs} between the global optimal individual and the global suboptimal individual.
- 7: Calculate the guiding force $AF_{gs,G}$ through Eq. (4).
- 8: Generate the matrix B by Eq. (5).
- 9: Evolve individuals in each group using Eq. (6).
- 10: Calculate the fitness value of each individual.
- 11: for $i = 1$ to SN
- 12: if $f(x_i) < f(x_{pbest_i})$
- 13: $x_{pbest_i} = x_i$
- 14: end if
- 15: end for
- 16: Update the current optimal individual x_{gbest} and the current suboptimal individual x_{sbest} .
- 17: Generate the new global optimal matrix $X_{gbest,G}$ and global suboptimal matrix $X_{sbest,G}$.
- 18: end for
- 19: Output: Individual of global optimal solution x_{gbest} and global optimal value $f(x_{gbest})$.

ALGORITHM 1: QUATRE-DEG.

strategy. Different subpopulations use different disturbance modes, and the linear descent parameter F is used to enhance the global optimization ability of the algorithm.

3. Proposed Approach

The original QUATRE algorithm has an inherent shortcoming of paying too much attention to development, which makes it easy to fall into the local optimal solution. The diversity of the population can make up for this shortcoming and enhance the search algorithm. In the iterative process, the global optimal solution of each iteration is widely used in the operation mechanism of a large number of algorithms. It can greatly enhance the convergence of the algorithm, make it easier to find the real global optimal, and also can improve the operation speed of the algorithm. And the excellent individuals in each iteration are also worthy of reasonable utilization, such as the global suboptimal solution. In this study, we propose a new QUATRE algorithm based on double excellent guidance (QUATRE-DEG) which divides the population into two subgroups, including the optimal guided group and the suboptimal guided group. It makes the possible to use the global suboptimal solution to enhance the performance of the algorithm. The details of the new QUATRE algorithm are as follows.

3.1. Initialization. Generate a new population and obtain the current optimal solution and suboptimal solution, as well as set a maximum number of iterations.

3.2. Double Population Structure. Firstly, all individuals are divided into two fixed subpopulations, including the optimal solution population X_1 and suboptimal solution population

X_2 . The number of individuals of optimal solution population and suboptimal solution population is determined by Eq. (3). In the population of optimal solution, B in Eq. (1) still takes the location of the optimal solution x_{gbest} as the center for mutation strategy. But in the population of suboptimal solution, B is the mutation centered on the position of suboptimal solution x_{sbest} . In this way, the individuals in the population will search in two central regions in each iteration, so as to explore more environment regions.

$$\begin{cases} SN_g = (1 - z) * SN, \\ SN_s = z * SN, \end{cases} \quad (3)$$

where SN is the total population and z is the parameter which are used to determine the individual number SN_g of the optimal solution population and SN_s of the suboptimal solution population.

3.3. Relation between Optimal Solution and Suboptimal Solution. For the global optimization algorithm, the task is to find the global optimal solution. In the process of updating and iteration, excellent individuals can play an important role. It should be concerned that the neighborhood of the current global optimal solution may not be better than the one of the global suboptimal solution, which means that both the global suboptimal solution and the global optimal solution contain extremely valuable information. If the information can be shared between them, it can enhance the performance of the algorithm, since there is a strong information interaction between individuals in the grasshopper optimization algorithm (GOA). Inspired by the GOA, this study proposes a model to determine the guiding force

TABLE 1: Comparison on 10D optimization under CEC2013 test suite with different values of parameter z .

QUATRE-DEG	$z = 0.1$	$z = 0.2$	$z = 0.3$	$z = 0.4$	$z = 0.5$	$z = 0.6$
f	Mean/std	Mean/std	Mean/std	Mean/std	Mean/std	Mean/std
f_{a1}	5.80E-04/8.5E-05	5.90E-04/1.2E-04	5.97E-04/7.9E-05	6.20E-04/1.2E-04	6.21E-04/6.9E-05	6.73E-04/1.0E-04
f_{a2}	2.68E+05/1.1E+05	3.40E+05/1.1E+05	4.23E+05/1.9E+05	2.63E+05/1.6E+05	3.29E+05/1.1E+05	3.55E+05/1.4E+05
f_{a3}	9.18E+06/2.6E+07	5.44E+05/1.0E+06	6.73E+05/1.0E+06	2.89E+05/5.4E+05	3.20E+05/4.9E+05	2.02E+06/3.2E+06
f_{a4}	2.01E+01/1.5E+01	2.48E+01/1.8E+01	1.79E+01/1.1E+01	2.77E+01/1.7E+01	2.12E+01/1.5E+01	3.45E+01/2.8E+01
f_{a5}	2.82E-04/6.7E-05	2.93E-04/3.8E-05	2.99E-04/4.7E-05	2.80E-04/3.5E-05	3.30E-04/6.2E-05	3.72E-04/7.1E-05
f_{a6}	6.98E+00/1.0E+01	9.43E+00/1.1E+01	3.35E+00/1.4E+00	4.92E+00/7.5E+00	4.47E+00/7.7E+00	4.97E+00/7.6E+00
f_{a7}	1.67E+01/8.5E+00	1.80E+01/1.4E+01	1.86E+01/1.0E+01	2.19E+01/1.5E+01	1.89E+01/1.2E+01	1.30E+01/1.2E+01
f_{a8}	2.10E+01/5.3E-02	2.09E+01/7.2E-02	2.10E+01/6.0E-02	2.10E+01/5.1E-02	2.09E+01/5.9E-02	2.10E+01/3.1E-02
f_{a9}	1.58E+01/2.9E+00	1.41E+01/5.5E+00	1.67E+01/5.9E+00	1.71E+01/4.6E+00	1.79E+01/3.9E+00	1.42E+01/5.2E+00
f_{a10}	6.09E-02/1.4E-02	5.72E-02/1.4E-02	5.72E-02/1.4E-02	5.68E-02/1.6E-02	6.47E-02/1.4E-02	5.88E-02/1.0E-02
f_{a11}	2.23E+01/5.0E+00	2.62E+01/8.8E+00	2.39E+01/7.5E+00	2.03E+01/5.9E+00	1.87E+01/3.8E+00	2.05E+01/6.7E+00
f_{a12}	6.10E+01/9.1E+00	5.16E+01/1.5E+01	6.41E+01/1.5E+01	5.51E+01/1.6E+01	5.64E+01/2.0E+01	5.35E+01/1.5E+01
f_{a13}	1.22E+02/3.1E+01	1.16E+02/3.1E+01	9.83E+01/3.3E+01	1.13E+02/2.9E+01	1.21E+02/3.4E+01	1.03E+02/3.5E+01
f_{a14}	7.67E+02/1.9E+02	6.38E+02/2.2E+02	6.50E+02/1.7E+02	7.10E+02/1.4E+02	7.66E+02/2.5E+02	7.07E+02/2.5E+02
f_{a15}	3.57E+03/5.9E+02	3.85E+03/6.2E+02	3.46E+03/6.8E+02	4.02E+03/5.2E+02	3.75E+03/7.2E+02	3.92E+03/5.5E+02
f_{a16}	3.11E-01/2.0E-01	3.34E-01/1.1E-01	3.59E-01/1.2E-01	3.40E-01/1.0E-01	3.20E-01/1.5E-01	2.42E-01/1.1E-01
f_{a17}	5.41E+01/4.8E+00	5.15E+01/6.2E+00	4.89E+01/3.5E+00	5.02E+01/6.1E+00	4.59E+01/4.6E+00	4.63E+01/5.3E+00
f_{a18}	8.44E+01/2.0E+01	8.87E+01/1.7E+01	8.49E+01/1.0E+01	7.62E+01/1.8E+01	8.12E+01/1.1E+01	7.65E+01/1.0E+01
f_{a19}	3.04E+00/7.1E-01	2.49E+00/2.9E-01	3.05E+00/5.8E-01	2.19E+00/5.3E-01	2.49E+00/4.6E-01	2.41E+00/5.0E-01
f_{a20}	1.16E+01/5.5E-01	1.17E+01/4.9E-01	1.17E+01/3.4E-01	1.12E+01/6.4E-01	1.13E+01/4.8E-01	1.14E+01/6.0E-01
f_{a21}	2.75E+02/7.7E+01	3.18E+02/1.1E+02	3.33E+02/8.2E+01	3.33E+02/8.2E+01	3.28E+02/1.0E+02	2.65E+02/8.0E+01
f_{a22}	1.09E+03/2.5E+02	6.64E+02/2.7E+02	6.45E+02/1.3E+02	7.37E+02/2.1E+02	6.98E+02/1.6E+02	5.98E+02/2.5E+02
f_{a23}	3.86E+03/4.1E+02	4.10E+03/5.3E+02	3.96E+03/2.9E+02	3.90E+03/8.2E+02	3.93E+03/4.9E+02	4.08E+03/8.0E+02
f_{a24}	2.33E+02/1.2E+01	2.30E+02/1.2E+01	2.37E+02/8.2E+00	2.24E+02/1.4E+01	2.29E+02/1.6E+01	2.25E+02/1.3E+01
f_{a25}	2.60E+02/9.5E+00	2.54E+02/7.2E+00	2.55E+02/3.5E+00	2.54E+02/8.0E+00	2.50E+02/7.8E+00	2.52E+02/9.3E+00
f_{a26}	2.69E+02/7.2E+01	2.38E+02/6.1E+01	2.13E+02/4.0E+01	2.51E+02/6.5E+01	2.23E+02/4.7E+01	2.27E+02/5.6E+01
f_{a27}	6.49E+02/1.0E+02	6.39E+02/7.1E+01	6.61E+02/1.3E+02	6.10E+02/5.8E+01	6.11E+02/7.3E+01	6.50E+02/7.7E+01
f_{a28}	3.97E+02/3.0E+02	3.01E+02/5.1E-02	3.01E+02/6.9E-02	3.01E+02/9.1E-02	3.01E+02/5.2E-02	3.01E+02/1.1E-01
Best quantity	2	4	5	9	4	4

between the global optimal solution and the global suboptimal solution by the distance between them, as shown in Eq. (4).

$$\begin{cases} f(r) = a_1 * e^{-r/a_2} - e^{-r}, \\ r = \frac{d_{gs}}{D+2}, \\ AF_{gs} = \frac{1}{2} * (ub - lb) * f(r) * \frac{x_{gbest} - x_{sbest}}{d_{gs}}, \end{cases} \quad (4)$$

where d_{gs} is the Euclidean distance between the global optimal solution and the global suboptimal solution and D represents the dimension. In the function of $f(r)$, the parameter values of a_1 and a_2 are recommended to be set to 0.5 and 1.5,

respectively. The AF_{sg} represents the attraction of the global optimal solution to the global suboptimal solution. The ub and lb represent the upper and lower bounds of the search space, respectively. In the $f(r)$ function, when r is equal to 2.079, the function value is 0, and when r value increases, the function value will first increase to the maximum value of 0.02 and then gradually decrease to 0. A distance-based attractiveness model is constructed by using the characteristics of $f(r)$. When the distance between the global optimal solution and the global suboptimal solution is close, the attractiveness AF_{gs} of the global optimal solution and the global suboptimal solution is positively correlated with the distance d_{gs} . However, when the distance between the global optimal solution and the global suboptimal solution is far, the attraction AF_{gs} is negatively correlated with the distance d_{gs} .

TABLE 2: Comparison on 10D optimization under CEC2013 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C-QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{a1}	1.32E-04/5.1E-05	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{a2}	8.42E-01/7.4E-01	1.98E-04/1.4E-03(>)	5.29E-03/1.8E-02(>)	1.48E+06/5.5E+05(<)	1.51E+06/6.6E+05(<)
f_{a3}	7.64E+01/8.7E+01	5.60E-01/1.7E+00(>)	1.63E-01/8.9E-01(>)	7.18E+06/6.5E+06(<)	5.54E+07/3.6E+07(<)
f_{a4}	1.82E-02/8.3E-03	6.92E-10/1.4E-09(>)	4.53E-07/8.7E-07(>)	1.27E+04/3.3E+03(<)	5.63E+03/1.7E+03(<)
f_{a5}	7.75E-05/3.5E-05	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{a6}	4.90E+00/4.9E+00	3.17E+00/4.6E+00(>)	5.26E+00/1.0E+01(<)	4.87E+00/3.6E+00(>)	3.22E+00/4.2E+00(>)
f_{a7}	6.14E-01/2.1E+00	2.37E+00/9.4E+00(<)	7.23E-01/2.2E+00(<)	1.54E+01/3.3E+00(<)	1.89E+01/4.8E+00(<)
f_{a8}	2.03E+01/9.1E-02	2.05E+01/9.7E-02(<)	2.05E+01/8.8E-02(<)	2.04E+01/8.2E-02(<)	2.04E+01/7.7E-02(<)
f_{a9}	1.71E+00/1.2E+00	1.74E+00/1.3E+00(<)	1.98E+00/1.3E+00(<)	5.05E+00/5.8E-01(<)	4.15E+00/7.8E-01(<)
f_{a10}	1.73E-01/7.6E-02	1.75E-01/1.1E-01(<)	1.82E-01/1.2E-01(<)	7.61E-01/1.6E-01(<)	1.03E+00/2.2E-01(<)
f_{a11}	2.56E+00/1.7E+00	2.98E+00/1.6E+00(<)	2.75E+00/1.7E+00(<)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{a12}	1.23E+01/5.5E+00	1.47E+01/7.9E+00(<)	1.38E+01/5.9E+00(<)	1.77E+01/4.2E+00(<)	1.57E+01/4.6E+00(<)
f_{a13}	1.75E+01/8.1E+00	2.16E+01/7.3E+00(<)	2.21E+01/8.5E+00(<)	1.90E+01/4.5E+00(<)	1.77E+01/5.1E+00(<)
f_{a14}	6.05E+01/7.3E+01	9.90E+01/9.7E+01(<)	9.80E+01/8.3E+01(<)	3.89E+01/2.7E+01(>)	6.99E+00/4.2E+00(<)
f_{a15}	8.24E+02/2.7E+02	1.08E+03/3.0E+02(<)	9.81E+02/3.0E+02(<)	1.12E+03/1.6E+02(<)	7.15E+02/1.8E+02(>)
f_{a16}	1.15E-01/1.0E-01	1.27E+00/3.4E-01(<)	1.19E+00/3.7E-01(<)	1.04E+00/1.6E-01(<)	1.01E+00/1.6E-01(<)
f_{a17}	1.03E+01/3.1E+00	1.01E+01/4.0E+00(>)	1.12E+01/2.5E+00(<)	1.22E+01/5.7E-01(<)	1.01E+01/1.3E-02(>)
f_{a18}	2.25E+01/5.7E+00	3.27E+01/9.0E+00(<)	3.10E+01/7.0E+00(<)	3.70E+01/3.3E+00(<)	3.42E+01/5.8E+00(<)
f_{a19}	5.23E-01/1.9E-01	6.87E-01/2.0E-01(<)	6.64E-01/2.7E-01(<)	5.24E-01/1.9E-01(<)	5.78E-01/1.8E-01(<)
f_{a20}	2.78E+00/5.7E-01	3.04E+00/5.4E-01(<)	2.92E+00/6.0E-01(<)	3.21E+00/2.3E-01(<)	2.79E+00/3.1E-01(<)
f_{a21}	3.88E+02/4.8E+01	3.57E+02/9.0E+01(>)	3.79E+02/6.1E+01(>)	2.41E+02/9.6E+01(>)	3.68E+02/8.6E+01(>)
f_{a22}	1.40E+02/1.0E+02	1.74E+02/9.3E+01(<)	1.96E+02/1.4E+02(<)	1.21E+02/6.5E+01(>)	3.80E+01/5.9E+01(>)
f_{a23}	8.82E+02/3.3E+02	1.05E+03/3.5E+02(<)	9.66E+02/3.8E+02(<)	1.26E+03/1.5E+02(<)	9.46E+02/2.1E+02(<)
f_{a24}	2.04E+02/5.2E+00	2.05E+02/4.7E+00(<)	2.06E+02/5.2E+00(<)	1.49E+02/9.1E+00(>)	1.38E+02/2.8E+01(>)
f_{a25}	2.04E+02/3.8E+00	2.05E+02/4.3E+00(<)	2.00E+02/1.9E+01(>)	1.84E+02/1.4E+01(>)	1.91E+02/2.3E+01(>)
f_{a26}	1.62E+02/4.4E+01	1.72E+02/5.9E+01(<)	1.75E+02/5.4E+01(<)	1.39E+02/7.7E+00(>)	1.32E+02/2.7E+01(>)
f_{a27}	3.31E+02/2.9E+01	3.55E+02/9.0E+01(<)	3.37E+02/7.8E+01(<)	3.33E+02/3.2E+01(<)	3.45E+02/3.8E+00(<)
f_{a28}	2.89E+02/4.8E+01	2.84E+02/5.4E+01(>)	2.92E+02/5.2E+01(<)	2.43E+02/6.3E+01(>)	2.93E+02/1.6E+02(<)
$w/t/l$		19/0/9	21/0/7	17/0/11	16/0/12

3.4. *Arithmetic Strategy.* The model of $AF_{gs,G}$ establishes the relationship between the global optimal solution and the global suboptimal solution so that the algorithm can effectively search the neighborhood between the global optimal solution and the global suboptimal solution. The characteristics of the function $f(r)$ can help us balance the search and development of the algorithm. The new arithmetic strategy is shown in Eq. (5) and Eq. (6).

$$\begin{cases} B_g = X_{g\text{best},G} + F \bullet (X_{r1,G} - X_{r2,G}) - AF_{gs,G}, \\ B_s = X_{s\text{best},G} + F \bullet (X_{r1,G} - X_{r2,G}) + AF_{gs,G}, \\ B = [B_g; B_s], \end{cases} \quad (5)$$

$$\begin{cases} X = [X_1; X_2], \\ X \leftarrow \bar{M} \otimes X + M \otimes B, \end{cases} \quad (6)$$

where $X_{g\text{best},G}$ and $X_{s\text{best},G}$ are the matrices generated, respectively, by taking the position coordinates of the global optimal solution and the global suboptimal solution as the row vector elements, and the number of columns of the matrix is determined by the number of individuals in the population. The $AF_{gs,G}$ matrix can be obtained by the same principle. As shown in Eq. (6), the matrix M and the crossover strategy remain unchanged. The algorithm assigns individuals to two fixed subpopulations to increase the diversity of the population. Each subpopulation has different operation

TABLE 3: Comparison on 30D optimization under CEC2013 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C- QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{a1}	6.17E-04/1.1E-04	0.00E+00/0.0E+0(>)	2.23E-14/6.8E-1(>)	0.00E+00/0.0E+0(>)	0.00E+00/0.0E+0(>)
f_{a2}	3.16E+05/1.5E+05	5.42E+05/2.8E+0(<)	2.60E+05/1.0E+0(>)	6.89E+07/1.3E+0(<)	2.13E+07/3.9E+0(<)
f_{a3}	1.62E+06/4.2E+06	4.55E+05/1.8E+0(>)	2.41E+06/4.8E+0(<)	3.35E+09/8.6E+0(<)	4.53E+09/1.2E+0(<)
f_{a4}	2.39E+01/1.6E+01	8.33E+01/4.9E+0(<)	1.59E+01/1.0E+0(>)	8.06E+04/1.0E+0(<)	4.69E+04/7.1E+0(<)
f_{a5}	3.08E-04/5.0E-05	9.59E-14/4.2E-1(>)	1.11E-13/1.6E-1(>)	0.00E+00/0.0E+0(>)	0.00E+00/0.0E+0(>)
f_{a6}	5.01E+00/8.0E+00	1.05E+01/1.3E+0(<)	6.51E+00/1.1E+0(<)	1.76E+01/8.9E+0(<)	3.59E+01/2.4E+0(<)
f_{a7}	1.39E+01/1.0E+01	8.53E+00/7.8E+0(>)	2.17E+01/1.5E+0(<)	9.79E+01/8.1E+0(<)	9.64E+01/8.2E+0(<)
f_{a8}	2.09E+01/5.6E-02	2.10E+01/5.9E-0(<)	2.10E+01/5.0E-0(<)	2.09E+01/5.3E-0(=)	2.09E+01/6.1E-0(=)
f_{a9}	1.65E+01/5.3E+00	1.51E+01/5.6E+0(>)	1.69E+01/6.3E+0(<)	3.43E+01/1.2E+0(<)	2.92E+01/1.9E+0(<)
f_{a10}	5.98E-02/1.4E-02	2.69E-02/1.7E-0(<)	2.48E-02/1.6E-0(>)	6.22E-01/5.0E-0(<)	2.25E+00/7.4E-0(<)
f_{a11}	2.25E+01/5.6E+00	1.91E+01/5.1E+0(>)	2.69E+01/7.2E+0(<)	9.60E+01/8.0E+0(<)	0.00E+00/0.0E+0(>)
f_{a12}	6.19E+01/1.9E+01	8.75E+01/2.4E+0(<)	7.87E+01/2.5E+0(<)	2.28E+02/1.2E+0(<)	2.84E+02/1.2E+0(<)
f_{a13}	1.12E+02/2.8E+01	1.13E+02/3.0E+0(<)	1.25E+02/3.0E+0(<)	2.25E+02/1.4E+0(<)	2.83E+02/2.4E+0(<)
f_{a14}	6.88E+02/2.7E+02	6.48E+02/2.8E+0(>)	8.74E+02/3.2E+0(<)	4.14E+03/2.8E+0(<)	9.08E+01/3.2E+0(>)
f_{a15}	3.77E+03/6.1E+02	5.59E+03/8.8E+0(<)	5.23E+03/6.2E+0(<)	7.37E+03/2.9E+0(<)	5.00E+03/4.2E+0(<)
f_{a16}	3.01E-01/1.6E-01	2.43E+00/3.6E-0(<)	2.51E+00/4.2E-0(<)	2.43E+00/2.7E-0(<)	2.20E+00/2.5E-0(<)
f_{a17}	4.71E+01/4.9E+00	4.94E+01/5.5E+0(<)	5.46E+01/6.8E+0(<)	1.43E+02/8.9E+0(<)	3.08E+01/1.8E-0(>)
f_{a18}	8.61E+01/2.5E+01	1.74E+02/2.3E+0(<)	1.69E+02/2.4E+0(<)	2.52E+02/9.2E+0(<)	3.30E+02/2.8E+0(<)
f_{a19}	2.53E+00/5.5E-01	3.23E+00/7.4E-0(<)	3.47E+00/7.4E-0(<)	1.33E+01/1.0E+0(<)	2.75E+00/5.1E-0(<)
f_{a20}	1.14E+01/6.0E-01	1.19E+01/5.6E-0(<)	1.18E+01/5.8E-0(<)	1.38E+01/2.7E-0(<)	1.30E+01/3.0E-0(<)
f_{a21}	2.93E+02/7.5E+01	2.93E+02/8.8E+0(=)	2.84E+02/6.5E+0(>)	2.61E+02/4.9E+0(>)	3.28E+02/5.8E+0(<)
f_{a22}	6.09E+02/2.6E+02	6.47E+02/2.4E+0(<)	8.62E+02/3.0E+0(<)	4.68E+03/2.2E+0(<)	1.23E+02/3.0E+0(>)
f_{a23}	3.92E+03/7.6E+02	5.58E+03/6.3E+0(<)	5.37E+03/8.0E+0(<)	7.86E+03/3.0E+0(<)	5.82E+03/5.1E+0(<)
f_{a24}	2.31E+02/1.1E+01	2.32E+02/1.4E+0(<)	2.37E+02/1.2E+0(<)	2.87E+02/3.4E+0(<)	2.53E+02/5.9E+0(<)
f_{a25}	2.51E+02/7.4E+00	2.53E+02/7.0E+0(<)	2.59E+02/7.9E+0(<)	2.94E+02/4.2E+0(<)	2.63E+02/2.6E+0(<)
f_{a26}	2.47E+02/6.5E+01	2.56E+02/6.5E+0(<)	2.58E+02/6.8E+0(<)	2.08E+02/1.5E+0(>)	2.00E+02/5.4E-0(>)
f_{a27}	6.43E+02/1.1E+02	6.34E+02/1.0E+0(>)	6.51E+02/1.0E+0(<)	1.12E+03/1.5E+0(<)	8.71E+02/1.0E+0(<)
f_{a28}	3.22E+02/1.5E+02	3.83E+02/2.9E+0(<)	3.16E+02/1.4E+0(>)	3.00E+02/3.4E-1(>)	2.80E+02/6.0E+0(>)
$w/t/l$		18/1/9	21/0/7	22/1/5	19/1/8

strategies and can share information in the process of searching. When searching with the global optimization as the center, it will be guided by the global suboptimal force; similarly, when searching with the global suboptimal as the center, it will be guided by the global optimal force. This expands the algorithm's search for excellent regions and avoids falling into the local optimum. The double optimal guidance mechanism not only ensures the convergence of both global optimal position and global suboptimal position, but also brings more exploration to benchmark environments. Accordingly, the proposed algorithm can be an effective approach to enhance the search performance of the algorithm.

3.5. Algorithm Process. The process of QUATRE-DEG algorithm is given as follows.

4. Experiment and Analysis

In order to verify the performance of the algorithm, we carried out experiments with 58 real-parameter single objective optimizations in the CEC2013 and CEC2014 test suites. All benchmark functions are considered as black-box problems, and the search range is set to $[-100, 100]$. In CEC2013, there are 28 test functions, including unimodal functions (fa1-fa5), basic multimodal functions (fa6-f20), and composition functions (fa21-fa28). In CEC2014, there are 30 test functions including four different categories: unimodal function (fb1-fb3), simple multimodal function (fb4-fb16), hybrid function 1 (fb17-fb22), and composition function (fb23-fb30).

We have learned that the QUATRE algorithm focuses on the development, but the search is slightly inadequate. Our

TABLE 4: Comparison on 50D optimization under CEC2013 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C-QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{a1}	1.31E-03/1.4E-04	1.25E-13/1.1E-13(>)	2.27E-13/0.0E+00(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{a2}	1.06E+06/3.1E+05	8.70E+05/2.8E+05(>)	1.01E+06/4.3E+05(>)	2.99E+08/3.3E+07(<)	3.66E+07/5.6E+06(<)
f_{a3}	1.76E+07/1.7E+07	3.27E+07/4.3E+07(<)	2.46E+07/3.1E+07(<)	5.73E+10/6.6E+09(<)	1.48E+10/2.3E+09(<)
f_{a4}	1.45E+02/6.1E+01	8.71E+01/6.1E+01(>)	1.36E+02/8.2E+01(>)	1.54E+05/1.1E+04(<)	8.10E+04/8.9E+03(<)
f_{a5}	6.06E-04/8.1E-05	1.18E-13/2.2E-14(>)	1.78E-13/6.5E-14(>)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{a6}	4.36E+01/8.1E-01	4.39E+01/1.5E+00(<)	4.37E+01/8.0E-01(<)	4.45E+01/9.7E-01(<)	7.77E+01/2.2E+01(<)
f_{a7}	4.42E+01/1.6E+01	4.32E+01/1.7E+01(>)	6.56E+01/2.6E+01(<)	1.79E+02/8.7E+00(<)	1.13E+02/7.7E+00(<)
f_{a8}	2.11E+01/3.3E-02	2.12E+01/3.7E-02(<)	2.12E+01/3.1E-02(<)	2.11E+01/3.5E-02(=)	2.11E+01/5.0E-02(=)
f_{a9}	3.58E+01/1.0E+01	3.72E+01/1.1E+01(<)	3.58E+01/1.1E+01(=)	6.89E+01/1.3E+00(<)	5.66E+01/2.4E+00(<)
f_{a10}	1.08E-01/2.1E-02	6.63E-02/3.2E-02(>)	4.57E-02/2.8E-02(>)	3.30E+02/4.1E+01(<)	3.25E+00/6.0E-01(<)
f_{a11}	6.59E+01/1.4E+01	7.60E+01/1.9E+01(<)	8.56E+01/1.6E+01(<)	2.91E+02/1.6E+01(<)	1.73E-01/6.1E-01(>)
f_{a12}	1.06E+02/2.6E+01	1.84E+02/4.0E+01(<)	1.80E+02/4.7E+01(<)	4.83E+02/1.6E+01(<)	4.52E+02/3.4E+01(<)
f_{a13}	2.24E+02/4.5E+01	2.58E+02/5.4E+01(<)	2.62E+02/4.2E+01(<)	4.84E+02/1.8E+01(<)	5.70E+02/4.0E+01(<)
f_{a14}	1.67E+03/4.6E+02	2.29E+03/4.6E+02(<)	2.32E+03/4.8E+02(<)	9.50E+03/3.2E+02(<)	1.44E+02/5.8E+01(>)
f_{a15}	7.13E+03/8.1E+02	1.14E+04/1.3E+03(<)	1.11E+04/1.0E+03(<)	1.42E+04/3.4E+02(<)	1.06E+04/6.4E+02(<)
f_{a16}	5.13E-01/1.7E-01	3.34E+00/5.1E-01(<)	3.20E+00/5.3E-01(<)	3.38E+00/2.7E-01(<)	2.85E+00/3.2E-01(<)
f_{a17}	1.12E+02/1.4E+01	1.39E+02/1.5E+01(<)	1.40E+02/1.6E+01(<)	3.61E+02/1.5E+01(<)	5.24E+01/7.5E-01(>)
f_{a18}	1.47E+02/2.2E+01	3.58E+02/4.6E+01(<)	3.37E+02/3.9E+01(<)	5.30E+02/1.4E+01(<)	6.98E+02/5.0E+01(<)
f_{a19}	6.17E+00/1.6E+00	8.69E+00/1.9E+00(<)	8.79E+00/1.9E+00(<)	3.63E+01/1.7E+00(<)	6.89E+00/7.3E-01(<)
f_{a20}	2.10E+01/6.3E-01	2.18E+01/5.6E-01(<)	2.16E+01/7.2E-01(<)	2.38E+01/2.3E-01(<)	2.21E+01/4.3E-01(<)
f_{a21}	7.23E+02/4.2E+02	6.85E+02/4.4E+02(>)	7.39E+02/4.3E+02(<)	9.43E+02/3.3E+02(<)	8.98E+02/1.2E+02(<)
f_{a22}	1.95E+03/5.0E+02	2.27E+03/4.8E+02(<)	2.46E+03/5.0E+02(<)	9.99E+03/2.9E+02(<)	7.92E+01/5.7E+01(>)
f_{a23}	7.60E+03/9.7E+02	1.13E+04/1.1E+03(<)	1.14E+04/1.2E+03(<)	1.49E+04/3.9E+02(<)	1.18E+04/7.0E+02(<)
f_{a24}	2.72E+02/1.5E+01	2.73E+02/1.3E+01(<)	2.80E+02/1.1E+01(<)	3.72E+02/4.0E+00(<)	3.33E+02/1.0E+01(<)
f_{a25}	3.06E+02/1.2E+01	3.13E+02/1.5E+01(<)	3.13E+02/1.0E+01(<)	3.92E+02/4.4E+00(<)	4.20E+02/1.2E+01(<)
f_{a26}	3.63E+02/5.1E+01	3.65E+02/3.7E+01(<)	3.58E+02/6.0E+01(>)	2.35E+02/5.4E+00(>)	2.01E+02/1.5E-01(>)
f_{a27}	1.04E+03/1.6E+02	1.08E+03/1.7E+02(<)	1.13E+03/1.5E+02(<)	2.03E+03/3.4E+01(<)	1.70E+03/7.6E+01(<)
f_{a28}	8.13E+02/1.0E+03	9.90E+02/1.2E+03(<)	8.73E+02/1.1E+03(<)	4.23E+02/1.6E+02(>)	4.04E+02/2.8E+01(>)
$w/t/l$		21/0/7	21/1/6	23/1/4	19/1/8

QUATRE-DEG algorithm focuses on improving the search ability of the algorithm to improve the overall performance of the algorithm. In the experiments, to verify the overall performance of the algorithm, it is not only compared with the advanced QUATRE variant but also with the famous variants of the ABC algorithm, which is famous for its strong search ability. The new QUATRE-DEG algorithm is a further improvement of QUATRE algorithm, and the search strategy of ABC algorithm determines its strong exploration performance. In QUATRE-DEG, the performance of the algorithm is mainly improved by enhancing the exploration ability. Therefore, we use some famous ABC variants and advanced QUATRE variants to compare with our proposed algorithm to verify that the performance of QUATRE-DEG is reliable. The comparison includes the original QUATRE [14], C-QUATRE [22], IABC [25], MABC [26], and QUATRE-

DEG algorithms. In all QUATRE-based algorithms, the population size is set to 100, while the population size of all ABC algorithms is set to 50, and the parameter limit is set to $D^*SN/2$, where D is the dimension size and SN is the population size. In addition, in the QUATRE, C-QUATRE, and QUATRE-DEG algorithms, the value of parameter F is set to the recommended value of 0.7. In the IABC algorithm, MR is set to 0.3.

The experiments were complete on Windows 10 Professional system with 8 GB RAM and Intel i5-4210 m processor. The evaluation tool is MATLAB 2019a and the $maxFES$ is set to $10e4 * D$.

4.1. Parameter Setting. There is an important parameter z in the QUATRE-DEG algorithm, which determines the size of the population allocated to the global optimal and global

TABLE 5: Comparison on 10D optimization under CEC2014 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C-QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{b1}	2.84E-01/2.0E-01	1.20E-10/1.8E-10(>)	6.59E-08/7.5E-08(>)	1.34E+05/6.7E+04(<)	3.14E+05/1.3E+05(<)
f_{b2}	5.82E+00/2.9E+01	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	1.08E+02/1.5E+02(<)	4.47E+02/5.1E+02(<)
f_{b3}	1.14E-03/3.8E-04	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)	1.50E+02/2.5E+02(<)	1.44E+02/8.5E+01(<)
f_{b4}	2.36E+01/1.6E+01	2.10E+01/1.7E+01(>)	2.17E+01/1.6E+01(>)	1.16E+00/1.4E+00(>)	1.12E+00/1.7E+00(>)
f_{b5}	1.96E+01/2.8E+00	2.02E+01/5.5E-02(<)	1.90E+01/4.8E+00(>)	1.99E+01/1.2E+00(<)	1.40E+01/8.8E+00(>)
f_{b6}	2.33E-01/3.6E-01	6.54E-01/7.8E-01(<)	4.09E-01/6.2E-01(<)	6.54E-01/4.2E-01(<)	3.61E-01/2.9E-01(<)
f_{b7}	9.98E-02/4.8E-02	8.97E-02/6.8E-02(>)	1.03E-01/6.1E-02(<)	1.36E-02/1.5E-02(>)	1.36E-02/1.3E-02(>)
f_{b8}	2.69E+00/1.5E+00	3.02E+00/1.4E+00(<)	3.06E+00/1.7E+00(<)	0.00E+00/0.0E+00(>)	0.00E+00/0.0E+00(>)
f_{b9}	7.92E+00/4.1E+00	9.30E+00/4.0E+00(<)	9.35E+00/4.4E+00(<)	9.07E+00/2.0E+00(<)	5.25E+00/1.5E+00(>)
f_{b10}	4.23E+01/6.4E+01	7.09E+01/7.5E+01(<)	6.07E+01/6.3E+01(<)	1.88E+00/2.9E+00(>)	1.63E-01/7.2E-02(>)
f_{b11}	3.48E+02/2.1E+02	3.89E+02/1.9E+02(<)	3.32E+02/2.2E+02(>)	5.28E+02/1.1E+02(<)	1.50E+02/8.6E+01(>)
f_{b12}	2.30E-02/2.6E-02	3.16E-01/1.8E-01(<)	2.75E-01/1.9E-01(<)	4.50E-01/1.1E-01(<)	3.56E-01/8.0E-02(<)
f_{b13}	1.17E-01/3.2E-02	1.48E-01/7.8E-02(<)	1.48E-01/6.1E-02(<)	1.83E-01/3.1E-02(<)	1.71E-01/2.6E-02(<)
f_{b14}	1.07E-01/4.2E-02	1.52E-01/5.9E-02(<)	1.58E-01/5.2E-02(<)	1.86E-01/3.1E-02(<)	1.44E-01/2.6E-02(<)
f_{b15}	8.24E-01/2.7E-01	1.30E+00/4.5E-01(<)	1.06E+00/4.2E-01(<)	1.33E+00/2.2E-01(<)	1.00E+00/1.9E-01(<)
f_{b16}	1.79E+00/5.6E-01	1.93E+00/5.5E-01(<)	1.94E+00/5.5E-01(<)	2.72E+00/1.6E-01(<)	1.80E+00/3.2E-01(<)
f_{b17}	8.07E+01/7.4E+01	8.26E+01/8.0E+01(<)	9.87E+01/6.7E+01(<)	9.45E+03/6.1E+03(<)	1.82E+04/1.7E+04(<)
f_{b18}	2.74E+00/2.9E+00	2.39E+00/2.0E+00(<)	3.87E+00/4.4E+00(<)	3.67E+02/2.9E+02(<)	2.45E+03/1.4E+03(<)
f_{b19}	3.94E-01/4.6E-01	8.26E-01/9.7E-01(<)	7.46E-01/7.6E-01(<)	8.59E-01/2.4E-01(<)	3.95E-01/3.1E-01(<)
f_{b20}	1.09E+00/8.2E-01	1.15E+00/7.1E-01(<)	1.92E+00/2.5E+00(<)	1.14E+02/1.0E+02(<)	3.72E+02/3.6E+02(<)
f_{b21}	3.13E+01/4.7E+01	5.02E+01/6.0E+01(<)	5.19E+01/6.2E+01(<)	1.02E+03/9.4E+02(<)	2.82E+03/1.9E+03(<)
f_{b22}	1.15E+01/1.1E+01	8.59E+00/1.1E+01(>)	1.60E+01/3.0E+01(<)	5.96E-01/7.3E-01(>)	5.77E+00/1.8E+01(>)
f_{b23}	3.29E+02/8.1E-07	3.29E+02/2.9E-13(=)	3.29E+02/2.9E-13(=)	3.18E+02/4.8E+01(>)	3.19E+02/6.5E+01(>)
f_{b24}	1.19E+02/4.9E+00	1.18E+02/6.1E+00(>)	1.19E+02/5.8E+00(<)	1.16E+02/3.4E+00(>)	1.09E+02/3.6E+00(>)
f_{b25}	1.73E+02/3.8E+01	1.85E+02/3.3E+01(<)	1.76E+02/3.7E+01(<)	1.52E+02/7.3E+00(>)	1.34E+02/6.3E+00(>)
f_{b26}	1.00E+02/4.1E-02	1.00E+02/7.4E-02(=)	1.00E+02/7.0E-02(=)	1.00E+02/4.8E-02(=)	1.00E+02/2.4E-02(=)
f_{b27}	2.25E+02/1.8E+02	2.28E+02/1.9E+02(<)	2.30E+02/1.7E+02(<)	3.38E+01/7.2E+01(>)	4.69E+01/1.2E+02(>)
f_{b28}	4.06E+02/5.4E+01	4.24E+02/7.6E+01(<)	4.32E+02/6.0E+01(<)	3.62E+02/4.9E+00(>)	3.69E+02/8.9E+01(>)
f_{b29}	4.20E+04/3.0E+05	1.17E+05/4.7E+05(<)	1.22E+05/4.9E+05(<)	3.56E+02/5.4E+01(>)	4.04E+02/7.0E+01(>)
f_{b30}	5.18E+02/7.5E+01	5.56E+02/1.9E+02(<)	5.46E+02/1.3E+02(<)	5.23E+02/3.2E+01(<)	5.46E+02/4.6E+01(<)
$w/t/l$		20/2/8	22/2/6	18/1/11	15/1/14

suboptimal subpopulations. The size of the population will affect the performance of the algorithm, especially when the two populations affect each other. Accordingly, it is necessary to determine the size of the two subpopulations first. Ideologically, it usually does not allow the size of the global optimal subpopulation to be lower than that of the global suboptimal. The population should be dominated by the global optimal subpopulation, which conforms to the basic logic of finding the global optimal solution. In our setting, when z value is larger, the more individuals are allocated to the global optimal subpopulation and more development will be brought to the global optimal position; on the contrary, when the z value is smaller, the less individuals are allocated to the global

optimal subpopulation and the less the development will be. Therefore, in this experiment, the parameter z is set to 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, respectively, and the dimension is set to 30D. The algorithms run 10 times for each different z value. The results are shown in Table 1. The evaluation compares the average and variance of the running results. It is obvious that when z is set to 0.4, the algorithm QUATRE-DEG is the most competitive. It can also be analyzed that when z is larger, the algorithm will pay more attention to exploration, and when z is smaller, the algorithm will have a better development. It also shows that when $z = 0.4$ the exploration and development of the algorithm will tend to a balance; it is undoubtedly a good choice of the z value.

TABLE 6: Comparison on 30D optimization under CEC2014 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C-QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{b1}	1.15E+05/8.1E+04	2.10E+05/1.7E+05(<)	1.37E+05/1.0E+05(<)	6.65E+07/1.1E+07(<)	2.07E+07/3.9E+06(<)
f_{b2}	1.86E+01/5.1E+01	5.57E-16/4.0E-15(>)	1.39E-14/1.4E-14(>)	2.79E+01/4.3E+01(<)	7.21E+03/3.9E+03(<)
f_{b3}	3.84E-03/5.8E-04	1.00E-14/2.2E-14(>)	3.34E-14/2.8E-14(>)	4.69E+02/3.0E+02(<)	7.78E+02/3.2E+02(<)
f_{b4}	3.78E+00/1.5E+01	6.42E+00/1.9E+01(<)	2.53E+00/1.2E+01(>)	7.65E+01/1.5E+01(<)	7.86E+01/1.3E+01(<)
f_{b5}	2.00E+01/4.1E-02	2.05E+01/8.3E-02(<)	2.05E+01/9.4E-02(<)	2.09E+01/4.7E-02(<)	2.05E+01/4.9E-02(<)
f_{b6}	2.60E+00/2.0E+00	2.08E+00/1.8E+00(>)	3.69E+00/2.3E+00(<)	2.65E+01/1.1E+00(<)	9.34E+00/1.9E+00(<)
f_{b7}	5.90E-03/6.3E-03	4.16E-03/7.6E-03(>)	7.77E-03/1.2E-02(<)	0.00E+00/0.0E+00(>)	1.20E-04/2.1E-04(>)
f_{b8}	2.20E+01/5.4E+00	1.89E+01/4.8E+00(>)	2.85E+01/7.1E+00(<)	8.68E+01/8.0E+00(<)	8.31E-02/2.7E-01(>)
f_{b9}	5.53E+01/1.4E+01	7.11E+01/2.3E+01(<)	7.07E+01/2.0E+01(<)	1.84E+02/1.2E+01(<)	6.98E+01/9.2E+00(<)
f_{b10}	3.86E+02/1.8E+02	3.15E+02/1.7E+02(>)	4.99E+02/2.5E+02(<)	3.26E+03/1.8E+02(<)	2.45E+00/1.3E+00(>)
f_{b11}	2.41E+03/6.5E+02	3.30E+03/7.2E+02(<)	3.35E+03/6.9E+02(<)	6.15E+03/2.4E+02(<)	2.88E+03/3.7E+02(<)
f_{b12}	8.04E-02/3.7E-02	5.04E-01/2.2E-01(<)	4.43E-01/2.3E-01(<)	1.75E+00/1.5E-01(<)	6.95E-01/7.6E-02(<)
f_{b13}	2.82E-01/7.2E-02	3.39E-01/7.1E-02(<)	3.10E-01/8.4E-02(<)	4.26E-01/4.1E-02(<)	2.58E-01/3.0E-02(<)
f_{b14}	2.80E-01/1.1E-01	2.70E-01/6.7E-02(>)	3.38E-01/1.3E-01(<)	2.47E-01/2.9E-02(>)	1.36E-01/1.8E-02(>)
f_{b15}	3.97E+00/1.0E+00	8.37E+00/2.2E+00(<)	7.41E+00/2.1E+00(<)	1.75E+01/1.2E+00(<)	1.28E+01/1.4E+00(<)
f_{b16}	1.03E+01/6.3E-01	1.04E+01/7.6E-01(<)	1.03E+01/6.8E-01(=)	1.26E+01/1.9E-01(<)	1.05E+01/3.5E-01(<)
f_{b17}	1.68E+03/8.1E+02	2.66E+03/2.7E+03(<)	2.31E+03/2.6E+03(<)	1.69E+06/4.9E+05(<)	1.41E+06/5.6E+05(<)
f_{b18}	6.05E+01/3.4E+01	3.85E+01/2.1E+01(>)	8.77E+01/4.6E+01(<)	4.06E+02/1.4E+02(<)	1.90E+04/2.2E+04(<)
f_{b19}	4.11E+00/1.5E+00	4.46E+00/1.7E+00(<)	4.75E+00/1.4E+00(<)	9.25E+00/2.8E-01(<)	8.57E+00/1.5E+00(<)
f_{b20}	2.57E+01/1.6E+01	2.59E+01/1.2E+01(<)	3.89E+01/3.7E+01(<)	4.63E+03/1.5E+03(<)	2.70E+03/1.4E+03(<)
f_{b21}	4.60E+02/2.0E+02	5.74E+02/5.5E+02(<)	4.94E+02/2.7E+02(<)	2.22E+05/7.6E+04(<)	2.06E+05/7.0E+04(<)
f_{b22}	2.91E+02/1.9E+02	3.27E+02/1.6E+02(<)	3.14E+02/1.5E+02(<)	1.77E+02/5.1E+01(>)	2.27E+02/6.9E+01(>)
f_{b23}	3.15E+02/4.4E-07	3.15E+02/4.0E-13(=)	3.15E+02/4.0E-13(=)	3.15E+02/4.4E-13(=)	3.15E+02/3.6E-01(=)
f_{b24}	2.24E+02/3.9E+00	2.25E+02/3.2E+00(<)	2.28E+02/5.8E+00(<)	2.24E+02/1.5E+00(=)	2.00E+02/3.1E-03(>)
f_{b25}	2.03E+02/4.7E-01	2.03E+02/6.5E-01(=)	2.03E+02/8.4E-01(=)	2.15E+02/2.0E+00(<)	2.00E+02/0.0E+00(>)
f_{b26}	1.00E+02/2.0E+01	1.00E+02/8.3E-02(=)	1.00E+02/1.0E-01(=)	1.00E+02/5.2E-02(=)	1.00E+02/3.7E-02(=)
f_{b27}	3.86E+02/5.3E+01	3.93E+02/4.5E+01(<)	4.17E+02/7.1E+01(<)	5.00E+02/2.1E+01(<)	4.05E+02/1.7E+00(<)
f_{b28}	8.51E+02/7.6E+01	8.73E+02/1.2E+02(<)	9.20E+02/1.5E+02(<)	9.44E+02/2.0E+01(<)	8.93E+02/4.7E+01(<)
f_{b29}	1.70E+05/1.2E+06	1.77E+05/1.3E+06(<)	1.81E+05/1.3E+06(<)	2.37E+03/6.4E+02(>)	3.56E+03/1.7E+03(>)
f_{b30}	1.53E+03/7.5E+02	1.58E+03/6.4E+02(<)	1.80E+03/7.5E+02(<)	4.52E+03/6.2E+02(<)	5.40E+03/9.4E+02(<)
$w/t/l$		19/3/8	24/3/3	24/2/4	20/2/8

4.2. *Comparison of Optimization Accuracy.* For the data reliability, 51 independent experiments were carried out with the comparison of the five algorithms. The mean and variance of the outcome of each algorithm are also compared and analyzed. Firstly, the dimensions were set to 10D, 30D, and 50D for the CEC2013 test suite, and then, 28 benchmark functions were carried out. The comparative experimental results are shown in Tables 2–4 for the dimensions of 10D, 30D, and 50D, respectively. Table 2 shows that the proposed QUATRE-DEG obtains better performance in 19 functions compared with the C-QUATRE algorithm, better performance in 21 functions compared with the QUATRE algorithm, and better performance in 17 functions compared

with both of the MABC and IABC algorithms. Table 3 shows the QUATRE-DEG obtains better performance in 22 functions as well as similar performance in 1 function compared with the MABC, better performance in 21 functions compared with the QUATRE, better performance in 19 functions as well as similar performance in 1 function compared with the IABC algorithms, and better performance in 18 functions as well as similar performance in 1 function compared with the C-QUATRE. Table 4 shows the QUATRE-DEG obtains better performance in 23 functions as well as similar performance in 1 function compared with the MABC, better performance in 21 functions as well as similar performance in 1 function compared with the QUATRE, better performance

TABLE 7: Comparison on 50D optimization under CEC2014 test suite with different algorithms.

Algorithms f	QUATRE-DEG Mean/std	C-QUATRE Mean/std	QUATRE Mean/std	MABC Mean/std	IABC Mean/std
f_{b1}	8.15E+05/3.3E+05	6.41E+05/2.5E+05(>)	6.64E+05/3.1E+05(>)	2.55E+08/2.9E+07(<)	1.84E+07/3.4E+06(<)
f_{b2}	6.95E+03/6.5E+03	1.37E-03/3.4E-03(>)	1.46E-04/3.4E-04(>)	3.78E+03/3.5E+03(>)	1.93E+04/1.4E+04(<)
f_{b3}	2.73E-01/2.0E-01	1.50E-03/4.7E-03(>)	3.16E-04/9.1E-04(>)	9.86E+04/8.6E+03(<)	2.87E+03/8.4E+02(<)
f_{b4}	4.96E+01/4.1E+01	4.93E+01/3.7E+01(>)	5.05E+01/4.2E+01(<)	9.64E+01/2.1E+00(<)	1.33E+02/3.3E+01(<)
f_{b5}	2.00E+01/9.3E-02	2.07E+01/9.0E-02(<)	2.07E+01/7.6E-02(<)	2.11E+01/3.6E-02(<)	2.06E+01/4.6E-02(<)
f_{b6}	8.63E+00/4.6E+00	9.40E+00/3.6E+00(<)	1.13E+01/4.3E+00(<)	5.74E+01/1.3E+00(<)	1.94E+01/2.4E+00(<)
f_{b7}	6.60E-03/5.5E-03	2.08E-03/4.2E-03(>)	2.27E-03/5.2E-03(>)	0.00E+00/0.0E+00(>)	8.38E-03/1.2E-02(<)
f_{b8}	5.91E+01/1.4E+01	6.87E+01/1.6E+01(<)	7.09E+01/1.4E+01(<)	2.56E+02/1.2E+01(<)	1.44E+00/2.7E+00(>)
f_{b9}	1.12E+02/2.3E+01	1.40E+02/3.5E+01(<)	1.59E+02/3.9E+01(<)	4.21E+02/1.4E+01(<)	1.63E+02/1.5E+01(<)
f_{b10}	1.36E+03/3.5E+02	1.67E+03/5.4E+02(<)	1.69E+03/5.2E+02(<)	8.74E+03/3.3E+02(<)	6.20E+00/1.9E+00(>)
f_{b11}	5.07E+03/8.8E+02	7.34E+03/1.3E+03(<)	7.14E+03/1.0E+03(<)	1.26E+04/3.6E+02(<)	6.60E+03/6.2E+02(<)
f_{b12}	1.62E-01/6.2E-02	4.81E-01/2.5E-01(<)	4.72E-01/2.5E-01(<)	2.85E+00/2.2E-01(<)	8.41E-01/1.2E-01(<)
f_{b13}	3.94E-01/5.7E-02	4.50E-01/9.5E-02(<)	4.64E-01/1.0E-01(<)	5.86E-01/3.6E-02(<)	4.43E-01/3.2E-02(<)
f_{b14}	4.34E-01/2.2E-01	4.40E-01/2.2E-01(<)	4.39E-01/2.1E-01(<)	3.00E-01/2.7E-02(>)	2.18E-01/1.7E-02(>)
f_{b15}	8.42E+00/2.2E+00	1.67E+01/3.7E+00(<)	1.69E+01/4.3E+00(<)	4.08E+01/1.7E+00(<)	3.09E+01/2.4E+00(<)
f_{b16}	1.94E+01/9.8E-01	1.96E+01/8.6E-01(<)	1.99E+01/8.2E-01(<)	2.24E+01/1.8E-01(<)	1.91E+01/4.4E-01(>)
f_{b17}	2.35E+04/1.4E+04	2.65E+04/2.1E+04(<)	2.75E+04/1.4E+04(<)	1.35E+07/1.7E+06(<)	5.19E+06/1.1E+06(<)
f_{b18}	8.88E+02/9.2E+02	2.84E+02/5.8E+02(>)	2.42E+02/1.9E+02(>)	1.37E+03/8.9E+02(<)	4.43E+04/8.5E+04(<)
f_{b19}	8.65E+00/2.0E+00	8.20E+00/2.4E+00(>)	8.88E+00/2.4E+00(<)	3.19E+01/7.5E+00(<)	1.49E+01/6.3E+00(<)
f_{b20}	1.03E+02/6.3E+01	1.14E+02/6.3E+01(<)	1.32E+02/6.8E+01(<)	4.84E+04/8.5E+03(<)	6.29E+03/2.2E+03(<)
f_{b21}	5.63E+03/4.5E+03	1.55E+04/2.3E+04(<)	9.09E+03/7.4E+03(<)	4.20E+06/1.1E+06(<)	1.56E+06/4.6E+05(<)
f_{b22}	8.76E+02/2.4E+02	1.11E+03/2.9E+02(<)	1.07E+03/2.9E+02(<)	1.00E+03/1.3E+02(<)	7.93E+02/1.4E+02(>)
f_{b23}	3.44E+02/2.2E-07	3.44E+02/5.3E-13(=)	3.44E+02/4.4E-13(=)	3.44E+02/4.6E-13(=)	2.68E+02/7.3E+01(>)
f_{b24}	2.74E+02/2.9E+00	2.76E+02/3.2E+00(<)	2.76E+02/3.5E+00(<)	2.58E+02/1.1E+00(>)	2.00E+02/2.0E-03(>)
f_{b25}	2.07E+02/1.4E+00	2.09E+02/2.1E+00(<)	2.07E+02/1.7E+00(=)	2.53E+02/5.0E+00(<)	2.00E+02/0.0E+00(>)
f_{b26}	1.25E+02/5.4E+01	1.44E+02/7.9E+01(<)	1.28E+02/6.5E+01(<)	1.01E+02/4.6E-02(>)	1.49E+02/5.0E+01(<)
f_{b27}	5.69E+02/8.9E+01	6.08E+02/9.0E+01(<)	6.58E+02/9.0E+01(<)	1.57E+03/7.7E+01(<)	7.53E+02/9.6E+01(<)
f_{b28}	1.33E+03/2.8E+02	1.42E+03/3.7E+02(<)	1.44E+03/2.7E+02(<)	1.49E+03/3.7E+01(<)	1.46E+03/1.1E+02(<)
f_{b29}	7.21E+05/5.1E+06	1.77E+06/8.8E+06(<)	1.97E+06/9.9E+06(<)	3.22E+04/1.5E+04(>)	3.44E+03/9.0E+02(>)
f_{b30}	9.54E+03/9.8E+02	9.92E+03/9.4E+02(<)	9.94E+03/8.8E+02(<)	1.63E+04/1.7E+03(<)	1.26E+04/1.3E+03(<)
$w/t/l$		22/1/7	23/2/5	23/1/6	21/0/9

in 21 functions compared with the C-QUATRE, and better performance in 19 functions as well as similar performance in 1 function compared with the IABC. Based on the above results obtained from the CEC2013 test suite, it is obvious the proposed QUATRE-DEG performs better than the other algorithms.

The CEC2014 test suite also was used in the experiment for the performance evaluation of the proposed QUATRE-DEG. The results of 30 benchmark functions of CEC2014 are shown in Tables 5–7 for the dimensions of 10D, 30D, and 50D, respectively. Table 5 shows that the proposed QUATRE-DEG obtains better performance in 21 functions and similar performance in 3 function compared with the

QUATRE algorithm, better performance in 20 functions and similar performance in 2 function compared with the C-QUATRE algorithm, better performance in 18 functions and similar performance in 1 function compared with the MABC algorithm, and better performance in 15 functions and similar performance in 1 function compared with the IABC algorithm. Table 6 shows the QUATRE-DEG obtains better performance in 23 functions and similar performance in 4 functions compared with the QUATRE, better performance in 23 functions and similar performance in 3 functions compared with the MABC, better performance in 19 functions and similar performance in 2 functions compared with the IABC, and better performance in 19 functions and

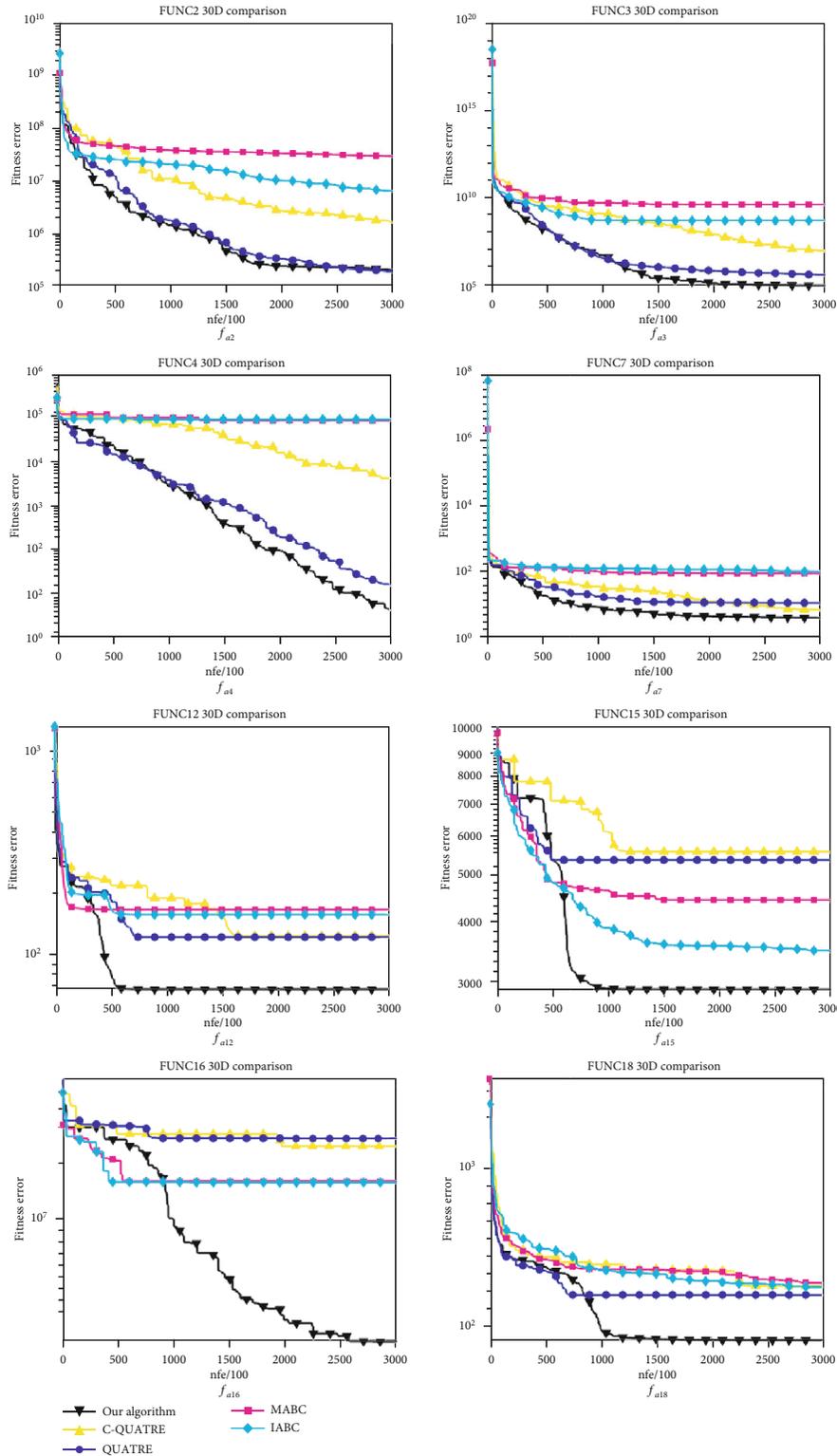


FIGURE 1: Comparison of convergence curves on 30D optimization under CEC2013 with different algorithms.

similar performance in 3 functions compared with the C-QUATRE. Table 7 shows the QUATRE-DEG obtains better performance in 23 functions and similar performance in 2 functions compared with the QUATRE, better performance

in 23 functions and similar performance in 1 function compared with the MABC, better performance in 22 functions and similar performance in 1 function compared with the C-QUATRE, and better performance in 21 functions

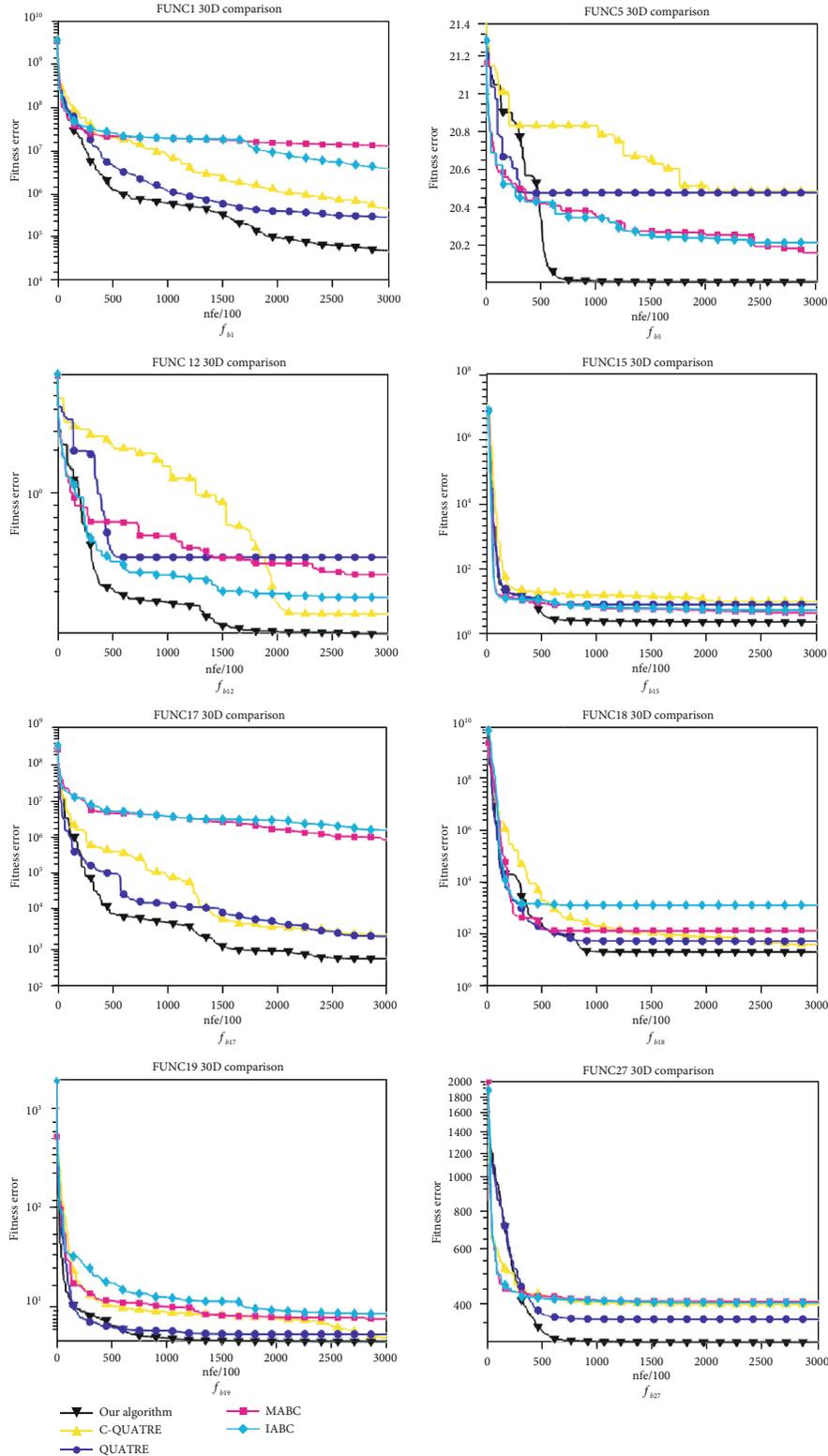


FIGURE 2: Comparison of convergence curves on 30D optimization under CEC2014 with different algorithms.

compared with the IABC. Based on the above results obtained from the CEC2014 test suite, it is also obvious the proposed QUATRE-DEG performs better than the other algorithms.

4.3. *Analysis of Convergence Rate.* The QUATRE-DEG algorithm was also evaluated from the perspective of convergence speed. Figures 1 and 2 show the convergence curves obtained by the representative benchmark functions in the CEC2013

TABLE 8: Wilcoxon's test among C-QUATRE, QUATRE, MABC, IABC, and QUATRE-DEG algorithms over all dimensions under CEC2013 test suite.

D	Algorithms	R^+	R^-	p value	+	=	-	Dec.
10	QUATRE-DEG vs. C-QUATRE	298	108	0.02941	19	0	9	+
	QUATRE-DEG vs. QUATRE	321	85	0.00606	21	0	7	+
	QUATRE-DEG vs. MABC	247	159	0.32721	17	0	11	=
	QUATRE-DEG vs. IABC	226	180	0.61364	16	0	12	=
30	QUATRE-DEG vs. C-QUATRE	273	105	0.04291	18	1	9	+
	QUATRE-DEG vs. QUATRE	327.5	78.5	0.0035	21	0	7	+
	QUATRE-DEG vs. MABC	345	33	4.74602E-5	22	1	5	+
	QUATRE-DEG vs. IABC	283	95	0.02294	19	1	8	+
50	QUATRE-DEG vs. C-QUATRE	329.5	76.5	0.00296	21	0	7	+
	QUATRE-DEG vs. QUATRE	325	53	5.9849E-4	21	1	6	+
	QUATRE-DEG vs. MABC	346	32	4.09335E-5	23	1	4	+
	QUATRE-DEG vs. IABC	280	98	0.02802	19	1	8	+

TABLE 9: Wilcoxon's test among C-QUATRE, QUATRE, MABC, IABC, and QUATRE-DEG algorithms over all dimensions under CEC2014 test suite.

D	Algorithms	R^+	R^-	p value	+	=	-	Dec.
10	QUATRE-DEG vs. C-QUATRE	313	93	0.01102	20	2	8	+
	QUATRE-DEG vs. QUATRE	299	79	0.007	21	3	6	+
	QUATRE-DEG vs. MABC	258	177	0.39265	18	1	11	=
	QUATRE-DEG vs. IABC	221	214	0.94907	15	1	14	=
30	QUATRE-DEG vs. C-QUATRE	291.5	86.5	0.01224	19	3	8	+
	QUATRE-DEG vs. QUATRE	325	26	3.18587E-5	23	4	3	+
	QUATRE-DEG vs. MABC	335.5	42.5	1.64852E-4	23	3	4	+
	QUATRE-DEG vs. IABC	307	99	0.01673	19	2	9	+
50	QUATRE-DEG vs. C-QUATRE	338	97	0.00801	22	1	7	+
	QUATRE-DEG vs. QUATRE	325	81	0.00441	23	2	5	+
	QUATRE-DEG vs. MABC	370	65	5.50576E-4	23	1	6	+
	QUATRE-DEG vs. IABC	345	120	0.01966	21	0	9	+

and CEC2014 test suites, respectively. Figure 1 shows the proposed QUATRE-DEG performs much better than the other algorithms in functions fa12, fa15, fa16, and fa18 and performs similar to the other algorithms in functions fa2, fa3, fa4, and fa7. Figure 2 shows the convergence speed of the proposed QUATRE-DEG is significantly better than the other algorithms in functions fb1, fb5, fb12, and fb17 and is slightly better than the other algorithms in fb15, fb18, fb19, and fb27 functions. The ABC variant is easy to converge prematurely and fall into local optimal solution, such as the performance of fa3, fa4, fa7, fa12, fb17, and fb27 benchmark functions. However, the performance of QUATRE variants in the functions of fa2, fa3, fa4, fa7, fb18, and fb19 is only slightly insufficient. In summary, the proposed QUATRE-DEG can perform with a very competitive convergence speed compared with the QUATRE variants and ABC variants.

4.4. The Wilcoxon Signed Rank Test. Wilcoxon signed rank test is needed when two results present two different populations [27]. Here, we also give the Wilcoxon signed rank test

results. Tables 8 and 9 give the Wilcoxon signed rank test results of CEC2013 test suites and CEC2014 test suites, respectively. The results are calculated by using Origin 2019b statistical analysis packages, and our proposed QUATRE-DEG algorithm is verified to be well performed.

The statistical analysis results of applying Wilcoxon's test between QUATRE-DEG and other comparison algorithms using CEC2013 for all dimensions are summarized in Table 8. As the table states, QUATRE-DEG shows a significant improvement over all comparison algorithms, with a level of significance $\alpha = 0.05$ in 30 and 50 dimensions, over QUATRE and C-QUATRE, with a level of significance $\alpha = 0.05$ in 10 dimensions. On the other hand, in all comparisons, QUATRE-DEG gets higher R^+ values than R^- in all dimensions. Therefore, the performance of QUATRE-DEG algorithm is proved to be better than the other algorithms in this benchmark.

The statistical analysis results of applying Wilcoxon's test between QUATRE-DEG and other comparison algorithms using CEC2014 for all dimensions are summarized in

TABLE 10: Time complexity comparison on 30D optimization under CEC2013 benchmark 14.

Algorithms	T_0	T_1	\widehat{T}_2	$\frac{\widehat{T}_2 - T_1}{T_0}$
QUATRE-DEG	0.0906	0.7336	2.7126	21.8433
C-QUATRE	0.0906	0.7336	2.2314	16.5320
QUATRE	0.0906	0.7336	2.0011	13.9901
MABC	0.0906	0.7336	5.7920	55.8322
IABC	0.0906	0.7336	4.7662	44.5099

Table 9. As the table states, QUATRE-DEG shows a significant improvement over all comparison algorithms, with a level of significance $\alpha = 0.05$ in 30 and 50 dimensions, over QUATRE and C-QUATRE, with a level of significance $\alpha = 0.05$ in 10 dimensions. On the other hand, in all comparisons, QUATRE-DEG gets higher R^+ values than R^- in all dimensions. Therefore, the performance of QUATRE-DEG algorithm is proved to be better than the other algorithms in this benchmark.

4.5. Comparison of Complexity. The evaluation method of algorithm complexity adopted by the CEC2013 competition was used to evaluate the proposed QUATRE-DEG algorithm. Let T_0 be the standard computation time for the reference benchmark given by CEC2013 and T_1 be the standard computation time of the specified function fa14 for 200000 evaluations with a dimension of 30D. Let T_2 be the computation time of function fa14 using a specified algorithm for 200000 evaluations with the same dimension of 30D, and \widehat{T}_2 is the average of T_2 values acquired by 51 runs. Finally, the algorithm complexity is determined by $(\widehat{T}_2 - T_1)/T_0$. Table 10 shows the result of comparing the proposed QUATRE-DEG with the algorithms. It indicates that the QUATRE-DEG can obtain lower complexity compared with either the MABC or IABC. However, the QUATRE-DEG takes more computation time compared with the QUATRE and C-QUATRE. It is reasonable due to the additional process in the proposed algorithm.

5. Conclusion

QUATRE is a relatively new algorithm proposed in recent years. Although it is powerful, it still can be improved in some concerns. For example, in the later stage of evolution, it is easier to fall into local optimal solutions and could not explore more. This study proposes a new QUATRE-DEG algorithm to solve this problem. In the proposed algorithm, excellent individuals in the current iteration process are fully utilized for the optimization. Not only the global optimal solution but also the global suboptimal solution is used to guide the evolution of individuals. According to the distance between the global optimal individuals and the global suboptimal individuals, a new mutation strategy is proposed to expand the current search environment. In addition, the population is divided into two subpopulations, including the subpopulation with the global optimal location as the search center and the subpopulation with the global suboptimal location as the search center. The expansion of search

coverage and the increase of population diversity provide better conditions for more exploration, so as to avoid falling into the local optimal solution. Although QUATRE-DEG has stronger search performance and can explore more benchmark environments, it also reduces its search speed and sometimes makes it not easy to completely converge to the global optimal solution, which indicates the possible limitation of the approach. In the experiments, we use CEC2013 and CEC2014 to verify the performance of the proposed QUATRE-DEG algorithm. The results show that the optimization accuracy and convergence speed are excellent, and the time complexity is tolerable. In future works, the proposed algorithm will be applied to several practical engineering problems such as UAV path planning [28, 29]. Moreover, CEC2017 benchmark suite will be used to evaluate the further works.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Fujian University of Technology under Grant GY-Z20016 and GY-Z18183, National Natural Science Foundation of China under Grant 61773415, and Fujian Provincial Natural Science Foundation under Grant 2017J01730.

References

- [1] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.
- [2] N. Liu, J. S. Pan, C. Sun, and S. C. Chu, "An efficient surrogate-assisted quasi-affine transformation evolutionary algorithm for expensive optimization problems," *Knowledge Based Systems*, vol. 209, article 106418, 2020.
- [3] S. Zhang and S. Liu, "A discrete improved artificial bee colony algorithm for 0–1 knapsack problem," *IEEE Access*, vol. 7, pp. 104982–104991, 2019.
- [4] X. Pan, Q. Zhang, and H. Pan, "Improved artificial bee colony algorithm and its application to fundus retinal blood vessel image binarization," *IEEE Access*, vol. 8, pp. 123726–123734, 2020.
- [5] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.
- [6] N. Liu, J.-S. Pan, J. Wang, and T.-T. Nguyen, "An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks," *Sensors*, vol. 19, no. 19, article 4112, 2019.

- [7] J. S. Pan, N. Liu, S. C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [9] R. Storn and K. Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," Technical Report TR-95-012, International Computer Science Institute, Vacaville, CA, USA, 1995.
- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948, Perth, WA, Australia, 2002.
- [11] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [12] M. Seyedali, M. Mohammad, and H. Abdolreza, "Multi-verse optimizer," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.
- [13] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [14] 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.
- [15] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, 2013.
- [16] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, 2013.
- [17] Z. Meng, Y. Chen, X. Li, C. Yang, and Y. Zhong, "Enhancing QUasi-Affine TRansformation Evolution (QUATRE) with adaptation scheme on numerical optimization," *Knowledge Based Systems*, vol. 197, article 105908, 2020.
- [18] S.-C. Chu, Y. Chen, F. Meng, C. Yang, J.-S. Pan, and Z. Meng, "Internal search of the evolution matrix in QUasi-Affine TRansformation Evolution (QUATRE) algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5673–5684, 2020.
- [19] B.-Q. Jiang and J.-S. Pan, "A Parallel Quasi-Affine Transformation Evolution algorithm for global optimization," *J. Netw. Intell.*, vol. 4, pp. 30–46, 2019.
- [20] 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.
- [21] B. Zhao, T. W. Sung, and X. Zhang, "A quasi-affine transformation artificial bee colony algorithm for global optimization," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 3, pp. 5527–5544, 2021.
- [22] Z. Meng and J.-S. Pan, "A Competitive QUasi-Affine TRansformation Evolutionary (C-QUATRE) Algorithm for global optimization," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1644–1649, Budapest, Hungary, 2016.
- [23] J.-S. Pan, Z. Meng, S.-C. Chu, and J. F. Roddick, "QUATRE Algorithm with Sort Strategy for Global Optimization in Comparison with DE and PSO Variants," in *The Euro-China Conference on Intelligent Data Analysis and Applications*, pp. 314–323, Springer, 2017.
- [24] N. Liu, J.-S. Pan, and T.-T. Nguyen, "A bi-population QUasi-Affine TRansformation Evolution algorithm for global optimization and its application to dynamic deployment in wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.
- [25] 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, no. S2, pp. 3011–3019, 2019.
- [26] W. Gao and S. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.
- [27] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [28] T.-W. Sung, L. Sun, and K.-C. Chang, "Multi-hop Route Planning Based on Environment Information for Path-Following UAVs," in *The International Conference on Artificial Intelligence and Computer Vision*, pp. 831–839, Cairo, Egypt, 2020.
- [29] T.-W. Sung, B. Zhao, and K.-C. Chang, "Experimental Comparison of Different Feature Detection Algorithms for UAV Obstacle Avoidance," in *The International Conference on Artificial Intelligence and Computer Vision*, pp. 840–849, Cairo, Egypt, 2020.