



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/160036/>

Version: Accepted Version

---

**Article:**

Cao, B., Fan, S., Zhao, J. et al. (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>

---

Article available under the terms of the CC-BY-NC-ND licence  
(<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

**Reuse**

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# Quantum-Enhanced Multiobjective Large-scale Optimization via Parallelism

Bin Cao<sup>a,b,c</sup>, Shanshan Fan<sup>a,b,c</sup>, Jianwei Zhao<sup>a,b,c,\*</sup>, Po Yang<sup>d,\*</sup>, Khan Muhammad<sup>e,\*</sup>, Mohammed Tanveer<sup>f</sup>

<sup>a</sup>State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300401, China.

<sup>b</sup>School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China.

<sup>c</sup>Hebei Provincial Key Laboratory of Big Data Calculation, 300401 Tianjin, China

<sup>d</sup>Department of Computer Science, Sheffield University, UK

<sup>e</sup>Department of Software, Sejong University, Seoul, Republic of Korea

<sup>f</sup>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:

- 139 1. We integrate the position update strategy based on 177  
140 the theory of quantum mechanics in QPSO into the 178  
141 DE operator of the DPCCMOLSEA framework to 179  
142 optimize the population. 180
- 143 2. Based on jDE and JADE, we propose the variants 181  
144 PMLEA-QjDE and PMLEA-QJADE, in which the 182  
145 adaptive parameters are quantized. 183
- 146 3. The integration of parallel operation based on the 184  
147 message passing interface (MPI) substantially re- 185  
148 duces the runtime of the quantum-enhanced algo- 186  
149 rithm. 187

150 The organization of this paper is as follows. Sec- 188  
151 tion 2 introduces the large-scale MOPs and the QPSO 189  
152 algorithm. The proposed methodology is described in 190  
153 Section 3. Section 4 reports the experimental compari- 191  
154 son results and provides an analysis. Finally, Section 5 192  
155 summarizes this paper.

## 156 2. Related Work

### 157 2.1. MOLSOPs

158 MOPs in which the decision variable number is 194  
159 greater than or equal to 100 are called MOLSOPs. In 195  
160 general, an MOP with  $N$  decision variables and  $M$  ob- 196  
161 jective variables can be described as follows [34, 45]:

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

162 where  $x$  is a decision vector in decision space  $\Omega$ ,  $N \geq$   
163 100, and  $F(x)$  is an objective vector located in the ob- 197  
164 jective space,  $M \leq 3$ .

### 165 2.2. Quantum-behaved Particle Swarm Optimization

166 The QPSO algorithm is based on the quantum poten- 201  
167 tial well model inspired by the principles of quantum 202  
168 mechanics. It establishes an attractive potential that af-  
169 fects the individuals in a population, in which each par-  
170 ticle is attracted by a quantum potential well whose cen-  
171 ter is located at its local attractor. The randomness of the  
172 particle position in QPSO improves its global search ca-  
173 pability.

174 In standard particle swarm optimization (PSO) [46], 205  
175 each particle moves in an  $N$ -dimensional space accord- 206  
176 ing to the following equations: 207

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

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

177 where  $V_i^g$  denotes the velocity vector,  $X_i^g$  denotes the po-  
178 sition vector,  $i \in \{1, 2, \dots, NP\}$  denotes the individual  
179 index,  $NP$  denotes the population size,  $j \in \{1, 2, \dots, N\}$   
180 denotes the variable index,  $g$  denotes the current gen-  
181 eration number,  $\omega$  denotes the inertia weight,  $c_1$  and  $c_2$   
182 are acceleration coefficients,  $P_i^g = (P_{i,1}^g, P_{i,2}^g, \dots, P_{i,N}^g)$   
183 is the best previous position of particle  $i$  and is re-  
184 ferred to as the personal best position (pbest), and  $G^g =$   
185  $(G_1^g, G_2^g, \dots, G_N^g)$  is the best particle position in the pop-  
186 ulation and is called the global best location (gbest).

187 Different from the particles in PSO, which are rep-  
188 resented by both position and velocity, only positional  
189 information is used to describe the particles in QPSO,  
190 and the local attractor of particle  $i$  is a random position.  
191 Specifically, for each dimension of particle  $i$ , the posi-  
192 tion of a random point is calculated first as follows:

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

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

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

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

197 where  $C_j^g$  is the average of the pbest positions of all par-  
198 ticles in the  $j$ -th dimension,  $\alpha$  denotes the contraction  
199 expansion (CE) coefficient controlling the convergence  
200 speed, and  $u_{i,j}^g = U(0, 1)$  and  $u_{i,j}^g > 0$  is a random num-  
201 ber.

## 202 3. The Proposed Quantum-enhanced Algorithm

203 In QPSO, the randomness of the particle position 204  
causes it to have better global search capability. There-  
205 fore, inspired by the theory of position update in QP-  
206 SO and based on the DPCCMOLSEA framework, we  
207 propose PMLEA-QDE, PMLEA-QjDE and PMLEA-  
208 QJADE.

209 DPCCMOEA [43] and DPCCMOLSEA [44] both re-  
210 ly on decomposition to solve MOLSOPs. In this sec-  
211 tion, we describe DPCCMOEA, DPCCMOLSEA, and  
212 the proposed quantum-enhanced algorithms.

### 213 3.1. DPCCMOEA

#### 214 3.1.1. Overall architecture

215 In the first layer, the variables are decomposed into  
216 several groups, each of which is optimized by a subpop-  
217 ulation. In the second layer, each CPU core is responsi-  
218 ble for the evolution and evaluation of the individuals.

#### 219 3.1.2. Optimization

220 Each individual relies on neighboring individuals or  
221 the whole subpopulation to share information. Howev-  
222 er, the individuals in each subpopulation are divided into  
223 multiple sets. To reduce the amount of communication,  
224 the set of individuals in each CPU core can obtain on-  
225 ly the information of the individual sets in the adjacent  
226 CPU cores. Each variable  $j$  of the partial trail vector  
227  $trail_{i,j}$  is as follows:

$$228 \quad \begin{aligned} trail_{i,j} &= p_{i,j} + F \times (p_{a_1,j} - p_{a_2,j}) & (7) \\ \text{s.t. } i &\in \{1, 2, \dots, NP\}, j \in S_{opt} \end{aligned}$$

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

#### 233 3.1.3. Crossover

234 To evaluate the fitness, the remaining variables of the  
235 trail vector should be generated to form a complete solu-  
236 tion. For which, the crossover strategy is employed as  
237 follows:

$$238 \quad \begin{aligned} trail_{i,j} &= \begin{cases} p_{i,j} & \text{if } j \notin S_{opt} \wedge r_1 < 0.5 \\ p_{b_1,j} & \text{if } j \notin S_{opt} \wedge r_1 > 0.5 \wedge r_2 \leq 0.5 \\ p_{b_2,j} & \text{otherwise} \end{cases} & (8) \end{aligned}$$

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

#### 241 3.1.4. Mutation

242 The generated  $trail_i$  vector is mutated with the proba-  
243 bility of  $1/N$  via polynomial mutation. Finally, the pop-  
244 ulation update refers to MOEA/D [47].

### 245 3.2. DPCCMOLSEA

#### 246 3.2.1. Overall architecture

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

#### 252 3.2.2. Optimization

253 In DPCCMOEA, each subpopulation is separated to  
254 several sets, each of which is in the charge of one CPU  
255 core. Therefore, all individual sets are evolved in paral-  
256 lel. Different from DPCCMOEA, in DPCCMOLSEA,  
257 all individuals in each subpopulation are evolved in one  
258 corresponding master CPU core in serial, resulting in  
259 better utilization of the information between individuals  
260 in each subpopulation.

#### 261 3.2.3. Crossover

262 Different from DPCCMOEA, which uses a fixed  
263 crossover rate, DPCCMOLSEA adopts an adaptive s-  
264 trategy [48]:

$$265 \quad CR_i = \text{GaussRand}(\mu_1, 0.1) \quad (9)$$

266 where  $CR_i$  represents the crossover probability of the  $i$ -  
267 th individual and satisfies the Gaussian distribution with  
268 mean value of  $\mu_1$  and a deviation factor of 0.1. The  
269 update of  $\mu_1$  satisfies the following equation:

$$270 \quad \mu_1 = (1 - c) \times \mu_1 + c \times \text{mean}_A(S_{CR}) \quad (10)$$

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

### 272 3.3. The Proposed Algorithm

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

#### 282 3.3.1. Parameter quantization: PMLEA-QDE

283 Considering the establishment of an attractive poten-  
284 tial that affects individuals in the population, the  $\delta$  po-  
285 tential well field produces a better effect [18]. To deter-  
286 mine the exact position of the individual, the quantum  
287 state must be collapsed to the classical state; then, the  
288 particle position is measured by a Monte Carlo stochas-  
289 tic simulation. Each variable of an individual moves in  
290 an one-dimensional  $\delta$  potential well centered at point  $p$   
291 [18], and its position can be calculated via the following  
292 stochastic equation:

$$293 \quad X = p \pm \frac{L}{2} \ln(1/u) \quad (11)$$

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

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

297 It is proven that in an  $N$ -dimensional space, the nec-  
 298 essary and sufficient condition for the position of indi-  
 299 vidual  $i$  evolved through the above process to converge  
 300 in probability to its attractor  $p_i^g = (p_{i,1}^g, p_{i,2}^g, \dots, p_{i,N}^g)$   
 301 is that each dimensional coordinate  $X_{i,j}^{g+1}$  converges in  
 302 probability to  $p_{i,j}^g$  [18].

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

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

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

$$L_{i,j}^g = 2\alpha \times |P_{ave,j}^g - X_{i,j}^g| \quad (14)$$

311 Finally, the evolutionary formula for an individual be-  
 312 comes:

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

313 where  $\alpha$  is the CE coefficient, and  $u_{i,j}^g = U(0, 1) \wedge u_{i,j}^g \neq$   
 314  $0$  is a random number.

315 When optimizing the population by DE, the scaling  
 316 factor  $F$  is a key coefficient in the optimization process.  
 317 If  $F$  is too large, then it contributes to population di-  
 318 versity but the convergence is slow, reducing the search  
 319 efficiency. In contrast,  $F$  that is too small causes prema-  
 320 ture convergence. In standard DE, each decision vector  
 321  $X_i^g$  ( $i = 1, 2, \dots, NP$ ) produces a mutation vector. The  
 322 mutation strategies include DE/rand/1, DE/current-to-  
 323 best/1, DE/best/1, etc., and the most commonly utilized  
 324 strategy is DE/rand/1:

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

325 where  $i$  is the index of the current individual,  $V_i^g$  is the  $i$ -  
 326 th variation vector generated in the  $g$ -th generation, and  
 327  $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$  are random numbers with  $r_1 \neq$

328  $r_2 \neq r_3 \neq i$ , and  $F_i$  is the mutation scale factor in  $(0, 1]$ ,  
 329 being fixed or varied with evolution.

330 The principle of individual evolution in QPSO is integ-  
 331 rated to the DE optimization process. In the quantum-  
 332 enhanced algorithm, mutated individuals are generated  
 333 as follows:

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

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

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

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

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

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

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

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

342 where  $r_j = U(0, 1)$  ( $j \in \{1, 2, 3, 4\}$ ) are random num-  
 343 bers, and  $\tau_1$  and  $\tau_2$  are parameters.

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

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

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

347 JADE uses a parameter strategy based on statistical  
 348 learning in which  $F$  and  $CR$  are dynamically adjust-  
 349 ed according to previous successful experiences [48].  
 350 Specifically,  $CR$  is updated as follows:

$$CR_i = \text{GaussRand}(\mu_1, 0.1) \quad (21)$$

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

351 where  $CR_i$  obeys the Gaussian distribution with mean  
 352 of  $\mu_1$  and a standard deviation of 0.1,  $c$  is a constant in  
 353  $(0, 1)$ ,  $\text{mean}_A(\cdot)$  denotes the usual arithmetic mean, and  
 354  $S_{CR}$  records the crossover probabilities  $CR_i$  that enable

the corresponding offsprings successfully entering the next generation.

The scaling factor  $F$  is updated as follows:

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

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

where  $F_i$  obeys the Cauchy distribution with location parameter of  $\mu_2$  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:

$$F'_i = \begin{cases} \ln[1/u_i] \times F_i & \text{if } r_6 \leq 0.5 \\ -\ln[1/u_i] \times F_i & \text{otherwise} \end{cases} \quad (25)$$

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 (LMOCSSO) [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:

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

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  $x$

446 and individuals in  $P^*$ . Therefore, a smaller IGD value  
 447 represents an approximated PF closer to the real PF, in-  
 448 dicating better performance.  
 449 In addition, we used a nonparametric test to evaluate  
 450 the algorithm more accurately [65, 66].

### 4.3. Algorithm Comparison

451 Table 1 lists the mean values of the IGD index val-  
 452 ues obtained by the algorithms on the 2- and 3-objective  
 453 DTLZ, WFG, LSMOP and MaOP test suites; the bold  
 454 numbers indicate two best values. The ranking re-  
 455 sults of all algorithms via the nonparametric Friedman  
 456 tests with respect to the average IGD indicator val-  
 457 ues are listed in Table 2. Overall, PMLEA-QJADE  
 458 achieves the best performance on the four test suites,  
 459 PMLEA-JADE ranks the second, and PMLEA-QDE  
 460 and PMLEA-QjDE perform well, while CCGDE3 per-  
 461 forms the worst. Therefore, the proposed quantum-  
 462 enhanced parallel multiobjective large-scale algorithms  
 463 are superior to other state-of-the-art multiobjective EAs.

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

464 Table 3 lists the ranking via the nonparametric Fried-  
 465 man tests with respect to the average IGD indicator  
 466 values on the 2- and 3-objective DTLZ and WFG test  
 467 suites. Furthermore, Figs. 1 and 2 illustrate the evolu-  
 468 tion curves of the IGD values of the 17 algorithms on the  
 469 2- and 3-objective DTLZ1, DTLZ3 and DTLZ6 func-  
 470 tions as well as the 2- and 3-objective WFG4, WFG6  
 471 and WFG9 test functions. PMLEA-QjDE achieves the  
 472 best performance on the DTLZ1-7 and WFG1-9 test  
 473 functions, PMLEA-QJADE performs well, and the per-  
 474 formance of CCGDE3 is the worst.

#### 4.3.2. Analysis of the Experimental Results on LSMOP

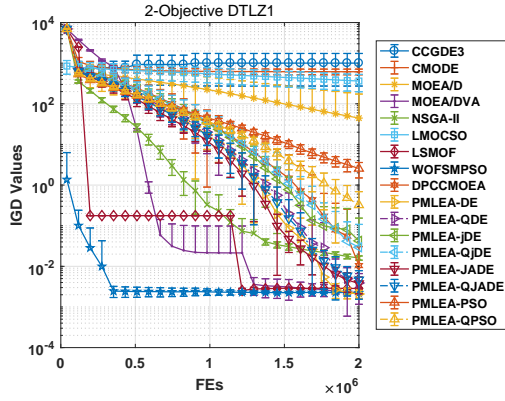
475 Table 4 lists the ranking via the nonparametric Fried-  
 476 man tests with respect to the average IGD indicator val-  
 477 ues on the 2- and 3-objective LSMOP3 and LSMOP6  
 478 benchmark functions, while Fig. 4 illustrates the evo-  
 479 lution curves of the IGD values of the 17 algorithms on  
 480 the 2- and 3-objective LSMOP3 and LSMOP6 test func-  
 481 tions. Fig. 4 and Table 4 show that PMLEA-QJADE is  
 482 superior to the other algorithms, WOF-SMPSO ranks  
 483 the second, and the performance of CCGDE3 is the  
 484 worst.

#### 4.3.3. Analysis of the Experimental Results on MaOP

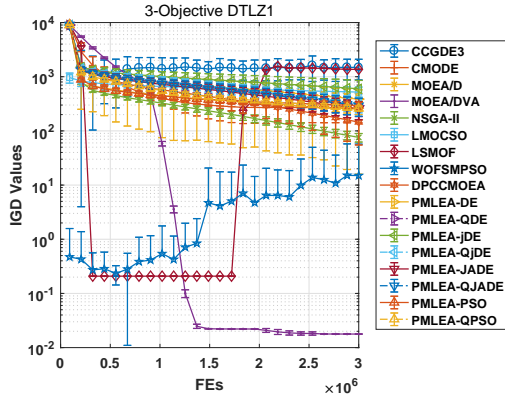
485 Table 1 lists the average IGD values of the 17 algo-  
 486 rithms, and Fig. 3 illustrates the IGD evolution curves

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

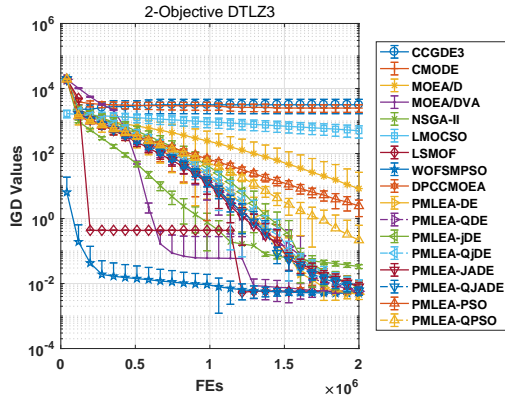
	PMLEA-DE	PMLEA-QjDE	PMLEA-JDE	PMLEA-QjDE	PMLEA-JADE	PMLEA-QJADE	CCGDE3	CMODE	MOEA/D	MOEA/DVA	NSGA-II	DPCCMOEA	PMLEA-PSO	PMLEA-QPSO	WOF-SMPSO	LMOCSO	LSMOP
DTLZ1_ORJ2_DIM200	<b>0.002088626</b>	0.003897537	0.04239312	0.029408533	0.00379159	0.00468736	1003.406583	616.1178447	44.47922058	0.00228169	0.016089766	0.010430225	2.545419854	0.322173084	<b>0.002243638</b>	363.9020117	0.002857475
DTLZ2_ORJ2_DIM200	<b>0.003962248</b>	0.003962274	0.003962268	0.003962273	<b>0.003962279</b>	0.004362279	0.495750902	0.004139474	0.004429535	0.004368042	0.005073273	0.003963892	0.004020412	0.003962295	0.005164346	0.003975439	0.005097113
DTLZ3_ORJ2_DIM200	<b>0.004701186</b>	0.005773431	0.007641549	0.009350826	0.008017119	0.006057534	3154.660993	2498.220851	0.033437566	0.005274935	0.033437566	0.008424056	2.42287943	0.216560816	<b>0.004965715</b>	502.3398919	0.005840367
DTLZ4_ORJ2_DIM200	0.003962273	<b>0.003962232</b>	0.003962242	0.003962237	<b>0.003962234</b>	0.003962235	8.85501731	0.521002531	0.410285924	0.004368082	0.078823882	0.003963655	0.004039368	0.003962292	0.005199681	0.040872964	0.005170641
DTLZ5_ORJ2_DIM200	<b>0.003962252</b>	0.003962272	0.003962269	0.003962276	<b>0.003962257</b>	0.003962278	0.499209087	0.004166437	0.004441294	0.004368085	0.005073272	0.003963749	0.004020412	0.003962294	0.005148535	0.003973627	0.005134977
DTLZ6_ORJ2_DIM200	0.177945919	0.078418803	0.101617367	0.118037797	0.151098733	0.10683858	9.96020306	0.00695701	0.004363507	32.63978502	2.143515854	0.003963344	<b>0.003962245</b>	0.00396673	0.005246118	<b>0.003962262</b>	0.00582253
DTLZ7_ORJ2_DIM200	0.005490612	0.005488959	0.005489146	0.005488909	0.005488947	0.005488909	0.021058609	<b>0.004453536</b>	0.246570448	0.006539698	<b>0.005282519</b>	0.005524644	0.005524644	0.00548916	0.005489039	0.049056182	0.443024999
DTLZ1_ORJ3_DIM300	386.4608212	291.2319667	599.2731884	317.8985895	253.6516648	<b>293.3638362</b>	1485.153821	263.6166657	251.5372332	<b>0.017785163</b>	<b>75.81964796</b>	145.2086576	294.6028285	259.9408876	14.97411308	520.0826899	1368.252975
DTLZ2_ORJ3_DIM300	0.046791473	0.04674174	0.046760854	0.046740389	0.046757795	<b>0.04673947</b>	4.948585084	0.048320948	0.049869268	<b>0.046712625</b>	0.065271194	0.047048248	0.048281674	0.046761849	0.07215715	0.046807113	0.168333925
DTLZ3_ORJ3_DIM300	238.65322841	131.8279829	375.7900119	239.2788294	149.0552529	<b>174.7218579</b>	2673.494982	719.3394218	436.33991128	<b>0.047037235</b>	31.96586412	104.8384879	478.9310625	477.9365544	<b>2.088308844</b>	1162.527811	39.64065528
DTLZ4_ORJ3_DIM300	0.046788522	<b>0.046744708</b>	0.046767202	<b>0.04674245</b>	0.046757391	0.046743665	1.612952183	0.307361313	0.307361313	0.049170596	0.065651491	0.047652249	0.047901163	0.046772897	0.066258829	0.295157158	0.154562965
DTLZ5_ORJ3_DIM300	0.01699633	0.016995984	0.016995967	0.016995985	0.016995917	0.01699603	3.643932706	<b>0.003543877</b>	0.018727002	0.018670471	<b>0.004823633</b>	0.016866995	0.017006291	0.016996074	0.00690476	0.028619585	0.02304066
DTLZ6_ORJ3_DIM300	0.053934206	0.042710966	0.031725574	0.022745961	0.031145931	0.034220235	156.7070152	18.41243835	0.017805599	52.07665714	65.62359904	0.016869305	0.017000589	0.018474823	<b>0.004541466</b>	0.028908375	<b>0.004361799</b>
DTLZ7_ORJ3_DIM300	0.073879312	0.073911746	0.073911746	0.073915231	0.073945562	0.073923297	1.268760969	<b>0.058408785</b>	0.252420613	<b>0.066951653</b>	0.073977572	0.073981414	0.074044339	0.081096256	0.2001172806	0.799993999	
WFG1_ORJ2_DIM200	0.685438703	0.329655028	0.5831579	0.378709538	0.727640801	0.328272622	1.29657094	<b>0.090290274</b>	1.21409339	1.024874453	0.323434384	0.480444262	1.09631096	0.945062898	1.173390379	0.96458305	<b>0.02385789</b>
WFG2_ORJ2_DIM200	0.034847151	0.035752034	0.030442621	0.030498099	0.025061727	0.10762162	0.25061727	0.10762162	0.033181551	1.145233567	0.184983373	0.029408866	0.054763163	0.052552902	<b>0.021443471</b>	0.065156777	<b>0.013206724</b>
WFG3_ORJ2_DIM200	0.042353075	0.025945526	<b>0.02396932</b>	0.028441103	0.027545705	<b>0.023308903</b>	0.293108191	0.08554004	0.032688642	1.128082276	0.092094149	0.02970188	0.043916281	0.05542859	0.02974612	0.061368269	0.0487054
WFG4_ORJ2_DIM200	0.015485831	0.013370275	0.014660398	<b>0.01311633</b>	0.01345554	<b>0.012822523</b>	0.166289546	0.023344185	0.068299357	1.349776066	0.018651367	0.017713489	0.018184855	0.017363884	0.01031623	0.017376864	0.0171317326
WFG5_ORJ2_DIM200	0.065211229	0.063570416	0.064160121	0.064011956	0.064055762	0.063305557	0.08458947	<b>0.062460661</b>	0.068414284	0.54804995	0.065628544	0.069392495	0.066653969	0.065108244	0.063967634	0.064933214	<b>0.025908695</b>
WFG6_ORJ2_DIM200	0.013000073	0.013157312	<b>0.012887952</b>	0.013296985	0.013532162	0.013258261	0.296283474	0.013678107	1.35109091	0.016865710	<b>0.012856693</b>	0.015647976	0.012981612	0.012981612	0.017679513	0.018076269	0.02102868
WFG7_ORJ2_DIM200	0.012219714	0.012220063	0.012219686	<b>0.012219216</b>	0.012219544	0.012219563	0.241179529	0.012742253	<b>0.011691608</b>	1.37170142	0.016449063	0.012226269	0.01279448	0.012233099	0.018831524	0.013709211	0.01616036
WFG8_ORJ2_DIM200	0.046985969	0.043250477	<b>0.041563646</b>	0.042631402	0.044277994	0.046054922	0.285366141	0.061719298	0.081955761	1.356007291	0.037368806	0.03388326	0.050943793	0.052864099	0.048766359	0.038075049	<b>0.038075049</b>
WFG9_ORJ2_DIM200	0.015711835	0.015260449	0.015659095	<b>0.014894123</b>	0.015317312	<b>0.015134659</b>	0.172090923	0.02766218	0.023780334	1.391257921	0.026816948	0.018854377	0.018859184	0.015631589	0.024970528	0.032046049	0.016390173
WFG1_ORJ3_DIM300	0.968864981	<b>0.713727208</b>	0.868145366	0.868145366	<b>0.713727208</b>	1.032071992	1.771142017	0.98819022	1.28792875	1.28792875	1.59941377	1.040826024	1.152600268	1.316221968	1.372466185	1.372466185	1.57855374
WFG2_ORJ3_DIM300	<b>0.183401842</b>	0.190007304	0.194264969	0.196624598	<b>0.176654958</b>	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_ORJ3_DIM300	0.112510662	0.128235056	0.126191695	0.141378371	0.084320156	0.483324967	0.184853254	0.184853254	0.179125899	<b>0.078962619</b>	0.185599933	0.16237527	0.095765363	0.146710293	<b>0.055761246</b>	0.222445421	0.093735661
WFG4_ORJ3_DIM300	0.195787268	0.191565499	0.193474785	<b>0.191134621</b>	<b>0.191298915</b>	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_ORJ3_DIM300	0.211416974	0.208686011	0.210871398	0.208755731	0.210589479	<b>0.208646458</b>	0.535813255	0.224679722	0.218118377	<b>0.203090555</b>	0.275668836	0.218180785	0.211944749	0.211056236	0.284420476	0.252663574	
WFG6_ORJ3_DIM300	0.189232999	0.189292708	0.189258727	<b>0.189315462</b>	<b>0.189340533</b>	<b>0.189305585</b>	1.395395041	0.193305885	0.191267186	0.189464899	0.254333746	0.189857445	0.205373637	0.189363476	0.275103491	0.205393272	0.242270803
WFG7_ORJ3_DIM300	0.189464885	0.189497439	0.189518307	0.189516689	0.189575508	0.189583308	0.882100089	0.193692221	0.210489459	<b>0.189012192</b>	0.242409785	0.192036325	0.20290145	<b>0.18944378</b>	0.258665633	0.205993993	0.33882526
WFG8_ORJ3_DIM300	0.236135683	0.235152749	0.235892849	0.233298249	0.236174043	0.236743198	0.895984972	0.238768085	0.273794007	<b>0.228582964</b>	0.293225288	0.247743263	0.241567796	0.238946108	0.28449675	<b>0.216951125</b>	0.575992556
WFG9_ORJ3_DIM300	0.218592418	0.215573786	0.214410577	0.219371776	0.21273806	<b>0.22047737</b>	0.694204715	0.232529152	<b>0.208901788</b>	<b>0.202111708</b>	0.291723442	0.218659914	0.213474212	0.21578303	0.246377922	0.208141798	0.284671889
LSMOP3_ORJ2_DIM206	7.005908079	7.361660026	1.258758083	1.414691135	0.486008939	<b>0.220116107</b>	114.7531486	7.322371631	1.334863805	0.681258589	0.85875914	<b>0.411901519</b>	14.36160461	8.210356189	0.005371976	0.70717098	1.354053085
LSMOP6_ORJ2_DIM206	0.440161754	0.457627519	0.432112255	0.433196895	0.451397961	0.440858070	290.8224493	0.491828277	0.546215024	<b>0.431916207</b>	0.578481326	0.573743295	0.474733076	0.43506646	0.035526209	0.459523	



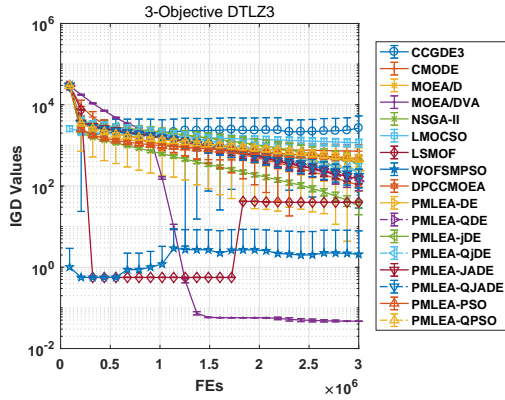
(a) Comparison of algorithms on 2-objective DTLZ1



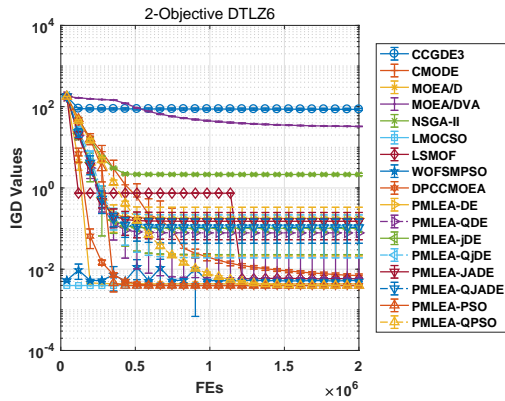
(b) Comparison of algorithms on 3-objective DTLZ1



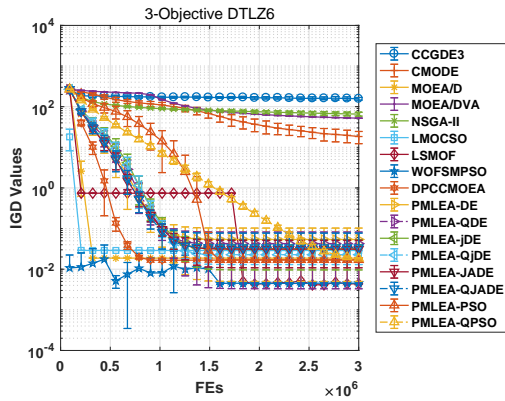
(c) Comparison of algorithms on 2-objective DTLZ3



(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.

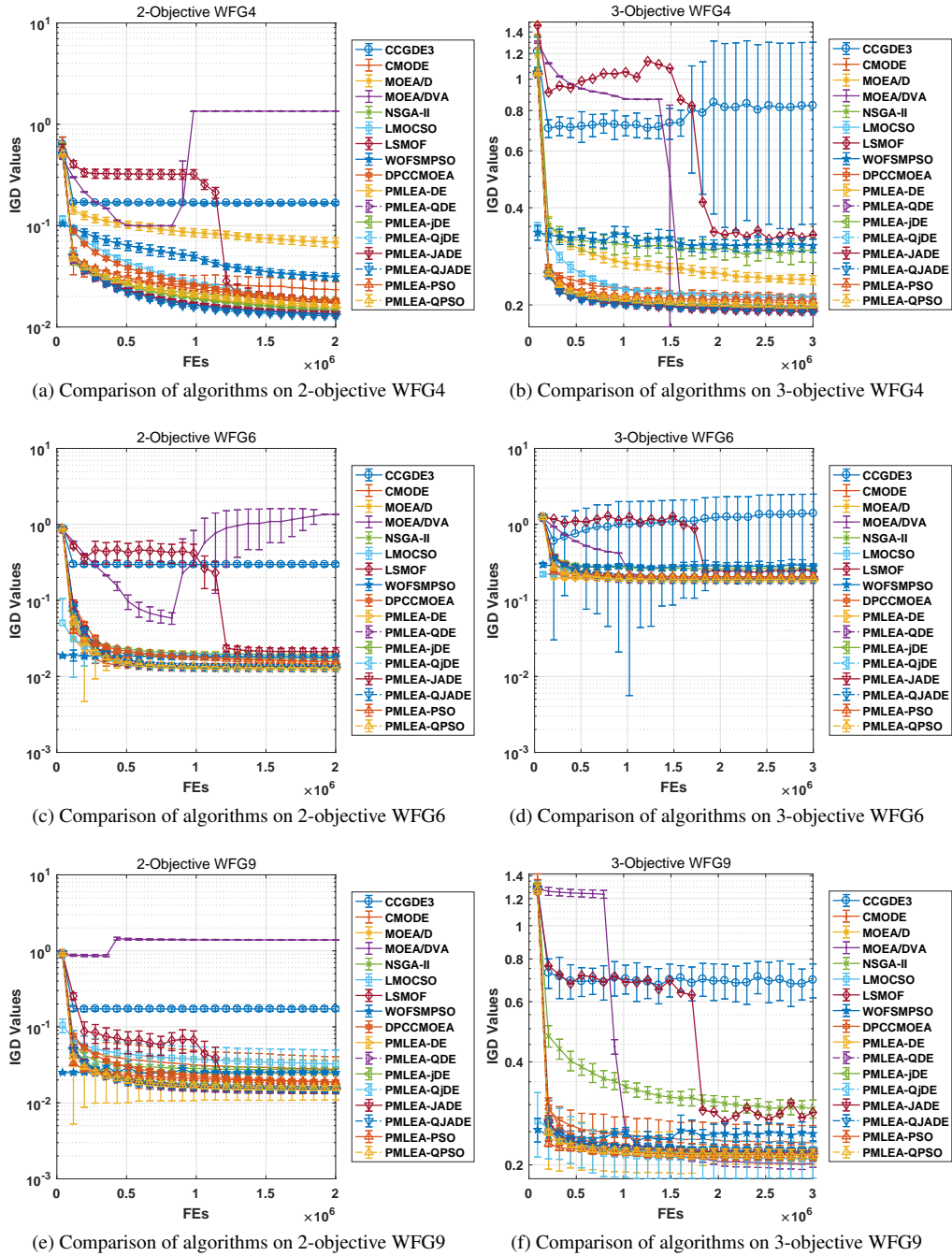


Figure 2: IGD evolution curves for different algorithms on the 2/3-objective WFG4, 6, and 9 functions.

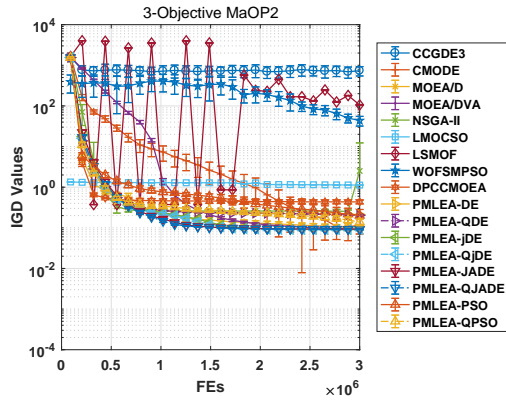
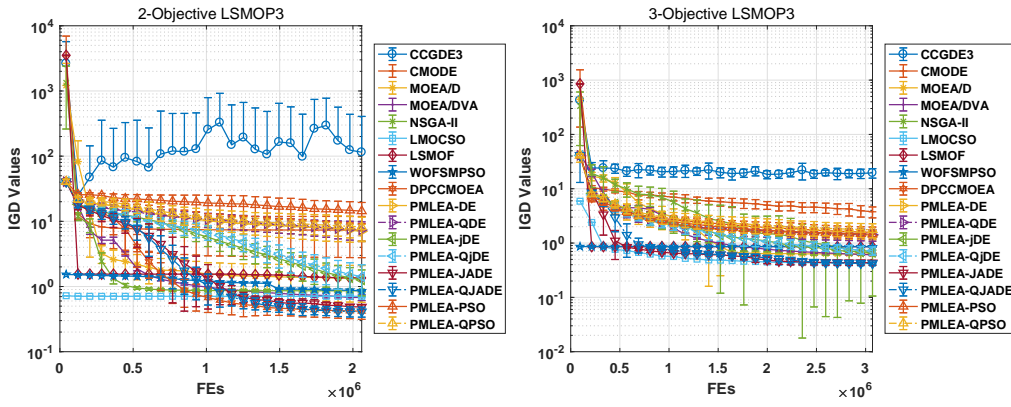
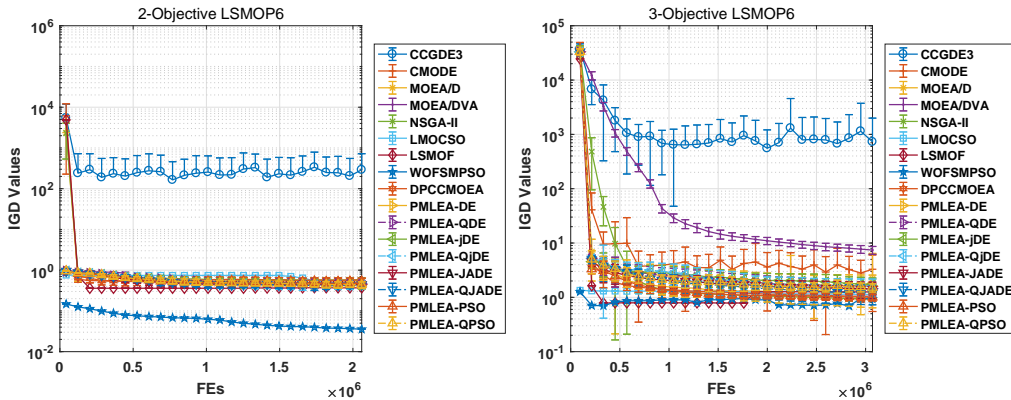


Figure 3: IGD evolution curves for different algorithms on the 3-objective MaOP2 function.



(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
LMOCSSO [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
LMOCSSO [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
LMOCSSO [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

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

## 495 5. Conclusions

496 Based on the DPCCMOLSEA framework, we  
497 proposed a series of quantum-enhanced algorithms:  
498 PMLEA-QDE, PMLEA-QjDE and PMLEA-QJADE.  
499 We combined parameter quantization and the DE op-  
500 erator to optimize the population. Moreover, in op-  
501 timizers of jDE and JADE, the adaptive parameters  
502 are enhanced by quantization. We used the multiob-  
503 jective test suites DTLZ, WFG, LSMOP and MaOP  
504 to compare the quantum-enhanced algorithms to oth-  
505 er state-of-the-art multiobjective algorithms and ranked  
506 the algorithms using nonparametric tests. The result-  
507 s showed that PMLEA-QJADE, PMLEA-QjDE and  
508 PMLEA-QDE achieve better optimization results than  
509 the other algorithms. The adoption of parallel operation  
510 in the MPI environment greatly reduced the time con-  
511 sumption of the algorithms. In future work, we will in-  
512 troduce the theory of quantum mechanics into differen-  
513 t stages of multiobjective large-scale EAs and propose  
514 new parameter-adaptive methods to improve the opti-  
515 mization efficiency. We will also use the improved mul-  
516 tiobjective large-scale EA to solve complex real-world  
517 optimization problems.

## 518 Acknowledgements

519 This work was supported in part by the National  
520 Natural Science Foundation of China (NSFC) under  
521 Grant No. 61976242, in part by the Opening Project of  
522 Guangdong Province Key Laboratory of Computational  
523 Science at the Sun Yat-sen University under Grant No.  
524 2018002, in part by the Open Fund of Key Laboratory  
525 of Intelligent Perception and Image Understanding of  
526 Ministry of Education, Xidian University, under Grant  
527 No. IPIU2019003, and in part by the State Key Pro-  
528 gram of National Natural Science Foundation of Chi-  
529 na under Grant No. 61836009. This work is also sup-  
530 ported in part by the National Supercomputer Center in  
531 Guangzhou.

## 532 References

533 [1] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm  
534 for a class of combinatorial optimization, *IEEE transactions*  
535 *on evolutionary computation* 6 (6) (2002) 580–593.  
536 [2] L. Wang, Advances in quantum-inspired evolutionary algo-  
537 rithms, *Control and Decision* 23 (12) (2008) 1321–1326.  
538 [3] H. Xiong, Z. Wu, H. Fan, G. Li, G. Jiang, Quantum rotation gate  
539 in quantum-inspired evolutionary algorithm: A review, analysis  
540 and comparison study, *Swarm and Evolutionary Computation*  
541 42 (2018) 43–57.  
542 [4] C. Patvardhan, S. Bansal, A. Srivastav, Parallel improved quan-  
543 tum inspired evolutionary algorithm to solve large size quadra-  
544 tic knapsack problems, *Swarm & Evolutionary Computation* 26  
545 (2016) 175–190.  
546 [5] B. Cao, J. Zhao, Y. Gu, Y. Ling, X. Ma, Applying graph-based di-  
547 fferential grouping for multiobjective large-scale optimization,  
548 *Swarm and Evolutionary Computation* 53.  
549 [6] X. Chang, Y.-L. Yu, Y. Yang, E. P. Xing, Semantic pooling for  
550 complex event analysis in untrimmed videos, *IEEE transaction-*  
551 *s on pattern analysis and machine intelligence* 39 (8) (2016)  
552 1617–1632.  
553 [7] T.-C. Lu, J.-C. Juang, Quantum-inspired space search algorithm  
554 (QSSA) for global numerical optimization, *Applied Mathemat-*  
555 *ics and Computation* 218 (6) (2011) 2516–2532.  
556 [8] B. Ji, X. Yuan, X. Li, Y. Huang, W. Li, Application of quantum-  
557 inspired binary gravitational search algorithm for thermal unit  
558 commitment with wind power integration, *Energy Conversion*  
559 *& Management* 87 (87) (2014) 589–598.  
560 [9] M. Fiasché, D. E. Liberati, S. Gualandi, M. Taisch, Quantum-  
561 inspired evolutionary multiobjective optimization for a dynam-  
562 ic production scheduling approach, in: *Multidisciplinary Ap-*  
563 *proaches to Neural Computing*, Springer, 2018, pp. 191–201.  
564 [10] L. Cui, L. Wang, J. Deng, J. Zhang, A new improved quantum  
565 evolution algorithm with local search procedure for capacitated  
566 vehicle routing problem, *Mathematical Problems in Engineering*  
567 2013 (159495) (2013) 1–17.  
568 [11] R. S. Pavithr, Gursaran, Quantum inspired social evolution  
569 (QSE) algorithm for 0-1 knapsack problem, *Swarm & Evolu-*  
570 *tionary Computation* 29 (2016) 33–46.  
571 [12] Z. A. E. M. Dahi, C. Mezoud, A. Draa, A quantum-inspired  
572 genetic algorithm for solving the antenna positioning problem,  
573 *Swarm & Evolutionary Computation* 31 (2016) 24–63.

574 [13] D. Alanis, P. Botsinis, S. X. Ng, L. Hanzo, Quantum-assisted  
575 routing optimization for self-organizing networks, *IEEE Access*  
576 2 (2014) 614–632.  
577 [14] B. Li, Z. Zhou, W. Zou, D. Li, Quantum memetic evolutionary  
578 algorithm-based low-complexity signal detection for underwater  
579 acoustic sensor networks, *IEEE Transactions on Systems & Man*  
580 *Cybernetics Part C* 42 (5) (2012) 626–640.  
581 [15] R. Storn, K. Price, Differential evolution—a simple and efficient  
582 heuristic for global optimization over continuous spaces, *Journal*  
583 *of global optimization* 11 (4) (1997) 341–359.  
584 [16] K. Price, R. M. Storn, J. A. Lampinen, *Differential evolution: A*  
585 *practical approach to global optimization*, Springer Science &  
586 *Business Media*, 2006.  
587 [17] G. Hu, S. Yang, Y. Li, S. U. Khan, A hybridized vector optimal  
588 algorithm for multi-objective optimal designs of electromagnet-  
589 ic devices, *IEEE Transactions on Magnetics* 52 (3) (2015) 1–4.  
590 [18] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, Quantum-behaved  
591 particle swarm optimization: Analysis of individual particle  
592 behavior and parameter selection, *Evolutionary computation*  
593 20 (3) (2012) 349–393.  
594 [19] S. Yang, G. Ni, Y. Li, B. Tian, R. Li, A universal tabu search  
595 algorithm for global optimization of multimodal functions with  
596 continuous variables in electromagnetics, *IEEE transactions on*  
597 *magnetics* 34 (5) (1998) 2901–2904.  
598 [20] A. SaiToh, R. Rahimi, M. Nakahara, A quantum genetic algo-  
599 rithm with quantum crossover and mutation operations, *Quant-*  
600 *um information processing* 13 (3) (2014) 737–755.  
601 [21] Z. Ren, R. Xiong, J. Chu, Hybrid quantum differential evolu-  
602 tionary algorithm and its applications, *Control Theory & Appli-*  
603 *cations* 28 (10) (2011) 1349–1355.  
604 [22] W. Ding, Q. Shi, S. Chen, Z. Guan, J. Wang, A novel quan-  
605 tum cooperative co-evolutionary algorithm for large-scale min-  
606 imum attribute reduction optimization, in: *2013 IEEE Sympo-*  
607 *sium on Computational Intelligence and Data Mining (CIDM)*,  
608 *IEEE*, 2013, pp. 280–286.  
609 [23] M. A. Potter, K. A. D. Jong, Cooperative coevolution: An ar-  
610 chitecture for evolving coadapted subcomponents, *Evolutionary*  
611 *computation* 8 (1) (2000) 1–29.  
612 [24] N. Tian, Quantum-behaved particle swarm optimization with  
613 cooperative coevolution for large scale optimization, in: *2015*  
614 *14th International Symposium on Distributed Computing and*  
615 *Applications for Business Engineering and Science (DCABES)*,  
616 *IEEE*, 2015, pp. 82–85.  
617 [25] W. Fang, L. Zhang, J. Zhou, X. Wu, J. Sun, A novel quantum-  
618 behaved particle swarm optimization with random selection for  
619 large scale optimization, in: *2017 IEEE Congress on Evolution-*  
620 *ary Computation (CEC)*, *IEEE*, 2017, pp. 2746–2751.  
621 [26] K. Praditwong, M. Harman, X. Yao, Software module cluster-  
622 ing as a multi-objective search problem, *IEEE Transactions on*  
623 *Software Engineering* 37 (2) (2010) 264–282.  
624 [27] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, Y. Yang, Rank-  
625 constrained spectral clustering with flexible embedding, *IEEE*  
626 *transactions on neural networks and learning systems* 29 (12)  
627 (2018) 6073–6082.  
628 [28] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C.  
629 Coello, A survey of multiobjective evolutionary algorithms for  
630 data mining: Part I, *IEEE Transactions on Evolutionary Comput-*  
631 *ation* 18 (1) (2013) 4–19.  
632 [29] L. M. Antonio, C. A. C. Coello, Use of cooperative coevolu-  
633 tion for solving large scale multiobjective optimization prob-  
634 lems, in: *2013 IEEE Congress on Evolutionary Computation*,  
635 *IEEE*, 2013, pp. 2758–2765.  
636 [30] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, M. Gong,  
637 A multiobjective evolutionary algorithm based on decision vari-  
638 able 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 co-evolutionary 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 multi-objective 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 multi-objective 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.

- 769 [63] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. D.  
770 Fonseca, Performance assessment of multiobjective optimizers:  
771 An analysis and review, *IEEE Transactions on Evolutionary*  
772 *Computation* 7 (2) (2003) 117–132.
- 773 [64] S. Cheng, Y. Shi, Q. Qin, On the performance metrics of multi-  
774 objective optimization, in: *International Conference in Swarm*  
775 *Intelligence*, Springer, 2012, pp. 504–512.
- 776 [65] J. Derrac, S. Garca, D. Molina, F. Herrera, A practical tutorial  
777 on the use of nonparametric statistical tests as a methodology  
778 for comparing evolutionary and swarm intelligence algorithms,  
779 *Swarm & Evolutionary Computation* 1 (1) (2011) 3–18.
- 780 [66] J. Alcal-Fdez, L. Snchez, S. Garca, M. J. D. Jesus, S. Ventura,  
781 J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas,  
782 KEEL: a software tool to assess evolutionary algorithms for data  
783 mining problems, *Soft Computing* 13 (3) (2009) 307–318.