

Research Article

An Improved Multiobjective Quantum-Behaved Particle Swarm Optimization Based on Double Search Strategy and Circular Transposon Mechanism

Fei Han, Yu-Wen-Tian Sun , and Qing-Hua Ling

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, China

Correspondence should be addressed to Yu-Wen-Tian Sun; sunyuwentian@163.com

Received 7 July 2018; Revised 23 September 2018; Accepted 26 September 2018; Published 1 November 2018

Guest Editor: Yimin Zhou

Copyright © 2018 Fei Han 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.

Although multiobjective particle swarm optimization (MOPSO) has good performance in solving multiobjective optimization problems, how to obtain more accurate solutions as well as improve the distribution of the solutions set is still a challenge. In this paper, to improve the convergence performance of MOPSO, an improved multiobjective quantum-behaved particle swarm optimization based on double search strategy and circular transposon mechanism (MOQPSO-DSCT) is proposed. On one hand, to solve the problem of the dramatic diversity reduction of the solutions set in later iterations due to the single search pattern used in quantum-behaved particle swarm optimization (QPSO), the double search strategy is proposed in MOQPSO-DSCT. The particles mainly learn from their personal best position in earlier iterations and then the particles mainly learn from the global best position in later iterations to balance the exploration and exploitation ability of the swarm. Moreover, to alleviate the problem of the swarm converging to local minima during the local search, an improved attractor construction mechanism based on opposition-based learning is introduced to further search a better position locally as a new attractor for each particle. On the other hand, to improve the accuracy of the solutions set, the circular transposon mechanism is introduced into the external archive to improve the communication ability of the particles, which could guide the population toward the true Pareto front (*PF*). The proposed algorithm could generate a set of more accurate and well-distributed solutions compared to the traditional MOPSO. Finally, the experiments on a set of benchmark test functions have verified that the proposed algorithm has better convergence performance than some state-of-the-art multiobjective optimization algorithms.

1. Introduction

Optimization problems widely exist in fields including economics [1], natural science [2], and industrial production [3]. Compared to single-objective optimization problems (SOPs), the theories and methods of multiobjective optimization problems (MOPs) are more complicated and usually incomplete. Moreover, it is difficult to find the feasible Pareto front (*PF*) for most optimization algorithms due to the equality and inequality constraints as well as multivariable constraints in a MOP [1]. Since the objectives in MOPs are conflict in many cases, there exists a set of solutions which are superior to the rest of solutions in the search space when all objectives are considered but are inferior to other solutions in the space in one or more objectives [4]. These solutions

are termed nondominated solutions. How to obtain a set of well-distributed nondominated solutions is a crucial task of multiobjective optimization.

As an efficient evolutionary algorithm, particle swarm optimization (PSO) algorithm has been widely used for various optimization problems [5]. Due to its fast convergence speed and good convergence accuracy, multiobjective optimization based on PSO has attracted much more attention. In [6], the multiobjective particle swarm optimization (MOPSO) algorithm was first proposed to handle various kinds of MOPs, in which Pareto dominance was used to generate nondominated solutions. In MOPSO, a secondary archive was first used to store the nondominated solutions found by the algorithm to guide the particles toward the true *PF*. Although the convergence performance of the

algorithm is improved greatly by this mechanism, the distribution of the PF generated by the algorithm could not be guaranteed. In [7], the mechanism of crowding distance computation was introduced into MOPSO to maintain the external archive, which could make the algorithm obtain a set of well-distributed nondominated solutions. Based on these theories, the external archive was widely used in different kinds of MOPSO and many mechanisms were applied to maintain the external archive. In [8], a new archive called nondominated local set was proposed to store the personal best solutions obtained by one particle during the search process, which showed a superiority in terms of capturing the shape of the true PF and obtaining nondominated solutions with satisfactory diversity characteristics. In [9], a new multiobjective particle swarm optimizer based on ε -Dominance was proposed to utilize ε -Dominance to generate a new archive and the crowding distance to maintain the archive. In [10], a novel hybrid multiobjective particle swarm optimization was proposed, which used a teaching-learning-based optimization algorithm to promote the diversity of the algorithm and circular crowded sorting to select g_{best} and teacher solution from the archive. In [11], a niching multiobjective particle swarm optimization was proposed to solve the MOPs of contractors in circuits, in which the entropy weight ideal point theory was proposed to ensure the diversity of the nondominated solutions. Different from the algorithms mentioned above, a multiobjective particle swarm optimization algorithm based on a decomposition approach (dMOPSO) was proposed in [12], which decomposed a continuous and constrained MOP into many SOPs. Although this algorithm does not need an external archive to store the nondominated solutions, the solutions generated by the algorithm cannot cover the entire true PF for some complicated MOPs. It can be seen that an external archive is crucial for most algorithms to select the leading particle.

The traditional search strategy of the particles in PSO has the deficiency of the particles not being able to traverse the entire search space, which may lead to an incomplete search in the search space and miss some solutions. To overcome this deficiency, a new PSO based on probability called quantum-behaved particle swarm optimization (QPSO) was proposed in [13], in which particles could appear in any position of the feasible region. Since the faster convergence speed of QPSO than classical PSO, multiobjective optimization algorithms based on QPSO have been studied by many researchers. In [14], QPSO and adaptive grid were introduced to improve the performance of MOPSO (MOQPSO-AG), which showed a superior convergence performance when compared with other algorithms, especially in terms of convergence speed. In [15], a novel quantum Delta potential well with two local attractors named double-potential well was established for QPSO and applied to deal with MOPs, which showed a better convergence and distribution performance when handling high-dimensional MOPs. In [16], QPSO was used to handle constrained multiobjective optimization problems, in which two strategies were investigated to combine constraint processing mechanism with a multiobjective QPSO. It can be seen that QPSO has a great potential to deal with MOPs.

Despite the fact that QPSO shows superior performance over PSO in terms of convergence speed when handling MOPs, the leadership of the average best position may be reduced dramatically due to the single search pattern in QPSO, which is not beneficial to obtain more potential solutions for MOPs. Moreover, the mechanism of constructing an attractor for each particle may reduce the ability of the particles to jump out of local optima when its personal best position is equal to the global best position. Finally, most MOPSO algorithms only use the external archive to select the global and local best individuals without fully utilizing the potential significant information hidden in the archive, which may reduce the convergence accuracy of the algorithm.

To alleviate the situation mentioned above, an improved multiobjective quantum-behaved particle swarm optimization based on double search strategy and circular transposon mechanism (MOQPSO-DSCT) is proposed in this paper. On one hand, the double search strategy is proposed to adjust the relationship between the exploration and the exploitation ability, which is beneficial for the diversity of the solutions set as well as the convergence speed. To implement the double search strategy, a search probability parameter is proposed to make the two search patterns execute alternatively to maintain the diversity of the swarm. Moreover, an improved attractor construction mechanism based on opposition-based learning is used to construct a better attractor for one particle when its personal best position is equal to the global best position, which improves the ability of the particles to escape from local optimum. On the other hand, the circular transposon mechanism is introduced into the external archive to help the particles find more useful information between each other, which can make the population move toward the true PF with a higher probability and maintain the distribution of the PF . Therefore, the MOQPSO-DSCT algorithm has improved the performance in convergence accuracy and kept the reasonable distribution of the solutions.

The reminder of the paper is organized as follows. In Section 2, the concepts of multiobjective optimization, the basic model of QPSO, and some related strategies are briefly described. The details of the proposed algorithm (MOQPSO-DSCT) are depicted in Section 3. The experiment results on several benchmark test functions are given in Section 4. Finally, the conclusions and our future work are included in Section 5.

2. Preliminaries

2.1. Multiobjective Optimization. In general, the model of a minimization MOP can be defined as follows:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & h_j(x) = 0, \quad j = 1, 2, \dots, p \end{aligned} \quad (1)$$

where x is a vector with D decision variables, k is the number of objectives, $f_k(x)$ is the function value of the k -th objective,

and m and p are the number of inequality and equality constraints, respectively.

To better understand how a given MOP $F(x)$ produces a large number of nondominated solutions in the feasible region Ω , some definitions related to Pareto Optimality are described as follows [17].

Definition 1 (Pareto dominance). A vector $u = (u_1, u_2, \dots, u_k)$ is said to dominate another vector $v = (v_1, v_2, \dots, v_k)$, denoted by $u < v$, if and only if

$$\begin{aligned} \forall i \in \{1, 2, \dots, k\}, \quad u_i \leq v_i \wedge \exists i \in \{1, 2, \dots, k\}, \\ u_i < v_i \end{aligned} \quad (2)$$

Definition 2 (Pareto optimal). A solution $x^* \in \Omega$ is said to be a Pareto optimal or a nondominated solution, if and only if

$$\begin{aligned} \neg \exists x \in \Omega, \\ x < x^* \end{aligned} \quad (3)$$

Definition 3 (Pareto optimal set). A set P containing all the Pareto optimal is a Pareto optimal set, which can be defined as

$$P = \{x^* \in \Omega \mid \neg \exists x' \in \Omega : F(x') < F(x^*)\} \quad (4)$$

Definition 4 (Pareto front). The region PF generated by the objective function values of all Pareto optimal solutions is Pareto front, which can be defined as

$$PF = \{u = F(x) \mid x \in P\} \quad (5)$$

2.2. Quantum-Behaved Particle Swarm Optimization. In 2004, a quantum-behaved particle swarm optimization algorithm was proposed to overcome the deficiency of PSO not being able to traverse the entire search space. Particles in QPSO only have positions without velocities, which is extremely different from the particles in PSO. In quantum world, the positions of particles can not be determined because of the uncertainty principle so that particles can appear anywhere in the feasible search region with a certain probability, which guarantees the ability of QPSO to converge to the global optimal.

Each particle in PSO is considered as an individual with two attributes in D -dimensional search space. The position and velocity of each particle can be represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ $i = (1, 2, \dots, M)$, respectively, where M is the size of population. Then the position and velocity of each particle can be updated by the following equations [18]:

$$\begin{aligned} v_{ij}(t+1) = \omega * v_{ij}(t) + c_1 r_1 [p_{ij} - x_{ij}(t)] \\ + c_2 r_2 [G_j(t) - x_{ij}(t)] \end{aligned} \quad (6)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (7)$$

where $j = (1, 2, \dots, D)$; $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ is the personal best position of the i -th particle; $G(t) = (g_1, g_2, \dots, g_D)$ is the

global best position in the current iteration; ω is the inertial weight of PSO; t denotes the iteration number; c_1 and c_2 are two acceleration factors which can balance the impact of p_i and $G(t)$; r_1 and r_2 are two numbers randomly generated in $[0, 1]$.

Different from the traditional PSO, the main idea of QPSO is to establish a potential field to bind the particles. In [19], the convergence behavior of particles was analyzed by Clerc M and Kennedy J, which demonstrated that the i -th particle in PSO is attracted by an attractor $attractor_i = (a_{i1}, a_{i2}, \dots, a_{iD})$. Obviously, the attractor directly influences the convergence behavior of the particles in QPSO. The position of $attractor_i$ is described as

$$attractor_{ij}(t) = \varphi_{ij}(t) * p_{ij}(t) + [1 - \varphi_{ij}(t)] * G_j(t) \quad (8)$$

where $p_i(t)$ is the personal best position of the i -th particle; $G(t)$ is the global best position in the current iteration; $\varphi_{ij}(t) = c_1 r_{1j}(t) / [c_1 r_{1j}(t) + c_2 r_{2j}(t)]$. In QPSO, $\varphi_{ij}(t)$ is a uniformly distributed random number between $[0, 1]$. Then, the position of the i -th particle in QPSO can be updated as shown in

$$\begin{aligned} x_{ij}(t+1) = attractor_{ij}(t) - \frac{L_{ij}(t)}{2} \ln\left(\frac{1}{u_{ij}(t)}\right) \\ \text{if } u_{ij}(t) > 0.5 \end{aligned} \quad (9)$$

$$\begin{aligned} x_{ij}(t+1) = attractor_{ij}(t) + \frac{L_{ij}(t)}{2} \ln\left(\frac{1}{u_{ij}(t)}\right) \\ \text{otherwise} \end{aligned}$$

$$L_{ij}(t) = 2\alpha |c_j(t) - x_{ij}(t)| \quad (10)$$

$$c_j(t) = \frac{1}{M} \sum_{i=1}^M p_{ij}(t) \quad (11)$$

where $c(t)$ is the average best position in population; $L(t)$ is the length of the Delta potential well, which denotes the creativity of the individual in the population, $L(t)$ with a larger value usually means the particle has a relatively large search range and is more likely to find more solutions; $u_{ij}(t)$ is a uniformly distributed random number between $[0, 1]$; α is a contraction-expansion coefficient usually less than 1.782 [20]. In our experiments, α is set to be positive and linearly decreasing from 1.0 to 0.5 as proposed in [21].

$$\alpha = \alpha_0 + \frac{(\alpha_1 - \alpha_0)(t_{\max} - t)}{t_{\max}} \quad (12)$$

where α_0 and α_1 are set to 1.0 and 0.5, respectively; t is the current iteration; and t_{\max} is the maximum iteration number.

2.3. Opposition-Based Learning Strategy. In [22], the concepts of opposition-based learning were proposed, which demonstrated that the opposite solution has a higher probability to find a better solution than the current solution. On the other words, we can get the opposite position of the particle

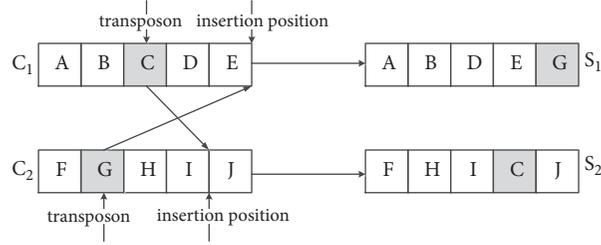


FIGURE 1: Cut-and-paste transposon operation on two particles.

according to opposition-based learning. If the opposite position is better than the original position, the original position will be replaced with the opposite position. Currently, this mechanism is often combined with intelligent algorithms to deal with optimization problems. In [23], opposition-based learning was introduced into a new intelligent algorithm named chicken swarm optimization to improve the diversity of the population. In [24], opposition-based learning was used to generate the transformed population in the population initialization, which enhanced the performance of constrained differential evolution algorithms. In [25], opposition-based learning was executed by the current best individual to generate an opposition search populations in its dynamic search boundaries, which improved the balance and exploring ability of the algorithm. In [26], the opposite process of the opposition-based learning strategy was combined with the refraction principle of light to improve the global search ability of PSO. Therefore, in this paper, opposition-based learning will be used to construct a new attractor for the particle when its personal best position is equal to the global best position to help the particle to escape from local optimum in this paper. Some definitions of opposition-based learning are as follows.

Definition 5 (opposite number). Let x be a real number defined on a certain interval $x \in [a, b]$. The opposite number \tilde{x} is defined as follows:

$$\tilde{x} = a + b - x \quad (13)$$

Definition 6 (opposite point). Let $P(x_1, x_2, \dots, x_D)$ be a point in a D -dimensional coordinate system and $x_j \in [a_j, b_j]$. The opposite point \tilde{P} is completely defined by its coordinates $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_D$ where

$$\tilde{x}_j = a_j + b_j - x_j \quad j = 1, 2, \dots, D \quad (14)$$

2.4. Transposon Mechanism. Transposon, also known as jumping genes, was a special chromosomal genetic phenomenon firstly discovered by Barbara McClintock in 1950 [27, 28]. In her work on maize chromosomes, she found that genes can jump from one position to another on the same chromosome when existing dissociation elements and genes can jump between different chromosomes when existing activator elements. She also found that there were two methods to make a gene move on the chromosome. One method is cut-and-paste transposon, in which one part of

DNA is cut then pasted to another part. The other is called copy-and-paste transposon, in which the information on one position in DNA is copied and pasted to RNA, then the information in one position is copied from RNA to another position in DNA. In [29], the transposon mechanism was first adopted to replace the operations of crossover, mutation, and selection in traditional evolutionary algorithms, which enhanced the diversity of the population. In PSO, a particle in the search space can be considered as a chromosome, then this operation can also be executed on a particle. In this paper, the copy-and-paste transposon operation will be performed on the individuals in the external archive, which can greatly improve the information exchange ability of the particles in the external archive and guide the population toward the true PF . This operation is easily implemented only with a probability parameter tp . For example, a transposon and an insertion position are generated for two particles C_1 and C_2 , respectively. Then, the transposon of C_1 is cut and pasted to the insertion position of C_2 ; the transposon of C_2 is cut and pasted to the insertion position of C_1 . Two new particles S_1 and S_2 can be generated after this operation. The copy-and-paste transposon operation on two particles is shown in Figure 1.

3. The Proposed MOQPSO-DSCT Algorithm

The proposed MOQPSO-DSCT algorithm is aimed at dealing with the two problems about how to find more potential solutions for different MOPs and how to improve the convergence accuracy and the distribution of the nondominated solutions. In the MOQPSO-DSCT algorithm, the double search strategy is proposed to balance the exploration and exploitation ability of the swarm and obtain a large number of solutions, in which an improved attractor construction mechanism based on opposition-based learning can help each particle to escape from local optima. The double search strategy combined with the opposition-based learning strategy could solve the former problem. To solve the latter problem, the circular transposon mechanism is proposed to improve the communication ability of the particles in the external archive, which can make the solutions have a higher accuracy and a better distribution. Therefore, the MOQPSO-DSCT algorithm can generate a set of well-distributed and accurate nondominated solutions. The double search strategy and circular transposon mechanism are depicted in detail in the following subsections.

3.1. The Double Search Strategy Combined with the Opposition-Based Learning. To improve the diversity of the solutions set and keep a balance between the exploration and exploitation ability of the swarm, the double search strategy is proposed in the MOQPSO-DSCT algorithm to replace the single search pattern in QPSO. In earlier iterations, the particles dominantly learn from their personal best position to improve the ability of the particles to discover new regions. In later iterations, the particles dominantly learn from the global best position to improve the convergence performance. A search probability parameter sp is used to control the two search patterns to be executed alternately.

Inspired by the social and individual cognition in PSO, two search patterns are proposed in the MOQPSO-DSCT algorithm to replace the single search pattern in QPSO. One search pattern called global search pattern is described in (15) as follows:

$$x_{ij}(t+1) = attractor_{ij}(t) - \alpha |p_{ij}(t) - x_{ij}(t)| \ln\left(\frac{1}{u_{ij}(t)}\right) \quad (15)$$

where $p_i(t)$ is the personal best position of the i -th particle. Obviously, this search pattern focuses on the individual cognition, which can make all the particles freely diverge into the whole search space and improve the probability of the particles to find more global best solutions. So the global search pattern is suitable to be executed in earlier iterations.

The other search pattern called local search pattern is described in (16) as follows:

$$x_{ij}(t+1) = attractor_{ij}(t) + \alpha |leader_j(t) - x_{ij}(t)| \ln\left(\frac{1}{u_{ij}(t)}\right) \quad (16)$$

where $leader(t)$ is a particle selected from the external archive with a large crowding distance. Obviously, this search pattern focuses on the social cognition, which can make the particles search around the elite individuals to improve the ability to converge to an optimum. Since $leader(t)$ has a large crowding distance, it can directly guide the particle to move toward the sparse region of the search space, which can keep the distribution of the nondominated solutions set. So the local search pattern is suitable to be executed in later iterations.

To keep a balance between the implementation of the two patterns, a search probability parameter sp is introduced. In [30], the exponential decrease rule and randomness were integrated into the dynamical control of the inertial weight ω instead of decreasing ω simply with iterations. Inspired by this idea, this method is introduced to dynamical adjust the value of sp , then sp is shown as

$$sp = \exp\left(\frac{1-t^2}{t_{\max}^2-1}\right) * \text{rand} \quad (17)$$

where t is the number of the current iteration and t_{\max} is the maximum number of iterations. It can be seen that the value of sp is decreased with the increasing of iterations

on the whole but maintains an oscillatory trend because the curve is affected by a random number between $[0, 1]$. If sp is more than 0.5, the swarm will perform the global search represented by (15); otherwise, the swarm will turn to the local search represented by (16). Thus, the particles in MOQPSO-DSCT can be updated as follows:

$$x_{ij}(t+1) = attractor_{ij}(t) - \alpha |p_{ij}(t) - x_{ij}(t)| \ln\left(\frac{1}{u_{ij}(t)}\right) \quad \text{if } sp > 0.5 \quad (18)$$

$$x_{ij}(t+1) = attractor_{ij}(t) + \alpha |leader_j(t) - x_{ij}(t)| \ln\left(\frac{1}{u_{ij}(t)}\right) \quad \text{otherwise}$$

Therefore, the two search patterns alternatively executed can not only avoid the situation that all the particles in QPSO converge to the same position $c(t)$, but also make the particles move toward the sparse region, which is beneficial for the diversity of the nondominated solutions set.

However, the particle will get trapped into local optimum if its personal best position is equal to the global best position in (8) because the attractor can not be updated for a long time. In MOQPSO-DSCT, opposition-based learning is adopted to further search a better attractor for an individual locally when its personal best position is equal to the global personal. According to (14), the opposite attractor $attractor^*(x_1^*, x_2^*, \dots, x_D^*)$ of the $attractor(x_1, x_2, \dots, x_D)$ can be described as

$$x_j^* = ua_j + lb_j - x_j, \quad j = 1, 2, \dots, D \quad (19)$$

where ua_j and lb_j are the upper bound and the lower bound of the j -th dimension of population, respectively. If $attractor^*$ is dominated by $attractor$, the current attractor will be kept; If $attractor$ is dominated by $attractor^*$, the current attractor will be replaced with $attractor^*$; otherwise, one of the two attractors will be selected randomly as the current attractor.

3.2. The Circular Transposon Mechanism. It is evident that the external archive is crucial for most MOPSO algorithms to store the nondominated solutions and select the best position of the population. However, individuals in the external archive can not exchange useful information between each other, which is not beneficial for the population to move toward the true PF . To improve the communication ability of the particles in the external archive, the circular transposon mechanism is introduced into the external archive in the proposed algorithm.

The flowchart of the circular transposon mechanism is given in Figure 2. To adopt this mechanism into the external archive, the 50% nondominated solutions with the largest crowding distance in the external archive A will be stored to a new archive E . For each solution $A_i = (x_{i1}, x_{i2}, \dots, x_{iD}) (i =$

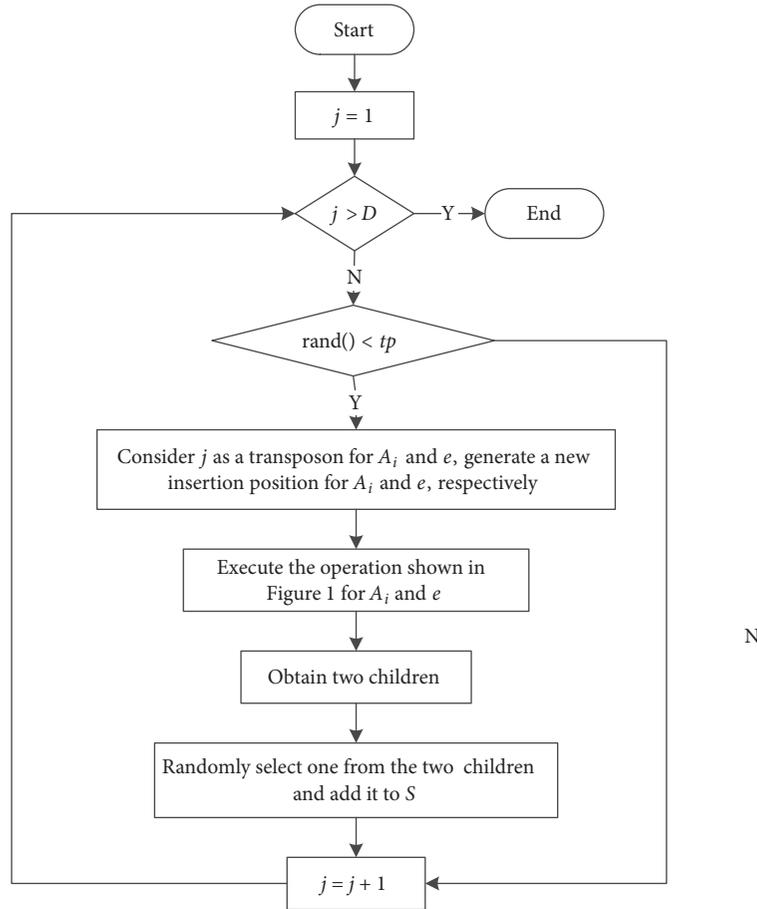


FIGURE 2: The flowchart of the circular transposon mechanism.

$1, 2, \dots, |A|$), a solution $e = (x_1, x_2, \dots, x_D)$ is randomly selected from E , then a probability parameter tp will be introduced to control the operation described as Figure 1 on A_i and e . For each dimension of A_i and e , if the value of tp is less than an uniformly distributed random number between $[0, 1]$, this operation will be performed on A_i and e for once. After this operation, a new set S containing the children of A_i and e will be generated. Therefore, the solutions in S are the results of the information exchange among the particles in A . Finally, the algorithm for updating the external archive is used to update the external archive A , which can be referred to Algorithm 3 in [31].

3.3. The Steps of the Proposed MOQPSO-DSCT Algorithm. The steps of the MOQPSO-DSCT algorithm are given in Figure 3, and the detailed steps are summarized as follows.

Step 1. Initialize the position of each particle, the corresponding parameters, and the external archive A .

Step 2. Evaluate all the particles and store the nondominated solutions to the external archive A according to Pareto dominance.

Step 3. For each particle in population, select one particle from the external archive A as a leader according to the crowding distance.

Step 4. If the personal best position of the current particle is equal to the best position of the population, opposition-based learning described as (19) will be used to construct an attractor for the current particle; otherwise, an attractor will be constructed according to (8) for the current particle.

Step 5. Calculate the value of sp according to (17) and update the position for each particle according to the double search strategy described as (18).

Step 6. Evaluate all the particles and update A ; the 50% solutions in A with the largest crowding distance are stored to another archive E .

Step 7. For each particle in A , randomly select one solution from E , perform the operation of Figure 2 on the two individuals, then randomly select one from the offspring generated by the circular transposon mechanism, and add it to a new set S .

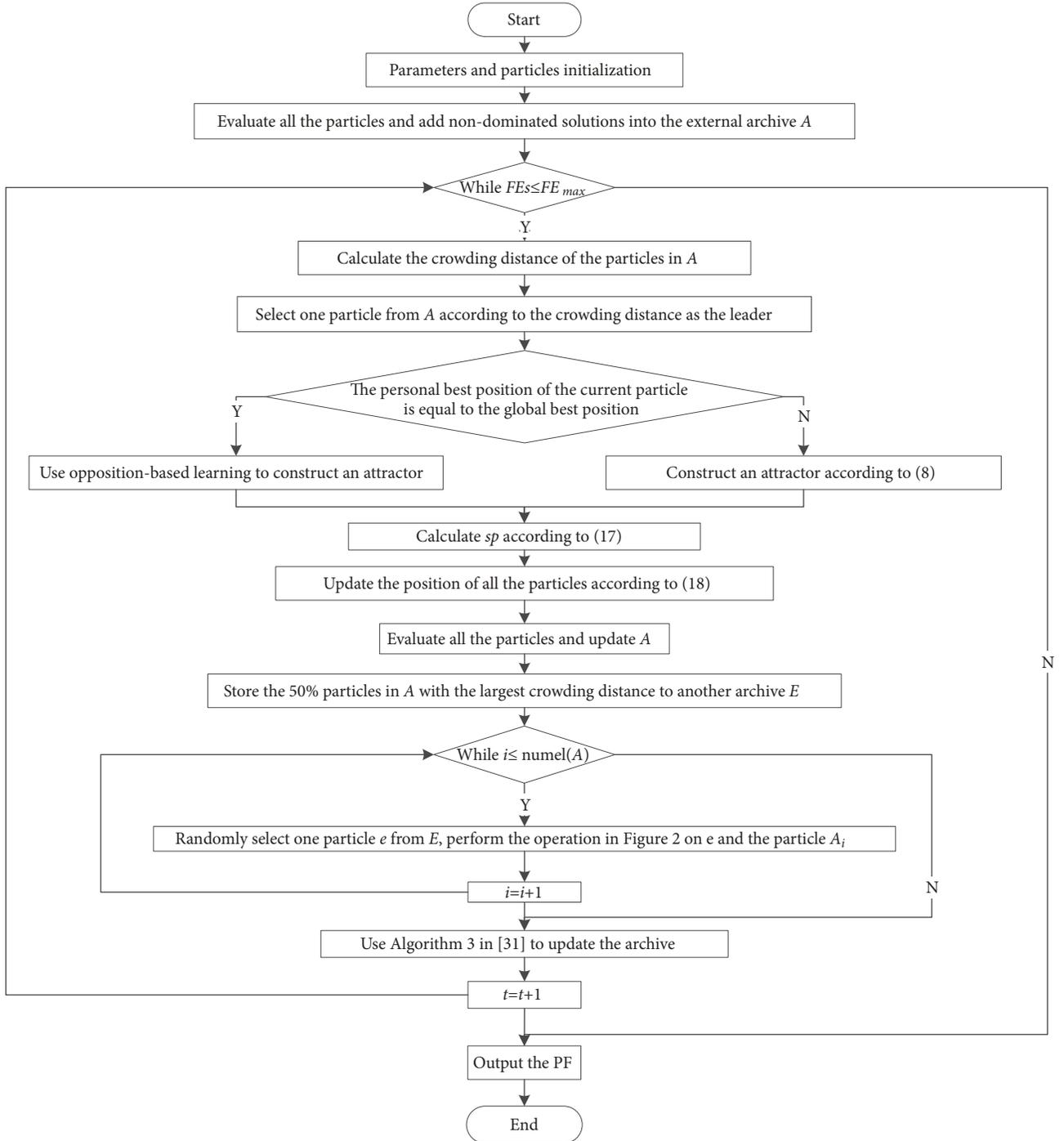


FIGURE 3: The flowchart of the MOQPSO-DSCT algorithm.

Step 8. Use Algorithm 3 in [31] to update A .

Step 9. If the termination condition is satisfied, the optima results are output; otherwise repeat Steps 3–8.

3.4. Time Complexity of the MOQPSO-DSCT Algorithm. According to Section 3.3, the time complexity of the MOQPSO-DSCT algorithm is determined by the main loop

from Steps 3 to 8. This main loop can also be divided into three parts. The time complexity of the proposed algorithm is mainly influenced by the number of the objectives k , the size of the population M , the size of the archive N , and the number of the decision variables D .

The first part is the double search strategy from Steps 3–6. The time complexity of this part is determined by the size of the population because the double search strategy is

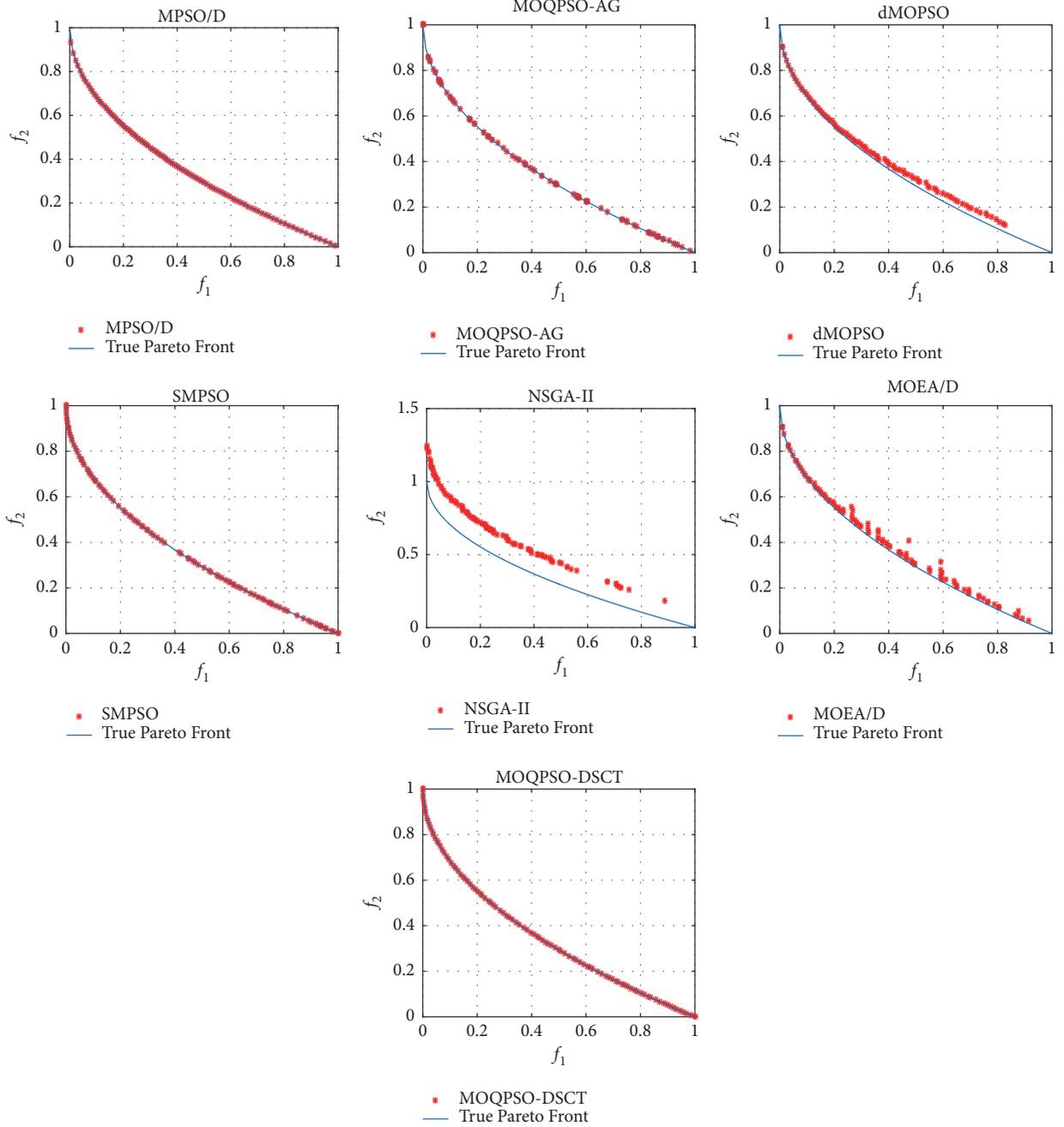


FIGURE 4: PFs obtained by seven algorithms on ZDT1.

performed on each particle. Therefore, the time complexity of this part is $O(kDM)$.

The second part is Step 7, which contains the circular transposon mechanism. It can be observed from Section 3.2 and Figure 3 that the time complexity of the circular transposon mechanism is determined by the size of the archives A and E and the number of decision variables. Since not all the decision variables are affected by this mechanism and the number of decision variables is much smaller than the size of A , the influence of the number of decision variables can be

ignored. Therefore, the time complexity of the second part is $O(kDN^2)$.

The last part is Step 8, in which Algorithm 3 in [31] is used to update the external archive. The time complexity of this part is also $O(kDN^2)$, which can be obtained from [31].

Therefore, the time complexity of the MOQPSO-DSCT algorithm is $O(kD * (N^2 + N^2 + M)) \sim O(kDN^2)$ in each iteration. Since the values of D and k are much smaller than M and N , the time complexity of the MOQPSO-DSCT algorithm is $O(N^2)$ in each iteration.

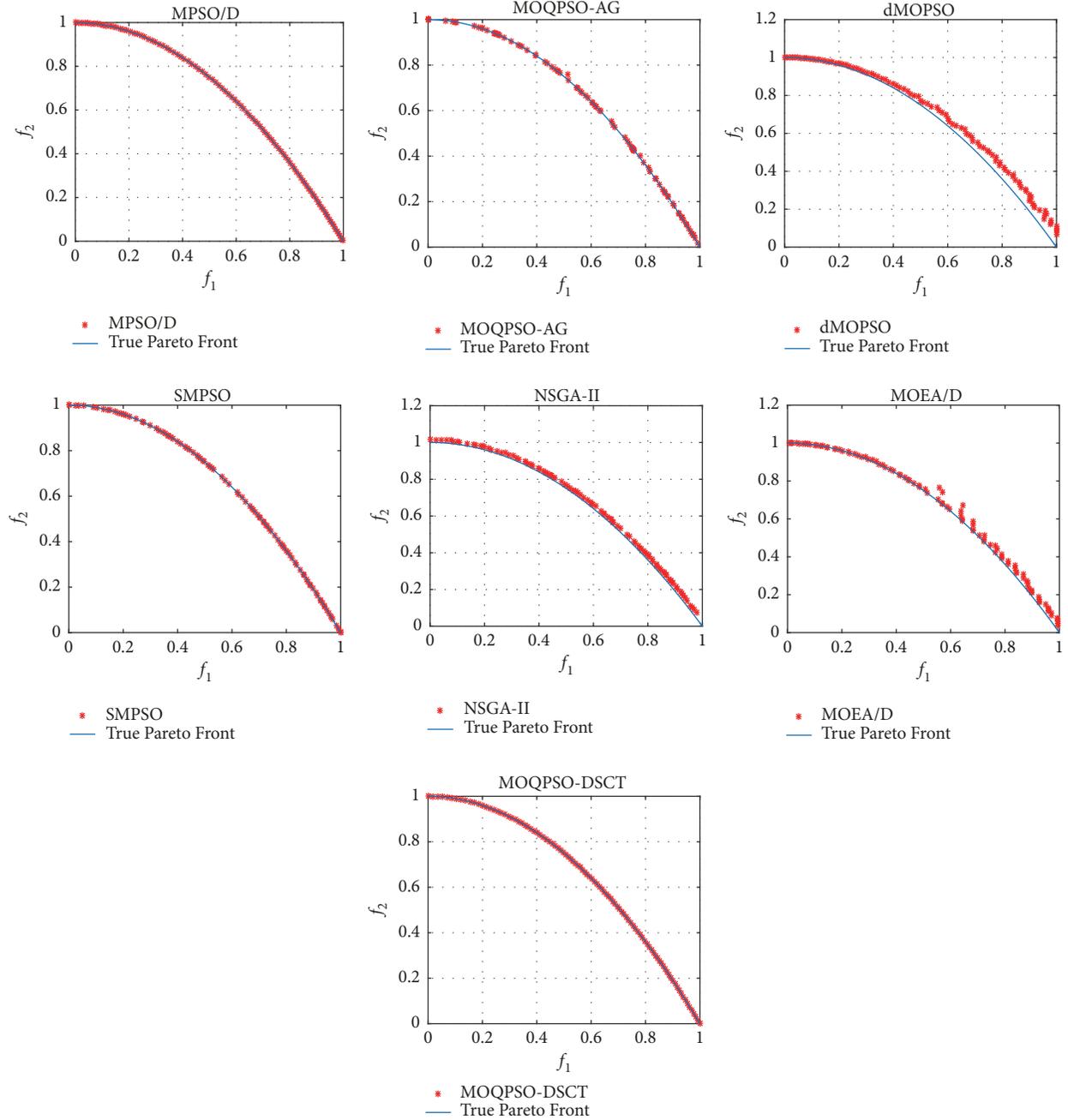


FIGURE 5: PFs obtained by seven algorithms on ZDT2.

4. Experiment Results and Discussion

4.1. Test Functions and Parameters Setting. In this section, to verify the performance of the proposed algorithm on ZDT (ZDT1~4, ZDT6) [32] and DTLZ (DTLZ2, DTLZ5, and DTLZ6) [33] function sets, the MOQPSO-DSCT algorithm will be compared to other multiobjective optimization algorithms. These algorithms can be divided into two categories, multiobjective PSO including MPSO/D [34], MOQPSO-AG [14], dMOPSO [12], SMPSO [35], and other multiobjective optimization algorithms including NSGA-II

[36] and MOEA/D [37]. MOQPSO-AG and dMOPSO are both mentioned above. MPSO/D is a multiobjective particle swarm optimization that uses objective space decomposition to maintain the diversity of the solutions. SMPSO is a new multiobjective swarm optimization algorithm, in which one mechanism is used to limit the velocity of the particles to enable the particles search the space sufficiently. NSGA-II is a multiobjective genetic algorithm without an external archive, which uses a fast nondominated sorting to make the algorithm find the true PF at a high speed and crowding distance computation to get a well-distributed solutions.

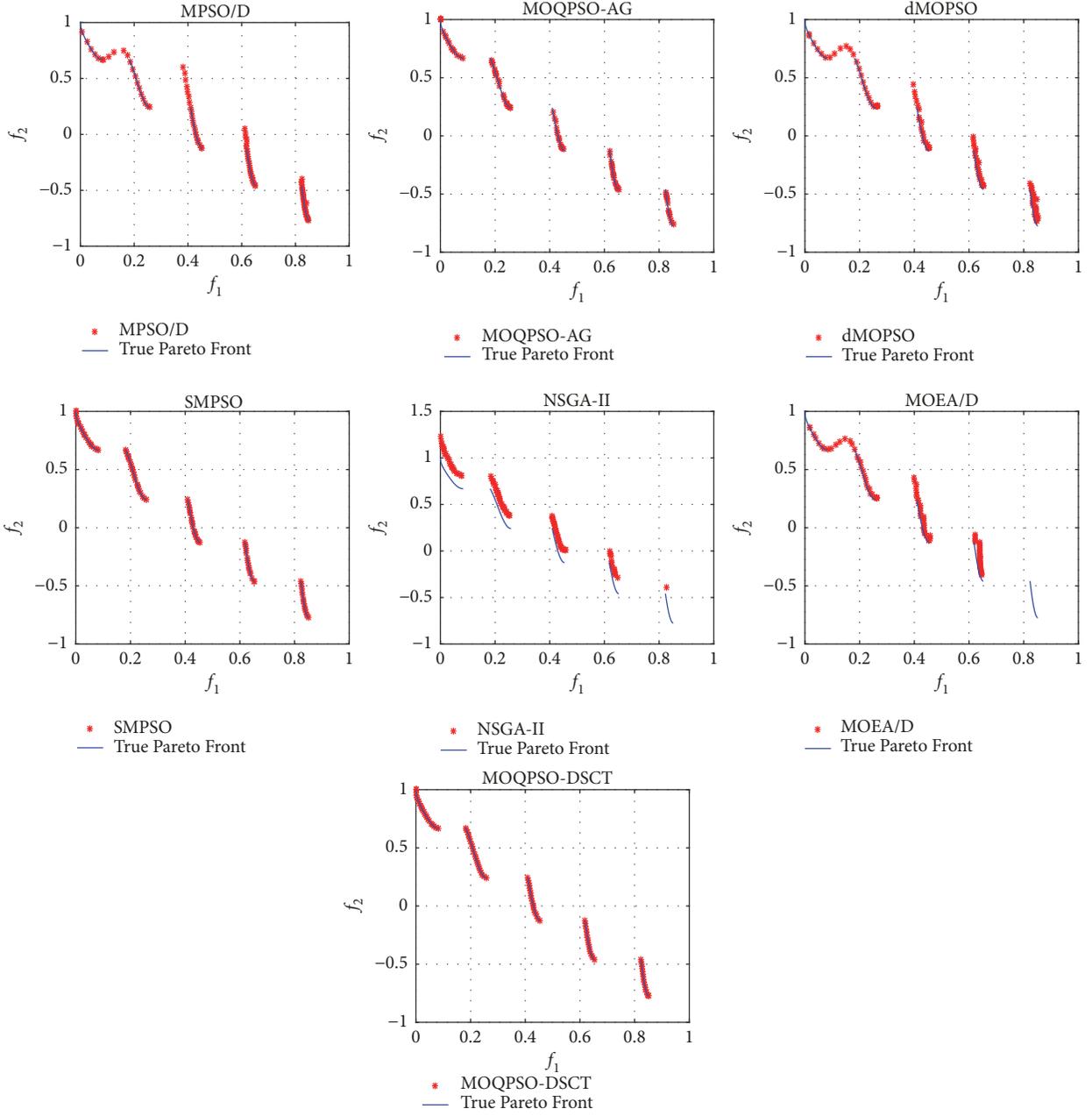


FIGURE 6: PFs obtained by seven algorithms on ZDT3.

MOEA/D is a multiobjective evolutionary algorithm based on decomposition, which can optimize a number of subproblems simultaneously instead of treating a MOP as a whole to deal with. It is necessary to compare the performance of MOQPSO-DSCT and MOQPSO-AG because they all based on QPSO. Since the particles in QPSO do not have velocities, the comparison between SMPSO and MOQPSO-DSCT can sufficiently test the performance of the quantum model. The comparison of SMPSO, MOQPSO-AG, and MOQPSO-DSCT can test the performance of the circular transposon mechanism because they all use an external archive. The performance difference between the algorithm with an external archive and the algorithm without an external archive can

be seen from the comparison of MOQPSO-DSCT, MPSO/D, and dMOPSO. The comparison of MOQPSO-DSCT, NSGA-II, and MOEA/D can test whether PSO is more competitive than other evolutionary algorithms. The details of test functions are listed in Table 1.

In the experiments, the size of population is 100, the maximum evaluation is 30000, and the maximum iteration is 300 for all the algorithms. The size of the external archive is 100 for ZDT and DTLZ function sets. Parameters in all the algorithms are shown in Table 2. 30 test runs are done for each test function in each algorithm. The algorithms are executed on MATLAB R2015b, Intel(R) Core(TM) i5-6500, 3.20GHz, 4GB RAM.

TABLE 1: The details of test functions.

Test functions	Objective functions	Features of PF	Variables	Range
ZDT1	$\min f_1(x) = x_1$ $\min f_2(x) = g(x) \left(1 - \sqrt{\frac{x_1}{g(x)}} \right)$ $\text{s.t. } g(x) = 1 + \frac{9 \left(\sum_{i=2}^D x_i \right)}{(D-1)}$	Convex	30	$x_i \in [0, 1]$
ZDT2	$\min f_1(x) = x_1$ $\min f_2(x) = g(x) \left(1 - \left(\frac{x_1}{g(x)} \right)^2 \right)$ $\text{s.t. } g(x) = 1 + \frac{9 \left(\sum_{i=2}^D x_i \right)}{(D-1)}$	Concave	30	$x_i \in [0, 1]$
ZDT3	$\min f_1(x) = x_1$ $\min f_2(x) = g(x) \left(1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1 \sin(10\pi x_1)}{g(x)} \right)$ $\text{s.t. } g(x) = 1 + \frac{9 \left(\sum_{i=2}^D x_i \right)}{(D-1)}$	Disconnected multi-modal	30	$x_i \in [0, 1]$
ZDT4	$\min f_1(x) = x_1$ $\min f_2(x) = g(x) \left(1 - \sqrt{\frac{x_1}{g(x)}} \right)$ $\text{s.t. } g(x) = 1 + 10(D-1) + \sum_{i=2}^D (x_i^2 - 10 \cos(4\pi x_i))$	Convex multi-modal	10	$x_1 \in [0, 1];$ $x_i \in [-5, 5];$ $i = 2, \dots, D$
ZDT6	$\min f_1(x) = 1 - \exp(-4x_1) \sin(6\pi x_1)^6$ $\min f_2(x) = g(x) \left(1 - \left(\frac{x_1}{g(x)} \right)^2 \right)$ $\text{s.t. } g(x) = 1 + 10(D-1) + \sum_{i=2}^D (x_i^2 - 10 \cos(4\pi x_i))$	Concave non-uniform disconnected	10	$x_i \in [0, 1]$
DTLZ2	$\min f_1(x) = (1 + g(x)) \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right)$ $\min f_2(x) = (1 + g(x)) \cos\left(\frac{\pi x_1}{2}\right) \sin\left(\frac{\pi x_2}{2}\right)$ $\min f_3(x) = (1 + g(x)) \sin\left(\frac{\pi x_1}{2}\right)$ $\text{s.t. } g(x) = \sum_{i=3}^D (x_i - 0.5)^2$	Concave	10	$x_i \in [0, 1]$
DTLZ5	$\min f_1(x) = (1 + g(x)) \cos\left(\frac{\theta_1 \pi}{2}\right) \cos\left(\frac{\theta_2 \pi}{2}\right)$ $\min f_2(x) = (1 + g(x)) \cos\left(\frac{\theta_1 \pi}{2}\right) \sin\left(\frac{\theta_2 \pi}{2}\right)$ $\min f_3(x) = (1 + g(x)) \sin\left(\frac{\theta_1 \pi}{2}\right)$ $\text{s.t. } g(x) = \sum_{i=1}^D (x_i - 0.5)^2$ $\theta_i = \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i)$	-	10	$x_i \in [0, 1]$
DTLZ6	$\min f_1(x) = (1 + g(x)) \cos\left(\frac{\theta_1 \pi}{2}\right) \cos\left(\frac{\theta_2 \pi}{2}\right)$ $\min f_2(x) = (1 + g(x)) \cos\left(\frac{\theta_1 \pi}{2}\right) \sin\left(\frac{\theta_2 \pi}{2}\right)$ $\min f_3(x) = (1 + g(x)) \sin\left(\frac{\theta_1 \pi}{2}\right)$ $\text{s.t. } g(x) = \sum_{i=1}^D x_i^{0.1}$ $\theta_i = \frac{\pi}{4(1 + g(x))} (1 + 2g(x)x_i)$	Non-uniform	10	$x_i \in [0, 1]$

TABLE 2: Parameters setting in the seven algorithms.

Algorithms	Parameters setting
MOQPSO-DSCT	$\alpha : 1.0 - 0.5, tr = 0.2$
MPSO/D	$\omega : 0.9 \rightarrow 0.1, c_1 = c_2 = 2.0, CR = 0.5, p_c = 1.0, p_m = 0.1, T = 20, F = 0.5$
MOQPSO-AG	$\alpha : 1.0 - 0.5, p_m = 0.1, nGrid = 30$
dMOPSO	$\omega : 0.9 \rightarrow 0.4, c_1 = c_2 = 1.5$
SMPSO	$\omega : 0.9 \rightarrow 0.4, c_1 = c_2 = 1.5, p_m = 1/n, \mu_m = 20$
NSGA-II	$p_c = 0.9, p_m = 1/n, \eta_c = \mu_m = 20$
MOEA/D	$p_c = 1.0, p_m = 1/n, \mu_m = 20, T = 20, F = 0.5$

TABLE 3: The IGD values of all the algorithms in the eight test functions.

		MOQPSO-DSCT	MPSO/D	MOQPSO-AG	dMOPSO	SMPSO	NSGA-II	MOEA/D
ZDT1	Mean	2.8115e-3	3.8276e-3	7.8213e-3	1.4360e-2	5.0019e-3	5.9674e-1	1.5134e-2
	Std	5.84e-5	4.54e-3	6.64e-4	6.57e-3	9.55e-4	3.46e-2	1.46e-2
ZDT2	Mean	3.9237e-3	4.7723e-3	8.3342e-3	4.0957e-1	4.8600e-3	1.3533e-1	1.0239e-2
	Std	3.57e-5	5.64e-3	7.56e-4	3.01e-1	1.21e-4	7.63e-2	7.85e-3
ZDT3	Meas	4.4523e-3	9.8854e-3	9.7834e-3	1.5570e-2	5.2183e-3	9.8741e-2	2.0824e-2
	Std	3.28e-5	5.39e-3	1.46e-3	1.14e-3	2.49e-4	5.28e-2	1.08e-2
ZDT4	Mean	3.7716e-3	2.2094e+1	1.6469e+1	8.0307e-3	4.8343e-3	3.0449e-1	1.8829e-2
	Std	4.55e-5	5.01e+0	5.67e+0	3.06e-3	5.14e-3	1.69e-1	8.19e-3
ZDT6	Mean	3.3158e-3	3.7869e-3	7.2344e-3	3.3020e-3	4.1433e-3	8.8106e-3	4.2676e-3
	Std	2.45e-5	3.84e-6	6.07e-4	1.30e-6	4.86e-4	5.07e-3	9.92e-4
DTLZ2	Mean	6.6844e-2	5.4732e-2	1.1113e-1	1.3527e-1	7.5454e-2	6.9344e-2	5.4467e-2
	Std	2.02e-3	1.37e-4	5.42e-3	4.79e-3	3.75e-3	2.11e-3	1.58e-6
DTLZ5	Mean	4.3522e-3	3.7595e-2	1.9358e-2	4.1346e-1	5.4218e-3	5.7790e-3	8.3766e-1
	Std	2.20e-5	2.23e-3	3.34e-3	6.06e-3	3.02e-4	2.95e-4	3.53e-5
DTLZ6	Mean	4.0435e-3	3.1749e-2	1.5366e-2	3.3606e-2	8.6849e-3	6.8228e-3	3.3847e-2
	std	2.31e-5	8.97e-4	5.87e-3	7.81e-5	2.82e-3	3.82e-4	7.85e-5

4.2. *Performance Indicator.* In order to better evaluate the performance of MOQPSO-DSCT, IGD (Inverted Generational Distance) [38] will be used as an indicator in this paper. IGD is the distance from the *PF* obtained by the algorithm to the true *PF*. Let P be the true Pareto optimal set and A be the solutions set obtained by our proposed algorithm. The IGD of A to P can be calculated by the following equation:

$$IGD = \frac{1}{|P|} \sum_{i=1}^{|P|} Dist_i \quad (20)$$

where $Dist_i = \min_{j=1}^{|A|} \sqrt{\sum_{l=1}^k ((f_l(p_i) - f_l(a_j)) / (f_l^{\max} - f_l^{\min}))^2}$ is the minimum normalization Euclidean distance; f_l^{\max} and f_l^{\min} are the maximum and minimum value of the l -th objective in P , respectively; $l = 1, 2, \dots, k$; k is the number of the objectives; $p_i \in P, i = 1, 2, \dots, |P|$; $a_j \in A, j = 1, 2, \dots, |A|$; $|P|$ is the size of the solutions set obtained by our proposed algorithm. The value of $|P|$ is set to roughly 1000 for ZDT test functions and 5000 for DTLZ test functions. It can be seen that a value of zero for IGD means the *PF* obtained by the algorithm has completely coincided with the true *PF*.

4.3. *Experimental Results and Discussion.* The mean values and standard deviations (std) of IGD for all the algorithms in each test function are all summarized in Table 3, where the best values are highlighted in bold and italic font. The *PFs* obtained by all algorithms in each test function are shown in Figures 4–11. The convergence curves of all algorithms in each test function are shown in Figure 12. It can be seen from Table 3 that MOQPSO-DSCT obtains best results on six test functions in terms of both mean and std values, including ZDT1, ZDT2, ZDT3, ZDT4, DTLZ5, and DTLZ6, which demonstrates that MOQPSO-DSCT has a better convergence and distribution performance than other compared algorithms.

ZDT1 is a two-objective function with a convex *PF* and ZDT2 is a two-objective function with a concave *PF*. Almost all the algorithms can converge to the *PF*. It can be seen from Table 3 that MOQPSO-DSCT, MPSO/D, MOQPSO-AG, and SMPSO all perform better than dMOPSO, NSGA-II, and MOEA/D. As shown in Figures 4, 5, and 12, MOQPSO-AG has a relatively faster convergence speed than other algorithms, but it can not generate a set of well-distributed solutions, which will lead to a slightly worse IGD value than other algorithms. NSGA-II has the slowest convergence

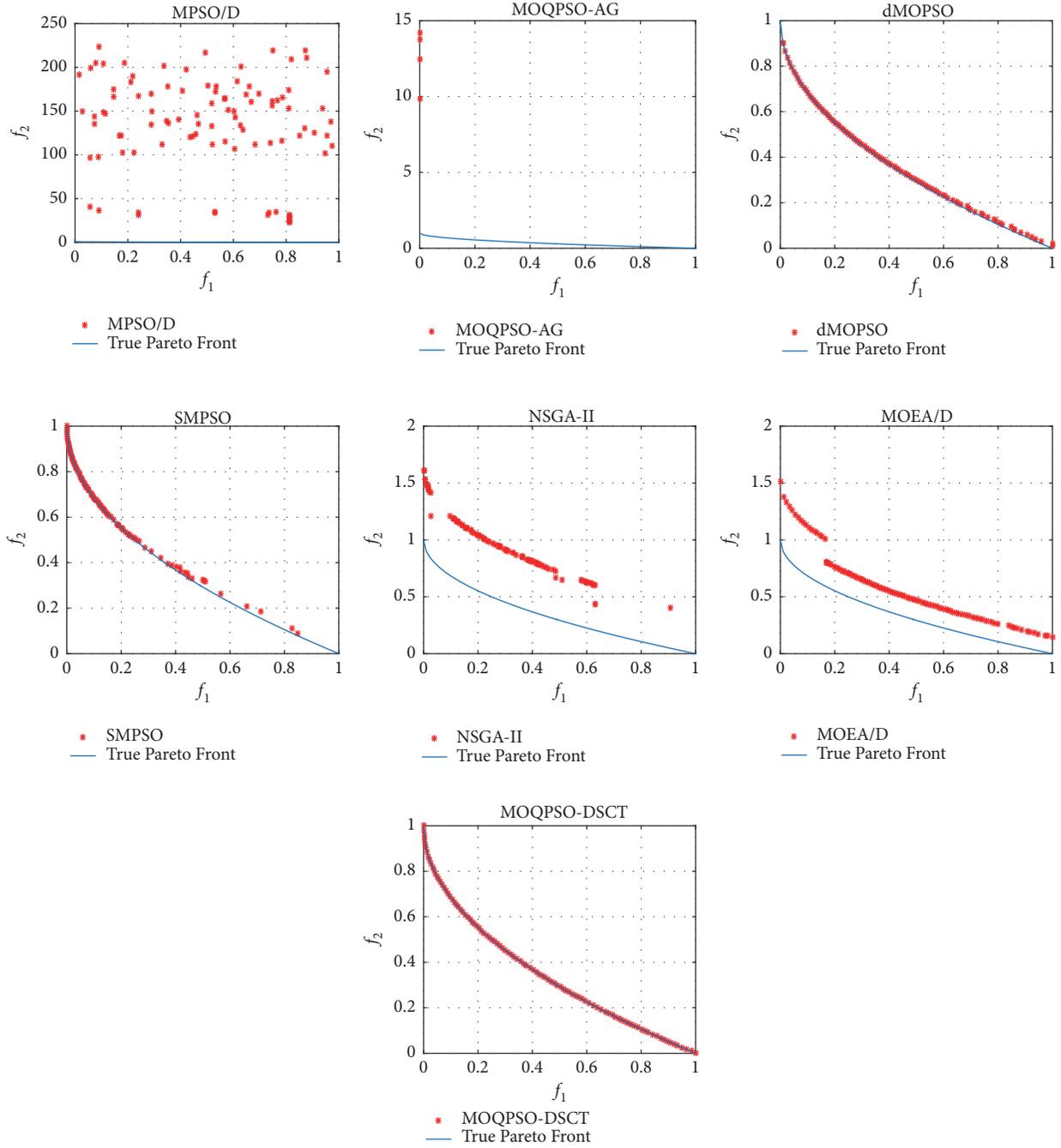


FIGURE 7: PFs obtained by seven algorithms on ZDT4.

speed in all the algorithms and cannot find the true *PF* on ZDT1. Although the IGD value of MOQPSO-DSCT, SMPSO, and MPSO/D are very close, the std value of MOQPSO-DSCT is significantly better than SMPSO and MPSO/D in terms of magnitude. Therefore, MOQPSO-DSCT has the best performance on ZDT1 and ZDT2.

ZDT3 is a two-objective function with a disconnected *PF*. It can be observed from Figure 6 that the worst performance on ZDT3 is NSGA-II, which is unable to find the true *PF*; MOEA/D, dMOPSO, and MPSO/D all have the similar performance; they can converge to the true *PF* but fail to

obtain a set of satisfactory solutions. It can be seen from Table 3 that the IDG values of MOQPSO-DSCT, MOQPSO-AG, and SMPSO are very close, but the std of MOQPSO-DSCT is significantly better than MOQPSO-AG and SMPSO in terms of magnitude. Therefore, MOQPSO-DSCT has the best performance on ZDT3.

ZDT4 is a two-objective function with 21^9 local *PFs* and a global *PF*. Many algorithms can not converge to the true *PF* because of their low ability to escape from local optimal. As shown in Figure 7, both MPSO/D and MOQPSO-AG get trapped at a local *PF* far away from the true *PF*; the

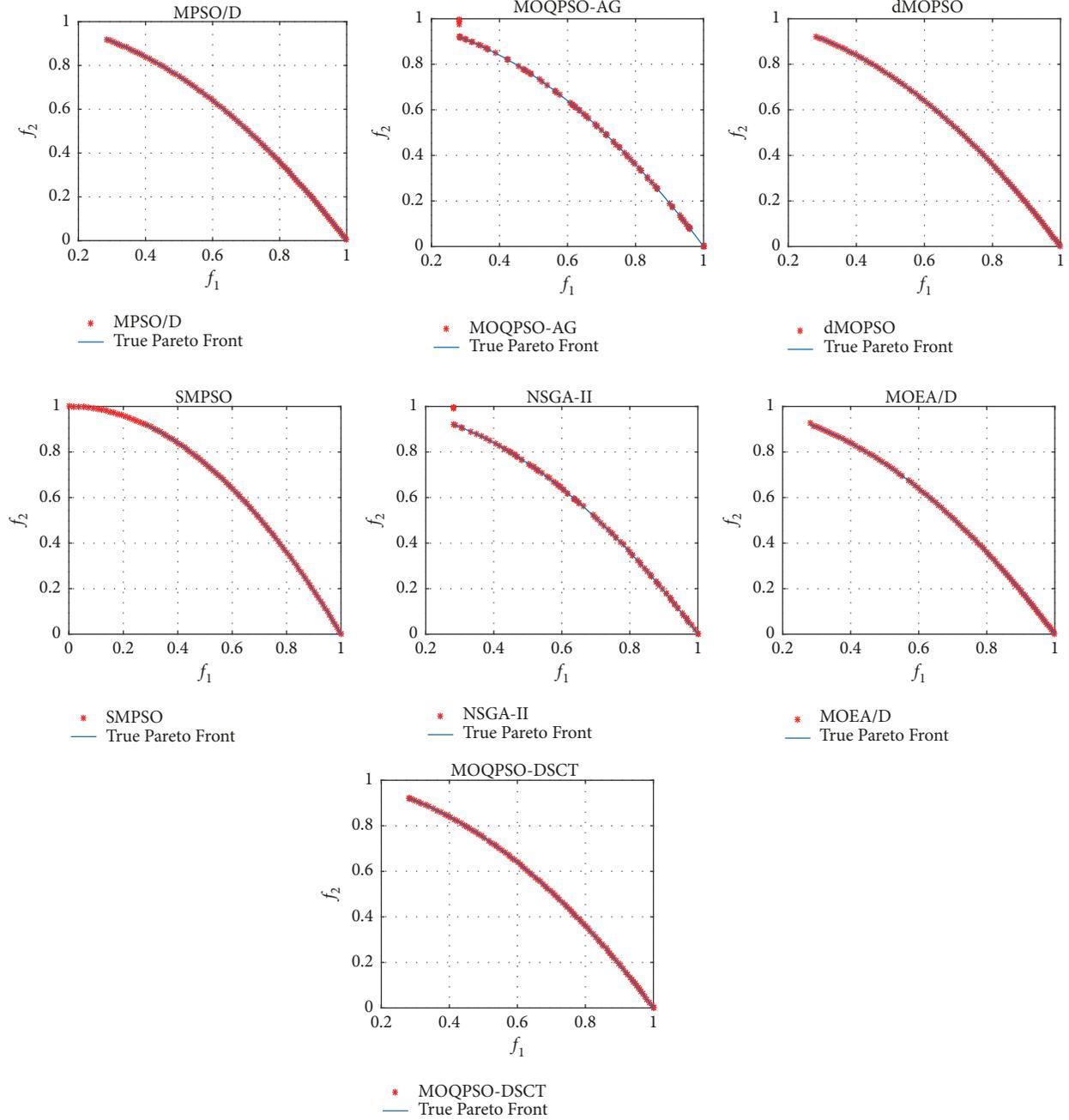


FIGURE 8: PFs obtained by seven algorithms on ZDT6.

PFs produced by NSGA-II and MOEA/D are closer to the true PF; it can be deduced that NSGA-II and MOEA/D have the potential to converge to the true PF if the maximum evaluation is set to a larger number. Only MOQPSO-DSCT, dMOPSO, and SMPSO can find the true PF. It can be seen from Figure 12 that the convergence speed of MOQPSO-DSCT is significantly faster than other algorithms. The values in Table 3 indicate that MOQPSO-DSCT can find best convergence and better spread solutions along the entire true PF than other compared algorithms with the lowest value of IGD.

ZDT6 is a two-objective function with a nonconvex and nonuniformly spaced PF. All the algorithms can converge easily to the true PF. It can be observed from Figure 8 that SMPSO and MOQPSO-AG cannot push all the solutions to the true PF. In Table 3, MOQPSO-DSCT, MPSO/D, dMOPSO, and MOEA/D have the similar IGD value. The best performance on ZDT6 is dMOPSO with the lowest std value of IGD. MOQPSO-DSCT performs the second best on ZDT6.

DTLZ2 is a three-objective function with a spherical PF. As shown in Figure 9, the PFs obtained by dMOPSO are far away from the true PF; MOQPSO-DSCT and NSGA-II have

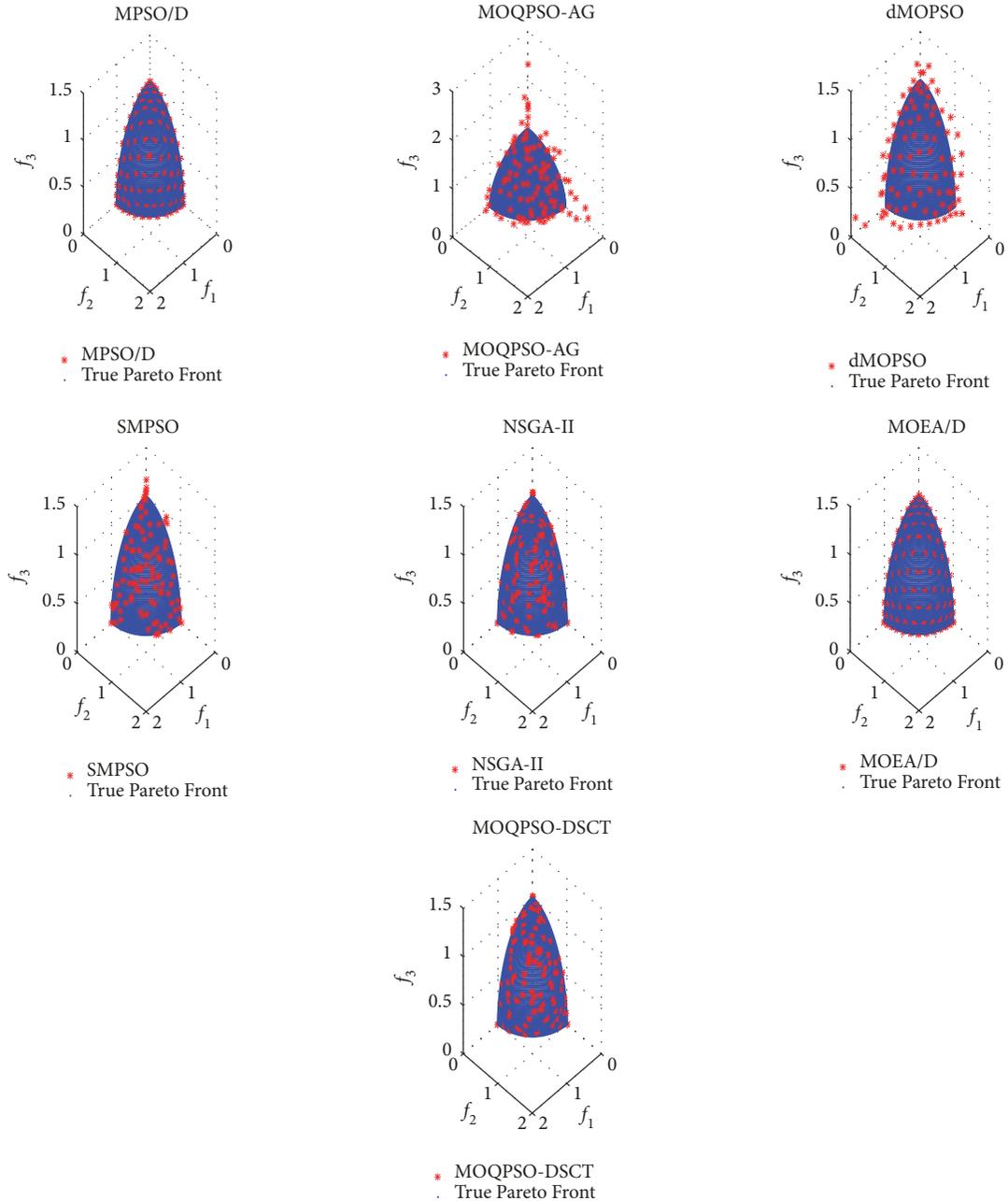


FIGURE 9: PFs obtained by seven algorithms on DTLZ2.

similar performance such that they can converge to the true PF but cannot make solutions spread well along the entire PF; MOQPSO-AG and SMPSO can find the true PF but fail to obtain a set of satisfactory solutions. It can be seen from Table 3 and Figure 9 that both MOEA/D and MPSO/D can not only converge easily to the true PF but also generate a set of well-distributed solutions. MOEA/D performs best on DTLZ2 with the lowest IGD value and MOQPSO-DSCT performs the third best on DTLZ2.

DTLZ5 and DTLZ6 are all 3-objective functions that test the ability to converge into a degenerated curve and the search ability in disconnected area, respectively. As shown in Figures

10 and 11, NSGA-II and SMPSO can converge to the true PF but cannot keep the distribution of the solutions; MPSO/D, MOEA/D and dMOPSO all have a bad performance on DTLZ5 and DTLZ6. MOQPSO-AG has good performance on DTLZ5 while it cannot push all the solutions to the true PF on DTLZ6. The values in Table 3 indicate that MOQPSO-DSCT performs significantly better than other algorithms.

4.4. The Comparison of the Diversity of the Solutions. The diversity of the nondominated solutions set indicates the number of the nondominated solutions obtained by the algorithm and the distribution of the nondominated solutions

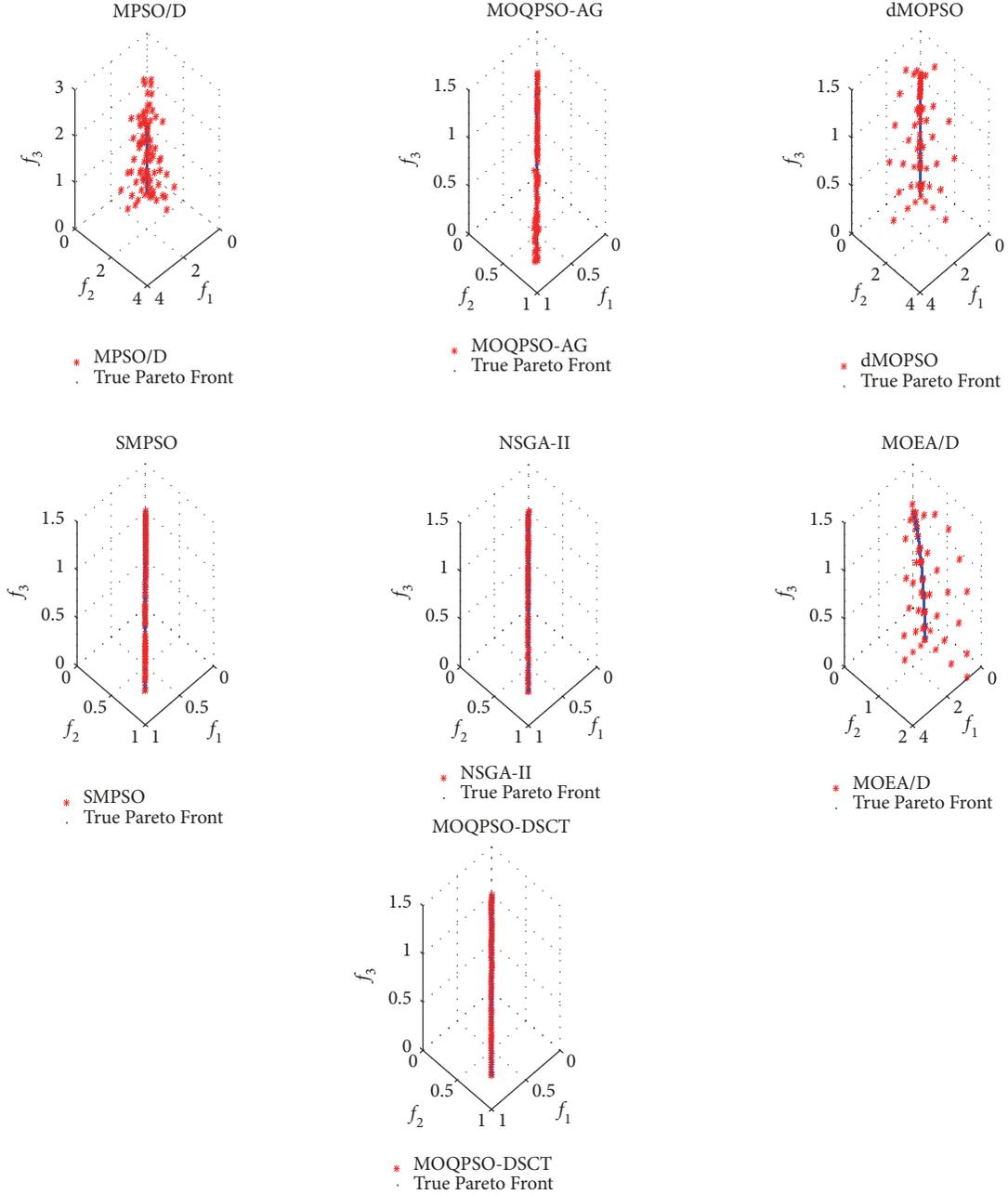


FIGURE 10: PFs obtained by seven algorithms on DTLZ5.

set. To measure the diversity of the nondominated solutions set, SP (Spacing Metric) [38] is introduced as an indicator. Let $A = (a_1, a_2, \dots, a_{|A|})$ be the nondominated solutions set obtained by the algorithm, so the indicator SP can be defined as

$$SP = \sqrt{\frac{1}{|A| - 1} \sum_{i=1}^{|A|} (\bar{d} - d_i)^2} \quad (21)$$

where $d_i = \min_{j=1}^{|A|} \{\sum_{l=1}^k |f_l(a_i) - f_l(a_j)|\}$, $i = 1, 2, \dots, |A|$; $|A|$ is the size of A ; \bar{d} is the mean value of d_i ($1 < i < |A|$); k is the

number of the objectives. A value of zero for SP means a good diversity of the nondominated solutions set. To demonstrate that the double search strategy in MOQPSO-DSCT can make the particles find more nondominated solutions and improve the diversity of the nondominated solutions set, we will compare the SP values of MOQPSO-DSCT and MOQPSO-AG on each test function. To highlight the effectiveness of the double search strategy, the value of tp will be set to 0 to exclude the influence of the circular transposition mechanism on the diversity of the nondominated solutions set. 30 test runs are done for each algorithm. The mean and std values of SP are shown in Table 4, where the best values are highlighted

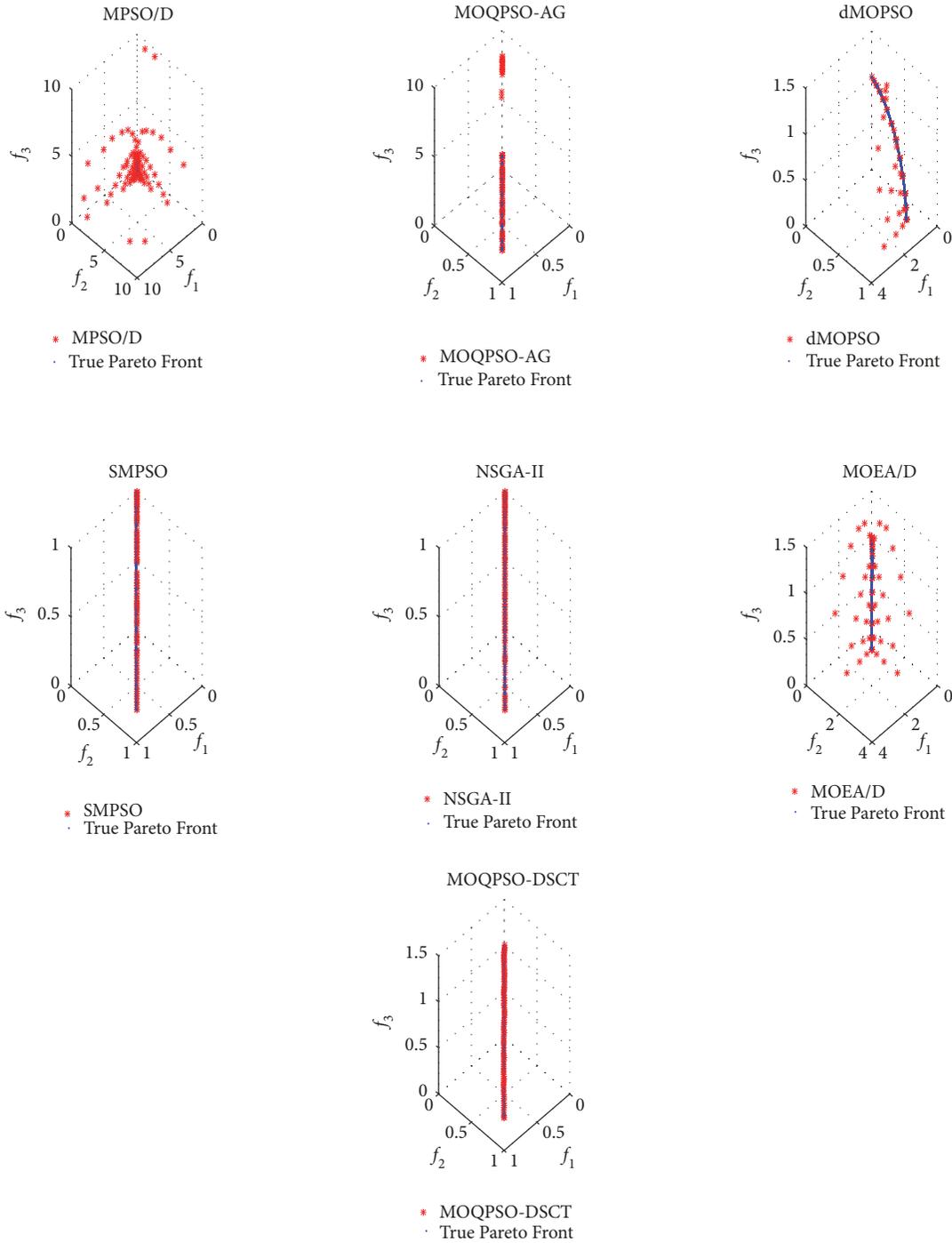


FIGURE 11: PFs obtained by seven algorithms on DTLZ6.

in bold and italic font. It can be seen from Table 4 that **MOQPSO-DSCT** get the best results in all test functions. Therefore, the double search strategy can be said to have the ability to improve the diversity of the nondominated solutions set.

4.5. Discussion on Probability Parameter. In our proposed algorithm, parameter tp is used to control the frequency of

the implementation of the circular transposon mechanism, which directly influences the information exchange ability of the particles in the archive. The mean values of IGD obtained by **MOQPSO-DSCT** with tp changing from 0 to 1.0 for ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ2, DTLZ5, and DTLZ6 are shown in Figure 13. 30 test runs are done for each test function. It can be seen that some test functions are insensitive to the change of tp , such as ZDT1, ZDT2, ZDT3, DTLZ2, and DTLZ5. It can be observed from Figures

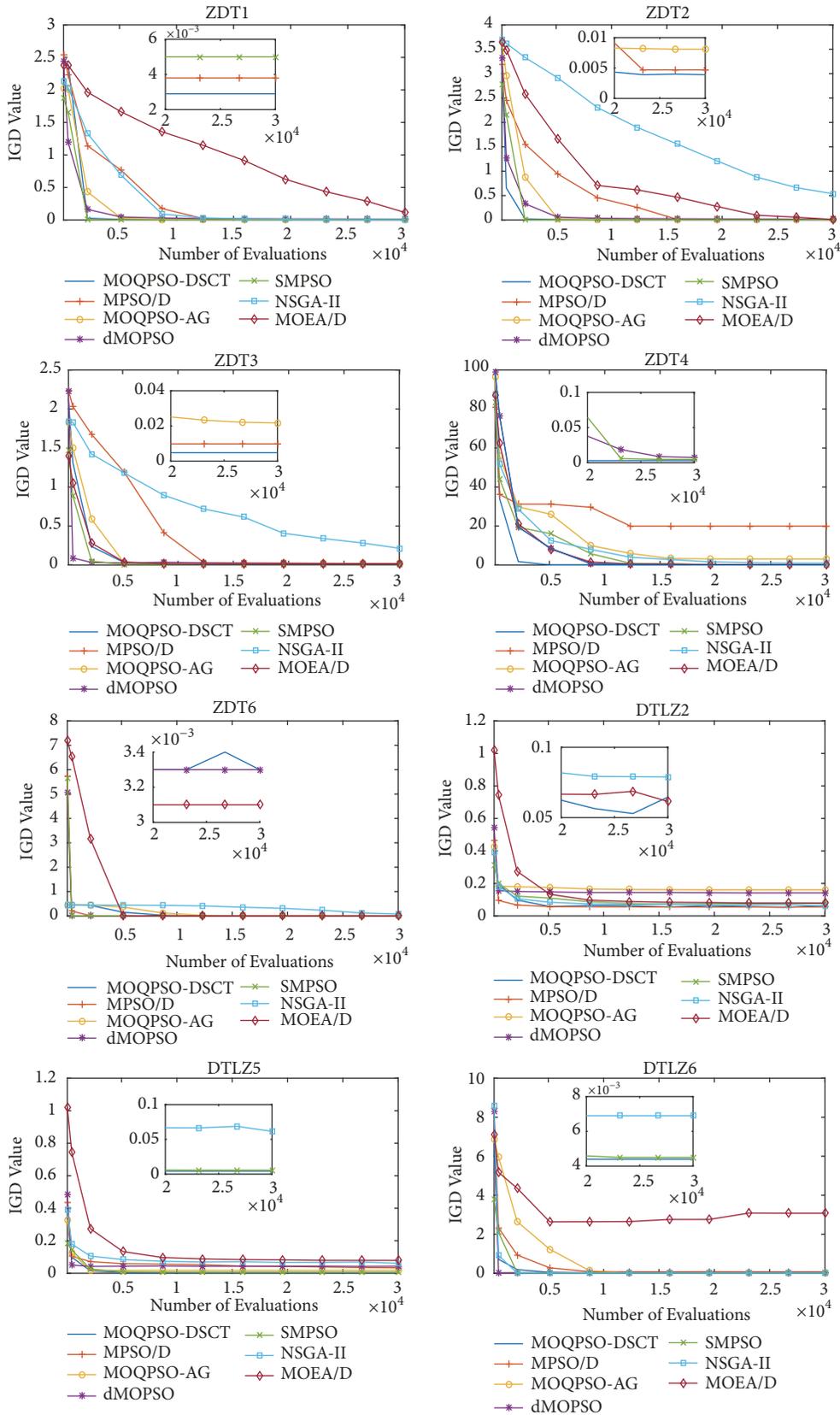


FIGURE 12: The convergence curves of seven algorithms on each test function.

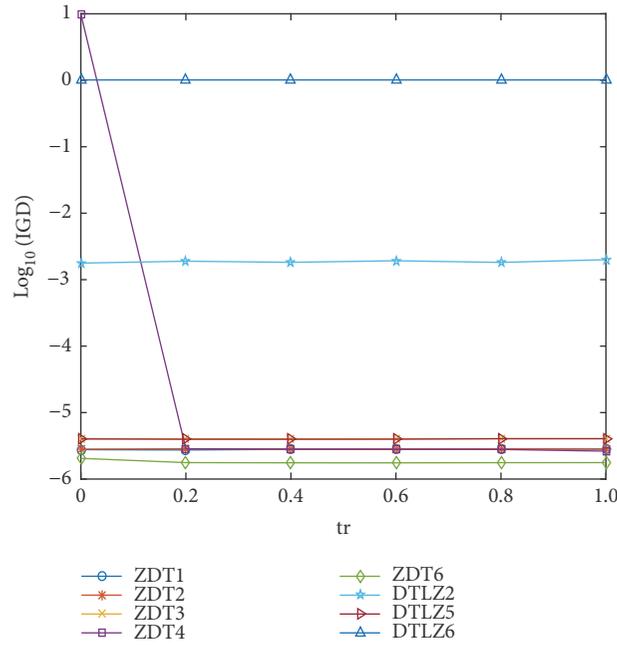


FIGURE 13: The mean values of IGD with tp changing from 0 to 1.0.

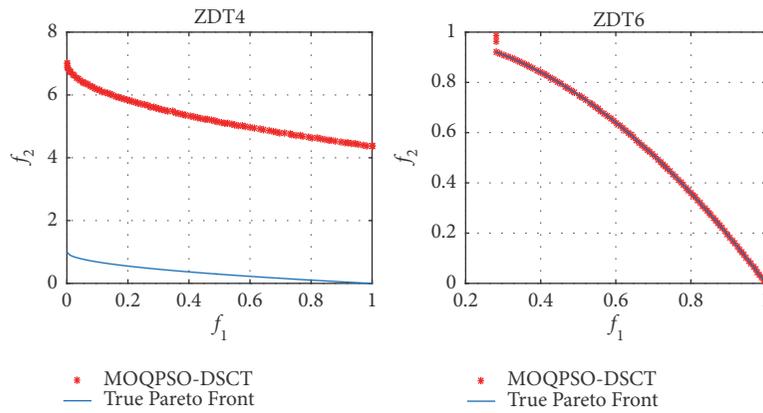


FIGURE 14: The PF obtained by MOQPSO-DSCT with $tp=0$ on ZDT4 and ZDT6.

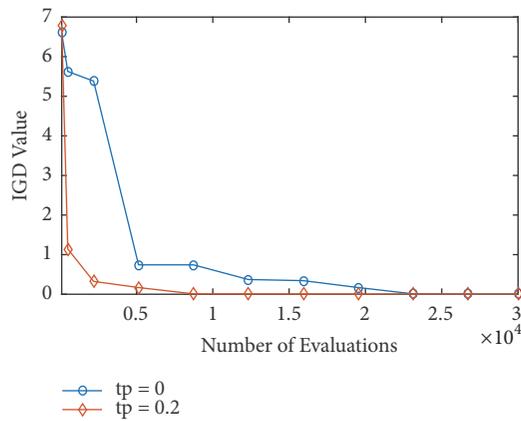


FIGURE 15: The convergence curves of MOQPSO-DSCT on DTLZ6 with $tp = 0$ and $tp = 0.2$.

TABLE 4: The SP values for MOQPSO-DSCT and MOQPSO-AG on eight test functions.

	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6		DTLZ2		DTLZ5		DTLZ6	
	Mean	Std														
MOQPSO-DSCT	3.3651e-3	6.36e-4	3.0114e-3	2.59e-4	5.2454e-3	6.51e-4	2.3574e-2	2.27e-2	4.1145e-3	1.69e-3	5.7050e-2	4.35e-3	5.0546e-3	6.96e-4	5.8231e-3	2.18e-4
MOQPSO-AG	1.0803e-2	1.12e-3	1.0634e-2	1.10e-3	1.1188e-2	1.79e-3	1.1946e-0	1.10e+0	9.3476e-3	2.42e-3	9.0943e-2	2.73e-2	1.5238e-2	1.67e-3	9.9771e-2	4.31e-2

14 and 15 that the value of tp has a great influence on ZDT4, ZDT6 and DTLZ6. MOQPSO-DSCT will not find the true PF of ZDT4 and some solutions cannot be pushed to the true PF of ZDT6 if tp is set to 0. The convergence speed of MOQPSO-DSCT on DTLZ6 can be greatly reduced without this mechanism. It can be seen from Figure 14 that the logarithm of IGD values of MOQPSO-DSCT on all test functions almost has no differences with tp changing from 0.2 to 1.0, but the computing resources will be severely consumed if tp is set to a large value. Therefore, the value of tp is set to 0.2 in this paper.

5. Conclusions

In this paper, an improved multiobjective quantum-behaved particle swarm optimization based on double search strategy and circular transposon mechanism (MOQPSO-DSCT) was proposed. In MOQPSO-DSCT, the double search strategy with a search probability parameter sp was used to update the position of all the particles. To help the particles escape from local optimal, opposition-based learning was introduced to construct a better attractor for each particle when its personal best position is equal to the global best position. The double search strategy can make the algorithm generate more solutions for MOPs. Then, the circular transposon mechanism was added to the external archive to exchange useful information between particles, which greatly improved the convergence accuracy of the algorithm and made the true PF be covered with the nondominated solutions evenly and dispersively. The experiment results have demonstrated the best performance of MOQPSO-DSCT in most MOPs when comparing with other multiobjective optimization algorithms. However, the experiment results on 3-objective problems (DTLZ function sets) are worse than the results on 2-objective problems (ZDT function sets) and 3-objective optimization problems are less involved in this paper. Therefore, our next improvements will be aimed at 3-objective problems. The future work will include two aspects. Firstly, some information hidden in the swarm will be considered to guide the particle to select the appropriate search pattern instead of introducing a new search probability parameter, because the new parameter will increase the complexity of designing the algorithm and adjusting the parameter. Secondly, this algorithm will be applied to solve some real-world problems on gene selection. For example, our proposed algorithm will be combined with other classical machine learning algorithms to deal with a multiobjective optimization model with two problems about how to obtain predictive genes with lower redundancy and higher prediction accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [Nos. 61572241 and 61271385], the National Key R&D Program of China [No. 2017YFC0806600], the Foundation of the Peak of Six Talents of Jiangsu Province [No. 2015-DZXX-024], the Fifth “333 High Level Talented Person Cultivating Project” of Jiangsu Province [No. (2016) III-0845], and the Research Innovation Program for College Graduates of Jiangsu Province [1291170030].

References

- [1] X. Li, J. Lai, and R. Tang, “A Hybrid Constraints Handling Strategy for Multiconstrained Multiobjective Optimization Problem of Microgrid Economical/Environmental Dispatch,” *Complexity*, vol. 2017, Article ID 6249432, 12 pages, 2017.
- [2] J. Lee, W. Seo, and D. W. Kim, “Effective Evolutionary Multilabel Feature Selection under a Budget Constraint,” *Complexity*, vol. 2018, Article ID 3241489, 2018.
- [3] M.-S. Casas-Ramírez, J.-F. Camacho-Vallejo, R. G. González-Ramírez, J.-A. Marmolejo-Saucedo, and J. Velarde-Cantú, “Optimizing a Biobjective Production-Distribution Planning Problem Using a GRASP,” *Complexity*, vol. 2018, Article ID 3418580, 13 pages, 2018.
- [4] N. Srinivas and K. Deb, “Multiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [5] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [6] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [7] C. R. Raquel and P. C. Naval Jr., “An effective use of crowding distance in multiobjective particle swarm optimization,” in *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, pp. 257–264, ACM, June 2005.
- [8] M. A. Abido, “Multiobjective particle swarm optimization with nondominated local and global sets,” *Natural Computing*, vol. 9, no. 3, pp. 747–766, 2010.
- [9] S. R. Margarita and C. Coello, “Improving PSO-based Multi-Objective Optimization Using Crowding, Mutation and ϵ -Dominance,” *Lecture Notes in Computer Science*, vol. 3410, pp. 505–519, 2005.
- [10] T. Cheng, M. Chen, P. J. Fleming, Z. Yang, and S. Gan, “A novel hybrid teaching learning based multi-objective particle swarm optimization,” *Neurocomputing*, vol. 222, pp. 11–25, 2017.
- [11] W. Yang, J. Guo, Y. Liu, and G. Zhai, “The design of contactors based on the niching multiobjective particle swarm optimization,” *Complexity*, vol. 2018, Article ID 9054623, 10 pages, 2018.
- [12] S. Z. Martínez and C. A. Coello Coello, “A multi-objective particle swarm optimizer based on decomposition,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, vol. 162, pp. 69–76, Dublin, Ireland, July 2011.
- [13] J. Sun, B. Feng, and W. Xu, “Particle swarm optimization with particles having quantum behavior,” in *Proceedings of the*

- Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, Portland, Ore, USA, June 2004.
- [14] Z. Shi and Q. W. Chen, “Multi-objective optimization algorithm based on quantum-behaved particle swarm and adaptive grid,” *Information and Control*, vol. 40, no. 2, pp. 214–220, 2011.
- [15] S.-H. Xu, X.-D. Mu, D. Chai, and P. Zhao, “Multi-objective quantum-behaved particle swarm optimization algorithm with double-potential well and share-learning,” *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 12, pp. 4921–4927, 2016.
- [16] H. Al-Baity, S. Meshoul, and A. Kaban, “Constrained multi-objective optimization using a quantum behaved particle swarm,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7665, no. 3, pp. 456–464, 2012.
- [17] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, NY, USA, 2007.
- [18] Y. H. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Anchorage, Alaska, USA, May 1998.
- [19] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [20] J. Sun, W. Xu, and B. Feng, “Adaptive parameter control for quantum-behaved particle swarm optimization on individual level,” in *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3049–3054, Waikoloa, HI, USA, 2006.
- [21] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, “Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection,” *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.
- [22] H. R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC '05)*, pp. 695–701, Vienna, Austria, November 2005.
- [23] C. Qu, S. Zhao, Y. Fu, and W. He, “Chicken swarm optimization based on elite opposition-based learning,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 2734362, 20 pages, 2017.
- [24] W. Wei, J. Zhou, F. Chen, and H. Yuan, “Constrained differential evolution using generalized opposition-based learning,” *Soft Computing*, vol. 20, no. 11, pp. 4413–4437, 2016.
- [25] P. Wang, W. Gao, X. Qian et al., “Hybrid fireworks explosion optimization algorithm using elite opposition-based learning,” *Journal of Computer Applications*, vol. 34, no. 10, pp. 2886–2890, 2014.
- [26] P. Shao, Z.-J. Wu, X.-Y. Zhou, and C.-S. Deng, “Improved particle swarm optimization algorithm based on opposite learning of refraction,” *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 43, no. 11, pp. 2137–2144, 2015.
- [27] B. McClintock, “The origin and behavior of mutable loci in maize,” *Proceeding of the National Academy of Science*, vol. 36, no. 6, pp. 344–355, 1950.
- [28] B. McClintock, “Chromosome organization and genic expression,” *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 16, pp. 13–47, 1951.
- [29] T. M. Chan, K. F. Man, K.-S. Tang, and S. Kwong, “A jumping gene paradigm for evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 143–159, 2008.
- [30] H.-B. Ouyang, L.-Q. Gao, S. Li, and X.-Y. Kong, “Improved global-best-guided particle swarm optimization with learning operation for global optimization problems,” *Applied Soft Computing*, vol. 52, pp. 987–1008, 2017.
- [31] Q. Lin, J. Li, Z. Du, J. Chen, and Z. Ming, “A novel multi-objective particle swarm optimization with multiple search strategies,” *European Journal of Operational Research*, vol. 247, no. 3, pp. 732–744, 2015.
- [32] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 825–830, May 2002.
- [34] C. Dai, Y. Wang, and M. Ye, “A new multi-objective particle swarm optimization algorithm based on decomposition,” *Information Sciences*, vol. 325, pp. 541–557, 2015.
- [35] A. J. Nebro, J. J. Durillo, G. Nieto, C. A. C. Coello, F. Luna, and E. Alba, “SMPSO: a new pso-based metaheuristic for multi-objective optimization,” in *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM '09)*, pp. 66–73, Nashville, Tenn, USA, April 2009.
- [36] 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.
- [37] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [38] J. Xiao, J.-J. Li, X.-X. Hong et al., “An Improved MOEA/D Based on Reference Distance for Software Project Portfolio Optimization,” *Complexity*, vol. 2018, Article ID 3051854, 16 pages, 2018.

