Article:

Cao, B., Fan, S., Zhao, J. et al. (3 more authors) (2020) Quantum-enhanced multiobjective large-scale optimization via parallelism. Swarm and Evolutionary Computation, 57. 100697. ISSN 2210-6502

https://doi.org/10.1016/j.swevo.2020.100697

Quantum-Enhanced Multiobjective Large-scale Optimization via Parallelism

Bin Cao[a,b,c], Shanshan Fan[a,b,c], Jianwei Zhao[a,b,c,*], Po Yang[d,*], Khan Muhammad[e,*], Mohammed Tanveer[f]

[a] *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300401, China.*
[b] *School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China.*
[c] *Hebei Provincial Key Laboratory of Big Data Calculation, 300401 Tianjin, China*
[d] *Department of Computer Science, Sheffield University, UK*
[e] *Department of Software, Sejong University, Seoul, Republic of Korea*
[f] *Discipline of Mathematics, Indian Institute of Technology, Indore, Republic of India*

Abstract

Traditional quantum-based evolutionary algorithms are intended to solve single-objective optimization problems or multiobjective small-scale optimization problems. However, multiobjective large-scale optimization problems are continuously emerging in the big-data era. Therefore, the research in this paper, which focuses on combining quantum mechanics with multiobjective large-scale optimization algorithms, will be beneficial to the study of quantum-based evolutionary algorithms. In traditional quantum-behaved particle swarm optimization (QPSO), particle position uncertainty prevents the algorithm from easily falling into a local optimum. Inspired by the uncertainty principle of position, the authors propose quantum-enhanced multiobjective large-scale algorithms, which are parallel multiobjective large-scale evolutionary algorithms (PMLEAs). Specifically, PMLEA-QDE, PMLEA-QjDE and PMLEA-QJADE are proposed by introducing the search mechanism of the individual particle from QPSO into differential evolution (DE), differential evolution with self-adapting control parameters (jDE) and adaptive differential evolution with optional external archive (JADE). Moreover, the proposed algorithms are implemented with parallelism to improve the optimization efficiency. Verifications performed on several test suites indicate that the proposed quantum-enhanced algorithms are superior to the state-of-the-art algorithms in terms of both effectiveness and efficiency.

Keywords: Quantum mechanics, Multiobjective large-scale optimization, Quantum-inspired evolutionary algorithm (QIEA), Large-scale optimization

1. Introduction

The quantum-inspired evolutionary algorithm (QIEA) combines the evolutionary algorithm (EA) and quantum computation, achieving a balance between exploration and exploitation [1]. Compared with EAs, QIEAs use the probability amplitude representation of qubits to encode chromosomes. The use of the quantum rotation gate update strategy allows QIEAs to converge more quickly [2]. Quantum gate updating is a key step in quantum evolutionary algorithms (QEAs). Xiong et al. [3] summarized the most commonly used quantum rotation gates. The superposition and entanglement of the quantum state provides QIEAs with the potential to apply parallelism in the process of evolution [4]. Patvardhan et al. [4] proposed a parallel improved quantum inspired evolutionary algorithm (IQIEA-P) with a high acceleration ratio for large-size quadratic knapsack problems, which have only one objective.

In addition to single-objective optimization problems, many real-world problems need to optimize multiple conflicting objectives simultaneously. Problems with two or three objectives are usually called multiobjective optimization problems (MOPs). Problems with more than three objectives are called many-objective optimization problems (MaOPs). Moreover, many practical optimization problems have hundreds of decision variables [5, 6], which are referred to as large-scale optimization problems. Problems with two or three objectives and a large number of decision variables (usually more than 100) are denoted as multiobjective large-scale optimization problems (MOLSOPs).

Considering the excellent diversity characteristics of quantum systems, many studies have combined quantum computation with single-objective EAs and applied them to numerical optimization [7], combinatorial opti-

*Corresponding authors.

mization [8], production scheduling [9], vehicle routing [10], and other fields. Pavithr [11] proposed a hybrid quantum-inspired social evolutionary algorithm (QSE) that performed well on the 0-1 knapsack problem. Dahi et al. [12] proposed a quantum-inspired genetic algorithm (QIGA) with new quantum gates to address the antenna positioning problem. Alanis et al. [13] proposed a nondominated quantum optimization algorithm (NDQO) to optimize a multiobjective routing problem. Li et al. [14] proposed a quantum memetic algorithm (QMA) by introducing cultural evolution. Some scholars have combined differential evolution (DE) [15, 16] with quantum computation. Hu et al. [17] combined quantum-behaved particle swarm optimization (QPSO) [18], DE and the tabu search algorithm [19], proposing the hybridized vector optimal algorithm QPSO-DET, which better balances the relationship between local search and global search. SaiToh et al. [20] showed that even with the introduction of quantum mutation operators, the algorithm is sometimes prone to fall into local search. Therefore, a quantum crossover process that crosses all chromosomes in each generation was proposed. Based on QEAs, Ren et al. [21] proposed a hybrid quantum differential evolution algorithm (HQDE) that updates quantum chromosomes by quantum differential evolution (QDE) and quantum harmony search (QHS).

However, quantum theory has rarely been applied to solve large-scale optimization problems. Ding et al. [22] proposed a single-objective quantum cooperative coevolution algorithm for attribute reduction (QCCAR) with respect to large data sets by combining the cooperative coevolutionary (CC) [23] framework with a QEA. Tian et al. [24] combined the QPSO algorithm with the CC framework and proposed the single-objective QPSO_CC framework to solve large-scale optimization problems. They used the random decomposition strategy to separate the search space and used QPSO to optimize each subgroup. Fang et al. [25] proposed a random selection decomposition strategy based on random dimension reduction to solve large-scale optimization problems and proposed the RSQPSO algorithm based on the QPSO and random selection strategy. The above three algorithms have applied the CC framework and QIEA for large-scale optimization but only been used for single-objective large-scale optimization problems.

Traditional EAs have been applied in many fields [26, 27, 28], but their optimization performance substantially decreases as the number of decision variables increases. Research on multiobjective large-scale EAs is both popular and difficult [29, 30, 31, 32]. Among these algorithms, the variable grouping and CC strategy are helpful in improving the optimization performance with respect to large-scale problems.

Some scholars have combined quantum mechanics with multiobjective EAs. Kumari et al. proposed a quantum heuristic multiobjective differential evolution algorithm (QMDEA) [33] and a multiobjective quantum heuristic hybrid differential evolution algorithm (MQHDE) [34] to balance exploration and exploitation. All these methods have combined DE with a genetic algorithm (GA) and quantum computation to form multiobjective frameworks, contributing to the balance between convergence and diversity in multiobjective optimization algorithms [35]. Li et al. [36] proposed the quantum behavioral discrete multiobjective particle swarm optimization (QDM-PSO) algorithm and applied it to a large-scale complex network clustering problem. Mouradian et al. [37] modeled task allocation for a large number of robots in a large-scale natural environment as a multiobjective problem and proposed the quantum multiobjective particle swarm optimization (QMOPSO) algorithm. Mousavi et al. [38] used a QEA to solve the computational complexity of coalition formation in large-scale unmanned aerial vehicle (UAV) networks. Tang et al. [39] proposed a QPSO with memetic algorithm and memory (SMQPSO) algorithm to solve continuous nonlinear large-scale problems.

Distributed and parallel algorithms [40] can capitalize on large numbers of computing resources and substantially reduce algorithm time consumption, improving algorithm efficiency [41]. Tan et al. [42] proposed a distributed coevolution multiobjective optimization algorithm. Cao et al. proposed a distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm (DPCCMOEA) [43] based on an improved variable analysis strategy and a distributed parallel cooperative coevolutionary multiobjective large-scale evolutionary algorithm (DPCCMOLSEA) [44] to solve MOLSOPs. Both algorithms are based on a decomposition strategy in which the variables are broken down into groups, and each group is optimized by one subpopulation using the DE operator [15, 16]. Based on DPCCMOLSEA, we propose the parallel multiobjective large-scale evolutionary algorithm(PMLEA) with either quantum-enhanced DE, quantum-enhanced differential evolution with self-adapting control parameters (jDE) or quantum-enhanced adaptive differential evolution with optional external archive (JADE), denoted as PMLEA-QDE, PMLEA-QjDE and PMLEA-QJADE, respectively.

The contributions of the present study include the following:

1. We integrate the position update strategy based on the theory of quantum mechanics in QPSO into the DE operator of the DPCCMOLSEA framework to optimize the population.
2. Based on jDE and JADE, we propose the variants PMLEA-QjDE and PMLEA-QJADE, in which the adaptive parameters are quantized.
3. The integration of parallel operation based on the message passing interface (MPI) substantially reduces the runtime of the quantum-enhanced algorithm.

The organization of this paper is as follows. Section 2 introduces the large-scale MOPs and the QPSO algorithm. The proposed methodology is described in Section 3. Section 4 reports the experimental comparison results and provides an analysis. Finally, Section 5 summarizes this paper.

2. Related Work

2.1. MOLSOPs

MOPs in which the decision variable number is greater than or equal to 100 are called MOLSOPs. In general, an MOP with N decision variables and M objective variables can be described as follows [34, 45]:

$$\min \mathrm{F}(\mathrm{x}) = (f_1(x), f_2(x), \cdots, f_M(x)) \in R^M \quad (1)$$
$$\text{s.t. } \mathrm{x} = \{x_1, \cdots, x_N\} \in \Omega \subset R^N$$

where x is a decision vector in decision space Ω, $N \geq 100$, and F(x) is an objective vector located in the objective space, $M \leq 3$.

2.2. Quantum-behaved Particle Swarm Optimization

The QPSO algorithm is based on the quantum potential well model inspired by the principles of quantum mechanics. It establishes an attractive potential that affects the individuals in a population, in which each particle is attracted by a quantum potential well whose center is located at its local attractor. The randomness of the particle position in QPSO improves its global search capability.

In standard particle swarm optimization (PSO) [46], each particle moves in an N-dimensional space according to the following equations:

$$V_{i,j}^{g+1} = \omega V_{i,j}^{g} + c_1 r_{i,j}^{g}\left(P_{i,j}^{g} - X_{i,j}^{g}\right) + c_2 R_{i,j}^{g}\left(G_{j}^{g} - X_{i,j}^{g}\right) \quad (2)$$

$$X_{i,j}^{g+1} = X_{i,j}^{g} + V_{i,j}^{g+1} \quad (3)$$

where V_i^g denotes the velocity vector, X_i^g denotes the position vector, $i \in \{1, 2, \ldots, NP\}$ denotes the individual index, NP denotes the population size, $j \in \{1, 2, \ldots, N\}$ denotes the variable index, g denotes the current generation number, ω denotes the inertia weight, c_1 and c_2 are acceleration coefficients, $P_i^g = \left(P_{i,1}^g, P_{i,2}^g, \ldots, P_{i,N}^g\right)$ is the best previous position of particle i and is referred to as the personal best position (pbest), and $G^g = \left(G_1^g, G_2^g, \ldots, G_N^g\right)$ is the best particle position in the population and is called the global best location (gbest).

Different from the particles in PSO, which are represented by both position and velocity, only positional information is used to describe the particles in QPSO, and the local attractor of particle i is a random position. Specifically, for each dimension of particle i, the position of a random point is calculated first as follows:

$$p_{i,j}^g = \varphi_{i,j}^g P_{i,j}^g + \left(1 - \varphi_{i,j}^g\right) G_j^g,\ \varphi_{i,j}^g = U(0,1) \quad (4)$$

where $\varphi_{i,j}^g$ denotes a random number, and $U(0,1)$ denotes a uniformly generated random number in $[0, 1)$. Then, the whole particle position can be calculated, and the corresponding offspring i is generated as follows:

$$X_{i,j}^{g+1} = p_{i,j}^g \pm \alpha \left|X_{i,j}^g - C_j^g\right| \ln\left(1/u_{i,j}^g\right) \quad (5)$$

$$C_j^g = \frac{1}{NP}\sum_{i=1}^{NP} P_{i,j}^g \ (1 \leq j \leq N) \quad (6)$$

where C_j^g is the average of the pbest positions of all particles in the j-th dimension, α denotes the contraction expansion (CE) coefficient controlling the convergence speed, and $u_{i,j}^g = U(0,1)$ and $u_{i,j}^g > 0$ is a random number.

3. The Proposed Quantum-enhanced Algorithm

In QPSO, the randomness of the particle position causes it to have better global search capability. Therefore, inspired by the theory of position update in QPSO and based on the DPCCMOLSEA framework, we propose PMLEA-QDE, PMLEA-QjDE and PMLEA-QJADE.

DPCCMOEA [43] and DPCCMOLSEA [44] both rely on decomposition to solve MOLSOPs. In this section, we describe DPCCMOEA, DPCCMOLSEA, and the proposed quantum-enhanced algorithms.

3.1. DPCCMOEA

3.1.1. Overall architecture

In the first layer, the variables are decomposed into several groups, each of which is optimized by a subpopulation. In the second layer, each CPU core is responsible for the evolution and evaluation of the individuals.

3.1.2. Optimization

Each individual relies on neighboring individuals or the whole subpopulation to share information. However, the individuals in each subpopulation are divided into multiple sets. To reduce the amount of communication, the set of individuals in each CPU core can obtain only the information of the individual sets in the adjacent CPU cores. Each variable j of the partial trail vector $trail_{i,j}$ is as follows:

$$trail_{i,j} = p_{i,j} + F \times \left(p_{a_1,j} - p_{a_2,j}\right) \quad \text{s.t. } i \in \{1, 2, \ldots, NP\},\ j \in S_{opt} \tag{7}$$

where i is selected by the binary tournament method, p_i is the decision vector, j is the index of the decision variable, a_1 and a_2 are randomly selected solutions, and S_{opt} is the variable group for optimization with respect to the current CPU core.

3.1.3. Crossover

To evaluate the fitness, the remaining variables of the trail vector should be generated to form a complete solution. For which, the crossover strategy is employed as follows:

$$trail_{i,j} = \begin{cases} p_{i,j} & \text{if } j \notin S_{opt} \wedge r_1 < 0.5 \\ p_{b_1,j} & \text{if } j \notin S_{opt} \wedge r_1 > 0.5 \wedge r_2 \leq 0.5 \\ p_{b_2,j} & \text{otherwise} \end{cases} \tag{8}$$

where $r_1, r_2 = U(0, 1)$ are random numbers, and b_1 and b_2 are randomly selected solutions satisfying $b_1 \neq b_2 \neq i$.

3.1.4. Mutation

The generated $trail_i$ vector is mutated with the probability of $1/N$ via polynomial mutation. Finally, the population update refers to MOEA/D [47].

3.2. DPCCMOLSEA

3.2.1. Overall architecture

In contrast to DPCCMOEA, in the second layer of DPCCMOLSEA, in each subpopulation, a master CPU core is responsible for the evolution of each subpopulation, while the computational burdens (i.e., the fitness evaluations) are shared across all the CPU cores.

3.2.2. Optimization

In DPCCMOEA, each subpopulation is separated to several sets, each of which is in the charge of one CPU core. Therefore, all individual sets are evolved in parallel. Different from DPCCMOEA, in DPCCMOLSEA, all individuals in each subpopulation are evolved in one corresponding master CPU core in serial, resulting in better utilization of the information between individuals in each subpopulation.

3.2.3. Crossover

Different from DPCCMOEA, which uses a fixed crossover rate, DPCCMOLSEA adopts an adaptive strategy [48]:

$$CR_i = \text{GaussRand}\,(\mu_1, 0.1) \tag{9}$$

where CR_i represents the crossover probability of the i-th individual and satisfies the Gaussian distribution with mean value of μ_1 and a deviation factor of 0.1. The update of μ_1 satisfies the following equation:

$$\mu_1 = (1 - c) \times \mu_1 + c \times \text{mean}_A\,(S_{CR}) \tag{10}$$

where c is 0.1, $\text{mean}_A\,(S_{CR})$ returns the mean of all elements in the set S_{CR}, and S_{CR} stores the CR values of successfully evolved individuals.

3.3. The Proposed Algorithm

Although DE converges quickly, it can easily fall into local optima. In QPSO, the bound-state particles described by the probability density function can appear in any interval throughout the feasible solution space with a certain probability. Based on the above considerations, we integrate the theory of position updating in QPSO into DE and its variants (jDE and JADE). The proposed quantum-enhanced algorithms are detailed as follows.

3.3.1. Parameter quantization: PMLEA-QDE

Considering the establishment of an attractive potential that affects individuals in the population, the δ potential well field produces a better effect [18]. To determine the exact position of the individual, the quantum state must be collapsed to the classical state; then, the particle position is measured by a Monte Carlo stochastic simulation. Each variable of an individual moves in an one-dimensional δ potential well centered at point p [18], and its position can be calculated via the following stochastic equation:

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

where L is the feature length of the δ potential well, and $u = U(0,1) \wedge u \neq 0$. The above results can be extended to the N-dimensional space. The basic evolution equation for the j-th variable of individual i is

$$X_{i,j}^{g+1} = p_{i,j}^{g} \pm \frac{L_{i,j}^{g}}{2} \ln\left(1/u_{i,j}^{g}\right) \tag{12}$$

It is proven that in an N-dimensional space, the necessary and sufficient condition for the position of individual i evolved through the above process to converge in probability to its attractor $p_i^g = \left(p_{i,1}^g, p_{i,2}^g, \cdots, p_{i,N}^g\right)$ is that each dimensional coordinate $X_{i,j}^{g+1}$ converges in probability to $p_{i,j}^g$ [18].

The necessary and sufficient condition for the position of an individual to converge in probability to the attractor is $\lim_{k\to\infty} L_{i,j}^g = 0$. Accordingly, to make the individual converge to the local attractor, controlling $L_{i,j}^g$ causes convergence to 0. Therefore, the average best position p_{ave}^g, that is, the average of the best positions of all individuals, is introduced into the algorithm [49, 50]:

$$p_{ave,j}^{g} = \frac{1}{NP} \sum_{i=1}^{NP} P_{i,j}^{g} \tag{13}$$

Then, $L_{i,j}^g$ is evaluated by the following formula:

$$L_{i,j}^{g} = 2\alpha \times \left| P_{ave,j}^{g} - X_{i,j}^{g} \right| \tag{14}$$

Finally, the evolutionary formula for an individual becomes:

$$X_{i,j}^{g+1} = p_{i,j}^{g} \pm \alpha \times \left(P_{ave,j}^{g} - X_{i,j}^{g}\right) \times \ln\left(1/u_{i,j}^{g}\right) \tag{15}$$

where α is the CE coefficient, and $u_{i,j}^g = U(0,1) \wedge u_{i,j}^g \neq 0$ is a random number.

When optimizing the population by DE, the scaling factor F is a key coefficient in the optimization process. If F is too large, then it contributes to population diversity but the convergence is slow, reducing the search efficiency. In contrast, F that is too small causes premature convergence. In standard DE, each decision vector X_i^g $(i = 1, 2, \ldots NP)$ produces a mutation vector. The mutation strategies include DE/rand/1, DE/current-to-best/1, DE/best/1, etc., and the most commonly utilized strategy is DE/rand/1:

$$V_i^g = X_{r_1}^g + F_i\left(X_{r_2}^g - X_{r_3}^g\right) \tag{16}$$

where i is the index of the current individual, V_i^g is the i-th variation vector generated in the g-th generation, and $r_1, r_2, r_3 \in \{1, 2, \cdots, NP\}$ are random numbers with $r_1 \neq r_2 \neq r_3 \neq i$, and F_i is the mutation scale factor in $(0, 1]$, being fixed or varied with evolution.

The principle of individual evolution in QPSO is integrated to the DE optimization process. In the quantum-enhanced algorithm, mutated individuals are generated as follows:

$$V_{i,j}^{g} = \begin{cases} X_{i,j}^{g} + \alpha \times \left(X_{r_1,j}^{g} - X_{r_2,j}^{g}\right) \times \ln\left(1/u_{i,j}^{g}\right) & \text{if } u_{i,j}^{g}{}' \leq 0.5 \\ X_{i,j}^{g} - \alpha \times \left(X_{r_1,j}^{g} - X_{r_2,j}^{g}\right) \times \ln\left(1/u_{i,j}^{g}\right) & \text{otherwise} \end{cases} \tag{17}$$

$$\text{s.t. } j \in S_{opt}$$

where $u_{i,j}^g{}' = U(0,1)$ denotes a random number.

3.3.2. Adaptive parameters with quantum: PMLEA-QjDE and PMLEA-QJADE

jDE [51] and JADE [48] are representative parameter-adaptive algorithms that can adjust both the crossover probability CR and the scaling factor F. Both jDE and JADE have achieved good optimization results on the standard test suites.

In jDE, before each generation, F and CR are updated using the following equations [51]:

$$F_i = \begin{cases} F_l + r_1 \times F_u & \text{if } r_2 < \tau_1 \\ F_i & \text{otherwise} \end{cases} \tag{18}$$

$$CR_i = \begin{cases} r_3 & \text{if } r_4 < \tau_2 \\ CR_i & \text{otherwise} \end{cases} \tag{19}$$

where $r_j = U(0,1)$ $(j \in \{1, 2, 3, 4\})$ are random numbers, and τ_1 and τ_2 are parameters.

By quantization, the final scale factor in use, F_i', in PMLEA-QjDE is as follows:

$$F_i' = \begin{cases} \ln\left[1/u_i\right] \times F_i & \text{if } r_5 \leq 0.5 \\ -\ln\left[1/u_i\right] \times F_i & \text{otherwise} \end{cases} \tag{20}$$

where $r_5 = U(0,1)$ is a random number.

JADE uses a parameter strategy based on statistical learning in which F and CR are dynamically adjusted according to previous successful experiences [48]. Specifically, CR is updated as follows:

$$CR_i = \text{GaussRand}\left(\mu_1, 0.1\right) \tag{21}$$

$$\mu_1 = (1 - c) \times \mu_1 + c \times \text{mean}_A\left(S_{CR}\right) \tag{22}$$

where CR_i obeys the Gaussian distribution with mean of μ_1 and a standard deviation of 0.1, c is a constant in $(0, 1)$, $\text{mean}_A(\cdot)$ denotes the usual arithmetic mean, and S_{CR} records the crossover probabilities CR_i that enable the corresponding offsprings successfully entering the next generation.

The scaling factor F is updated as follows:

$$F_i = \text{CauchyRand}(\mu_2, 0.1) \tag{23}$$

$$\mu_2 = (1 - c) \times \mu_2 + c \times \text{mean}_L (S_F) \tag{24}$$

where F_i obeys the Cauchy distribution with location parameter of μ_2 and a scale parameter of 0.1, c is a constant in $(0, 1)$, and $\text{mean}_L(\cdot)$ denotes the Lehmer mean. The scaling factor F_i that enables the corresponding offspring to successfully enter the next generation is recorded in S_F.

The strategy for quantizing F_i in PMLEA-QJADE is the same as in PMLEA-QjDE:

$$F_i' = \begin{cases} \ln[1/u_i] \times F_i & \text{if } r_6 \le 0.5 \\ -\ln[1/u_i] \times F_i & \text{otherwise} \end{cases} \tag{25}$$

where $r_6 = U(0, 1)$ is a random number.

4. Experimental Results and Analysis

4.1. Experimental Setup

We compared the proposed quantum-enhanced algorithms PMLEA-QDE, PMLEA-QjDE, and PMLEA-QJADE with PMLEA-DE, PMLEA-jDE, PMLEA-JADE, PMLEA-PSO, PMLEA-QPSO, the cooperative coevolutionary generalized differential evolution 3 (CCGDE3) algorithm [52], the multiobjective evolutionary algorithm based on decision variable analyses (MOEA/DVA) [53], MOEA/D [47], cooperative multiobjective differential evolution (CMODE) [54], nondominated sorting genetic algorithm II (NSGA-II) [55], weighted optimization framework-based speed-constrained multiobjective PSO (WOF-SMPSO) [56], large-scale multiobjective competitive swarm optimizer (LMOCSO) [57], large-scale multiobjective optimization framework (LSMOF) [58] and DPCCMOEA [43]. What should be mentioned is that, for multiobjective optimization with PMLEA-QPSO, there is not a global best individual simultaneously considering all objectives, and the central position averaging all personal best individuals may not contribute to the optimization of MOPs, therefore, in Eqs. 4 and 5, G^g and C^g are two distinct individuals, different from individual i, randomly selected in the niche or the whole population.

For the DE operator, we set F and CR, respectively, to 0.5 and 1.0; for the jDE and JADE operators, F and CR were both initially set to 0.5. CCGDE3 used a fixed grouping strategy, and the number of groups was set to 2, each of which are optimized by $NP/2$ individuals. In CMODE, 3 subpopulations were used when there were 3 objectives, and 2 subpopulations were used for 2 objectives. In addition, the size of each subpopulation was 20, and the archive sizes were 100 and 120 for 2 and 3 objectives, respectively. In MOEA/DVA, the repetition numbers of control variable analyses and interdependence analyses were set to 20 and 6, respectively. In DPCCMOEA, the above values were set to 20 and 1, which is the same for all PMLEA algorithms, and the group size threshold was set to 111, while it was 100 in all PMLEA algorithms. In MOEA/DVA, DPCCMOEA, all PMLEA algorithms, and MOEA/D, the niche size, the replacement limit of offspring individuals, and the probability of selecting a parent individual from niche were set to $0.1 \times NP$, 2, and 0.9, respectively.

MOEA/DVA and NSGA-II used simulated binary crossover (SBX) and polynomial mutation. MOEA/D, MOEA/DVA, DPCCMOEA and all PMLEA algorithms used polynomial mutation. The distribution indices of SBX and polynomial mutation were both set to 20, the crossover probability of SBX was 1.0, and the polynomial mutation probability was $1.0/N$.

The distributed parallel structure of the proposed algorithms was implemented via MPI and was tested on the Tianhe-2 supercomputer using a total of 72 CPU cores. All the comparison algorithms optimized each test instance for 20 times. In the experiments, we used the following test suites: DTLZ [59], WFG [60], LSMOP [61] and MaOP [62]. The numbers of variables in the DTLZ and WFG test problems were 200 and 300, respectively, for 2 and 3 objectives. For the 2-objective and 3-objective LSMOP instances, there were 206 and 307 variables, respectively. The number of variables in MaOP2 was 300, and the number of objectives was 3. We set the population size to 100 for algorithms with two objectives and 120 for algorithms with three objectives. The number of fitness evaluations was $N \times 10^4$.

4.2. Performance Measurement

Algorithm performance was measured by the inverted generational distance (IGD) [63, 64], which comprehensively measures the convergence and distribution of a generated Pareto front (PF). The IGD is defined as follows:

$$IGD(P, P^*) = \frac{\sum_{x=P} d(x, P^*)}{|P|} \tag{26}$$

where P is the point set uniformly sampled on the real PF, $|P|$ is the cardinality of set P, P^* denotes the Pareto solution set obtained by the optimized algorithm, and $d(x, P^*)$ is the minimum Euclidean distance between x

and individuals in P^*. Therefore, a smaller IGD value represents an approximated PF closer to the real PF, indicating better performance.

In addition, we used a nonparametric test to evaluate the algorithm more accurately [65, 66].

4.3. *Algorithm Comparison*

Table 1 lists the mean values of the IGD index values obtained by the algorithms on the 2- and 3-objective DTLZ, WFG, LSMOP and MaOP test suites; the bold numbers indicate two best values. The ranking results of all algorithms via the nonparametric Friedman tests with respect to the average IGD indicator values are listed in Table 2. Overall, PMLEA-QJADE achieves the best performance on the four test suites, PMLEA-JADE ranks the second, and PMLEA-QDE and PMLEA-QjDE perform well, while CCGDE3 performs the worst. Therefore, the proposed quantum-enhanced parallel multiobjective large-scale algorithms are superior to other state-of-the-art multiobjective EAs.

4.3.1. *Analysis of the experimental results on DTLZ and WFG*

Table 3 lists the ranking via the nonparametric Friedman tests with respect to the average IGD indicator values on the 2- and 3-objective DTLZ and WFG test suites. Furthermore, Figs. 1 and 2 illustrate the evolution curves of the IGD values of the 17 algorithms on the 2- and 3-objective DTLZ1, DTLZ3 and DTLZ6 functions as well as the 2- and 3-objective WFG4, WFG6 and WFG9 test functions. PMLEA-QDE achieves the best performance on the DTLZ1-7 and WFG1-9 test functions, PMLEA-QJADE performs well, and the performance of CCGDE3 is the worst.

4.3.2. *Analysis of the Experimental Results on LSMOP*

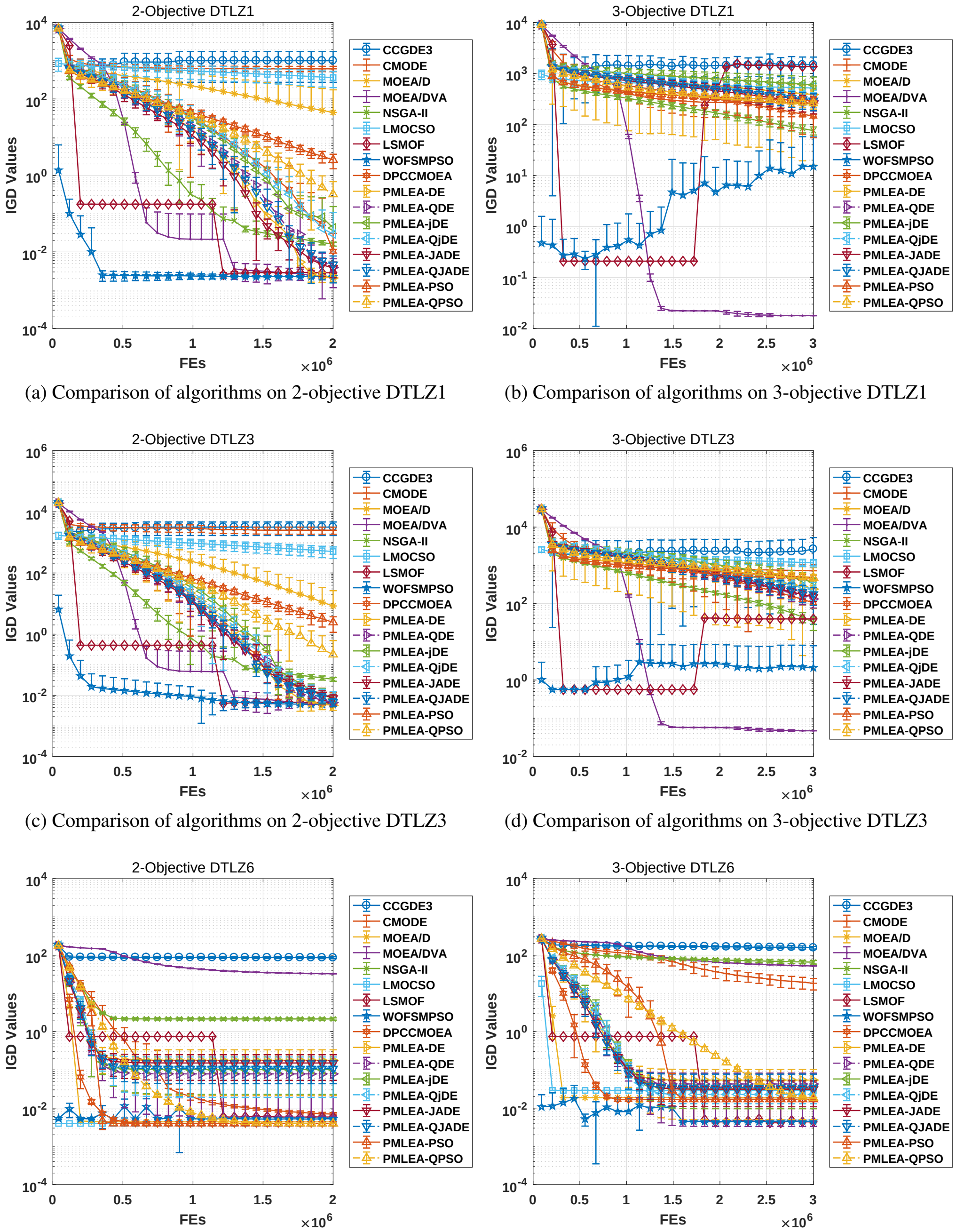
Table 4 lists the rankingvia the nonparametric Friedman tests with respect to the average IGD indicator values on the 2- and 3-objective LSMOP3 and LSMOP6 benchmark functions, while Fig. 4 illustrates the evolution curves of the IGD values of the 17 algorithms on the 2- and 3-objective LSMOP3 and LSMOP6 test functions. Fig. 4 and Table 4 show that PMLEA-QJADE is superior to the other algorithms, WOF-SMPSO ranks the second, and the performance of CCGDE3 is the worst.

4.3.3. *Analysis of the Experimental Results on MaOP*

Table 1 lists the average IGD values of the 17 algorithms, and Fig. 3 illustrates the IGD evolution curves

Table 1: IGD mean values of the algorithms on the DTLZ, WFG, LSMOP and MaOP test suites

	PMLEA-DE	PMLEA-QDE	PMLEA-jDE	PMLEA-QjDE	PMLEA-JADE	PMLEA-QJADE	CCGDE3	CMODE	MOEA/D	MOEA/DVA	NSGA-II	DPCCMOEA	PMLEA-PSO	PMLEA-QPSO	WOF-SMPSO	LMOCSO	LSMOF
DTLZ1_OBJ2_DIM200	**0.002088626**	0.003897537	0.04239312	0.029408533	0.00379159	0.00468736	1003.406583	616.1178447	44.47922058	0.00228169	0.016089766	0.010430225	2.545419854	0.322173084	**0.002243638**	363.9020117	0.002857475
DTLZ2_OBJ2_DIM200	**0.003962248**	0.003962274	0.003962268	0.003962273	**0.003962256**	0.003962279	0.495750902	0.004139474	0.004429535	0.004360842	0.005073273	0.003963892	0.004020412	0.003962295	0.005164346	0.003975439	0.00509713
DTLZ3_OBJ2_DIM200	**0.004701186**	0.005773431	0.007641549	0.009350826	0.008017119	0.006057534	3154.660993	2498.220851	8.340008267	0.005274935	0.033437566	0.008424056	2.42287943	0.215650816	**0.004965715**	502.3598191	0.005840367
DTLZ4_OBJ2_DIM200	0.003962273	**0.003962232**	0.003962242	0.003962237	**0.003962234**	0.003962235	0.855051731	0.521002531	0.410285924	0.00436082	0.078823882	0.003963655	0.004039368	0.003962292	0.005199681	0.040872964	0.005170641
DTLZ5_OBJ2_DIM200	**0.003962252**	0.003962272	0.003962269	0.003962276	**0.003962257**	0.003962278	0.499209087	0.004166437	0.004441294	0.00436085	0.005073272	0.003963749	0.004020412	0.003962294	0.005148535	0.003973987	0.005134977
DTLZ6_OBJ2_DIM200	0.177945919	0.078418803	0.101617367	0.118037797	0.151098733	0.10683858	85.96020306	0.00695701	0.004363507	32.63978502	2.143515854	0.003963344	**0.003962245**	0.00396673	0.005246118	**0.003962262**	0.00582253
DTLZ7_OBJ2_DIM200	0.005490612	0.005488959	0.005489146	0.005488893	0.005489047	0.00548895	0.210586009	**0.004453356**	0.246570448	0.006539698	**0.00542319**	0.005524645	0.00548916	0.005489039	0.049056182	0.02839576	0.443024999
DTLZ1_OBJ3_DIM300	386.4608212	291.2319967	599.2731884	317.8985895	253.6516648	293.3638362	1485.153821	263.6166657	251.5372332	**0.017785163**	**75.81964796**	145.2086576	294.6028285	259.9408876	14.97411308	520.0826899	1368.252975
DTLZ2_OBJ3_DIM300	0.046791473	0.04674174	0.046760854	0.046740389	0.046757795	**0.04673947**	4.948585084	0.048320948	0.049869268	**0.046713656**	0.06527194	0.047048248	0.048281674	0.046761849	0.07215715	0.046807113	0.168333925
DTLZ3_OBJ3_DIM300	238.6522841	131.8279829	375.7900119	239.2788294	149.055259	174.7218579	2673.494982	719.3394218	436.3399128	**0.047037235**	31.96586412	104.8384879	478.9310625	477.9365544	**2.088308844**	1162.527811	39.64065528
DTLZ4_OBJ3_DIM300	0.046788522	**0.046744708**	0.046767202	**0.04674245**	0.046757391	0.046743665	1.612952183	0.307361313	0.502670163	0.049170596	0.065651491	0.047652249	0.047901163	0.046772897	0.066258829	0.295157158	0.154562965
DTLZ5_OBJ3_DIM300	0.01699633	0.016995984	0.016995967	0.016995985	0.016995917	0.01699603	3.643932706	**0.003543877**	0.018727002	0.018670471	**0.004828363**	0.016866995	0.017006291	0.016996074	0.00690476	0.028619585	0.0204066
DTLZ6_OBJ3_DIM300	0.053934206	0.042710966	0.031725574	0.022745961	0.031145931	0.034220235	156.7070152	18.41243853	0.018700599	52.07665714	65.62259904	0.016869305	0.017000589	0.018474823	**0.004541466**	0.028908575	**0.004361799**
DTLZ7_OBJ3_DIM300	0.073879312	0.073911746	0.073951231	0.073945562	0.073898842	0.073923297	1.268760969	**0.058408785**	0.252420613	0.075671907	**0.066951653**	0.073977572	0.073981414	0.074044339	0.081096256	0.200172806	0.799999399
WFG1_OBJ2_DIM200	0.685438703	0.329655028	0.5831579	0.378709538	0.727640801	0.328272622	1.29657094	**0.090290274**	1.21409339	1.024874453	0.324364384	0.480444262	1.09631096	0.945062898	1.173390379	0.96458305	**0.02385789**
WFG2_OBJ2_DIM200	0.034847151	0.035752034	0.030442621	0.035422199	0.022143213	0.024099899	0.25061727	0.10762162	0.033181551	1.145233567	0.184983373	0.029408866	0.054763163	0.052552902	**0.021443471**	0.065156777	**0.013206724**
WFG3_OBJ2_DIM200	0.042353075	0.025945526	**0.02396932**	0.028441104	0.02754705	**0.023308903**	0.293108191	0.085406504	0.032686421	1.128082276	0.092094149	0.02970188	0.043916281	0.05542859	0.029746147	0.061368269	0.0487054
WFG4_OBJ2_DIM200	0.015485831	0.013370275	0.014660398	**0.01311633**	0.01345554	**0.012822253**	0.166289546	0.023344185	0.068299357	1.349776066	0.018651367	0.017713489	0.018184855	0.016303384	0.031051623	0.017736864	0.017317326
WFG5_OBJ2_DIM200	0.065211229	0.063570416	0.064160121	0.064011956	0.064055762	0.063305557	0.08458947	**0.062460661**	0.068414284	0.54804995	0.065628544	0.069392495	0.066653969	0.065108244	0.063967634	0.064932141	**0.025908695**
WFG6_OBJ2_DIM200	0.013000073	0.013157312	**0.012887952**	0.013296985	0.013532142	0.013258261	0.296283474	0.013678107	0.016865719	1.35109091	0.019352078	**0.012856693**	0.015647976	0.012981612	0.017679513	0.018067169	0.021202868
WFG7_OBJ2_DIM200	0.012219716	0.012220063	0.012219686	**0.012219216**	0.012219544	0.012219563	0.241179529	0.012742253	**0.011691608**	1.37170142	0.016449063	0.01222569	0.01279448	0.012233099	0.018831524	0.013709211	0.01661036
WFG8_OBJ2_DIM200	0.046985609	0.043250477	**0.041563646**	0.042631402	0.044277994	0.046054922	0.285366141	0.061719298	0.081955761	1.356007291	0.047368806	0.053388326	0.050943793	0.052864099	0.048766359	0.067963363	**0.038070549**
WFG9_OBJ2_DIM200	0.015711835	0.01526049	0.015659095	**0.014894123**	0.015317312	**0.015134659**	0.172090923	0.02766218	0.023780334	1.391257921	0.026816948	0.018845377	0.018859184	0.015631589	0.024970528	0.032460496	0.016390173
WFG1_OBJ3_DIM300	0.968864981	**0.708713239**	0.868145366	**0.713727208**	1.032071992	0.817551446	1.771142017	0.98819022	1.348973711	2.542213431	1.28792875	1.59941439	1.040826024	1.152600268	1.316221968	1.372466185	1.57855374
WFG2_OBJ3_DIM300	**0.183401842**	0.190007304	0.194264969	0.196624598	**0.176654958**	0.184833762	0.630461486	0.215225245	0.286705167	0.203555608	0.334809044	0.21561709	0.258711995	0.199580577	0.198318149	0.297549732	0.205349615
WFG3_OBJ3_DIM300	0.112510662	0.128235056	0.126191695	0.141378371	0.084302156	0.093445361	0.483324967	0.184853254	0.17912599	**0.078962619**	0.185599933	0.16237527	0.095763563	0.146710293	**0.055761246**	0.222445421	0.093735661
WFG4_OBJ3_DIM300	0.195787268	0.19156549	0.193474785	**0.191134621**	**0.191298915**	0.193480887	0.828577507	0.213191389	0.238860762	0.197158074	0.295702495	0.196304612	0.205527932	0.199302863	0.305018454	0.210849564	0.329953818
WFG5_OBJ3_DIM300	0.211416974	0.208686011	0.210871398	0.208755731	0.210589479	**0.208646458**	0.535813255	0.224679722	0.21811877	**0.203909555**	0.275668836	0.218180785	0.211947549	0.211056236	0.284420476	0.213149772	0.252663574
WFG6_OBJ3_DIM300	0.189329299	0.189292708	0.189358727	**0.189311569**	0.189340533	**0.189295297**	1.395395041	0.193300585	0.191267186	0.189464899	0.254333746	0.189857345	0.203673637	0.189363476	0.275103491	0.205393272	0.242270803
WFG7_OBJ3_DIM300	0.189486485	0.189497439	0.189518307	0.18951689	0.189575508	0.189538308	0.882100089	0.193692221	0.210489459	**0.189012192**	0.264209785	0.192036263	0.20290145	**0.18944378**	0.258665633	0.205993993	0.33858256
WFG8_OBJ3_DIM300	0.236135683	0.235152749	0.235892849	0.233296965	0.236174043	0.236743198	0.895984972	0.238768085	0.273794007	**0.228582964**	0.293222588	0.247743325	0.241567796	0.238946108	0.284494675	**0.216951125**	0.575999256
WFG9_OBJ3_DIM300	0.218592418	0.215573786	0.214410577	0.219371776	0.21773806	0.220247737	0.694204715	0.232529152	**0.208901788**	**0.202111708**	0.291273442	0.218659814	0.213474212	0.21578303	0.246377922	0.208141798	0.284671889
LSMOP3_OBJ2_DIM206	7.005908079	7.361660026	1.258758083	1.414691135	0.486008939	**0.410116107**	114.7531486	7.322371631	1.334863805	0.681258589	0.85875914	**0.419015199**	14.36160461	8.210356189	0.859813176	0.70710807	1.354053085
LSMOP6_OBJ2_DIM206	0.440161754	0.457627519	0.432122155	0.433196895	0.451397961	0.440856872	290.8234493	0.491828277	0.546215024	**0.431916207**	0.578481326	0.573743295	0.474733076	0.43506646	0.035526209	0.459523217	**0.369318611**
LSMOP3_OBJ2_DIM307	1.397418248	1.423409824	0.667814998	0.721408578	0.433195669	0.420998767	19.56388796	3.765370587	0.625228001	0.60831204	0.68678749	1.576548905	1.45409706	1.753105585	0.852571242	0.422210439	0.860717391
LSMOP6_OBJ2_DIM307	1.603398313	1.438718859	1.73095939	1.599899101	1.425281635	1.209051412	728.4863863	3.288035193	2.17752726	7.41286974	**0.92532105**	1.046371769	1.002720343	1.508917057	**0.933697427**	1.126726567	1.256966494
MaOP2_OBJ3_DIM300	0.212360645	0.20289491	**0.090216673**	**0.089920909**	0.092868501	0.091776921	736.9293743	0.117632583	0.102552483	0.09175609	2.501291402	0.434932841	0.201916816	0.138096762	43.59859211	1.068328047	105.8004428



(a) Comparison of algorithms on 2-objective DTLZ1

(b) Comparison of algorithms on 3-objective DTLZ1

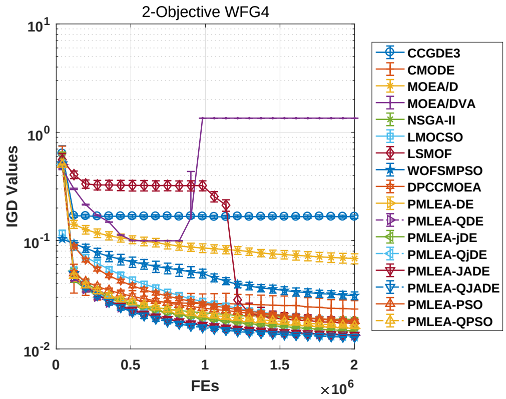
(c) Comparison of algorithms on 2-objective DTLZ3

(d) Comparison of algorithms on 3-objective DTLZ3

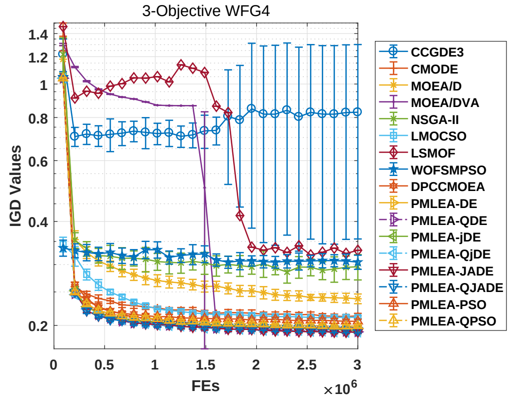
(e) Comparison of algorithms on 2-objective DTLZ6

(f) Comparison of algorithms on 3-objective DTLZ6

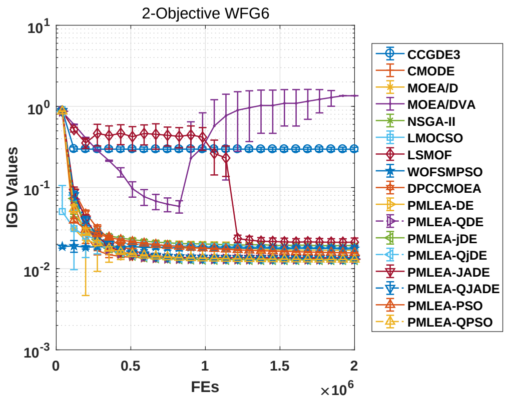
Figure 1: IGD evolution curves for different algorithms on the 2/3-objective DTLZ1, 3 and 6 functions.



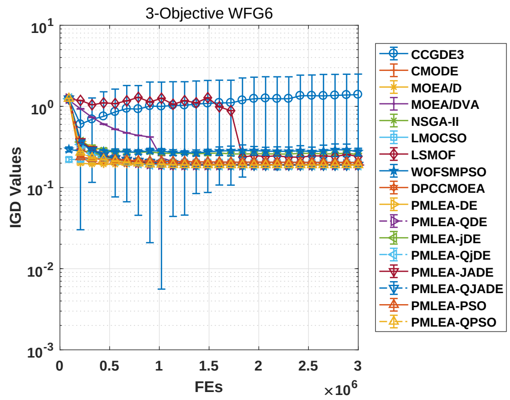
(a) Comparison of algorithms on 2-objective WFG4



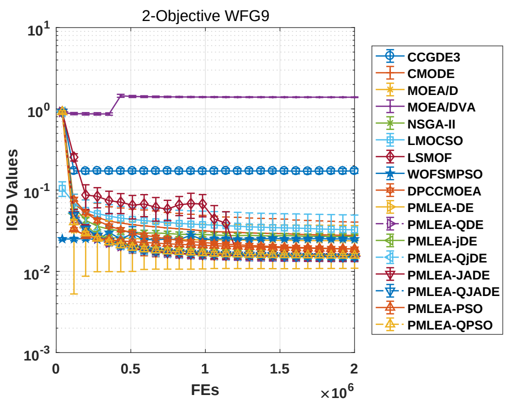
(b) Comparison of algorithms on 3-objective WFG4



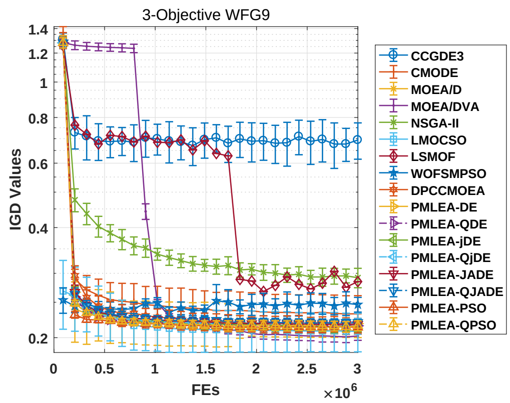
(c) Comparison of algorithms on 2-objective WFG6



(d) Comparison of algorithms on 3-objective WFG6



(e) Comparison of algorithms on 2-objective WFG9



(f) Comparison of algorithms on 3-objective WFG9

Figure 2: IGD evolution curves for different algorithms on the 2/3-objective WFG4, 6, and 9 functions.

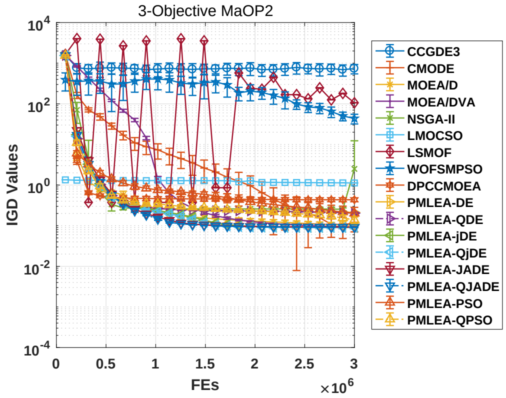
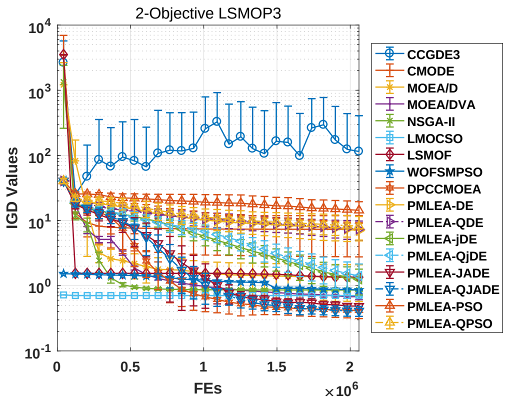
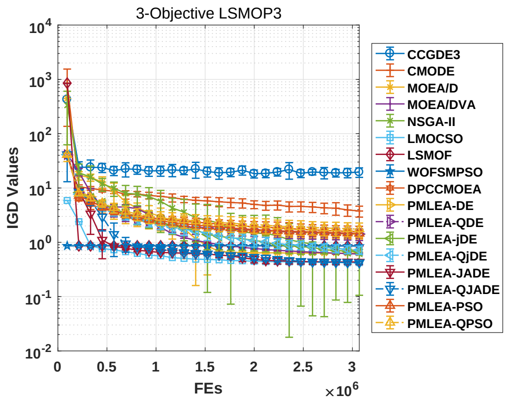


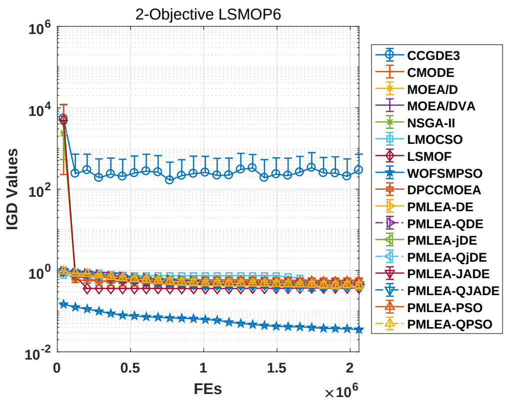
Figure 3: IGD evolution curves for different algorithms on the 3-objective MaOP2 function.



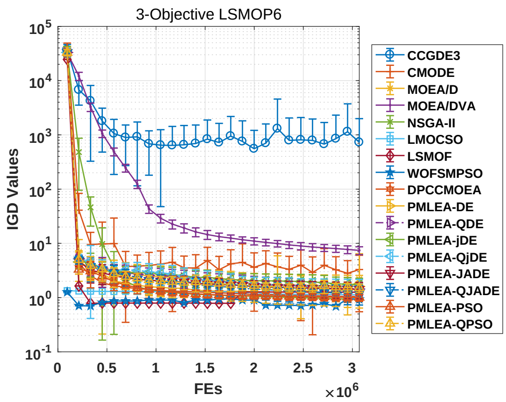
(a) Comparison of algorithms on 2-objective LSMOP3



(b) Comparison of algorithms on 3-objective LSMOP3



(c) Comparison of algorithms on 2-objective LSMOP6



(d) Comparison of algorithms on 3-objective LSMOP6

Figure 4: IGD evolution curves for different algorithms on the 2/3-objective LSMOP3 and LSMOP6 functions.

Table 2: The algorithm ranking via the nonparametric Friedman tests with respect to the average IGD values on the DTLZ, WFG, LSMOP and MaOP test suites

Algorithm	Ranking	Final Ranking
PMLEA-QJADE	5.0811	1
PMLEA-JADE [48]	5.4324	2
PMLEA-QDE	5.7838	3
PMLEA-QjDE	5.8919	4
PMLEA-jDE [51]	6.3243	5
PMLEA-DE [15]	7.2432	6
DPCCMOEA [43]	8.2973	7
PMLEA-QPSO [18]	8.5676	8
MOEA/DVA [53]	9.4324	9
WOF-SMPSO [56]	9.4595	10
PMLEA-PSO [46]	10.1892	11
LSMOF [58]	10.2703	12
CMODE [54]	10.7838	13
LMOCSO [57]	11	14
NSGA-II [55]	11.2162	15
MOEA/D [47]	11.3243	16
CCGDE3 [52]	16.7027	17

Table 3: The algorithm ranking via the nonparametric Friedman tests with respect to the average IGD values on the DTLZ and WFG test suites

Algorithm	Ranking	Final Ranking
PMLEA-QDE	4.9688	1
PMLEA-QJADE	5.25	2
PMLEA-JADE [48]	5.4062	3
PMLEA-QjDE	5.6875	4
PMLEA-jDE [51]	6.2812	5
PMLEA-DE [15]	6.7188	6
DPCCMOEA [43]	8.125	7
PMLEA-QPSO [18]	8.2188	8
WOF-SMPSO [56]	9.875	9
MOEA/DVA [53]	9.9688	10
PMLEA-PSO [46]	10.125	11
LSMOF [58]	10.4688	12
CMODE [54]	10.4688	12
MOEA/D [47]	11.5938	14
LMOCSO [57]	11.5938	14
NSGA-II [55]	11.5938	14
CCGDE3 [52]	16.6562	17

Table 4: The algorithm ranking via the nonparametric Friedman tests with respect to the average IGD values on the LSMOP test suite

Algorithm	Ranking	Final Ranking
PMLEA-QJADE	4	1
WOF-SMPSO [56]	4.75	2
PMLEA-JADE [48]	5.75	3
LMOCSO [57]	5.75	3
MOEA/DVA [53]	6.75	5
LSMOF [58]	7.25	6
NSGA-II [55]	7.5	7
PMLEA-jDE [51]	7.75	8
PMLEA-QjDE	8.75	9
DPCCMOEA [43]	8.75	9
MOEA/D [47]	10.5	11
PMLEA-DE [15]	10.5	11
PMLEA-PSO [46]	11	13
PMLEA-QDE	11.25	14
PMLEA-QPSO [18]	11.5	15
CMODE [54]	14.25	16
CCGDE3 [52]	17	17

on the 3-objective MaOP2 test function. PMLEA-QjDE performs the best, followed by PMLEA-jDE and MOEA/DVA, while CCGDE3 performs the worst.

5. Conclusions

Based on the DPCCMOLSEA framework, we proposed a series of quantum-enhanced algorithms: PMLEA-QDE, PMLEA-QjDE and PMLEA-QJADE. We combined parameter quantization and the DE operator to optimize the population. Moreover, in optimizers of jDE and JADE, the adaptive parameters are enhanced by quantization. We used the multiobjective test suites DTLZ, WFG, LSMOP and MaOP to compare the quantum-enhanced algorithms to other state-of-the-art multiobjective algorithms and ranked the algorithms using nonparametric tests. The results showed that PMLEA-QJADE, PMLEA-QjDE and PMLEA-QDE achieve better optimization results than the other algorithms. The adoption of parallel operation in the MPI environment greatly reduced the time consumption of the algorithms. In future work, we will introduce the theory of quantum mechanics into different stages of multiobjective large-scale EAs and propose new parameter-adaptive methods to improve the optimization efficiency. We will also use the improved multiobjective large-scale EA to solve complex real-world optimization problems.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61976242, in part by the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University under Grant No. 2018002, in part by the Open Fund of Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, under Grant No. IPIU2019003, and in part by the State Key Program of National Natural Science Foundation of China under Grant No. 61836009. This work is also supported in part by the National Supercomputer Center in Guangzhou.

References

[1] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE transactions on evolutionary computation 6 (6) (2002) 580–593.

[2] L. Wang, Advances in quantum-inspired evolutionary algorithms, Control and Decision 23 (12) (2008) 1321–1326.

[3] H. Xiong, Z. Wu, H. Fan, G. Li, G. Jiang, Quantum rotation gate in quantum-inspired evolutionary algorithm: A review, analysis and comparison study, Swarm and Evolutionary Computation 42 (2018) 43–57.

[4] C. Patvardhan, S. Bansal, A. Srivastav, Parallel improved quantum inspired evolutionary algorithm to solve large size quadratic knapsack problems, Swarm & Evolutionary Computation 26 (2016) 175–190.

[5] B. Cao, J. Zhao, Y. Gu, Y. Ling, X. Ma, Applying graph-based differential grouping for multiobjective large-scale optimization, Swarm and Evolutionary Computation 53.

[6] X. Chang, Y.-L. Yu, Y. Yang, E. P. Xing, Semantic pooling for complex event analysis in untrimmed videos, IEEE transactions on pattern analysis and machine intelligence 39 (8) (2016) 1617–1632.

[7] T.-C. Lu, J.-C. Juang, Quantum-inspired space search algorithm (QSSA) for global numerical optimization, Applied Mathematics and Computation 218 (6) (2011) 2516–2532.

[8] B. Ji, X. Yuan, X. Li, Y. Huang, W. Li, Application of quantum-inspired binary gravitational search algorithm for thermal unit commitment with wind power integration, Energy Conversion & Management 87 (87) (2014) 589–598.

[9] M. Fiasché, D. E. Liberati, S. Gualandi, M. Taisch, Quantum-inspired evolutionary multiobjective optimization for a dynamic production scheduling approach, in: Multidisciplinary Approaches to Neural Computing, Springer, 2018, pp. 191–201.

[10] L. Cui, L. Wang, J. Deng, J. Zhang, A new improved quantum evolution algorithm with local search procedure for capacitated vehicle routing problem, Mathematical Problems in Engineering 2013 (159495) (2013) 1–17.

[11] R. S. Pavithr, Gursaran, Quantum inspired social evolution (QSE) algorithm for 0-1 knapsack problem, Swarm & Evolutionary Computation 29 (2016) 33–46.

[12] Z. A. E. M. Dahi, C. Mezioud, A. Draa, A quantum-inspired genetic algorithm for solving the antenna positioning problem, Swarm & Evolutionary Computation 31 (2016) 24–63.

[13] D. Alanis, P. Botsinis, S. X. Ng, L. Hanzo, Quantum-assisted routing optimization for self-organizing networks, IEEE Access 2 (2014) 614–632.

[14] B. Li, Z. Zhou, W. Zou, D. Li, Quantum memetic evolutionary algorithm-based low-complexity signal detection for underwater acoustic sensor networks, IEEE Transactions on Systems & Man Cybernetics Part C 42 (5) (2012) 626–640.

[15] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization 11 (4) (1997) 341–359.

[16] K. Price, R. M. Storn, J. A. Lampinen, Differential evolution: A practical approach to global optimization, Springer Science & Business Media, 2006.

[17] G. Hu, S. Yang, Y. Li, S. U. Khan, A hybridized vector optimal algorithm for multi-objective optimal designs of electromagnetic devices, IEEE Transactions on Magnetics 52 (3) (2015) 1–4.

[18] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection, Evolutionary computation 20 (3) (2012) 349–393.

[19] S. Yang, G. Ni, Y. Li, B. Tian, R. Li, A universal tabu search algorithm for global optimization of multimodal functions with continuous variables in electromagnetics, IEEE transactions on magnetics 34 (5) (1998) 2901–2904.

[20] A. SaiToh, R. Rahimi, M. Nakahara, A quantum genetic algorithm with quantum crossover and mutation operations, Quantum information processing 13 (3) (2014) 737–755.

[21] Z. Ren, R. Xiong, J. Chu, Hybrid quantum differential evolutionary algorithm and its applications, Control Theory & Applications 28 (10) (2011) 1349–1355.

[22] W. Ding, Q. Shi, S. Chen, Z. Guan, J. Wang, A novel quantum cooperative co-evolutionary algorithm for large-scale minimum attribute reduction optimization, in: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2013, pp. 280–286.

[23] M. A. Potter, K. A. D. Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, Evolutionary computation 8 (1) (2000) 1–29.

[24] N. Tian, Quantum-behaved particle swarm optimization with cooperative coevolution for large scale optimization, in: 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), IEEE, 2015, pp. 82–85.

[25] W. Fang, L. Zhang, J. Zhou, X. Wu, J. Sun, A novel quantum-behaved particle swarm optimization with random selection for large scale optimization, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 2746–2751.

[26] K. Praditwong, M. Harman, X. Yao, Software module clustering as a multi-objective search problem, IEEE Transactions on Software Engineering 37 (2) (2010) 264–282.

[27] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, Y. Yang, Rank-constrained spectral clustering with flexible embedding, IEEE transactions on neural networks and learning systems 29 (12) (2018) 6073–6082.

[28] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C. Coello, A survey of multiobjective evolutionary algorithms for data mining: Part I, IEEE Transactions on Evolutionary Computation 18 (1) (2013) 4–19.

[29] L. M. Antonio, C. A. C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 2758–2765.

[30] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with

large-scale variables, IEEE Transactions on Evolutionary Computation 20 (2) (2015) 275–298.

[31] A. Song, Q. Yang, W.-N. Chen, J. Zhang, A random-based dynamic grouping strategy for large scale multi-objective optimization, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 468–475.

[32] X. Zhang, Y. Tian, R. Cheng, Y. Jin, A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization, IEEE Transactions on Evolutionary Computation 22 (1) (2016) 97–112.

[33] A. C. Kumari, K. Srinivas, M. Gupta, Multi-objective test suite minimisation using quantum-inspired multi-objective differential evolution algorithm, in: 2012 IEEE International Conference on Computational Intelligence and Computing Research, IEEE, 2012, pp. 1–7.

[34] A. C. Kumari, K. Srinivas, M. Gupta, Software requirements optimization using multi-objective quantum-inspired hybrid differential evolution, in: EVOLVE-a Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II, Springer, 2013, pp. 107–120.

[35] A. C. Kumari, K. Srinivas, Comparing the performance of quantum-inspired evolutionary algorithms for the solution of software requirements selection problem, Information and Software Technology 76 (2016) 31–64.

[36] L. Li, L. Jiao, J. Zhao, R. Shang, M. Gong, Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering, Pattern Recognition 63 (2017) 1–14.

[37] C. Mouradian, J. Sahoo, R. H. Glitho, M. J. Morrow, P. A. Polakos, A coalition formation algorithm for multi-robot task allocation in large-scale natural disasters, in: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2017, pp. 1909–1914.

[38] S. Mousavi, F. Afghah, J. D. Ashdown, K. Turck, Leader-follower based coalition formation in large-scale UAV networks, a quantum evolutionary approach, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2018, pp. 882–887.

[39] D. Tang, Y. Cai, J. Zhao, Y. Xue, A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems, Information Sciences 289 (2014) 162–189.

[40] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, J.-J. Li, Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, Applied Soft Computing 34 (2015) 286–300.

[41] B. Cao, J. Zhao, P. Yang, Y. Gu, K. Muhammad, J. J. P. C. Rodrigues, V. H. C. de Albuquerque, Multiobjective 3-D topology optimization of next-generation wireless data center network, IEEE Transactions on Industrial Informatics 16 (5) (2020) 3597–3605.

[42] K. C. Tan, Y. Yang, C. K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 527–549.

[43] B. Cao, J. Zhao, Z. Lv, X. Liu, A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization, IEEE Transactions on Industrial Informatics 13 (4) (2017) 2030–2038.

[44] B. Cao, J. Zhao, P. Yang, Z. Lv, X. Liu, G. Min, 3-D multiobjective deployment of an industrial wireless sensor network for maritime applications utilizing a distributed parallel algorithm, IEEE Transactions on Industrial Informatics 14 (12) (2018) 5487–5495.

[45] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art., Swarm & Evolutionary Computation 1 (1) (2011) 32–49.

[46] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 1998, pp. 69–73.

[47] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on evolutionary computation 11 (6) (2007) 712–731.

[48] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 945–958.

[49] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), Vol. 1, IEEE, 2004, pp. 325–331.

[50] J. Sun, W. Xu, B. Feng, A global search strategy of quantum-behaved particle swarm optimization, in: IEEE Conference on Cybernetics and Intelligent Systems, 2004., Vol. 1, IEEE, 2004, pp. 111–116.

[51] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657.

[52] L. M. Antonio, C. A. C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 2758–2765.

[53] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong, A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables, IEEE Transactions on Evolutionary Computation 20 (2) (2015) 275–298.

[54] J. Wang, W. Zhang, J. Zhang, Cooperative differential evolution with multiple populations for multiobjective optimization, IEEE transactions on cybernetics 46 (12) (2015) 2848–2861.

[55] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on evolutionary computation 6 (2) (2002) 182–197.

[56] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, A framework for large-scale multiobjective optimization based on problem transformation, IEEE Transactions on Evolutionary Computation 22 (2) (2018) 260–275.

[57] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Transactions on Cybernetics (2019) 1–13.

[58] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, IEEE Transactions on Evolutionary Computation 23 (6) (2019) 949–961.

[59] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: Evolutionary multiobjective optimization, Springer, 2005, pp. 105–145.

[60] S. Huband, P. Hingston, L. Barone, R. L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.

[61] R. Cheng, Y. Jin, M. Olhofer, B. sendhoff, Test problems for large-scale multiobjective and many-objective optimization, IEEE Transactions on Cybernetics 47 (12) (2017) 4108–4121.

[62] H. Li, K. Deb, Q. Zhang, P. Suganthan, L. Chen, Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, Swarm and Evolutionary Computation 46 (2019) 104–117.

[63] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. D. Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.
[64] S. Cheng, Y. Shi, Q. Qin, On the performance metrics of multiobjective optimization, in: International Conference in Swarm Intelligence, Springer, 2012, pp. 504–512.
[65] J. Derrac, S. Garca, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm & Evolutionary Computation 1 (1) (2011) 3–18.
[66] J. Alcal-Fdez, L. Snchez, S. Garca, M. J. D. Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2009) 307–318.