

Quantum-inspired multi-objective optimization evolutionary algorithm based on decomposition

Yang Wang¹ · Yangyang Li¹ · Licheng Jiao¹

Published online: 16 May 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract As an important multi-objective optimization algorithm, multi-objective evolutionary algorithm based on decomposition (MOEA/D) attracts more and more attention recently. In this paper, some methods inspired from quantum behavior are integrated in MOEA/D. A new algorithm, quantum-inspired MOEA/D (QMOEA/D), is proposed and proved to be effective to improve the performance of MOEA/D. In the new algorithm, a global solution (GS) and a local solution (LS) are stored for each subproblem. The attractor and characteristic length in quantum-inspired method are designed with GS and LS. The LS is selected as the attractor for each subproblem. And the characteristic length is associated with the difference between the LS and GS. The algorithm based on nondominated sorting is used for comparing firstly. Then the original and some advanced versions of MOEA/D are used as the comparison algorithms. Through the comparison it can be found that GS and LS are helpful to retain the diversity of the solutions. A wide Pareto front can be obtained on most of the test suites. And the quantum-inspired generator is effective to obtain better solutions with GS and LS.

Keywords Multi-objective optimization · Quantum-inspired method · Attractor · Characteristic length · MOEA/D

Communicated by V. Loia.

✉ Yang Wang
wangyang1990@aliyun.com

¹ Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

1 Introduction

As an important aspect in engineering application and scientific research, optimization is indispensable. In practical problems it is usual to optimize several objectives at the same time. These problems are multi-objective optimization problem (MOP). For any MOP, there are several points in decision space called Pareto optimal solutions (PS). Each objective of PS cannot be better without making deterioration of other objectives. The objective vectors of PS are Pareto front (PF). The purpose of an algorithm for solving MOP can be summarized as the following aspects: (1) searching solutions whose objective vectors are on or near PF; (2) searching solutions whose objective vectors distribute along PF as widely as possible.

Evolutionary computing has been widely used to solve MOP and is a great success solving many problems. Literatures [Deb et al. \(2002\)](#) and [Zitzler et al. \(2002\)](#) are some well-known multi-objective evolutionary algorithms (MOEAs). In [Deb et al. \(2002\)](#) a fast nondominated sorting approach is proposed. The solutions are ranked by their levels of nondomination firstly. Then with comparison of crowding distance, the population can be updated given convergence and diversity. In [Zitzler et al. \(2002\)](#) both dominating and dominated solutions are used to assign the fitness of one solution. In the updating of archive, the number of individuals is constant and the boundary solutions remain. Both algorithms treat the MOP as a whole and improve the quality of population by the domination relationship. Many advanced versions have been proposed to improve the performance in this framework. [Elhossini et al. \(2010\)](#) combine evolutionary algorithm (EA) and particle swarm optimization (PSO) to solve MOPs. Both EA operator and PSO operator are used. In [Zhang et al. \(2008\)](#), a regularity model is built to predict the promising area of optimal solutions in decision space.

Instead of treating the MOP as a whole, many traditional mathematical programming methods solve MOP by decomposing it into a number of single objective problems. Based on this idea, Zhang and Li (2007) proposed a new framework of MOEA, multi-objective optimization evolutionary algorithm based on decomposition (MOEA/D). It uses a set of single objective problems to solve MOP. This is a very different work from domination-based algorithms. The population of MOEA/D is composed by different single objective problems. The purpose of searching the PS for a MOP is transformed to search the best solutions of these single objective problems. In another words, the best solutions of these single objective problems are on the PS of the MOP. Once all the global optimal solutions of the single objective problems are obtained, the MOP is solved. In recent years, MOEA/D attracts more and more research attention. Zhang et al. (2010), Li and Landa-Silva (2011), Martínez and Coello Coello (2011), Sindhya et al. (2011), Ishibuchi and Nojima (2011), Cheng et al. (2012) are some of the studies.

In optimization of a single objective problem, avoiding the solutions trapping into local optimum is an important issue. It is also a key point in MOEA/D. The local optimum will lower the performance of MOEA/D from both the convergence and diversity of the solutions. From the view of convergence, it is obvious that the local optimal solution is not on the PS. From the view of diversity, the local optimum will assimilate the solutions of different subproblems. It will be described in part III in detail. In order to make the evolution of subproblems in MOEA/D more effective, some effort has been made in the last several years. Li and Zhang (2009), Zhang et al. (2009), Montañó et al. (2010) are some of them. In Li and Zhang (2009) a version combining DE operator and MOEA/D is proposed. A new mating pool is formed for generating of new solutions and updating of the subproblems. In Zhang et al. (2009), a strategy of allocation of computational resource is designed to make the algorithm focus more attention on more promising subproblems. In Montañó et al. (2010), the neighbors of one solution are determined with the Euclidean distance between solutions. The nondominated solution among its neighbors is defined as locally nondominated solution. These solutions are selected to generate new ones for maintaining the diversity.

Quantum-inspired evolutionary computing (QIEC) is a method based on concept and principle from quantum mechanics. Narayanan and Moore (1996) firstly combine evolutionary computation (EA) and quantum-inspired computation to solve the traveling salesman problem. After that a series of EA inspired by quantum computation appear, such as Han and Kim (2002), Abs da Cruz et al. (2004), Jiao et al. (2008). They are a kind of algorithm characterized by quantum bits and quantum gates. The quantum gates are used to change the quantum bits and generate new solutions through observing. Another kind of quantum-inspired computation

is proposed by Sun et al. (2004). It is an idea inspired by the behavior of particles in a potential field. The particles are bounded by the attractor. Meanwhile, they will appear in anyplace of the space with different probability density. Via setting potential well and solving Schrödinger equation, a new style to search space is built. Based on this point, Li et al. (2012) proposed an improved cooperative quantum-behaved particle swarm optimization method for solving real parameter optimization and obtained a good performance.

In this paper, we combine the attractor-based quantum method with MOEA/D to solve multi-objective optimization problems. It takes efforts on two sides: (1) define a so-called global solution (GS) and a local solution (LS) for each subproblem. The GS is the solution with the best scalar function value for the subproblem. The LS is the best solution among the solutions fitting the subproblem; (2) a new method inspired from quantum behavior for generating new solutions is designed for MOEA/D. New solutions are generated around LS to ensure wide searching in space. And with maximum likelihood estimation, GS keeps appearing probability as high as possible in the next generation. Through the new style of generating solutions, the GS and LS guide the evolution cooperatively. The whole paper is arranged as follow: Sect. 2 shows the background and basic methods related to our work. In Sect. 3, some features of MOEA/D are shown and the motivation and process of embedding quantum method in MOEA/D are explained. In Sect. 4, some experimental results are shown and analyzed. In Sect. 5, some conclusions and prospect are summarized.

2 Background and basic method

2.1 MOP decomposition and MOEA/D

In order to solve a MOP, many methods have been proposed to decompose a MOP into a number of scalar optimization problems. In Marler and Arora (2004) many styles can be found. In this chapter, three approaches mentioned in Zhang and Li (2007) are introduced.

The first one is the weighted sum approach which is effective for convex problems. $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$ is a weight vector, i.e., $\lambda_i \geq 0 (i = 1, 2, \dots, m)$ and $\sum_{i=1}^m \lambda_i = 1$. A subproblem constructed by a scalar function can be stated as:

$$\begin{aligned} \text{Minimize } g^{ws} \left(X | \vec{\lambda} \right) &= \sum_{i=1}^m \lambda_i f_i(x_i) \\ \text{subject to } X &\in \Omega^n \end{aligned} \quad (1)$$

Here $\vec{\lambda}$ is a coefficient of this scalar function. X is the variable vector to be optimized.

Another approach is Tchebycheff Approach. The form of the scalar function can be expressed as:

$$\begin{aligned} \text{Minimize } g^{te} \left(X \left| \vec{\lambda}, \vec{Z}^* \right. \right) &= \max_{1 \leq i \leq m} \{ \lambda_i | f_i(x_i) - z_i^* \} \\ \text{subject to } X &\in \Omega^n \end{aligned} \tag{2}$$

Here \vec{Z}^* is called reference point, i.e., $z_i^* = \min_{X \in \Omega^n} f_i(X)$.

The third method is penalty-based boundary intersection (PBI) approach. The form of the scalar function can be described as:

$$\begin{aligned} \text{Minimize } g^{pbi} \left(X \left| \vec{\lambda}, \vec{Z}^* \right. \right) &= d_1 + \theta d_2 \\ \text{subject to } X &\in \Omega^n \\ \text{where } d_1 &= \frac{\left\| \begin{matrix} \vec{\lambda} \\ (\vec{Z}^* - F(X))^T \vec{\lambda} \end{matrix} \right\|}{\left\| \vec{\lambda} \right\|} \\ \text{and } d_2 &= \left\| F(X) - (\vec{Z}^* - d_1 \vec{\lambda}) \right\| \end{aligned} \tag{3}$$

\vec{Z}^* is reference point as in Tchebycheff Approach. $\theta > 0$ is a penalty parameter. The details of this approach can be found in Zhang and Li (2007).

MOEA/D is a framework combining the population of subproblems with EA to solve a MOP. Because the subproblems are single objective problems, the values of scalar functions are used to evaluate solutions instead of using Pareto domination comparison. The process of MOEA/D is shown in Algorithm 1:

Algorithm 1: The framework of MOEA/D

Initialization:

1. The population size N (the number of subproblems);
2. N weight vectors;
3. N solutions and the value of responding scalar function for each weight vector;
4. T : the size of neighbors.

While (stopping condition is not true)

- step 1. Generate new solution for each subproblem;
- step 2. Calculate the objective vector of the new solution;
- step 3. Calculate the values of scalar functions of current subproblem and its neighbors;
- step 4. Update current subproblem and its neighbors. If the new value is better than the value stored before, replace the old solution with the new one and store the new value.

End While

Output: the solutions stored for the subproblems.

The framework of MOEA/D has some natural advantages than the non-domination-based MOEAs. The first one is its computational complexity. As well known, the computational complexity of fast nondominated sorting (Deb et al. 2002) is as high as $O(MN^2)$ (where M is the number of objectives and N is the population size). But in the frame of MOEA/D, for the nondominated sorting is avoided, the computation cost is low. Another advantage of MOEA/D is its ability of maintaining diversity. Each subproblem responds a region in PF. A uniform and broad searching can be offered with different subproblems. No special operators for diversity of solutions are needed.

2.2 Quantum-inspired method in EC

Quantum-inspired methods used in EC are a series of approaches coming from the concepts and equations of quantum mechanics. Usually these methods are associated with quantum gates, quantum chromosome, quantum bits, potential well, etc. In this paper, we mainly focus on the method based on attractor and characteristic length which is proposed by Sun et al. (2012).

In order to illustrate these approaches, some background knowledge is required. In classical physics, the state of motion of a particle is described by Newton equation based on Newton three laws of motion. The motion is deterministic, i.e., once the coordinate and momentum of a particle are given, its trajectory can be grasped. But in quantum space, there is wave particle duality. The motion trajectory of a particle is uncertain even if the coordinate and momentum is given. The particle can appear at any location of the space at the next time point. Instead of predicting the exact location of the particle, we just have the ability to give the probability density of appearing in the space. This prediction is achieved through wave function $\Psi(X, t)$, the square of whose modulus is probability density. Here a three-dimensional space acts as an example. For a given time point, some equations can be described as follows:

$$Q = |\Psi(x, y, z)|^2 \tag{4}$$

$$\int_{-\infty}^{+\infty} |\Psi|^2 dx dy dz = \int_{-\infty}^{+\infty} Q dx dy dz = 1 \tag{5}$$

x, y, z are coordinates describing the space. Q is the probability density function of appearing on any location for a particle.

In order to get the probability density function Q , Schrödinger equation should be solved. The form of the equation can be described as:

$$ih \frac{\partial}{\partial t} \psi(X, t) = \hat{H} \psi(X, t) \tag{6}$$

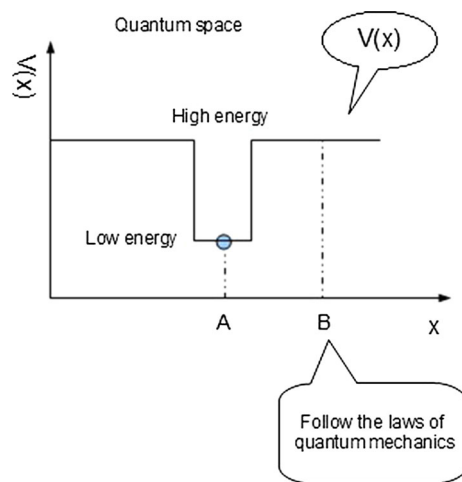


Fig. 1 Particles in quantum space

Here h is Planck constant. \hat{H} is Hamiltonian operator:

$$\hat{H} = -\frac{h^2}{2m}\nabla^2 + V(X) \quad (7)$$

m is the mass of the particle and $V(X)$ is the potential field where the particle is. ∇^2 is the Laplacian operator.

As shown in Fig. 1, in the quantum space, a point A and any point B have the low and high energy, respectively. This form of potential field is called potential well. Assuming $V(X)$ is one dimensional δ potential well:

$$V(X) = -\gamma\delta(X - p) = -\gamma\delta(Y) \quad (8)$$

Here $Y = X - p$. p is the coordinate of A . γ is an intensity coefficient. Then assuming the particle is in a certain energy state, the Schrödinger equation is equivalent to the following form:

$$\frac{d^2\phi(Y)}{dY^2} + \frac{2m}{h^2}[E + \gamma\delta(Y)]\phi(Y) = 0 \quad (9)$$

Here the probability density can be given as:

$$Q(Y) = |\phi(Y)|^2 = \frac{1}{\sqrt{L}}e^{-2|Y|/L} \quad (10)$$

$L = h^2/m\gamma$. Based on the probability density function, the location of a particle can be determined by a stochastic equation as:

$$X = p \pm \frac{L}{2} \ln(1/\mu) \quad (11)$$

$\mu \sim U(0, 1)$. The details of above derivation of the equations can be referenced in Sun et al. (2004).

According to Eq. 11, it can be found that a particle will appear around p , the bottom of the potential well, with high probability. And in other locations, the particle will appear with low probability. This point will be discussed more clearly in Sect 3. Here p and L are called as attractor and characteristic length, respectively. In Sun et al. (2012), a series of choice are given for the attractor and characteristic length. In the following section, special p and L are designed for MOEA/D according to its characters.

3 Motivation and quantum-inspired MOEA/D

3.1 Motivation

Assuming the set of subproblems in MOEA/D is $\{S_1, S_2, S_3, \dots, S_N\}$, the desired result is that every global optimum of the subproblems can be obtained to form the PF of the MOP. In the original framework of MOEA/D, a best solution is stored for each subproblem and it can be updated by new solutions generated from its neighbors. It means a new solution can be used on different subproblems. Or it can be said the solutions of several subproblems can be replaced by the same one. Obviously this cooperation of subproblems can improve the convergence of the solutions because more calculated resources are spend on several best solutions found so far. But on the other side, this cooperation blocks searching the global optimums of subproblems at some cases. Once a good solution is obtained by a subproblem, the solutions stored by its neighbors may be updated by the good one. This will lead the best solution of the neighbor to lose its trajectory. A visual example is shown in Fig. 2. For each subproblem, the best solution stored is represented as the hollow point. The ideal state is that the hollow point move along its arrow and reach a black point on the PF. Thus when all the hollow points reach the black points, the whole PF is obtained. But during the evolution, once a good solution like the red point appears, the best solutions of S_k, S_{k+1} will be updated by the red point. The moving strategy of these two hollow points will deviate from the intended track. The immediate result is that the black point responding these hollow points cannot be reached at the end of the evolution. From the point of PF, it will be observed that some parts of the PF cannot be found. Some studies have shown similar shortcomings. Liu et al. (2014) points out the simple replacement may miss some search regions. Li and Zhang (2009) indicates the loss of diversity at the early stage of search on some problems. And in Sect. 4, some visual results of the original MOEA/D can be seen.

Based on above characters of the original MOEA/D, a special quantum-inspired method is designed to relieve the side effects of the cooperation among subproblems. It focuses on the design of the attractor and the characteristic length

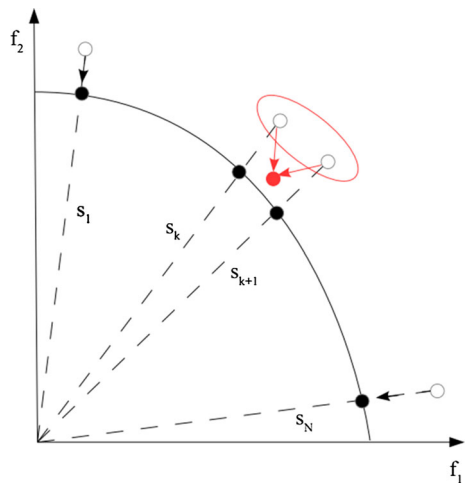


Fig. 2 The updated hollow points in oval with the red point (color figure online)

used in quantum method. The next two subsections will show the details.

3.2 The attractor and characteristic length for QMOEA/D

In order to construct the attractor and characteristic length of QMOEA/D, two solutions are stored for each subproblem. One is marked as global solution (GS) and the other is marked as local solution (LS). When a new solution is generated from one subproblem, the GSs of its neighbors will be updated. Then a most suitable subproblem for the new solution will be determined and the LSs of its neighbors will be updated. The visual difference between GS and LS is shown in Fig. 3. A new solution represented by a black point is generated from subproblem S_n . The GSs of the neighbor subproblems of S_n will be updated if the black point is better than the stored GSs. Obviously the black point is closer to the subproblem S_m . Then we will determine whether the LSs of the neighbors of S_m should be replaced by the new point according to the scalar function value.

Why are these two solutions stored for one subproblem? As what have been discussed, during the moving of one solution, if the objective vector deviates from the intended region, this solution is not suitable for the corresponding subproblem. It will mislead the subproblem to spend calculate resources on the regions focused by other subproblems, which is not helpful to obtain the global optimum. For a new solution which has a good scalar function value, if it deviates from the region of responding subproblem, it cannot update the LS of the subproblem. So the LSs are used to keep the solutions moving along the intended region. The GS is the same as the solutions in original MOEA/D. Except the scalar function value, no special information is used to guide the

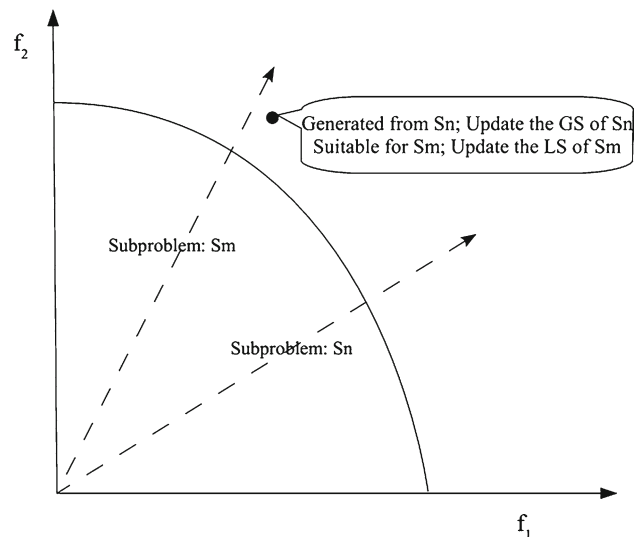


Fig. 3 The update of the GS and LS with a newly generated solution

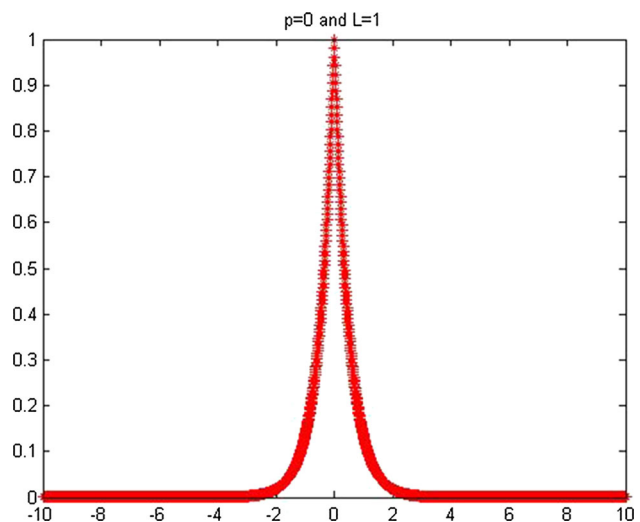


Fig. 4 The probability density with $p = 0, L = 1$

updating of GS. It is obvious that the solutions with best fitness should remain so to promote the evolution of populations. So with the use of LSs and GSs of these subproblems, the solutions are expected to move along the suitable direction to obtain the whole PF.

In quantum-inspired algorithms, the probability density function of the new solution deduced from Eq. 10 can be described as:

$$Q(X) = \frac{1}{\sqrt{L}} e^{-2|X-p|/L} \tag{12}$$

When $p = 0, L = 1$ and $p = 0, L = 5$, the probability density function of X is shown in Figs. 4 and 5, respectively. It is easy to find that solutions near attractor p are more likely to be generated. But the characteristic length L can be used

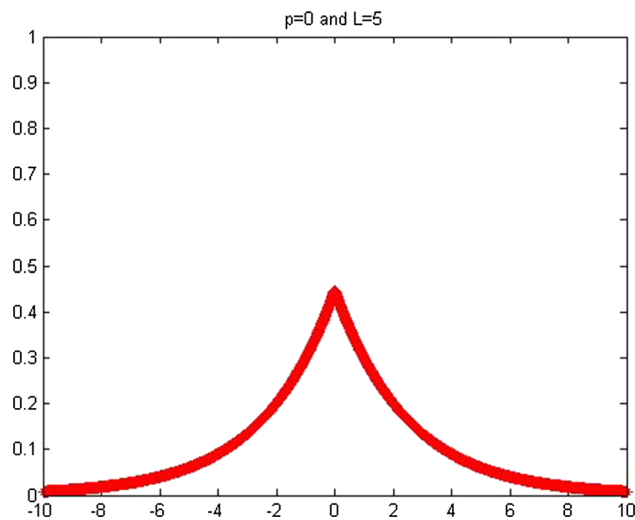


Fig. 5 The probability density with $p = 0$, $L = 5$

to modify the appearing probability of one location. When $L = 1$ and $L = 5$, the distributions are significantly different.

Taking both MOEA/D and quantum-inspired methods into account, the LS of each subproblem is selected as the attractor. Because the LS is the best one among the solutions fitting for the subproblem, more calculate resources should be spent around LS in generating more suitable solutions. Thus, Eq. 12 becomes the following form:

$$Q(X) = \frac{1}{\sqrt{L}} e^{-2|X-LS|/L} \quad (13)$$

In evolutionary algorithm, it is reasonable to remain the best solution. So for each subproblem, the solutions around GS which is the best solution obtained so far should have a high probability to appear. From Eq. 13 and maximum likelihood estimation, L can be obtained as:

$$L = 2 |GS - LS| \quad (14)$$

Thus the equation for QMOEA/D to generate new solution N_s can be described as:

$$N_s = LS \pm (GS - LS) * \ln(1/\mu) \quad (15)$$

3.3 Framework of QMOEA/D

In the above subsection, two main elements in quantum-inspired method are constructed for MOEA/D. Based on the previous attractor and characteristic length, the framework of QMOEA/D is shown as Algorithm 2:

Algorithm 2: The framework of QMOEA/D

Initialization:

1. The population size N (also the number of subproblems);
2. Uniform N weight vectors;
3. N solutions for each subproblem and stored as the GSs and LSs;
4. the number of neighbors, T

While (stopping condition is not true)

FOR $i=1: N$

step 1. Generate new solution N_s as Equation 15;

step 2. Polynomial mutation for N_s ;

step 3. Calculate the objective vector $F(N_s)$ of N_s ;

step 4. For every subproblem among S_i and its neighbors, calculate the scalar function value with $F(N_s)$. If the value is better than the stored GS, update the GS with N_s ;

step 5. Find a most suitable subproblem S_k for N_s ;

step 6. For every subproblem among S_k and its neighbors, calculate the scalar function value with $F(N_s)$. If the value is better than the stored LS, update the LS with N_s ;

End For

End While

Output: the GSs of the subproblems.

In the above framework, step 5 would be discussed in detail. For each subproblem S_i whose weight vector is $(\lambda_i^1, \lambda_i^2, \dots, \lambda_i^m)$, we normalize its weight vector to be $(1, \frac{\lambda_i^2}{\lambda_i^1}, \dots, \frac{\lambda_i^m}{\lambda_i^1})$. And the objective vector $F(N_s) = (f_{N_s}^1, f_{N_s}^2, \dots, f_{N_s}^m)$ is also normalized as $(1, \frac{f_{N_s}^2}{f_{N_s}^1}, \dots, \frac{f_{N_s}^m}{f_{N_s}^1})$. The Euclidean distance between these two normalized vectors is recorded as the distance between N_s and S_i . The subproblem with the shortest distance to N_s is selected as the most suitable one.

4 Comparison and discussion

In the following, the performance metric and test suits used in our experiments are introduced firstly. Then the comparison between QMOEA/D and nondominated sorting approach NSGA-II, QMOEA/D and original MOEA/D (MOEA/D-SBX), QMOEA/D and some other advanced MOEA/D are shown, respectively. And some discussion about QMOEA/D is made in the last subsection.

4.1 Performance metric

In the experiments, inverted generational distance (IGD) (Sierra and Coello Coello 2005) is used as the metric to measure the quality of solution sets. P^* is an objective vector set whose members distribute on the true PF uniformly. P is another objective vector set whose members are obtained by

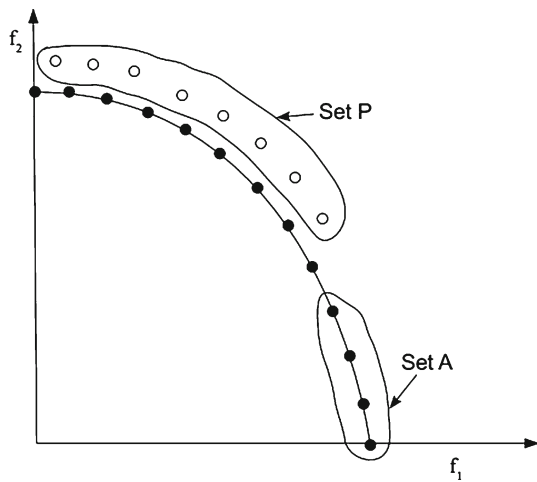


Fig. 6 A PF losing diversity

a multi-objective optimization algorithm. Then the IGD of P can be calculated as:

$$IGD(P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \tag{16}$$

Here $|P^*|$ is the number of members in P^* and set 500 in our experiments. $d(v, P)$ is the minimum distance from v to the set P .

It is obvious that the IGD is able to measure the convergence of solutions. Meanwhile, the metric can reflect the diversity of solutions like Fig. 6. When the objective vectors, represented by the hollow circles, are not distributed along PF widely, no solutions in set P is near with set A . So the sum of the distance in Eq. (16) will be large. And the IGD of the hollow circles could not be small.

For the comparison experiments in this section, several independent runs are performed with each algorithm. The average value, best value and standard deviation are shown and compared, respectively. To be more rigorous, the Wilcoxon signed-ranks test is performed on each problem. The significance level is set to be 0.05 (García et al. 2009). The sign (+) indicates the comparison algorithm is significantly better than QMOEA/D. The sign (-) indicates the comparison algorithm is significantly worse than QMOEA/D. The sign (=) indicates no significant difference.

4.2 Multi-objective suites

Eight test problems are used in the experiments. All the test suites are minimization problems. ZDT1 and ZDT2 come from Zitzler et al. (2000). ZZZJ1, ZZZJ2, ZZZJ5 and ZZZJ6 come from Zhang et al. (2008). MOP1 and MOP2 come from Liu et al. (2014). The details of the functions are as follows:

ZDT1:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n x_i \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

ZDT2:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n x_i \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

ZZJ1:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n (x_i - x_1)^2 \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

ZZJ2:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n (x_i - x_1)^2 \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

ZZJ3:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n (x_i^2 - x_1)^2 \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

ZZJ4:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right]$$

$$\text{where } g(x) = 1 + \frac{9 \left(\sum_{i=2}^n (x_i^2 - x_1)^2 \right)}{n-1}$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 30$.

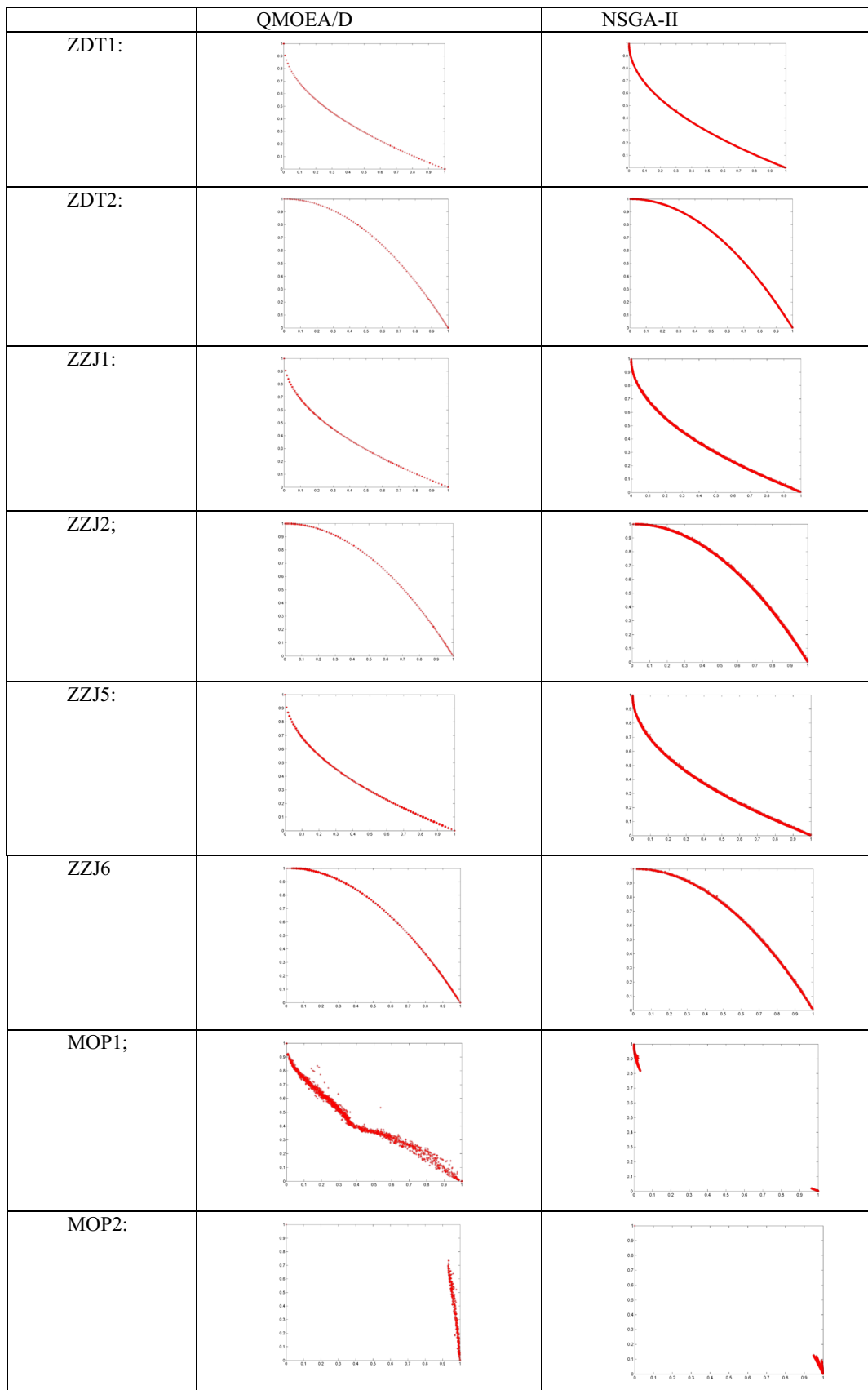


Fig. 7 The PFs from the 30 independent runs on the 8 test problems. The *left* are the results of QMOEA/D. The *right* are the results of NSGA-II

MOP1:

$$f_1(x) = (1 + g(x))x_1$$

$$f_2(x) = (1 + g(x))(1 - \sqrt{x_1})$$

where $g(x) = 2 \sin(\pi x_1) \sum_{i=2}^n (-0.9t_i^2 + |t_i|^{0.6})$,

$$t_i = x_i - \sin(0.5\pi x_1)$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 10$.

MOP2:

$$f_1(x) = (1 + g(x))x_1$$

$$f_2(x) = (1 + g(x))(1 - x_1^2)$$

where $g(x) = 10 \sin(\pi x_1) \sum_{i=2}^n \left(\frac{|t_i|}{1+e^{5|t_i|}} \right)$

$$t_i = x_i - \sin(0.5\pi x_1)$$

and $x_i \in [0, 1] \quad i = 1, \dots, n, n = 10$.

4.3 Comparison between QMOEA/D and NSGA-II

In this subsection, a comparison is made between QMOEA/D and NSGA-II which is the classical evolutionary algorithm for multi-objective optimization. The population size is 100 for both the algorithms like Deb et al. (2002). 3×10^5 fitness evaluations or 3000 generations are used as the termination condition. The neighbor size for QMOEA/D is 20. The parameters used for crossover and mutation keep the same with Deb et al. (2002). 30 independent experiments are run with each algorithm on each problem.

The PFs found in the 30 runs are plotted in Fig. 7. For each problem, all the PFs obtained with one algorithm are overlaid in the figure. The same style is used for plotting other figures of PFs in the following subsections.

Table 1 shows the average, best and std of IGDs on different problems, respectively. For ZDT1–ZZJ6, both the algorithms can obtain the whole PFs. But the quality of these PFs is quite different. QMOEA/D outperforms NSGA-II on all these six problems on both the average IGD and best IGD. Especially on ZDT2 and ZZJ6, the advantage is huge from the view of average IGD. And the std shows QMOEA/D has a more steady performance than NSGA-II. On MOP1 and MOP2, the PFs obtained by QMOEA/D are wider than the ones from NSGA-II which can be observed from Fig. 7.

With signed-ranks test, it can be observed that on all the eight problems QMOEA/D outperforms NSGA-II significantly. The different frameworks of the two algorithms may be a main reason for the different performance. Methods based on decomposition performs better than the methods based on domination ranking on many test problems, which

Table 1 The average, best and std of IGDs among the 30 independent runs with QMOEA/D and NSGA-II

	QMOEA/D	NSGA-II
ZDT1		
Average	0.0038	0.0050(–)
Best	0.0038	0.0044
Std	3.7183e–007	2.9886e–004
ZDT2		
Average	0.0038	0.0455(–)
Best	0.0038	0.0057
Std	4.0808e–007	0.1534
ZZJ1		
Average	0.0038	0.0059(–)
Best	0.0038	0.0053
Std	1.5117e–005	2.6504e–004
ZZJ2		
Average	0.0039	0.0060(–)
Best	0.0038	0.0056
Std	1.1305e–004	2.0953e–004
ZZJ5		
Average	0.0043	0.0059(–)
Best	0.0040	0.0054
Std	1.1727e–004	2.7980e–004
ZZJ6		
Average	0.0058	0.3485(–)
Best	0.0045	0.0056
Std	7.4539e–004	0.3038
MOP1		
Average	0.0602	0.3388(–)
Best	0.0355	0.3216
Std	0.0431	0.0078
MOP2		
Average	0.2558	0.3487(–)
Best	0.2344	0.3042
Std	0.0384	0.0148

The bold values show the better one in the comparison

has been shown in Zhang and Li (2007). The quantum-inspired operators for MOEA/D may be another important factor. In the next subsection, the influence of quantum-inspired operator will be shown through the comparison between QMOEA/D and the original MOEA/D.

4.4 Comparison between QMOEA/D and the original MOEA/D

The performance of QMOEA/D and the original MOEA/D (MOEA/D-SBX) are shown in this subsection. For fairness, the common parameters keep the same in both the algorithms. The population size is 200. The size of neighbor is

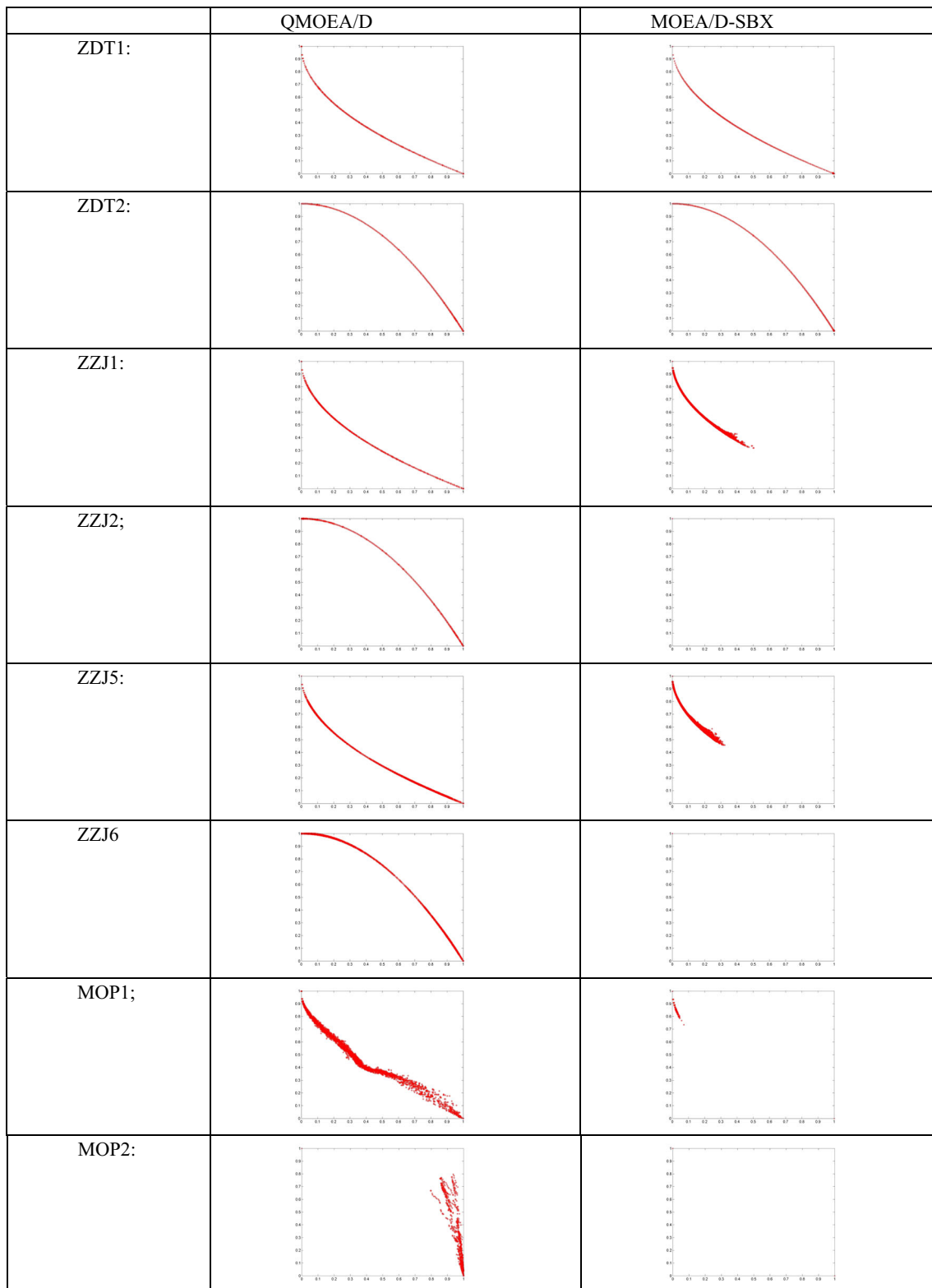


Fig. 8 The PFs from the 30 independent runs on the 8 test problems. The *left* is the results of QMOEA/D. The *right* is the results of MOEA/D-SBX

20. The decomposition approach used here is Tchebycheff. And 3×10^5 fitness evaluations are used as the termination condition. The distribution index for polynomial mutation is

20 and the mutation rate is $1/n$. n is the dimension of the decision space. In MOEA/D, the distribution index for simulated binary crossover (SBX) is set to 20. And the SBX rate is 1.

Both MOEA/D-SBX and QMOEA/D have been run 30 times independently on the 8 test problems, respectively. Figure 8 shows the Pareto fronts obtained from the 30 independent runs.

From the shape of the obtained PFs, an intuitive conclusion is that QMOEA/D can find more complete PFs. Both algorithms have good performance on ZDT1 and ZDT2. But on ZZZ1, ZZZ2, ZZZ5, ZZZ6 and MOP1, MOEA/D-SBX has little effect. Only a few points or parts near the PF are obtained. However QMOEA/D has better performance on these five problems. Except MOP1, the PFs from QMOEA/D are complete and smooth. For MOP1, although the convergence of the obtained PF is not good, the PF is relatively complete. On MOP2, a good PF is not obtained with both the algorithms. MOEA/D-SBX only obtains several points near the boundary of the PF. Meanwhile, a part with bad convergence of PF is found with QMOEA/D on this problem.

Generally speaking, the original MOEA/D (MOEA/D-SBX) is easy to lose the diversity of the solutions which has been proved by the shape of PF. Among the eight problems, the whole PF is obtained only on two problems. The results also tally with the motivation of our work. Some reasons for the poor performance on diversity have been shown in Part III. The solution cannot move along its own intended track in the framework of MOEA/D-SBX. Thus some parts of the PF cannot be found.

Compared with MOEA/D-SBX, QMOEA/D has a significant improvement. In order to show the results more precisely, Table 2 presents the average, best and std of IGDs with the 30 independent runs on the 8 problems. The better one is marked with bold style.

From the view of signed-ranks test, QMOEA/D outperforms MOEA/D-SBX on all the problems except ZDT2. Especially for ZZZ1, ZZZ2, ZZZ5, ZZZ6 and MOP1, the IGD values have great improvement than the original MOEA/D. Because IGD is an index evaluate the solutions from both the convergence and diversity. Generally speaking, the huge difference can be attributed to the failure of MOEA/D-SBX in obtaining the complete PF.

4.5 Comparison between QMOEA/D and some advanced versions of MOEA/D

In this subsection, two advanced versions of MOEA/D are used as the comparison algorithms. One is MOEA/D-DE (Li and Zhang 2009) and the other is MOEA/D-DRA (Zhang et al. 2009).

In MOEA/D-DE, three aspects are modified based on MOEA/D-SBX. (1) The differential evolution (DE) operator, instead of SBX, is used to generate new solutions; (2) the neighbors are used as the mating/update range with probability δ . Otherwise, the whole subproblems are used as the mating/update range; (3) the new generated solutions

Table 2 The average, best and std of IGDs among the 30 independent runs with QMOEA/D and MOEA/D-SBX

	QMOEA/D	MOEA/D-SBX
ZDT1		
Average	0.0019	0.0019 (-)
Best	0.0019	0.0019
Std	2.1970e-006	3.0931e-006
ZDT2		
Average	0.0019	0.0019 (+)
Best	0.0019	0.0019
Std	8.0367e-007	1.0515e-006
ZZJ1		
Average	0.0019	0.2406(-)
Best	0.0019	0.1563
Std	2.4867e-005	0.0285
ZZJ2		
Average	0.0019	0.6097(-)
Best	0.0019	0.6097
Std	1.5081e-005	1.1292e-016
ZZJ5		
Average	0.0025	0.3569(-)
Best	0.0023	0.2918
Std	1.0848e-004	0.0335
ZZJ6		
Average	0.0033	0.6097(-)
Best	0.0022	0.6097
Std	7.7697e-004	1.1292e-016
MOP1		
Average	0.0440	0.3263(-)
Best	0.0354	0.2856
Std	0.0059	0.0121
MOP2		
Average	0.2485	0.3543(-)
Best	0.1870	0.3543
Std	0.0428	8.1818e-017

can update n_r solutions stored in mating/update range at most. These are the differences between MOEA/D-DE and MOEA/D-SBX. The details of MOEA/D-DE can be found in Li and Zhang (2009).

In MOEA/D-DRA, except the three aspects mentioned above, a utility value is stored for each subproblem. The utility value is calculated based on the relative decrease of the scalar function value during several generations. It is used to allocate the compute resources to different subproblems. The details of MOEA/D-DRA can be found in Zhang et al. (2009).

In this subsection, the common parameters are the same as above subsection. The population size is 200. The neighbor size is 20. Tchebycheff approach is also used here. 3×10^5

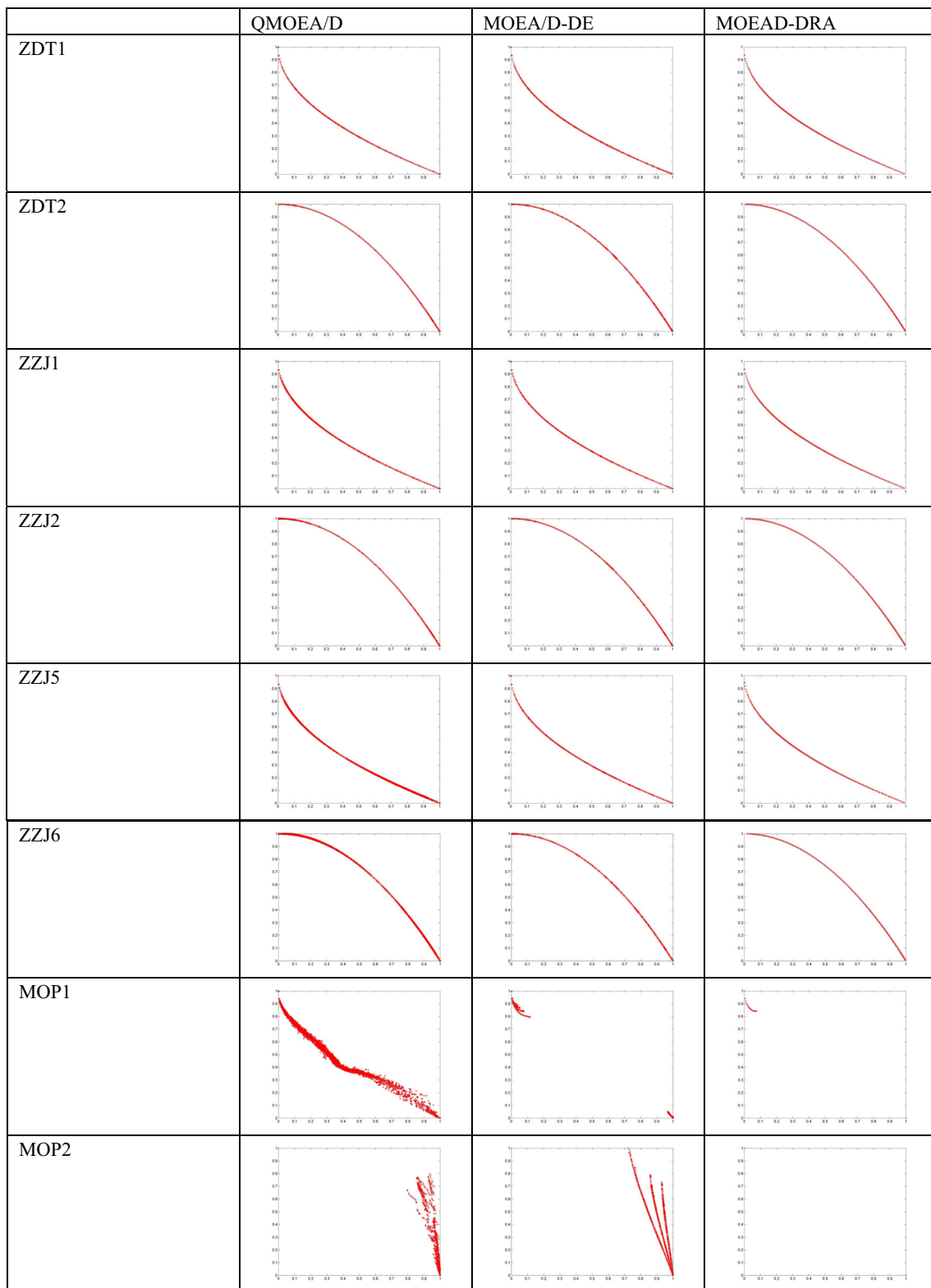


Fig. 9 The PFs from the 30 independent runs on the 8 test problems. The *left* is the results of QMOEA/D. The *middle* is the results of MOEA/D-DE. The *right* is the results of MOEA/D-DRA

fitness evaluations are used as the termination condition. The distribution index for polynomial mutation is 20 and the mutation rate is $1/n$. n is the dimension of decision space.

Here the F and CR in DE is set to 0.5 and 1.0, respectively, like Li and Zhang (2009). n_r and δ used in MOEA/D-DE and MOEA/D-DRA are set to 2 and 0.9, respectively. In

MOEA/D-DRA, the utility value is updated every 50 generations.

For these three algorithms, the experiments are run 30 times independently. Figure 9 shows all the PFs obtained in the 30 runs. From the left to the right, the figures respond to QMOEA/D, MOEA/D-DE and MOEA/D-DRA, respectively.

From Fig. 9, it can be found that all these three algorithms have better performance than MOEA/D-SBX. On the first six problems, all these three algorithms can obtain the complete PF. But on MOP1, only QMOEA/D obtains the relatively complete PF. On MOP2, the solutions from MOEA/D-DRA focus on the two ends of PF. From the view of PF shape, QMOEA/D and MOEA/D-DE have a similar performance on this problem.

In MOEA/D-DE, the new generated solution can update only n_r solutions instead of updating T solutions. This strategy may reduce the negative influence caused by other subproblems. It will help the solution move along its own track. The same mechanism also exists in MOEA/D-DRA. This is the reason why these two algorithms have a good performance on the first six problems. For the last two problems, there are some deception regions in their decision space. It is a challenge for many algorithms.

Some more precise values are given in the following. Table 3 shows the average, best and std IGDs of the 30 independent runs on each problems. The best one is marked with bold style.

For the average IGDs, QMOEA/D obtains 6 best results among the 8 ones. On ZZJ5 and ZZJ6, the IGD values of QMOEA/D are slightly worse than the best ones. On MOP1 and MOP2, the best ones com from QMOEA/D. Especially on MOP1, the advantage is greater than the results of the other two algorithms. For the best IGDs, QMOEA/D obtains 5 best results among the 8 ones. On ZZJ5, ZZJ6, and MOP2, the results of QMOEA/D are worse than the best one. But the shortage is slight. On MOP1, the best value also belongs to QMOEA/D with huge advantage.

Generally speaking, QMOEA/D has good performance on most of the test problems. Although not all the best results are from QMOEA/D, QMOEA/D can offer competitive results among these advanced versions of MOEA/D.

4.6 Some further discussion of the operators in QMOEA/D

As described in the above section, two changes are proposed in QMOEA/D. One is the stored GS and LS. The other is the quantum-inspired generator. In this subsection, the different effect of the two changes in QMOEA/D will be discussed based on some results of the comparison experiments.

A comparison algorithm called SMOEA/D is designed based on QMOEA/D. In SMOEA/D, GS and LS are also

Table 3 The average, best and std of IGDs among the 30 independent runs with QMOEA/D, MOEA/D-DE and MOEA/D-DRA

	QMOEA/D	MOEA/D-DE	MOEA/D-DRA
ZDT1			
Average	0.0019	0.0019 (-)	0.0021(-)
Best	0.0019	0.0019	0.0021
Std	2.1970e-006	5.8507e-006	8.8219e-019
ZDT2			
Average	0.0019	0.0019 (-)	0.0019 (-)
Best	0.0019	0.0019	0.0019
Std	8.0367e-007	2.4349e-006	6.6164e-019
ZZJ1			
Average	0.0019	0.0019 (+)	0.0021(-)
Best	0.0019	0.0019	0.0021
Std	2.4867e-005	2.2132e-006	4.4109e-019
ZZJ2			
Average	0.0019	0.0019 (+)	0.0019 (-)
Best	0.0019	0.0019	0.0019
Std	1.5081e-005	6.7863e-007	1.1027e-018
ZZJ5			
Average	0.0025	0.0019 (+)	0.0021(+)
Best	0.0023	0.0019	0.0021
Std	1.0848e-004	4.1794e-006	1.7644e-018
ZZJ6			
Average	0.0033	0.0829(-)	0.0021 (+)
Best	0.0022	0.0019	0.0021
Std	7.7697e-004	0.2101	8.8219e-019
MOP1			
Average	0.0440	0.3278(-)	0.3233(-)
Best	0.0354	0.2997	0.3233
Std	0.0059	0.0143	0
MOP2			
Average	0.2485	0.2604(=)	0.3543(-)
Best	0.1870	0.1710	0.3543
Std	0.0428	0.0657	5.6460e-017

stored and used as the parent solutions for each subproblem. But the new solutions are generated with simulated binary crossover (SBX), which means *step 1* in *Algorithm 2* is replaced with SBX.

The distribution index in SBX is set to be 20. For both QMOEA/D and SMOEA/D, the population size is 200. And the stopping condition is 3×10^5 fitness evaluations. 30 independent runs are also performed.

The PF shape of QMOEA/D and SMOEA/D are plotted in Fig. 10. Table 4 shows the average, best and std of IGDs on different problems, respectively. The better one is marked with bold style.

Firstly, the discussion is about the use of GS and LS. From Figs. 8 and 10, it can be found that the PFs obtained

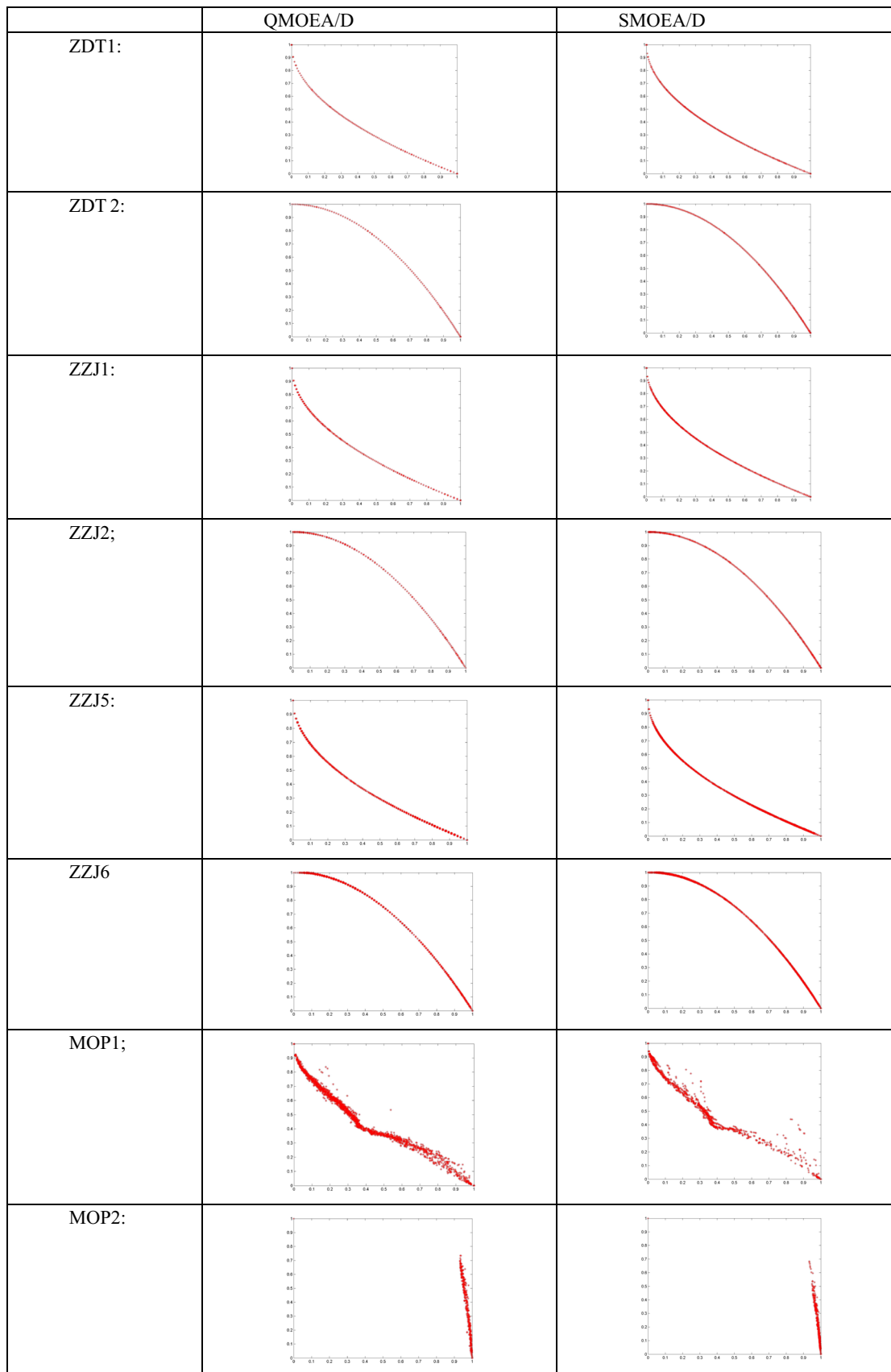


Fig. 10 The PFs from the 30 independent runs on the 8 test problems. The *left* is the results of QMOEA/D. The *right* is the results of SMOEA/D

by SMOEA/D and QMOEA/D are more complete than the PFs obtained by the original MOEA/D. Although the generators are different in QMOEA/D and SMOEA/D, in these two algorithms the GS and LS are stored and used for one subproblem. So it can be concluded that the use of LS and GS is helpful to retain the diversity of solutions in the framework of decomposition.

Secondly, the discussion is around the quantum-inspired generator. The main difference between QMOEA/D and SMOEA/D focuses on the different generators. In Table 4, different performance on the value of IGDs are shown. For ZDT1, ZDT2, ZZZ1, ZZZ2 and ZZZ6, the IGDs are similar for both the methods. With signed-rank test, it can be found that QMOEA/D is still better than SMOEA/D on two of them. For ZZZ5, MOP1 and MOP2, QMOEA/D is advantageous to SMOEA/D on both the average and best value. From the view of signed-rank test, QMOEA/D also outperforms SMOEA/D significantly on these three problems. Although the SMOEA/D has the ability to retain the relatively complete shape of PF. The quality of the solutions is not as good as the results obtained by QMOEA/D. As described above, the only difference between these two algorithms is the generator. With the same parents, GS and LS, the solutions obtained by QMOEA/D have better performance than the solutions offered by SMOEA/D. The advantages should be contributed to the generator used in QMOEA/D. It means the quantum-inspired generator is more effective to improve the performance of original MOEA/D with GS and LS.

5 Summary

MOEA/D decomposes a MOP into a number of single objective optimization problems. This framework provides us great convenience to the use of some methods which are widely applied in single objective optimization. But subproblems in MOEA/D are not totally the same as the simple single objective problems. In order to use the approaches in this framework effectively, more special features of MOEA/D should be mined.

In this paper, we combine the quantum-inspired method with MOEA/D because of the poor performance on the diversity. The subproblems in MOEA/D are some associated problems. The neighboring weight vectors respond to the similar subproblems. The information from the neighbor problems may make the solution deviate from its own track and reduce the diversity of the solutions. So we store LS and GS for each subproblem. Besides, special attractor and characteristic length are designed for MOEA/D.

And the experimental results show that the QMOEA/D is an effective and competitive algorithm for solving MOPs. But on some test problems such as MOP2, no excellent results are obtained. More study about MOEA/D are needed. Some

Table 4 The average, best and std of IGDs among the 30 independent runs with QMOEA/D and SMOEA/D

	QMOEA/D	SMOEA/D
ZDT1		
Average	0.0019	0.0019 (-)
Best	0.0019	0.0019
Std	2.1970e-006	2.3930e-006
ZDT2		
Average	0.0019	0.0019 (=)
Best	0.0019	0.0019
Std	8.0367e-007	4.2073e-007
ZZJ1		
Average	0.0019	0.0020(-)
Best	0.0019	0.0019
Std	2.1867e-005	2.4046e-005
ZZJ2		
Average	0.0019	0.0019 (=)
Best	0.0019	0.0019
Std	1.5081e-005	3.9973e-005
ZZJ5		
Average	0.0025	0.0028(-)
Best	0.0023	0.0026
Std	1.0848e-004	8.5506e-005
ZZJ6		
Average	0.0033	0.0033 (=)
Best	0.0022	0.0023
Std	7.7697e-004	4.8834e-004
MOP1		
Average	0.0440	0.1415(-)
Best	0.0354	0.0430
Std	0.0059	0.0512
MOP2		
Average	0.2485	0.2628(-)
Best	0.1870	0.2360
Std	0.0428	0.0186

The bold values show the better one in the comparison

more special behavior characters should be discovered for designing more reasonable approach.

Acknowledgments This work was supported by the Cheung Kong Scholars Program of China (Grant No. K5051302050), the Program for New Century Excellent Talents in University (No. NCET-12-0920), the Program for New Scientific and Technological Star of Shaanxi Province (No. 2014KJXX-45), the National Natural Science Foundation of China (Nos. 61272279, 61272282, 61001202 and 61203303), the Fundamental Research Funds for the Central Universities (Nos. K5051302049, K5051302023, K50511020011, K5051302002 and K5051302028), the Provincial Natural Science Foundation of Shaanxi of China (No. 2011JQ8020), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048) and EU IRSES project (No. 247619).

References

- Abs da Cruz AV, Barbosa CRH, Pacheco MAC, Vellasco MBR (2004) Quantum-inspired evolutionary algorithms and its application to numerical optimization problems. *Lecture notes in computer science*, pp 212–217
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Elhossini A, Areibi S, Dony R (2010) Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evol Comput* 18(1):127–156
- García S, Fernández A, Luengo J et al (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability[J]. *Soft Comput* 13(2):959–977
- Han K-H, Kim J-H (2002) Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans Evol Comput* 6:580–593
- Ishibuchi H, Nojima Y (2011) Performance evaluation of evolutionary multiobjective optimization algorithms for multiobjective fuzzy genetics-based machine learning. *Soft Comput* 15(12):2415–2434
- Jiao L, Li Y, Gong M, Zhang X (2008) Quantum-inspired immune clonal algorithm for global numerical optimization. *IEEE Trans Syst Man Cybern Part B Cybern* 38(5):1234–1253
- Jixiang C, Gexiang Z, Zhidan L, Yuquan L (2012) Multi-objective ant colony optimization based on decomposition for bi-objective traveling salesman problems. *Soft Comput* 16(4):597–614
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets. MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302
- Li H, Landa-Silva D (2011) An adaptive evolutionary multi-objective approach based on simulated annealing. *Evol Comput* 19(4):561–595
- Li Y, Xiang R, Jiao L, L Ruochen (2012) An improved cooperative quantum-behaved particle swarm optimization. *Soft Comput* 16(6):1061–1069
- Liu HL, Gu F, Zhang Q (2014) Decomposition of a multiobjective optimization problem into a number of simple multiobjective sub-problems[J]. *IEEE Trans Evol Comput* 18(3):450–455
- Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscipl Optim* 26(6):369–395
- Montaño AA, Coello Coello CA, Mezura-Montes E (2010) MODE-LD+SS: a novel differential evolution algorithm incorporating local dominance and scalar selection mechanisms for multi-objective optimization. In: 2010 IEEE Congress on evolutionary computation (CEC'2010), Barcelona, Spain, July 18–23. IEEE Press, pp 3284–3291
- Narayanan A, Moore M (1996) Quantum-inspired genetic algorithms. In: Proceedings of the 1996 IEEE international conference on evolutionary computation. IEEE Press, Piscataway, pp 61–66
- Sierra MR, Coello Coello CA (2005) A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In: Proceedings of Congress on evolutionary computation (CEC 2005), Edinburgh, UK, pp 65–72
- Sindhya K, Ruuska S, Haanp T, Miettinen K (2011) A new hybrid mutation operator for multiobjective optimization with differential evolution. *Soft Comput* 15(2):2041–2055
- Sun J et al (2004) Particle swarm optimization with particles having quantum behavior. In: Proceedings of 2004 Congress on evolutionary computation, pp 325–331
- Sun J, Fang W, Wu X, Palade V, Xu W (2012) Quantum-behaved particle swarm optimization: analysis of the individual particle's behavior and parameter selection. *Evol Comput* 20(3):349–393
- Zapotecas Martínez S, Coello Coello CA (2011) A multi-objective particle swarm optimizer based on decomposition. In: Proceedings of the 13th annual genetic and evolutionary computation conference (GECCO'2011), Dublin, Ireland, July 2011. ACM, pp 69–76
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
- Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans Evol Comput* 14(3):456–474
- Zhang Q, Liu W, Li H (2009) The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, Tech. Rep. CES-491, The School of Computer Science and Electronic Engineering, University of Essex
- Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
- Zitzler E, Laumanns M, Thiele L (2002) SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary methods for design optimisation and control*, pp 95–100