DMO-QPSO: A multi-objective quantum-behaved particle swarm optimization algorithm based on decomposition with diversity control

DOI 10.3390/math9161959
ISSN 2227-7390

Publisher: MDPI

mathematics

MDPI

Article

DMO-QPSO: A Multi-Objective Quantum-Behaved Particle Swarm Optimization Algorithm Based on Decomposition with Diversity Control

Qi You [1], Jun Sun [1,*], Feng Pan [1], Vasile Palade [2] and Bilal Ahmad [1]

[1] Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, Wuxi 214122, China; 7171905005@stu.jiangnan.edu.cn (Q.Y.); pan_feng_63@jiangnan.edu.cn (F.P.); bilalahmad@stu.jiangnan.edu.cn (B.A.)

[2] Faculty of Engineering and Computing, Coventry University, Coventry CV1 5FB, UK; ab5839@coventry.ac.uk

* Correspondence: junsun@jiangnan.edu.cn

Abstract: The decomposition-based multi-objective evolutionary algorithm (MOEA/D) has shown remarkable effectiveness in solving multi-objective problems (MOPs). In this paper, we integrate the quantum-behaved particle swarm optimization (QPSO) algorithm with the MOEA/D framework in order to make the QPSO be able to solve MOPs effectively, with the advantage of the QPSO being fully used. We also employ a diversity controlling mechanism to avoid the premature convergence especially at the later stage of the search process, and thus further improve the performance of our proposed algorithm. In addition, we introduce a number of nondominated solutions to generate the global best for guiding other particles in the swarm. Experiments are conducted to compare the proposed algorithm, DMO-QPSO, with four multi-objective particle swarm optimization algorithms and one multi-objective evolutionary algorithm on 15 test functions, including both bi-objective and tri-objective problems. The results show that the performance of the proposed DMO-QPSO is better than other five algorithms in solving most of these test problems. Moreover, we further study the impact of two different decomposition approaches, i.e., the penalty-based boundary intersection (PBI) and Tchebycheff (TCH) approaches, as well as the polynomial mutation operator on the algorithmic performance of DMO-QPSO.

Keywords: decomposition; multi-objective optimization; quantum-behaved particle swarm optimization; premature convergence; diversity control

check for updates

Citation: You, Q.; Sun, J.; Pan, F.; Palade, V.; Ahmad, B. DMO-QPSO: A Multi-Objective Quantum-Behaved Particle Swarm Optimization Algorithm Based on Decomposition with Diversity Control. *Mathematics* **2021**, *9*, 1959. https://doi.org/10.3390/math9161959

Academic Editor: Andreas (Andrzej) Wichert

Received: 17 May 2021
Accepted: 12 August 2021
Published: 16 August 2021

1. Introduction

The particle swarm optimization (PSO) algorithm, originally proposed by Kennedy and Eberhart in 1995, is a population-based metaheuristic that imitates the social behavior of birds flocking [1]. In PSO, each particle is treated as a potential solution, and all particles follow their own experiences and the current optimal particle to fly through the solution space. As it requires fewer parameters to adjust and can be easily implemented, PSO has been rapidly developed in solving real-world optimization problems, including circuit design [2], job scheduling [3], data mining [4], path planning [5,6] and protein-ligand docking [7]. In 1999, Moore and Chapman extended PSO to solve multi-objective problems (MOPs) for the first time in [8]. Since then, a great interest has been aroused among researchers from different communities to tackle MOPs by using PSO. For example, Coello and Lechuga [9] introduced a proposal for multi-objective PSO, noted as MOPSO, which determines particles' flight directions by using the concept of Pareto dominance and adopts a global repository to store previously found nondominated solutions. Later in 2004, Coello et al. [10] presented an enhanced version of MOPSO which employs a mutation operator and a constraint-handling mechanism to improve the algorithmic performance of the original MOPSO. Raquel and Prospero [11] proposed the MOPSO-CD algorithm that selects the global best and updates the external archive of nondominated solutions by calculating the crowding distance of particles. In 2008, Peng and Zhang developed a new MOPSO algorithm adopting a decomposition approach, called MOPSO/D [12]. It is based on a framework, named as MOEA/D [13], which converts an MOP into a number of single-objective optimization sub-problems and then simultaneously solves all these sub-problems. In MOPSO/D, the particle's global best is defined by the solutions located within a certain neighborhood. Moubayed et al. [14] proposed a novel smart MOPSO based on decomposition (SDMOPSO) that realizes the information exchange between neighboring particles with fewer objective function evaluations and stores the leaders of the whole particle swarm using a crowding archive. dMOPSO, proposed by Martinez and Coello [15], selects the global best from a set of solutions according to the decomposition approach and thus update each particle's position. Moubayed et al. [16] realized a MOPSO, called D2MOPSO, which hybrids the approach of dominance and decomposition and introduces an archiving technique using crowding distance. There have also been some other MOPSOs proposed in recent years that have proved to be effective in solving complex MOPs, such as MPSO/D [17], MMOPSO [18], AgMOPSO [19], CMOPSO [20] and CMaPSO [21].

The quantum-behaved PSO (QPSO), proposed by Sun et al. [22], is a variant of PSO inspired by quantum mechanics and the trajectory analysis of PSO. The trajectory analysis clarified the idea that each particle in PSO is in a bound state. Specifically, each particle in PSO oscillates around and converges to its local attractor [23]. In QPSO, the particle is assumed to have quantum behavior and further assumed to be attracted by a quantum delta potential well centered on its local attractor. Additionally, the concept of the mean best position was defined and employed in this algorithm to update particles' positions. In the terms of the update equation, which is different from that of PSO, on the one hand, QPSO has no velocity vectors for each particle to update, on the other hand, QPSO requires fewer parameters to adjust [24]. Due to these advantages of QPSO, we incorporate it into the original MOEA/D framework for the purpose of obtaining a more effective algorithm for solving MOPs than other decomposition-based MOPSOs that uses the canonical PSO.

QPSO and other PSO variants generally have fast convergence speed due to more information exchange among particles. This is why such kinds of algorithms are more efficient to solve optimization problems than other population-based random search algorithms. Fast convergence speed means rapid diversity decline, which is desirable for the algorithm to find satisfying solutions quickly during the early stage of the search process. However, rapid diversity decline during the later stage of the search process results in aggregation of particles around the global best position and in turn the stagnation of the whole particle swarm (i.e., premature convergence).

Diversity maintenance is also essential when extending PSO to solve MOPs. During the past decade, researchers have done a lot of work on developing novel techniques to maintain diversity in their MOPSOs. For example, Qiu et al. [25] introduced a novel global best selection method, which is based on proportional distribution and K-means algorithm, to make particles converge to the Pareto front in a fast speed while maintaining diversity. Cheng et al. [26] presented the IMOPSO-PS algorithm, in which the preference strategy is applied for optimal distributed generation (DG) integration into the distribution system. This algorithm uses a circular nondominated selection of particles from one iteration to the next and performs mutation on particles to enhance the swarm diversity during the search process.

In this paper, we propose a multi-objective quantum-behaved particle swarm optimization algorithm based on decomposition, named DMO-QPSO, which integrates the QPSO with the original MOEA/D framework and uses a strategy of diversity control. As in the literature [12,13], a neighborhood relationship is defined according to distances between the weight vectors of different sub-problems. Each sub-problem is solved utilizing the information only from its neighboring sub-problems. However, with the increasing number of iterations, the current best solutions to the neighbors of a sub-problem may get close to each other. This may result in a diversity loss of the new population produced in the next iteration, particularly at the later stage of the search process. Therefore, in DMO-QPSO, we do not adopt the neighborhood relationship described in the framework of MOEA/D and MOPSO/D. Meanwhile, we introduce a two-phased diversity controlling mechanism to make particles alternate between attraction and explosion states according to the swarm diversity. Particles move through the search space in the phase of attraction unless the swarm diversity declines to a threshold value that triggers the phase of explosion. Additionally, unlike MOPSO/D in which the global best is updated according to a decomposition approach, the proposed DMO-QPSO uses a vector set to store a pre-defined number of nondominated solutions and then randomly picks one as the current global best. All solutions in this vector set would have a chance to guide the movement of the whole particle swarm. The penalty-based boundary intersection (PBI) approach [27] is used in the algorithm owing to its advantage over other decomposition methods including the weighted sum (WS) and the Tchebycheff (TCH) [13].

The rest of this paper is organized as follows. Some preliminaries of MOP, PSO, QPSO and the framework of MOEA/D are given in Section 2. Section 3 describes the procedure of our proposed DMO-QPSO algorithm in detail. Section 4 presents the experimental results and analysis. Some further discussion on DMO-QPSO are introduced in Section 5. Finally, the paper is concluded in the last section.

2. Preliminaries

In this section, we first state the definition of MOPs and then describe the basic principles of the canonical PSO and QPSO. After that, some of the most commonly used decomposition methods and the original MOEA/D framework are presented.

2.1. Multi-Objective Optimization

A multi-objective optimization problem (MOP) can be stated as follows.

$$\begin{array}{c} \text{minimize } \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}),\ \ldots,\ f_m(\boldsymbol{x})) \\ s.t.\ \boldsymbol{x} \in \Omega \end{array} \tag{1}$$

where $\boldsymbol{x}$ is the decision variable vector, Ω is the decision (variable) space, and m is the number of the real-valued objective functions. $\boldsymbol{F}:\ \Omega \to R^m$ is the objective function vector, where R^m is the objective space. The objectives in an MOP are mutually conflicting, so no one solution can minimize all the objectives at the same time. Improvement of one objective may lead to deterioration of another. In this situation, the Pareto optimal solutions become the best tradeoffs among different objectives. Therefore, most of multi-objective optimization algorithms are designed to find a finite number of Pareto optimal solutions to approximate the Pareto front (PF), which could be good representatives of the whole PF [28–31]. In order to better understand the concept of Pareto optimality [32], some definitions are provided as follows.

Definition 1. *Let $\boldsymbol{x} = (x_1,\ \ldots,\ x_m)$, $\boldsymbol{y} = (y_1,\ \ldots,\ y_m) \in R^m$. For all $i = 1,\ \ldots,\ m$ and $\boldsymbol{x} \neq \boldsymbol{y}$, if and only if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$, then $\boldsymbol{x}$ dominates $\boldsymbol{y}$, denoted as $\boldsymbol{x} \prec \boldsymbol{y}$.*

Definition 2. *Let $\boldsymbol{x}^* \in \Omega$. If no solution $\boldsymbol{x}$ exists in Ω such that $\boldsymbol{F}(\boldsymbol{x})$ dominates $\boldsymbol{F}(\boldsymbol{x}^*)$, then $\boldsymbol{x}^*$ is a Pareto optimal solution to MOP in Equation (1), and $\boldsymbol{F}(\boldsymbol{x}^*)$ is a Pareto optimal (objective) vector. The set of all the Pareto optimal solutions is called the Pareto set (PS), denoted by $PS = \{\boldsymbol{x} \in \Omega | \boldsymbol{x}\ \text{is a Pareto optimal solution}\}$. The set of all the Pareto optimal (objective) vectors is called the Pareto front (PF), denoted by $PF = \{\boldsymbol{F}(\boldsymbol{x}) | \boldsymbol{x} \in PS\}$.*

2.2. Particle Swarm Optimization

In the canonical PSO algorithm with N particles, each particle i $(i = 1,\ \ldots,\ N)$ has a position vector $\boldsymbol{X}_i = (X_{i,1},\ \ldots, X_{i,D})$ and a velocity vector $\boldsymbol{V}_i = (V_{i,1},\ \ldots, V_{i,D})$. D is the dimension of the search space. During each iteration t, particle i in the swarm is updated according to its personal best (*pbest*) position $\boldsymbol{P}_i = (P_{i,1}, \ldots, P_{i,D})$ and the global best (*gbest*) position $\boldsymbol{P}_g = \left(P_{g,1}, \ldots, P_{g,D}\right)$ found by the whole swarm. The update strategies are presented as follows.

$$V_{i,j}(t+1) = wV_{i,j}(t) + c_1 r_1 \left(P_{i,j}(t) - X_{i,j}(t)\right) + c_2 r_2 \left(P_{g,j}(t) - X_{i,j}(t)\right) \tag{2}$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1) \tag{3}$$

for $i = 1, 2, \cdots, N; j = 1, 2, \cdots, D$, where w is the inertia weight, c_1, c_2 are the learning factors, r_1, r_2 are two random variables uniformly distributed on (0, 1).

2.3. Quantum-Behaved Particle Swarm Optimization

In the quantum-behaved PSO (QPSO) algorithm with N particles, each particle i $(i = 1, \ldots, N)$ has a position vector $\boldsymbol{X}_i = (X_{i,1}, \ldots, X_{i,D})$ and a personal best (*pbest*) position $\boldsymbol{P}_i = (P_{i,1}, \ldots, P_{i,D})$. D is the dimension of the search space. During each iteration t, particle i in the swarm is updated as follows.

$$X_{i,j}(t+1) = q_{i,j}(t) \pm \alpha \left|C_j(t) - X_{i,j}(t)\right| \cdot \ln\left(1/u_{i,j}(\mathrm{t})\right) \tag{4}$$

for $j = 1, 2, \cdots, D$, where $u_{i,j}$ is a random number distributed on (0, 1) uniformly, $\boldsymbol{q}_i = \left(q_{i,1}, \ldots, q_{i,D}\right)$ is the local attractor of particle i, calculated by

$$\boldsymbol{q}_i = \varphi \cdot \boldsymbol{P}_i + (1 - \varphi) \cdot \boldsymbol{P}_g, \tag{5}$$

φ is a random number distributed on (0, 1) uniformly and $\boldsymbol{P}_g = \left(P_{g,1}, \ldots, P_{g,D}\right)$ is the global best (*gbest*) position found by particles during the search process. The contraction-expansion coefficient α was designed to control the convergence speed of the QPSO algorithm. C is the mean of the personal best positions of all the particles, namely, the *mbest* position, and it can be calculated as below.

$$\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{P}_i = \left(\frac{1}{N}\sum_{i=1}^{N} P_{i,1}, \frac{1}{N}\sum_{i=1}^{N} P_{i,2}, \ldots, \frac{1}{N}\sum_{i=1}^{N} P_{i,D}\right) \tag{6}$$

2.4. The Decomposition Approaches

In the state-of-the-art multi-objective optimization algorithms based on decomposition, the most commonly used decomposition approaches are the Tchebycheff (TCH), the weighted sum (WS), and the penalty-based boundary intersection (PBI) approaches [13,28]. These methods are supposed to decompose an MOP into a finite group of single-objective optimization sub-problems, so that a certain algorithm can solve these sub-problems effectively and efficiently. Let $\boldsymbol{\lambda}^i = (\lambda_1^i, \ldots, \lambda_m^i)^T$ be a weight vector for the ith sub-problem $(i = 1, \ldots, N)$, satisfying $\sum_{j=1}^{m} \lambda_j^i = 1$ and $\lambda_j^i > 0$ for all $j = 1, 2, \ldots, m$; and $\boldsymbol{z}^* = (z_1^*, \ldots, z_m^*)^T$ be a reference point. Below are the definitions of TCH and PBI approaches which will be used later in this paper.

- Tchebycheff (TCH) approach:

  In the TCH approach, the sub-problem i is defined as

$$\begin{aligned} \text{minimize } g^{tch}\left(\boldsymbol{x} \middle| \boldsymbol{\lambda}^i, \boldsymbol{z}^*\right) &= \max_{1 \le j \le m} \left\{\lambda_j^i \left|f_j(x) - z_j^*\right|\right\} \\ \text{s.t. } \boldsymbol{x} \in \Omega \end{aligned} \tag{7}$$

- Penalty-based boundary intersection (PBI) approach:

  In PBI approach, the sub-problem i is defined as

$$\begin{array}{c} \text{minimize } g^{pbi}\left(\boldsymbol{x}\middle|\boldsymbol{\lambda}^{i},\boldsymbol{z}^{*}\right)=d_1+\theta d_2 \\ \text{where } d_1=\left\|\left(\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{z}^{*}\right)^{T}\boldsymbol{\lambda}^{i}\right\|/\left\|\boldsymbol{\lambda}^{i}\right\| \\ d_2=\left\|\boldsymbol{F}(\boldsymbol{x})-\boldsymbol{z}^{*}-\left(d_1^{i}/\left\|\boldsymbol{\lambda}^{i}\right\|\right)\boldsymbol{\lambda}^{i}\right\| \\ \text{s.t. } \boldsymbol{x}\in\Omega \end{array} \tag{8}$$

where $\|\cdot\|$ means the $L_2 - norm$, and $\theta > 0$ is a penalty parameter.

2.5. MOEA/D

MOEA/D divides an MOP into N single-objective optimization sub-problems and attempts to simultaneously optimize all these sub-problems rather than directly solving the MOP. These sub-problems are linked together by their neighborhoods. The neighborhood of sub-problem i is defined as the sub-problems whose weight vectors are the T closest ones to its weight vector $\boldsymbol{\lambda}^{i}$ and thus the neighborhood size of sub-problem i is T.

The MOEA/D algorithm maintains a population of N solutions $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N \in \Omega$, where $\boldsymbol{x}^i$ $(i = 1, \ldots, N)$ is a feasible solution to the sub-problem i. $\boldsymbol{FV}^i$ is the F-value (i.e., the fitness value) of $\boldsymbol{x}^i$, that is, $\boldsymbol{FV}^i = \boldsymbol{F}(\boldsymbol{x}^i)$. $\boldsymbol{z} = (z_1, \ldots, z_m)^T$ is a reference point and z_j $(j = 1, \ldots, m)$ is the minimal value of objective f_j found so far. EP is an external population used to tore non-dominated solutions found during the search process. The main framework of MOEA/D is described in Algorithm 1.

Algorithm 1 Framework of MOEA/D

```
Input: The number of sub-problems, i.e., the population size, $N$; The set of weight vectors, $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N$; The neighborhood size, $T$;
Output: $EP$;
$EP = \varnothing$;
Calculate the Euclidean distances between any two weight vectors;
Successively select $T$ weight vectors which are the closest to the weight vector $\lambda^i$ and store the indexes of these $T$ weight vectors in $B(i) = \{i_1, \ldots, i_T\}$, $i = 1, \ldots, N$;
Generate an initial population randomly;
Evaluate $\boldsymbol{FV}^i, i = 1, \ldots, N$;
Initialize $\boldsymbol{z} = (z_1, \ldots, z_m)^T$;
while termination criterion is not fulfilled do
  for $i = 1, \ldots, N$ do
    Select two indexes $k, l$ randomly from $B(i)$;
    Use the genetic operators to produce a new solution y from $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$;
    Repair $\boldsymbol{y}$;
    for $j = 1, \ldots, m$ do
      if $z_j > f_j(\boldsymbol{y})$ then
        $z_j = f_j(\boldsymbol{y})$;
      end if
    end for
  for each $j \in B(i)$ do
    if $\mathrm{g}\left(\boldsymbol{y}\middle|\lambda^i, \boldsymbol{z}\right) \le \mathrm{g}\left(\boldsymbol{x}^j\middle|\lambda^i, \boldsymbol{z}\right)$ then
      $\boldsymbol{x}^j = \boldsymbol{y}$; $\boldsymbol{FV}^i = \boldsymbol{F}(\boldsymbol{y})$;
    end if
  end for
end for
Update $EP$;
end while
```

3. The Proposed DMO-QPSO

In this section, we propose an improved multi-objective quantum-behaved particle swarm optimization algorithm based on decomposition, named as DMO-QPSO, which integrates the QPSO algorithm with the MOEA/D framework and adopts a mechanism to control the swarm diversity during the search process so as to avoid premature convergence and escape the local optimal area with a higher probability.

At the beginning of the proposed algorithm DMO-QPSO, we need to define a set of well-distributed weight vectors and then use a certain approach to decompose the original MOP into a group of single objective sub-problems. More precisely, let $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N$ to be the weight vectors, and the PBI approach is employed in this paper owing to its advantage over other decomposition approaches.

In DMO-QPSO, the swarm $\mathcal{P}$ with N particles is randomly initialized. Each particle i has a position vector $\boldsymbol{X}_i$ and a personal best position $\boldsymbol{P}_i$. $\boldsymbol{P}_i$ is initially set to be equal to $\boldsymbol{X}_i$. Then the mean best position of all particles can be easily obtained according to Equation (6). The global best position $\boldsymbol{P}_g$ is produced in a natural way according to the Pareto dominance relationship among different personal best positions. More specifically, we firstly define a vector set GS, and the size of GS is pre-set, denoted as nGS. Then, the fast nondominated sorting approach [33] is applied to sort the set of all the personal best positions and GS. The lower nondomination rank of a solution is, the better it is. Therefore, we only select the ones in the lower nondomination ranks and then store them in GS. All of the solutions in GS are regarded as candidates for the global best employed in the next iteration for updating particles' positions.

It should be noted that the neighborhood in the original MOEA/D framework is formed according to distances between the weight vectors of different sub-problems. That is to say, the neighborhood of a sub-problem includes all sub-problems with the closest weight vectors. Hence, on the basis of this definition, solutions to neighboring sub-problems would be close in the decision space. It may enable the algorithm to converge faster at the early stage but brings the risk of diversity loss and premature convergence at the later stage. For this reason, we do not adopt in DMO-QPSO the neighborhood relationship stated in the original MOEA/D framework.

Furthermore, we measure the swarm diversity during the search process and make the swarm alternate between two phases, i.e., attraction and explosion, according to its diversity. At each iteration, the diversity of the particle swarm is calculated as below.

$$diversity(\mathcal{P}) = \frac{1}{|\mathcal{P}|} \cdot \frac{1}{|A|} \cdot \sum_{i=1}^{|\mathcal{P}|} \sqrt{\sum_{j=1}^{D} \left(X_{i,j} - \overline{X_j}\right)^2} \tag{9}$$

where D is the dimensionality of the problem, $|A|$ is the length of longest diagonal in the search space, $\mathcal{P}$ is the particle swarm, $|\mathcal{P}| = N$ is the population size, $X_{i,j}$ is the jth value of particle i and $\overline{X_j}$ is the jth value of the average point. According to the literature [34,35], the particle converges when the contraction-expansion coefficient α is less than 1.778 and otherwise it diverges. Therefore, we set a threshold, denoted as $dlow$, to the swarm diversity. When the diversity drops below $dlow$ (i.e., in the explosion phase), the value of α will be reset to a constant α_0, larger than 1.778, to make particles diverge and thus increase the swarm diversity. Otherwise, α linearly decreases between the predefined interval [a, b] (i.e., in the attraction phase).

Like MOEA/D, we also use an external population EP in the DMO-QPSO to store the nondominated solutions found during the search process. In each iteration step, we check the Pareto dominance relationship between the new generated solutions and the solutions in EP. Solutions in EP dominated by a new generated solution will be removed from EP and this new generated solution will be added to EP if no one in EP dominates it. The main process of the DMO-QPSO algorithm is presented in Algorithm 2.

Algorithm 2 DMO-QPSO

```
Input: The number of sub-problems, i.e., the population size, $N$; The set of weight vectors, $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N$; The maximal number of iterations, $MaxIter$;
Output: $EP$;
$EP = \varnothing$, $GD = \varnothing$;
for $i = 1, \cdots, N$ do
    Randomly initialize the position vector $\boldsymbol{X}_i$ of particle $i$;
    Set the personal best position $\boldsymbol{P}_i$ of particle $i$ as $\boldsymbol{P}_i = \boldsymbol{X}_i$;
    Evaluate the fitness value $\boldsymbol{F}(\boldsymbol{X}_i)$;
end for
Update $GS$;
Initialize $\boldsymbol{z} = (z_1, \ldots, z_m)^T$;
for $t = 1, \cdots, MaxIter$ do
    Compute the mean best position $C$ of all the particles according to Equation (6);
    Measure $diversity(\mathcal{P})$;
    if $diversity(\mathcal{P}) < dlow$ do
        $\alpha = \alpha_0$;
    else
        Set $\alpha$ linearly decreasing between the interval $[a, b]$;
    end if
    for $i = 1, \cdots, N$ do
        Update the position vector $\boldsymbol{X}_i(t+1)$ using Equation (4);
        Repair $\boldsymbol{X}_i(t+1)$;
        Evaluate $\boldsymbol{F}(\boldsymbol{X}_i(t+1))$;
        for $j = 1, \cdots, m$ do
            if $z_i > f_j(\boldsymbol{X}_i(t+1))$ then
                $z_i = f_j(\boldsymbol{X}_i(t+1))$;
            end if
        end for
        if $g(\boldsymbol{X}_i(t+1) \big| \boldsymbol{\lambda}^i, \boldsymbol{z}) \leq g(\boldsymbol{P}_i(t) \big| \boldsymbol{\lambda}^i, \boldsymbol{z})$ then
            $\boldsymbol{P}_i(t+1) = \boldsymbol{X}_i(t+1)$;
        end if
    end for
    Update $GS$;
    Update $EP$;
end for
```

4. Experimental Studies

This section presents the experiments conducted to investigate the performance of our proposed DMO-QPSO algorithm. Firstly, we introduce a set of MOPs used as benchmark functions. Next, the parameter settings for different algorithms and two performance metrics are described in detail. Finally, the comparison experiments and results analysis are presented. More precisely, we compared the DMO-QPSO with two recently proposed multi-objective PSOs (i.e., MMOPSO and CMOPSO) and other three multi-objective optimization algorithms, namely, MOPSO, MOPSO/D and MOEA/D-DE [36]. The PBI approach is used in four decomposition-based algorithms (i.e., DMO-QPSO, MOPSO/D, MOEA/D-DE and MMOPSO).

4.1. Test Functions

We selected 15 test functions whose PFs have different characteristics including concavity, convexity, multi-frontality and disconnections. Twelve of these test functions are bi-objective (i.e., F1, F2, F3, F4, F5, F7, F8, F9 from the F test set [36], UF4, UF5, UF6, UF7 from the UF test set [37]) and the rest of them are tri-objective (i.e., F6 from the F test set and UF9, UF10 from the UF test set). As shown in references [36,37], the F test set and the UF test set are two sets of test instances for facilitating the study of the ability of MOEAs to solve problems with complicated PS shapes. Besides, we used 30 decision variables for the

UF test set, problems from F1 to F5, and F9. Problems from F6 to F8 were tested by using 10 decision variables.

4.2. Parameter Setting

The setting of weight vectors $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N$ is decided by an integer H [36]. More precisely, each individual weight in $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \ldots, \boldsymbol{\lambda}^N$ takes a value from $\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}$. Therefore, the population size N can be presented by $N = C_{H+m-1}^{m-1}$, where m is the number of objectives. H was 299 for the bi-objective test functions and 33 for the tri-objective ones. Consequently, the population size N was 300 for the bi-objective test functions and 595 for the tri-objective ones. The maximal number of iterations $MaxIter$ was 500 and each algorithm runs 30 independent times for each test function. The size of external population nEP was set to be 100. Besides, the penalty factor θ in PBI was 5.0.

The polynomial mutation [29] was employed in MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO and CMOPSO. Two parameters η_m, p_m in this mutation operator were 20 and $1/D$, respectively.

For DMO-QPSO, the size of GS was 10, and the contraction-expansion coefficient α in the attraction phase varied linearly from 1.0 to 0.5. The value of the lower bound of diversity $dlow$ was set to be 0.05. When the diversity drops below $dlow$, we set the parameter $\alpha = \alpha_0 = 2.0$.

The details are listed in Table 1.

Table 1. Parameters setting for different algorithms.

Algorithms	Parameters Setting
MOPSO	$\omega = 0.5,\ c_1 = c_2 = 1.0,\ \eta_m = 20,\ p_m = \frac{1}{D}$, 30 divisions for the adaptive grid
MOPSO/D	$\omega = 0.4,\ c_1 = c_2 = 2.0, \eta_m = 20,\ p_m = 0.05,\ T = 20$
MOEA/D-DE	$\eta_m = 20,\ p_m = \frac{1}{D},\ T = 20,\ CR = 1.0,\ F = 0.5$
MMOPSO	$\omega \in [0.1, 0.5],\ c_1, c_2 \in [1.5, 2.0], \eta_m = 20,\ p_m = \frac{1}{D},\ \eta_c = 20,\ p_c = 0.9, \delta = 0.9$
CMOPSO	$\eta_m = 20,\ p_m = \frac{1}{D},\ \gamma = 10$
DMO-QPSO	$\alpha = 1.0- > 0.5,\ \alpha_0 = 2.0,\ dlow = 0.05,\ nGS = 10$

4.3. Performance Metrics

In our experiments, the following performance metrics were used.

- The inverted generational distance (IGD) [36]: It is proposed as a method of estimating the distance between the elements in a set of nondominated vectors and those in the Pareto optimal set, and can be stated as:

$$\mathrm{IGD}(P^*,\ P) = \textstyle\sum_{v \in P^*} d(v,P) / |P^*| \tag{10}$$

where P^* is a number of points which are evenly distributed in the objective space along the PF, and P is an approximation to the PF. $d(v,\ P)$ is the minimal Euclidean distance between v and the points in P, and $|P^*|$ is the size of the set P^*. P must be as close as possible to the PF of a certain test problem and cannot miss any part of the whole PF, so as to obtain a low value of $\mathrm{IGD}(P^*,\ P)$. It can reflect both the diversity and the convergence of the obtained solutions to the real PF.

- Coverage (C-metric) [13]: It can be stated as:

$$C(A, B) = |\{u \in B | \exists v \in A : v \text{ dominants } u\}| / |B| \tag{11}$$

where $A,\ B$ are two approximations to the real PF of an MOP, $C(A, B)$ is defined as the proportion of the solutions in B that are dominated by at least one solution in A.

4.4. Results and Discussion

Tables 2 and 3 present the average, minimum and standard deviation (SD) of the IGD values of 30 final populations on different test functions that were produced by MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO, CMOPSO and DMO-QPSO. It is clear that our proposed algorithm, DMO-QPSO, performed better than the other five algorithms on most of the test problems. It yielded the best mean IGD values on all the problems except on F7, F8, UF4, UF6 and UF10. According to Tables 2 and 3, both the mean and minimal IGD values on F7, F8, UF4 and UF10 obtained by DMO-QPSO are worse than those obtained by MOEA/D-DE. It should be noted that the performance of DMO-QPSO is still acceptable on UF6. The mean IGD value of DMO-QPSO on UF6 are slightly worse than that of MMOPSO, but DMO-QPSO got the lowest minimal IGD value among all of these algorithms. Besides, it is obvious that MOPSO/D performed the worst on almost all of the test problems except on F6 and UF4.

Table 2. The mean, minimum and standard deviation of IGD values on F problems, where the best value for each test case is highlighted with a bold background.

IGD	F1	F2	F3	F4	F5	F6	F7	F8	F9	Total	Final Rank
MOPSO	0.0816	0.4617	0.1897	0.1832	0.1401	0.6410	0.5036	0.3653	0.4299		
	0.0661	0.3105	0.1378	0.1442	0.1118	0.4491	0.3329	0.3041	0.1763		
	0.0108	0.1070	0.0412	0.0226	0.0213	0.0533	0.2048	0.0490	0.1221		
	3−	5−	3−	4−	2−	4−	3−	2+	4−	30	4
MOPSO/D	0.2313	1.0223	0.5754	0.5748	0.5558	0.9781	1.5032	1.1271	1.0358		
	0.2159	0.8869	0.5013	0.5030	0.4757	0.6580	1.0491	0.8681	0.8319		
	0.0084	0.0757	0.0449	0.0331	0.0351	0.1810	0.2076	0.1334	0.0964		
	6−	6−	6−	6−	6−	5−	6−	6−	6−	53	6
MOEA/D-DE	0.1180	0.2394	0.1987	0.1702	0.1405	1.5602	**0.2139**	**0.3046**	0.1982		
	0.0820	0.1492	0.1464	0.1159	0.0891	1.4111	**0.0792**	**0.2053**	0.0971		
	0.0175	0.0775	0.0357	0.0294	0.0324	0.0577	0.0734	0.0645	0.0498		
	4−	3−	4−	3−	3−	6−	1+	1+	2−	27	3
MMOPSO	0.0481	0.1615	0.1761	0.1626	0.2365	0.3007	0.5140	0.4509	0.2285		
	0.0369	0.1156	0.0815	0.1072	0.0942	**0.1632**	0.3255	0.2344	0.1135		
	0.0102	0.0625	0.0784	0.0233	0.0688	0.0856	0.1059	0.1238	0.1574		
	2≈	2−	2−	2−	4−	2−	4−	3≈	3−	24	2
CMOPSO	0.1769	0.4350	0.3142	0.3467	0.2820	0.4895	0.6141	0.6237	0.4303		
	0.1540	0.3675	0.2762	0.2876	0.2626	0.3850	0.5204	0.5402	0.3003		
	0.0087	0.0365	0.0207	0.0269	0.0133	0.0643	0.0635	0.0542	0.0426		
	5−	4−	5−	5−	5−	3−	5−	5−	5−	42	5
DMO-QPSO	**0.0404**	**0.1023**	**0.0884**	**0.0898**	**0.0696**	**0.2608**	0.4283	0.4676	**0.0938**		
	0.0359	**0.0962**	**0.0787**	**0.0810**	**0.0635**	0.1634	0.2210	0.2686	**0.0803**		
	0.0030	0.0036	0.0071	0.0088	0.0035	0.0248	0.1160	0.0907	0.0108		
	1	1	1	1	1	1	2	4	1	13	1

$+, -, \approx$ denote that the performance of the corresponding algorithm is significantly better than, worse than, and similar to DMO-QPSO respectively by Wilcoxon rank sum test with $\alpha = 0.05$.

Table 3. The mean, minimum and standard deviation of IGD values on UF problems, where the best value for each test case is highlighted with a bold background.

IGD	UF4	UF5	UF6	UF7	UF9	UF10	Total	Final Rank
MOPSO	0.1640	2.8297	2.8264	0.5303	0.6590	1.6258		
	0.1483	2.2756	2.3398	0.2036	0.5393	1.0761		
	0.0057	0.3514	0.3608	0.1343	0.0644	0.3764		
	4−	5−	5−	4−	4−	2+	24	4
MOPSO/D	0.1672	4.3807	4.3793	1.0652	2.5708	12.7750		
	0.1592	3.1642	3.1732	0.7175	2.0591	10.8881		
	0.0035	0.3636	0.3645	0.1386	0.2999	0.9072		
	5−	6−	6−	6−	6−	6−	35	6
MOEA/D-DE	**0.0994**	1.7532	1.7616	0.1817	0.5971	**1.4303**		
	0.0879	0.9801	0.9003	0.0883	0.4705	**1.0272**		
	0.0052	0.5473	0.5481	0.0717	0.0640	0.3025		
	1+	3−	3−	2−	3−	1+	13	2
MMOPSO	0.7820	0.9039	**0.8107**	0.2905	0.4661	2.3204		
	0.5234	0.6526	0.5599	0.1019	0.3527	1.7783		
	0.0993	0.1261	0.1336	0.1706	0.0500	0.4342		
	6−	2−	1≈	3−	2−	4−	18	3
CMOPSO	0.1582	2.7759	2.7580	0.5738	1.4510	8.1102		
	0.1491	2.2412	2.1983	0.5187	0.9924	6.0312		
	0.0031	0.1728	0.1794	0.0343	0.1715	0.7816		
	3−	4−	4−	5−	5−	5−	25	5
DMO-QPSO	0.1108	**0.8548**	0.8562	**0.0765**	**0.3002**	1.8106		
	0.1052	**0.2345**	**0.2400**	**0.0602**	**0.2049**	1.5177		
	0.0029	0.6681	0.6658	0.0216	0.0552	0.1658		
	2	1	2	1	1	3	10	1

+, −, ≈ denote that the performance of the corresponding algorithm is significantly better than, worse than, and similar to DMO-QPSO respectively by Wilcoxon rank sum test with $\alpha = 0.05$.

In addition, the statistics by the Wilcoxon rank sum tests in Tables 2 and 3 also indicate that DMO-QPSO outperformed other five algorithms. The MMOPSO was the second best on F problems and the third best on UF problems respectively, while the CMOPSO was the second worst on both F and UF problems. Table 4 illustrates the total ranks of these algorithms on F and UF problems and gives their final ranks in the last column of the table. As shown in this table, MOEA/D-DE is the second-best algorithm, followed by MMOPSO. In contrast, MOPSO/D is the worst algorithm, followed by CMOPSO.

Table 4. The final rank of different algorithms on F and UF problems.

Algorithms	F [a]	UF [b]	Total [c]	Final Rank
MOPSO	30	24	54	4
MOPSO/D	53	35	88	6
MOEA/D-DE	27	13	40	2
MMOPSO	24	18	42	3
CMOPSO	42	25	67	5
DMO-QPSO	13	10	23	1

[a] The numbers in this column are derived from those in column "Total" in Table 4. [b] The numbers in this column are derived from those in column "Total" in Table 5. [c] The numbers in this column are the sum of columns "F" and "UF".

Table 5. Average C-metric between DMO-QPSO (A) and MOPSO (B).

		DMO-QPSO	**MOPSO**
Problems	F1	**0.0676**	0
	F2	**0.5282**	0
	F3	**0.0195**	0.0027
	F4	**0.0381**	0.0025
	F5	**0.0445**	0.0033
	F6	**0.0200**	**0.0075**
	F7	**0.1795**	0.0995
	F8	**0**	0.2179
	F9	**0.3643**	0
	UF4	**0.0692**	0
	UF5	**0.7500**	0
	UF6	**0.7320**	0
	UF7	**0.2573**	0
	UF9	**0.0250**	0.0230
	UF10	0.0256	**0.1763**

The average C-metric values are shown in Tables 5–9 which confirm the results above. It can be seen from these tables that the final solutions obtained by DMO-QPSO is better than those obtained by MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO and CMOPSO for most of the test functions.

Table 6. Average C-metric between DMO-QPSO (A) and MOPSO/D (B).

		DMO-QPSO	**MOPSO/D**
Problems	F1	**0.1900**	0
	F2	**0.8125**	0
	F3	**0.5473**	0
	F4	**0.3550**	0
	F5	**0.5248**	0
	F6	**0.5165**	0
	F7	**0.5589**	0
	F8	**0.2445**	0
	F9	**0.7254**	0
	UF4	**0.0800**	0
	UF5	**0.8675**	0
	UF6	**0.8537**	0
	UF7	**0.7155**	0
	UF9	**0.9965**	0
	UF10	**0.9970**	0

Table 7. Average C-metric between DMO-QPSO (A) and MOEA/D-DE (B).

		DMO-QPSO	**MOEA/D-DE**
Problems	F1	**0.0440**	0.0066
	F2	**0.1405**	0.0024
	F3	**0.0345**	0.0020
	F4	**0.0179**	0.0085
	F5	**0.0250**	0.0045
	F6	**0.9265**	0
	F7	0.0005	**0.1944**
	F8	0	**0.2444**
	F9	**0.1930**	0
	UF4	0.0005	**0.0426**
	UF5	**0.4765**	0.1103
	UF6	**0.4628**	0.1054
	UF7	**0.0870**	0.0079
	UF9	**0.0170**	0.0010
	UF10	0.0010	**0.2475**

Table 8. Average C-metric between DMO-QPSO (A) and MMOPSO (B).

		DMO-QPSO	**MMOPSO**
Problems	F1	0.0151	0.0135
	F2	**0.1031**	0.0132
	F3	**0.1200**	0.0732
	F4	**0.1053**	0.0779
	F5	**0.0400**	0.0065
	F6	**0.0235**	0.0020
	F7	**0.1595**	0.1459
	F8	0.0005	**0.0687**
	F9	**0.2510**	0.0662
	UF4	**0.0505**	0.0010
	UF5	**0.2351**	0.1693
	UF6	0.1398	**0.1720**
	UF7	**0.1042**	0.0299
	UF9	**0.2505**	0.0025
	UF10	**0.0280**	0.0171

Table 9. Average C-metric between DMO-QPSO (A) and CMOPSO (B).

		DMO-QPSO	**CMOPSO**
Problems	F1	**0.1290**	0.0045
	F2	**0.3180**	0
	F3	**0.2517**	0
	F4	**0.1511**	0.0139
	F5	**0.2698**	0
	F6	**0.1811**	0
	F7	**0.2605**	0.0050
	F8	**0.0619**	0.0406
	F9	**0.2874**	0
	UF4	**0.0420**	0
	UF5	**0.8011**	0
	UF6	**0.7857**	0
	UF7	**0.3145**	0
	UF9	**0.8122**	0
	UF10	**0.9210**	0

The results of the trial runs which have the lowest IGD values on 15 test functions produced by MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO, CMOPSO and DMO-QPSO are selected, respectively, and then plotted in Figures 1 and 2. The figures in Figures 1 and 2 clearly show the evolution of the IGD values for different algorithms versus the number of iterations on both F and UF problems. It can be seen that the results in these figures are in consistence with those in Tables 2 and 3. For the F problems, DMO-QPSO performed the best except on F6, F7 and F8. As we can see in Figure 1, MOEA/D-DE was the best on problems F7 and F8, the IGD values of which drop quickly at the early stage and then converge to values close to 0.08 and 0.20, respectively. MMOPSO obtained the second minimal IGD value on F8, followed by DMO-QPSO, but it declines even faster than MOEA/D-DE during the first 200 iteration steps. On F6, the IGD value of DMO-QPSO fluctuates for about 400 iteration steps during the whole search process and then reaches the value (0.1634) just slightly larger than that of MMOPSO (0.1632), which may be related to the variation of the swarm diversity. MOPSO/D had the worst performance on all of F problems except on F6.

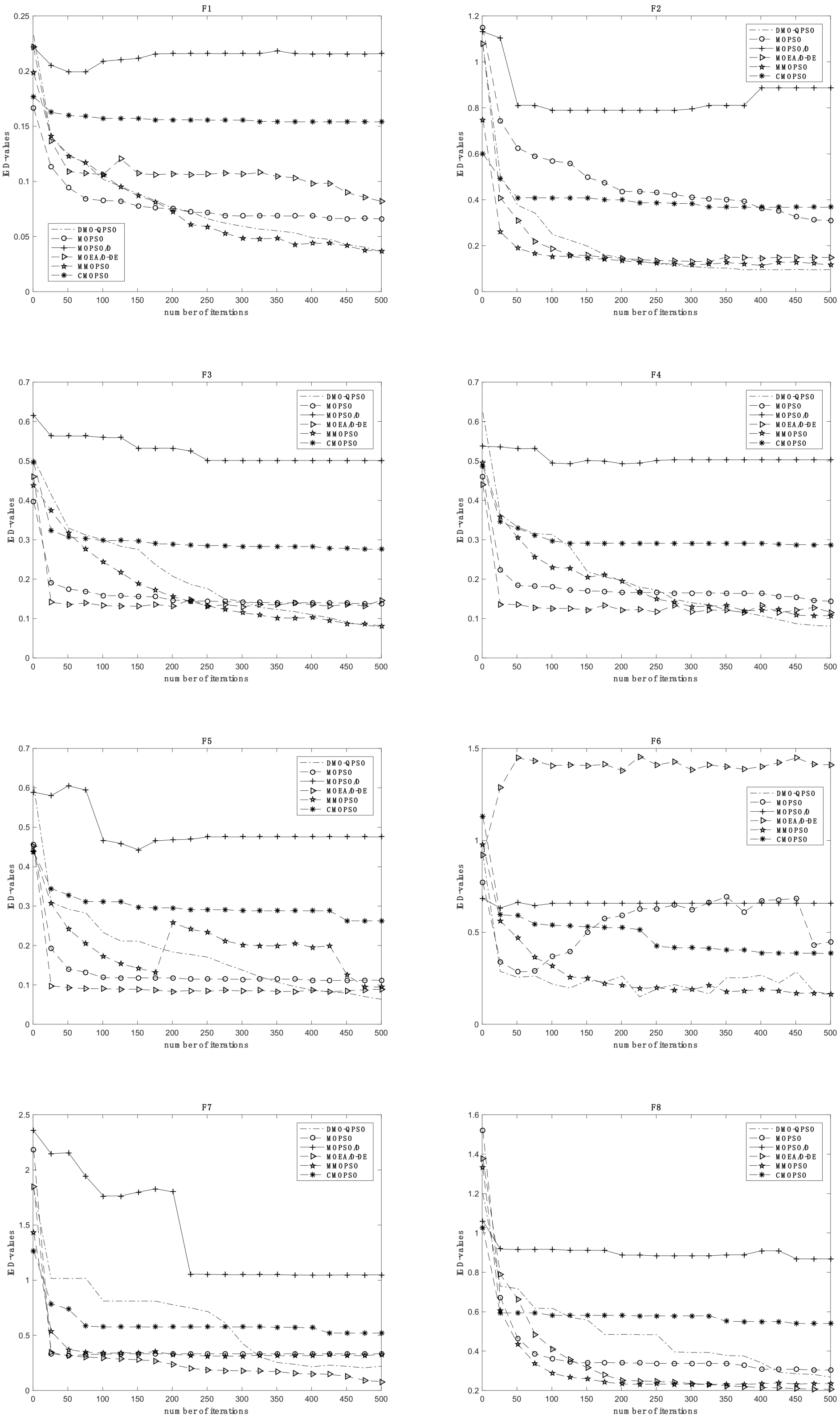

Figure 1. *Cont.*

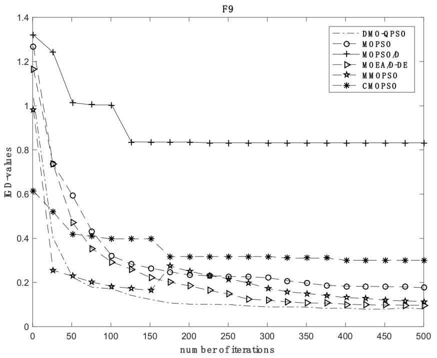

Figure 1. Evolution of the IGD values of MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO, CMOPSO and DMO-QPSO versus the number of iterations on F problems.

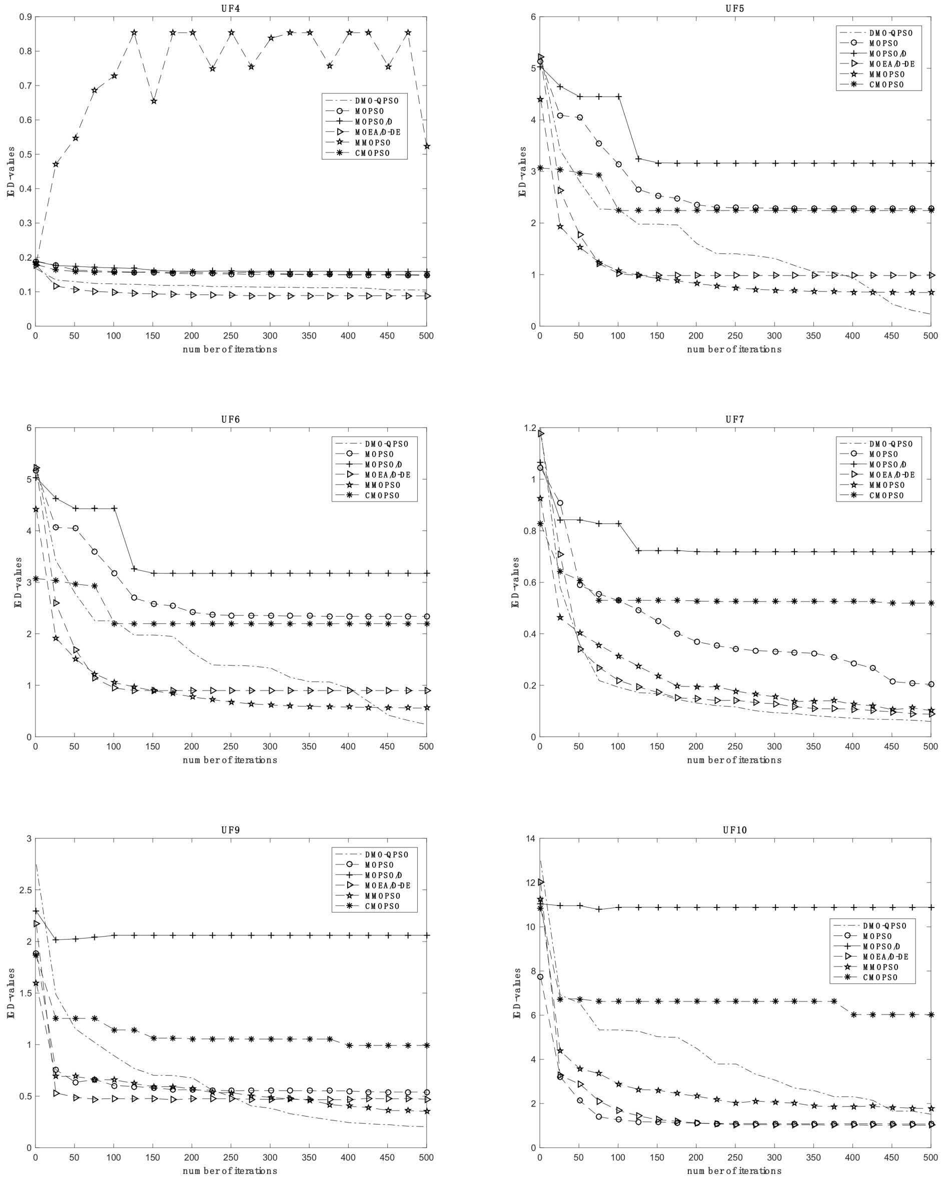

Figure 2. Evolution of the IGD values of MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO, CMOPSO and DMO-QPSO versus the number of iterations on UF problems.

For the UF problems, DMO-QPSO still performed the best except on UF4 and UF10. It can be seen that the MOEA/D-DE was the best on UF4, the IGD value of which decreases rapidly to a value just below 0.1 while the value of DMO-QPSO is larger than 0.1. By contrast, MMOPSO had the worst performance on UF4, the IGD value of which fluctuates significantly between 0.5 and 0.9. On UF10, MOEA/D-DE and MOPSO had the similar performance, for the IGD values of them decrease rapidly during the first 80 iteration steps and then gradually converge to values around 1.0. DMO-QPSO was the third-best, followed by MMOPSO. MOPSO/D had the worst performance on all of UF problems except on UF4. Additionally, from all the figures in Figures 1 and 2, we can observe that within the same iteration steps, the IGD value of DMO-QPSO declines slowly at the early stage on several test problems compared to MOEA/D-DE but gets to a much smaller value at the later stage. If we perform a greater amount of iterations, the DMO-QPSO may obtain much better results than the other compared algorithms.

In summary, DMO-QPSO has better performance on most of the test functions compared to other tested algorithms, and it is promising in solving MOPs with complicated PS shapes.

5. Further Discussions

In this section, we further study the impact of different decomposition approaches (i.e., TCH and PBI) and the polynomial mutation on DMO-QPSO. Some comparison experiments were also conducted.

5.1. The Impact of Different Decomposition Approaches

As described in Section 2.4, TCH and PBI are two commonly used approaches for decomposition-based multi-objective optimization algorithms. In terms of solution uniformness, the TCH approach may perform worse than the PBI approach, especially for the problems having more than two objectives. Therefore, we tested MOPSO/D, MOEA/D-DE and DMO-QPSO using TCH or PBI as the decomposition method respectively. The parameter settings for each algorithm are the same as those presented in Section 4.2. Tables 10 and 11 present the average, minimum and standard deviation (SD) of the IGD values of 30 final populations on different test functions that were produced by each algorithm. In these tables, MOPSO/D-TCH, MOEA/D-DE-TCH and DMO-QPSO-TCH stand for the variants of algorithms MOPSO/D, MOEA/D-DE and DMO-QPSO using the TCH approach, respectively.

Table 10. The mean, minimum and standard deviation of IGD values for different algorithms with TCH or PBI on F problems.

IGD	MOPSO/D-TCH	MOPSO/D	MOEA/D-DE-TCH	MOEA/D-DE	DMO-QPSO-TCH	DMO-QPSO
F1	0.2300 0.2066 0.0102	0.2313 0.2159 0.0084	0.1074 0.0763 0.0169	0.1180 0.0820 0.0175	0.0435 0.0405 0.0022	0.0404 0.0359 0.0030
F2	0.9740 0.7179 0.0904	1.0223 0.8869 0.0757	0.1396 0.0896 0.0295	0.2394 0.1492 0.0775	0.1015 0.0863 0.0132	0.1023 0.0962 0.0036
F3	0.5624 0.4750 0.0416	0.5754 0.5013 0.0449	0.1399 0.0885 0.0317	0.1987 0.1464 0.0357	0.0862 0.0816 0.0029	0.0884 0.0787 0.0071
F4	0.5712 0.5041 0.0320	0.5748 0.5030 0.0331	0.1313 0.0981 0.0217	0.1702 0.1159 0.0294	0.0922 0.0851 0.0043	0.0898 0.0810 0.0088
F5	0.5616 0.4807 0.0388	0.5558 0.4757 0.0351	0.1070 0.0786 0.0196	0.1405 0.0891 0.0324	0.0695 0.0635 0.0033	0.0696 0.0635 0.0035
F6	1.0426 0.7149 0.1945	0.9781 0.6580 0.1810	1.1151 0.9453 0.1279	1.5602 1.4111 0.0577	0.2699 0.1326 0.0572	0.2608 0.1634 0.0248

Table 10. *Cont.*

IGD	MOPSO/D-TCH	MOPSO/D	MOEA/D-DE-TCH	MOEA/D-DE	DMO-QPSO-TCH	DMO-QPSO
F7	1.4695 0.9858 0.2208	1.5032 1.0491 0.2076	0.1696 0.0306 0.0708	0.2139 0.0792 0.0734	0.4544 0.2444 0.1042	0.4283 0.2210 0.1160
F8	1.1057 0.8696 0.0949	1.1271 0.8681 0.1334	0.3246 0.1830 0.0668	0.3046 0.2053 0.0645	0.4861 0.3763 0.0602	0.4676 0.2686 0.0907
F9	0.9959 0.8894 0.0620	1.0358 0.8319 0.0964	0.1489 0.0916 0.0566	0.1982 0.0971 0.0498	0.1061 0.0964 0.0103	0.0938 0.0803 0.0108

Table 11. The mean, minimum and standard deviation of IGD values for different algorithms with TCH or PBI on UF problems.

IGD	MOPSO/D-TCH	MOPSO/D	MOEA/D-DE-TCH	MOEA/D-DE	DMO-QPSO-TCH	DMO-QPSO
UF4	0.1633 0.1535 0.0045	0.1672 0.1592 0.0035	0.0972 0.0865 0.0075	0.0994 0.0879 0.0052	0.1092 0.1030 0.0041	0.1108 0.1052 0.0029
UF5	4.3795 3.5695 0.2964	4.3807 3.1642 0.3636	1.7784 0.9730 0.3492	1.7532 0.9801 0.5473	0.9481 0.1929 0.6835	0.8548 0.2345 0.6681
UF6	4.3849 3.5693 0.2963	4.3793 3.1732 0.3645	1.7752 0.9120 0.3462	1.7616 0.9003 0.5481	0.9482 0.1989 0.6848	0.8562 0.2400 0.6658
UF7	1.0646 0.8402 0.1267	1.0652 0.7175 0.1386	0.1278 0.0902 0.0293	0.1817 0.0883 0.0717	0.0706 0.0584 0.0225	0.0765 0.0602 0.0216
UF9	2.7186 2.1546 0.3487	2.5708 2.0591 0.2999	0.5259 0.4504 0.0608	0.5971 0.4705 0.0640	0.4036 0.3203 0.0549	0.3002 0.2049 0.0552
UF10	13.368 11.118 0.9952	12.7750 10.8881 0.9072	1.7087 1.0282 0.4163	1.4303 1.0272 0.3025	2.3519 2.0301 0.1609	1.8106 1.5177 0.1658

According to Tables 10 and 11, algorithms (i.e., MOPSO/D and DMO-QPSO) using PBI performed better than those using TCH, particularly for solving tri-objective problems. As for DMO-QPSO, using PBI, to some extent, could help the algorithm to acquire better Pareto optimal solutions to approximate the entire PF. However, applying PBI to MOEA/D-DE does not show significant improvement compared to MOEA/D-DE using TCH. It may be related to the unique characteristics of the DE operators employed in MOEA/D-DE.

5.2. The Impact of Polynomial Mutation

Polynomial mutation was adopted in both MOPSO/D, MOEA/D-DE, MMOPSO and CMOPSO for producing new solutions as well as maintaining the population diversity, as stated in the literature [12,18,20,36]. In order to investigate the impact of polynomial mutation on DMO-QPSO, we tested four DMO-QPSO variants, i.e., DMO-QPSO, DMO-QPSO-TCH, DMO-QPSO-pm and DMO-QPSO-TCH-pm, on different problems here. DMO-QPSO and DMO-QPSO-TCH used PBI and TCH as the decomposition method, respectively. DMO-QPSO-pm and DMO-QPSO-TCH-pm are two variants using the polynomial mutation operator on the basis of DMO-QPSO and DMO-QPSO-TCH respectively. The parameter settings for each algorithm are the same as those presented in Section 4.2. Tables 12 and 13 present the average, minimum and standard deviation (SD) of the IGD values of 30 final populations on different test functions that were produced by different variants of DMO-QPSO.

Table 12. The mean, minimum and standard deviation of IGD values for four DMO-QPSO variants on F problems.

IGD	DMO-QPSO-TCH	DMO-QPSO-TCH-pm	DMO-QPSO	DMO-QPSO-pm
F1	0.0435	0.2766	0.0404	0.0407
	0.0405	0.2739	0.0359	0.0371
	0.0022	0.0016	0.0030	0.0028
F2	0.1015	0.0971	0.1023	0.0974
	0.0863	0.0862	0.0962	0.0815
	0.0132	0.0069	0.0036	0.0049
F3	0.0862	0.0845	0.0884	0.0836
	0.0816	0.0748	0.0787	0.0775
	0.0029	0.0045	0.0071	0.0042
F4	0.0922	0.0952	0.0898	0.0944
	0.0851	0.0859	0.0810	0.0791
	0.0043	0.0073	0.0088	0.0130
F5	0.0695	0.0697	0.0696	0.0688
	0.0635	0.0646	0.0635	0.0643
	0.0033	0.0027	0.0035	0.0030
F6	0.2699	0.2781	0.2608	0.2480
	0.1326	0.1395	0.1634	0.1315
	0.0572	0.0579	0.0248	0.0486
F7	0.4544	0.4231	0.4283	0.3838
	0.2444	0.2020	0.2210	0.2104
	0.1042	0.1469	0.1160	0.1201
F8	0.4861	0.5126	0.4676	0.4673
	0.3763	0.3843	0.2686	0.3712
	0.0602	0.0589	0.0907	0.0618
F9	0.1061	0.1025	0.0938	0.0919
	0.0964	0.0905	0.0803	0.0789
	0.0103	0.0065	0.0108	0.0074

'pm' in the table stands for the polynomial mutation, 'DMO-QPSO' means DMO-QPSO using PBI, and 'DMO-QPSO-pm' means the DMO-QPSO using both PBI and polynomial mutation.

Table 13. The mean, minimum and standard deviation of IGD values for four DMO-QPSO variants on UF problems.

IGD	DMO-QPSO-TCH	DMO-QPSO-TCH-pm	DMO-QPSO	DMO-QPSO-pm
UF4	0.1092	0.1080	0.1108	0.1117
	0.1030	0.0978	0.1052	0.1068
	0.0041	0.0043	0.0029	0.0025
UF5	0.9481	1.1249	0.8548	0.6837
	0.1929	0.2737	0.2345	0.2278
	0.6835	0.6946	0.6681	0.6642
UF6	0.9482	1.1217	0.8562	0.6886
	0.1989	0.2738	0.2400	0.2260
	0.6848	0.6896	0.6658	0.6654
UF7	0.0706	0.1134	0.0765	0.0767
	0.0584	0.0577	0.0602	0.0628
	0.0225	0.1043	0.0216	0.0416
UF9	0.4036	0.4015	0.3002	0.3029
	0.3203	0.3204	0.2049	0.2151
	0.0549	0.0542	0.0552	0.0480
UF10	2.3519	2.3720	1.8106	1.7806
	2.0301	1.9064	1.5177	1.4807
	0.1609	0.2306	0.1658	0.1562

'pm' in the table stands for the polynomial mutation, 'DMO-QPSO' means DMO-QPSO using PBI, and 'DMO-QPSO-pm' means the DMO-QPSO using both PBI and polynomial mutation.

As we can see from Tables 12 and 13, the DMO-QPSO variants using PBI, i.e., DMO-QPSO and DMO-QPSO-pm, outperformed the DMO-QPSO variants using TCH, i.e., DMO-QPSO-TCH and DMO-QPSO-TCH-pm, on most of the test problems. These results also confirm the conclusion presented in Section 5.1. Besides, DMO-QPSO-pm shows an advantage over DMO-QPSO on several test problems. It should be pointed out that adopting the PBI approach and the polynomial mutation at the same time can effectively improve the algorithmic performance of DMO-QPSO.

6. Conclusions

This paper has proposed a multi-objective quantum-behaved particle swarm optimization algorithm based on decomposition, named DMO-QPSO, which integrates the QPSO algorithm with the original MOEA/D framework and adopts a strategy to control the swarm diversity. Without using the neighboring relationship defined in the MOEA/D framework, we employed a two-phased diversity controlling mechanism to avoid the premature convergence and make the algorithm escape sub-optimal solutions with a higher probability. In addition, we used a set of nondominated solutions to produce the global best so as to update the particle's position. The comparison experiments were carried out among six algorithms, namely, MOPSO, MOPSO/D, MOEA/D-DE, MMOPSO, CMOPSO and DMO-QPSO, on 15 test functions with complicated PS shapes. The experimental results show that the proposed DMO-QPSO algorithm has an advantage over other five algorithms on most of the test problems. It has a slower convergence speed than MOEA/D-DE on some test problems at the early stage, but has better balance between exploration and exploitation, finally obtaining better solutions to an MOP. In addition, we further investigated the impact of different decomposition approaches, i.e., the TCH and PBI approaches, as well as the polynomial mutation on DMO-QPSO. It was shown that using PBI and the polynomial mutation can enhance the algorithmic performance of DMO-QPSO, particularly when the tri-objective problems were being solved.

In the future, we will focus on studying new methods for generating a set of weight vectors that are as uniformly distributed as possible, modifying the mechanism of diversity control in the DMO-QPSO algorithm for dealing with more complicated test problems, and improving the quality of the solutions obtained. In addition, we will extend the proposed algorithm to problems having more than three objectives.

Author Contributions: Q.Y., J.S., F.P., V.P. and B.A. contributed to the study conception and design; Q.Y. performed the experiment, analyzed the experimental results and wrote the first draft of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (Projects Numbers: 61673194, 61672263, 61672265), and in part by the national first-class discipline program of Light Industry Technology and Engineering (Project Number: LITE2018-25).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the test functions can be found in papers doi: 10.1109/TEVC.2008.925798 and https://www.researchgate.net/publication/265432807_Multiobjective_optimization_Test_Instances_for_the_CEC_2009_Special_Session_and_Competition, accessed on 1 May 2021.

Acknowledgments: The authors would like to express their sincere thanks to the anonymous referees for their great efforts to improve this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
2. Mallick, S.; Kar, R.; Ghoshal, S.P.; Mandal, D. Optimal sizing and design of cmos analogue amplifier circuits using craziness-based particle swarm optimization. *Int. J. Numer. Model.-Electron. Netw. Devices Fields* **2016**, *29*, 943–966. [CrossRef]
3. Singh, M.R.; Singh, M.; Mahapatra, S.; Jagadev, N. Particle swarm optimization algorithm embedded with maximum deviation theory for solving multi-objective flexible job shop scheduling problem. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 2353–2366. [CrossRef]

4. Sousa, T.; Silva, A.; Neves, A. Particle swarm based data mining algorithms for classification tasks. *Parallel Comput.* **2004**, *30*, 767–783. [CrossRef]
5. Zhang, Y.; Wu, L.; Wang, S. Ucav path planning by fitness-scaling adaptive chaotic particle swarm optimization. *Math. Probl. Eng.* **2013**, *2013*, 705238. [CrossRef]
6. Zhang, Y.; Jun, Y.; Wei, G.; Wu, L. Find multi-objective paths in stochastic networks via chaotic immune pso. *Expert Syst. Appl.* **2010**, *37*, 1911–1919. [CrossRef]
7. Ng, M.C.; Fong, S.; Siu, S.W. Psovina: The hybrid particle swarm optimization algorithm for protein-ligand docking. *J. Bioinform. Comput. Biol.* **2015**, *13*, 1541007. [CrossRef] [PubMed]
8. Moore, J.; Chapman, R. *Application of Particle Swarm to Multiobjective Optimization*; Unpublished Work; Department of Computer Science and Software Engineering, Auburn University: Auburn, Alabama, 1999.
9. Coello, C.A.C.; Lechuga, M.S. Mopso: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02 (Cat. No.02TH8600), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1051–1056.
10. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [CrossRef]
11. Raquel, C.R.; Prospero, C.N. An effective use of crowding distance in multiobjective particle swarm optimization. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05), Washington, DC, USA, 25–29 June 2005.
12. Peng, W.; Zhang, Q. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In Proceedings of the 2008 IEEE International Conference on Granular Computing (GrC 2008), Hangzhou, China, 26–28 August 2008.
13. Zhang, Q.; Li, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2008**, *11*, 712–731. [CrossRef]
14. Moubayed, N.; Petrovski, A.; Mccall, J. A novel smart multi-objective particle swarm optimisation using decomposition. In *International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–10.
15. Mart´ınez, Z.; Coello, C.A.C. A multi-objective particle swarm optimizer based on decomposition. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11), Dublin, Ireland, 12–16 July 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 69–76.
16. Moubayed, N.; Petrovski, A.; Mccall, J. D2mopso: Mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evol. Comput.* **2014**, *22*, 47–77. [CrossRef]
17. Dai, C.; Wang, Y.; Ye, M. A new multi-objective particle swarm optimization algorithm based on decomposition. *Inf. Sci.* **2015**, *325*, 541–557. [CrossRef]
18. Lin, Q.; Li, J.; Du, Z.; Chen, J.; Ming, Z. A novel multi-objective particle swarm optimization with multiple search strategies. *Eur. J. Oper. Res.* **2015**, *247*, 732–744. [CrossRef]
19. Zhu, Q.; Lin, Q.; Chen, W.; Wong, K.; Coello, C.A.C.; Li, J.; Chen, J.; Zhang, J. An external archive-guided multiobjective particle swarm optimization algorithm. *IEEE Trans. Cybern.* **2017**, *47*, 2794–2808. [CrossRef]
20. Zhang, X.; Zheng, X.; Cheng, R.; Qiu, J.; Jin, Y. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf. Sci.* **2018**, *427*, 63–76. [CrossRef]
21. Yang, W.; Chen, L.; Wang, Y.; Zhang, M. Multi/many-objective particle swarm optimization algorithm based on competition mechanism. *Comput. Intell. Neurosci.* **2020**, *2020*, 5132803. [CrossRef]
22. Sun, J.; Feng, B.; Xu, W. Particle swarm optimization with particles having quantum behavior. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 325–331.
23. Sun, J.; Fang, W.; Wu, X.; Palade, V. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. *Evol. Comput.* **2012**, *20*, 349–393. [CrossRef] [PubMed]
24. Sun, J.; Wu, X.; Palade, V.; Fang, W.; Lai, C.H.; Xu, W. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Inf. Sci.* **2012**, *193*, 81–103. [CrossRef]
25. Qiu, C.; Wang, C.; Zuo, X. A novel multi-objective particle swarm optimization with k-means based global best selection strategy. *Int. J. Comput. Intell. Syst.* **2013**, *6*, 822–835. [CrossRef]
26. Cheng, S.; Chen, M.; Peter, J. Improved multi-objective particle swarm optimization with preference strategy for optimal dg integration into the distribution system. *Neurocomputing* **2015**, *148*, 23–29. [CrossRef]
27. Trivedi, A.; Srinivasan, D.; Sanyal, K.; Ghosh, A. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Trans. Evol. Comput.* **2017**, *21*, 440–462. [CrossRef]
28. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: Hoboken, NJ, USA, 2001.
29. Coello, C.A.C.; Veldhuizen, D.A.V.; Lamont, G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed.; Springer: Boston, MA, USA, 2007.
30. Tan, K.C.; Khor, E.F.; Lee, T.H. *Multiobjective Evolutionary Algorithms and Applications (Advanced Information and Knowledge Processing)*; Springer: Berlin/Heidelberg, Germany, 2005.
31. Miettinem, K. *Nonlinear Multiobjective Optimization*; Springer: Boston, MA, USA, 1999.
32. Stadler, W. A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960. *J. Optim. Theory Appl.* **1979**, *29*, 1–52. [CrossRef]

33. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsgaii. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
34. Sun, J.; Xu, W.; Fang, W. Quantum-behaved particle swarm optimization algorithm with controlled diversity. In *International Conference on Computational Science*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 2006, pp. 847–854.
35. Sun, J.; Xu, W.; Feng, B. 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, Waikoloa, HI, USA, 12 October 2005.
36. Li, H.; Zhang, Q. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [CrossRef]
37. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthan, P.N.; Tiwari, S. *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*; Mechanical Engineering, University of Essex: Colchester, UK; Nanyang Technological University: Singapore, 2008.